INFORMATIK
BERICHTE

363 – 04/2012

A Generalized Iterative Scaling Algorithm for Maximum Entropy Reasoning in Relational Probabilistic Conditional Logic Under Aggregation Semantics

Marc Finthammer

FernUniversität in Hagen

Fakultät für Mathematik und Informatik
Postfach 940
D-58084 Hagen

A Generalized Iterative Scaling Algorithm for Maximum Entropy Reasoning in Relational Probabilistic Conditional Logic Under Aggregation Semantics

Marc Finthammer

Dept. of Computer Science, FernUniversität in Hagen

Abstract. Recently, different semantics for relational probabilistic conditionals and corresponding maximum entropy (ME) inference operators have been proposed. In this paper, we study the so-called aggregation semantics that covers both notions of a statistical and subjective view. The computation of its inference operator requires the calculation of the ME-distribution satisfying all probabilistic conditionals. Since each conditional induces a linear constraint on the probability distribution, the optimization problem to solve is the calculation of the probability distribution with maximum entropy under linear constraints. We demonstrate how the well-known Generalized Iterative Scaling (GIS) algorithm technique can be applied to this optimization problem to calculate the maximum entropy distribution in an iterative way. We show how the linear constrains are transformed into normalized feature functions to meet the requirements of GIS and present a practical algorithm which is tailor-made for the computation of the ME-inference operator based on aggregation semantics. We also present a practical implementation of the developed algorithm.

1 Introduction

There exist many approaches which combine propositional logic with probability theory to express uncertain knowledge and allow uncertain reasoning, e. g. Bayes Nets and Markov Nets [16] or probabilistic conditional logic [17]. Some of these approaches have been extended to first-order logic, e. g. Bayesian logic programs [13], Markov logic networks [8], and relational probabilistic conditional logic [9, 12] which introduces relational probabilistic conditionals.

Recently, different semantics for relational probabilistic conditionals and corresponding maximum entropy (ME) inference operators have been proposed. One of these approaches is the so-called aggregation semantics presented in [12]. This semantics has some nice properties, as it allows to cover both notions of a statistical and a subjective point of view. It can also handle statements about exceptional individuals without running into imminent inconsistencies. The following example taken from [12] (and inspired by [7]) illustrates some difficulties which can arise from a knowledge base in probabilistic first-order logic. Let X, Y

denote variables and let $el(X)$ mean that X is an elephant, $ke(X)$ means that X is a zookeeper, and $likes(X, Y)$ expresses that X likes Y.

Example 1.
$$r_1 : (likes(X, Y) \mid el(X) \land ke(Y))\,[0.6]$$
$$r_2 : (likes(X, fred) \mid el(X) \land ke(fred))\,[0.4]$$
$$r_3 : (likes(clyde, fred) \mid el(clyde) \land ke(fred))\,[0.7]$$

The first probabilistic rule (or probabilistic conditional) r_1 expresses that for an arbitrary chosen elephant and keeper (from some given population), there is a 0.6 probability that the elephant likes the keeper. But r_2 states that there is an (exceptional) keeper *fred*, for whom there is just a 0.4 probability that an arbitrary elephants likes him. Rule r_3 makes a statement about two exceptional individuals, i. e. the probability that the elephant *clyde* likes the keeper *fred* is even 0.7. Rule r_1 express statistical knowledge which holds in some given population, whereas r_3 expresses individual belief, and r_2 is a mixture of both. A simple semantical approach would be to ground all first-order rules (according to a given universe) and define semantics on grounded probabilistic rules. But in the above example, this would already cause severe problems, because the grounding $(likes(clyde, fred) \mid el(clyde) \land ke(fred))\,[0.6]$ of r_1 is in conflict with r_3 and also with the grounding of $(likes(clyde, fred) \mid el(clyde) \land ke(fred))\,[0.4]$ of r_2. However, the aggregation semantics is capable of handling such conflicts if an appropriate universe is provided, so that the probabilities of exceptional and generic individuals can be balanced.

The model-based inference operator $\mathrm{ME}_\odot$ for aggregation semantics presented in [12] is based on the principle of maximum entropy. The principle of maximum entropy exhibits excellent properties for commonsense reasoning [15, 10, 11] and allows to complete uncertain and incomplete information in an information-theoretic optimal way, and also the ME-based inference operator $\mathrm{ME}_\odot$ features many desirable properties of a rational inference operator [12]. However, up to now no practical implementation of $\mathrm{ME}_\odot$ inference has been developed. The determination of $\mathrm{ME}_\odot\,(\mathcal{R})$ for a set $\mathcal{R}$ of probabilistic conditionals requires the calculation of the ME-distribution satisfying all probabilistic conditionals in $\mathcal{R}$. This induces a convex optimization problem, so general techniques for solving convex optimization problems could be applied to compute a solution. Instead of employing such general techniques, in this paper we present the first practical algorithm for computing $\mathrm{ME}_\odot\,(\mathcal{R})$ which is tailor-made for the problem. We employ the technique of the well-known Generalized Iterative Scaling (GIS) algorithm [6], which allows to compute the ME-distribution under linear constraints.

The the rest of the paper is structured as follows. In Sec. 2 we give a compact overview of the syntax of relational probabilistic conditional logic and present the aggregation semantics. In Sec. 3 feature functions are introduced to represent the entailment relation in more compact way. In Sec. 4 the ME-inference operator $\mathrm{ME}_\odot$ and the corresponding optimization problem are defined. In Sec. 5 the Generalized Iterative Scaling (GIS) algorithm and its requirement are discussed. We demonstrate how the optimization problem can be transformed into a normalized form. This enables us to employ the GIS technique in a concrete algorithm which determines $\mathrm{ME}_{\odot}$. In Sec. 6 we present a practical implementation of the algorithm. We conclude in Sec. 7 with a summary and some discussion of related and future work.

2 Background: Aggregation Semantics

In this section, we will give a brief introduction to the syntax of relational probabilistic condition logic and the aggregation semantics.

2.1 Syntax

We consider a first-order signature $\Sigma := (\mathit{Pred}, \mathit{Const})$ consisting of a set of first order predicates *Pred* and a finite set of constants *Const*. So Σ is a restricted signature since it only contains functions with an arity of zero. Let p/k denote the predicate $p \in \mathit{Pred}$ with arity k. The set of atoms $\mathcal{A}$ over *Pred* with respect to a set of variables *Var* and *Const* is defined in the usual way by $p(t_1, \dots, t_k) \in \mathcal{A}$ iff $p/k \in \mathit{Pred}, t_i \in (\mathit{Var} \cup \mathit{Const})$ for $1 \leq i \leq k$. For better readability we will usually omit the referring indices.

Let $\mathcal{L}$ be a quantifier-free first-order language defined over Σ in the usual way, that is, $A \in \mathcal{L}$ if $A \in \mathcal{A}$, and if $A, B \in \mathcal{L}$ then $\neg A, A \wedge B \in \mathcal{L}$. Let $A \vee B$ be the shorthand for $\neg(\neg A \wedge \neg B)$. If it is clear from context, we use the short notation AB to abbreviate a conjunction $A \wedge B$.

Let $\mathrm{gnd}(A)$ be a grounding function which maps a formula A to its respective set of ground instances in the usual way.

Definition 1 (Conditional). *Let $A(\boldsymbol{X}), B(\boldsymbol{X}) \in \mathcal{L}$ be first-order formulas with $\boldsymbol{X}$ containing the variables of A and B. $(B(\boldsymbol{X})|A(\boldsymbol{X}))$ is called a* conditional*. A is the antecedence and B the consequence of the conditional. The set of all conditionals over $\mathcal{L}$ is denoted by $(\mathcal{L}|\mathcal{L})$.*

Definition 2 (Probabilistic Conditional). *Let $(B(\boldsymbol{X})|A(\boldsymbol{X})) \in (\mathcal{L}|\mathcal{L})$ be a conditional and let $d \in [0,1]$ be a real value. $(B(\boldsymbol{X})|A(\boldsymbol{X}))\ [d]$ is called a* probabilistic conditional *with probability d. If $d \in \{0,1\}$ then the probabilistic conditional is called* hard*, otherwise it is a called* soft*. The set of all probabilistic conditionals over $\mathcal{L}$ is denoted by $(\mathcal{L}|\mathcal{L})^{prob}$.*

A set of probabilistic conditionals is also called a *knowledge base*. (Probabilistic) conditionals are also called *(probabilistic) rules*. If it is clear from context, we will omit the "probabilistic" and just use the term "conditional".

Let $\mathcal{H}$ denote the Herbrand base, i.e. the set containing all ground atoms constructible from *Pred* and *Const*. A Herbrand interpretation ω is a subset of the ground atoms, that is $\omega \subseteq \mathcal{H}$. Using a closed world assumption, each ground atom $p_{\mathrm{gnd}} \in \omega$ is interpreted as true and each $p_{\mathrm{gnd}} \notin \omega$ is interpreted as false; in this way a Herbrand interpretation is similar to a complete conjunction in propositional logic. Let Ω denote the set of all possible worlds (i. e. Herbrand interpretations), that is, $\Omega := \mathfrak{P}(\mathcal{H})$ (with $\mathfrak{P}$ denoting the power set).

Definition 3 (Set of Grounding Vectors). *For a conditional* $(B(\boldsymbol{X})|A(\boldsymbol{X})) \in (\mathcal{L}|\mathcal{L})$*, the set of all constant vectors* $\boldsymbol{a}$ *which can be used for proper groundings of* $(B(\boldsymbol{X})|A(\boldsymbol{X}))$ *is defined as:*

$$\mathcal{H}^{\boldsymbol{x}(A,B)} := \{\boldsymbol{a} = (a_1, \ldots, a_s) \mid a_1, \ldots, a_s \in Const \\ \quad and\ (B(\boldsymbol{a})|A(\boldsymbol{a})) \in \operatorname{gnd}((B(\boldsymbol{X})|A(\boldsymbol{X})))\}$$

2.2 Aggregation Semantics

Let $P : \Omega \to [0, 1]$ be a probability distribution over possible worlds and let $\mathcal{P}_\Omega$ be the set of all such distributions. P is extended to ground formulas $A(\boldsymbol{a})$, with $\boldsymbol{a} \in \mathcal{H}^{\boldsymbol{x}(A)}$, by defining

$$P(A(\boldsymbol{a})) := \sum_{\omega \models A(\boldsymbol{a})} P(\omega)$$

In [12] the *aggregating semantics* is introduced which defines the entailment relation between probability distributions and probabilistic conditionals as follows:

Definition 4 (Aggregation Semantics Entailment Relation [12]). *The entailment relation* $\models_\odot$ *between a probability distribution* $P \in \mathcal{P}_\Omega$ *and a probabilistic conditional* $(B(\boldsymbol{X})|A(\boldsymbol{X}))\ [d] \in (\mathcal{L}|\mathcal{L})^{prob}$ *with* $\sum_{\boldsymbol{a} \in \mathcal{H}^{\boldsymbol{x}(A,B)}} P(A(\boldsymbol{a})) > 0$ *is defined as:*

$$P \models_\odot (B(\boldsymbol{X})|A(\boldsymbol{X}))\ [d] \qquad \text{iff} \qquad \frac{\sum\limits_{\boldsymbol{a} \in \mathcal{H}^{\boldsymbol{x}(A,B)}} P(A(\boldsymbol{a})B(\boldsymbol{a}))}{\sum\limits_{\boldsymbol{a} \in \mathcal{H}^{\boldsymbol{x}(A,B)}} P(A(\boldsymbol{a}))} = d \tag{1}$$

Note that both sums of the fraction run over the same set of grounding vectors and therefore the same number of ground instances, i. e. a particular probability $P(A(\boldsymbol{a}))$ can be contained multiple times in the denominator sum. If $P \models_\odot r$ holds for a conditional r, we say that P *satisfies* r or that P *is a model of* r.

Thus, the aggregation semantics resembles the definition of a conditional probability by summing up the probabilities of all respective ground formulas. The entailment relation $\models_\odot$ is extended to a set $\mathcal{R}$ of probabilistic conditionals by defining

$$P \models_\odot \mathcal{R} \quad \text{iff} \quad \forall r \in \mathcal{R} : P \models_\odot r$$

Let $\mathcal{S}(\mathcal{R}) := \{P \in \mathcal{P}_\Omega : P \models_\odot \mathcal{R}\}$ denote the set of all probability distributions which satisfy $\mathcal{R}$. $\mathcal{R}$ is *consistent* iff $\mathcal{S}(\mathcal{R}) \neq \emptyset$, i. e. there exists a probability distribution which satisfies all conditionals in $\mathcal{R}$. Accordingly, a probabilistic conditional r is called consistent (or *satisfiable*), if there exists a distribution which satisfies r.

3 Feature Functions

For propositional conditionals, the satisfaction relation can be expressed by using feature functions (e. g. [9]). The following definition introduces feature functions for the relational case where the groundings have to be taken into account.

Definition 5 (Feature Function). *For a probabilistic conditional $r_i := (B_i(\boldsymbol{X})|A_i(\boldsymbol{X}))\ [d_i]$ define the functions $v_i^{\#}, f_i^{\#} : \Omega \to \mathbb{N}_0$ with*

$$v_i^{\#}(\omega) := \left|\left\{\boldsymbol{a} \in \mathcal{H}^{\boldsymbol{x}(A_i,B_i)} \mid \omega \models A_i(\boldsymbol{a})B_i(\boldsymbol{a})\right\}\right| \text{ and}$$
$$f_i^{\#}(\omega) := \left|\left\{\boldsymbol{a} \in \mathcal{H}^{\boldsymbol{x}(A_i,B_i)} \mid \omega \models A_i(\boldsymbol{a})\overline{B_i(\boldsymbol{a})}\right\}\right| \tag{2}$$

$v_i^{\#}(\omega)$ indicates the number of groundings which verify *r_i for a certain $\omega \in \Omega$, whereas $f_i^{\#}(\omega)$ specifies the number of groundings which* falsify *r_i.*

The linear function function $\sigma_i : \Omega \to \mathbb{R}$ with

$$\sigma_i(\omega) := v_i^{\#}(\omega)(1-d_i) - f_i^{\#}(\omega)d_i \tag{3}$$

is called the feature function *of the probabilistic conditional r_i.*

Now that we have defined the feature function σ_i of a probabilistic conditional, the entailment relation of the aggregation semantics can be expressed in terms of a linear constraint involving σ_i.

Proposition 1. *Let $(B_i(\boldsymbol{X})|A_i(\boldsymbol{X}))\ [d_i]$ be a probabilistic conditional and let σ_i be its feature function according to Definition 5. Then it holds for $P \in \mathcal{P}_\Omega$ with $\sum_{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}} P(A(\boldsymbol{a})) > 0$:*

$$P \models_{\odot} (B_i(\boldsymbol{X})|A_i(\boldsymbol{X}))\ [d_i] \quad \text{iff} \quad \sum_{\omega\in\Omega} P(\omega)\sigma_i(\omega) = 0 \tag{4}$$

Proof. For ease of readability, we omit the index i of conditional r_i.

$$\begin{aligned}
&P \models_{\odot} (B(\boldsymbol{X})|A(\boldsymbol{X}))\ [d] \\
\text{iff}\ &\sum_{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}} P(A(\boldsymbol{a})B(\boldsymbol{a})) = d \sum_{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}} P(A(\boldsymbol{a})) \\
\text{iff}\ &\sum_{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}} P(A(\boldsymbol{a})B(\boldsymbol{a})) = d \sum_{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}} \left[P(A(\boldsymbol{a})B(\boldsymbol{a})) + P\left(A(\boldsymbol{a})\overline{B(\boldsymbol{a})}\right)\right] \\
\text{iff}\ &\sum_{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}} \left[(1-d)P(A(\boldsymbol{a})B(\boldsymbol{a})) - dP\left(A(\boldsymbol{a})\overline{B(\boldsymbol{a})}\right)\right] = 0 \\
\text{iff}\ &\left((1-d)\sum_{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}} \sum_{\substack{\omega\in\Omega:\\ \omega\models A(\boldsymbol{a})B(\boldsymbol{a})}} P(\omega)\right) - d\sum_{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}} \sum_{\substack{\omega\in\Omega:\\ \omega\models A(\boldsymbol{a})\overline{B(\boldsymbol{a})}}} P(\omega) = 0 \\
\text{iff}\ &\left((1-d)\sum_{\omega\in\Omega} \sum_{\substack{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}:\\ \omega\models A(\boldsymbol{a})B(\boldsymbol{a})}} P(\omega)\right) - d\sum_{\omega\in\Omega} \sum_{\substack{\boldsymbol{a}\in\mathcal{H}^{\boldsymbol{x}(A,B)}:\\ \omega\models A(\boldsymbol{a})\overline{B(\boldsymbol{a})}}} P(\omega) = 0
\end{aligned}$$

$$\text{iff} \sum_{\omega \in \Omega} \big(v^{\#}(\omega)(1-d)P(\omega) - f^{\#}(\omega)dP(\omega)\big) = 0$$

$$\text{iff} \sum_{\omega \in \Omega} \big(v^{\#}(\omega)(1-d) - f^{\#}(\omega)d\big) \; P(\omega) = 0$$

$$\text{iff} \sum_{\omega \in \Omega} \sigma(\omega)P(\omega) = 0 \tag{5}$$

Proposition 1 shows that under aggregation semantics a conditional induces a linear constraint which has to be met by a satisfying probability distribution. The *expected value* $\mathbb{E}(\sigma_i, P)$ of a function σ_i under a distribution P is defined as $\mathbb{E}(\sigma_i, P) := \sum_{\omega \in \Omega} \sigma_i(\omega)P(\omega)$. Thus (5) states that the expected value of the feature function σ_i must be 0 under every satisfying distribution.

4 Maximum Entropy Inference for Aggregation Semantics

The *entropy*

$$H(P) := -\sum_{\omega \in \Omega} P(\omega) \log P(\omega)$$

of a probability distribution P measures the indifference within the distribution.

In [15] and [10], it has already been shown for a propositional framework, that the principle of *maximum entropy (ME)* provides some desirable properties for commonsense reasoning. The ME-inference operator for propositional conditional logic from [10] allows to perform probabilistic reasoning on uncertain and incomplete knowledge in an information-theoretic optimal way.

4.1 ME-Inference Operator based on Aggregation Semantics

A ME-inference operator based on aggregation semantics is introduced in [12] as follows.

Definition 6 (ME$_\odot$ Inference Operator [12]). *Let $\mathcal{R}$ be a consistent set of probabilistic conditionals. The ME-inference operator* ME$_\odot$ *based on aggregation semantics is defined as*

$$\mathrm{ME}_\odot(\mathcal{R}) := \arg \max_{P \in \mathcal{P}_\Omega : P \models_\odot \mathcal{R}} H(P) \tag{6}$$

From all distributions satisfying $\mathcal{R}$, this model-based inference operator chooses the unique distribution with maximum entropy as a model for $\mathcal{R}$. ME$_\odot$ represents the incomplete (and uncertain) knowledge from $\mathcal{R}$ inductively completed to a full distribution by applying the ME-principle and respecting the aggregation semantics. It also features several desirable properties of a rational inference operator as shown in [12].

In [20] it is shown that (6) has a unique solution and describes a *convex optimization problem*, since the solutions to $P \models_\odot \mathcal{R}$ form a convex set and $H(P)$ is a strictly concave function. Thus, ME$_\odot(\mathcal{R})$ is well defined.

4.2 ME-Inference Optimization Problem

To avoid cumbersome distinctions of cases, in the following we consider only soft probabilistic conditionals. Therefore, for the rest of the paper let

$$\mathcal{R} := \{r_1, \ldots, r_m\} \tag{7}$$

be a consistent set of m soft probabilistic conditionals

$$r_i = (B_i(\boldsymbol{X}) | A_i(\boldsymbol{X}))\,[d_i]\,, \text{ with } d_i \in (0,1), 1 \leq i \leq m \tag{8}$$

and let σ_i denote the feature function of r_i according to Definition 5.

Proposition 2. *For any consistent set of soft probabilistic conditionals $\mathcal{R}$ as given by (7) and (8), it holds:*

$$\exists P \in \mathcal{S}(\mathcal{R}) : \forall \omega \in \Omega : P(\omega) > 0,$$

that is, there exists a positive probability distribution which satisfies $\mathcal{R}$.

From Definition 6 and Proposition 1 it follows that the determination of $\mathrm{ME}_{\odot}(R)$ requires to solve the following optimization problem with objective function $H(P)$ and m linear constraints induced by the m conditionals of R:

Definition 7 (Optimization Problem OptAgg($\mathcal{R}$)). *Let $\sigma_i, 1 \leq i \leq m$, be the feature functions for $\mathcal{R}$. Then the optimization problem* $\textsc{OptAgg}(\mathcal{R})$ *is defined as:*

$$\begin{array}{ll} \textit{maximize}\ H(P) & \\ \textit{subject to}\ \sum_{\omega \in \Omega} P(\omega)\sigma_i(\omega) & = 0,\ 1 \leq i \leq m \\ \qquad\qquad\ \ \sum_{\omega \in \Omega} P(\omega) & = 1 \\ \qquad\qquad\ \ P(\omega) & \geq 0, \forall \omega \in \Omega \end{array} \tag{9}$$

The two latter constraints ensure that the solution is a proper probability distribution.

Proposition 3. *The solution of the optimization problem* $\textsc{OptAgg}(\mathcal{R})$ *for a given set $\mathcal{R}$ is* $\mathrm{ME}_{\odot}(\mathcal{R})$.

5 Computing the Maximum Entropy Distribution

There exist several algorithms to calculate the solution of a general convex optimization problem [4], i. e. these algorithms can be applied to a convex optimization problem with an arbitrary (convex) objective function. In this paper, we investigate an algorithm which is tailor-made for a convex optimization problem of the form (9), i. e. for a convex optimization problem with entropy $H(P)$ as objective function. Since this algorithm is specialized to entropy optimization, it can take advantage of certain characteristics of the entropy function, whereas general algorithms for convex optimization problems can just utilize the convexity of the objective function.

5.1 Generalized Iterative Scaling

The so-called *Generalized Iterative Scaling (GIS)* algorithm presented in [6] computes the ME-distribution under linear constraints, i. e. it iteratively calculates a sequence of distributions which converges to the solution. To be precise, the GIS algorithm allows to compute the distribution with minimum relative entropy (also see [5] for an alternative proof of the algorithm). The *relative entropy* (also called *Kullback-Leibler divergence* or *information divergence*) between two distributions P and Q is defined as

$$K(P, Q) := \sum_{\omega \in \Omega} P(\omega) \log \frac{P(\omega)}{Q(\omega)}$$

Let $P_U(\omega) := \frac{1}{|\Omega|}$ for all $\omega \in \Omega$ denote the uniform distribution over Ω. It is easy to see that

$$K(P, P_U) = \log |\Omega| - H(P) \tag{10}$$

holds, i. e. entropy is just a special of relative entropy.

Proposition 4. *Let P_U be the uniform distribution and $\mathcal{S}$ be a set of probability distributions over Ω. Then it holds:*

$$\arg \min_{P \in \mathcal{S}} K(P, P_U) = \arg \max_{P \in \mathcal{S}} H(P)$$

The proof of 4 follows directly from (10). Therefore, instead of maximizing the entropy of a distribution, we will consider minimizing the relative entropy of a distribution with respect to the uniform distribution.

The general from of the optimization problem solved by the GIS algorithm is as follows:

Definition 8 (Optimization Problem OptGis(EQ)).
Let $Q \in \mathcal{P}_\Omega$ be a given probability distribution. For $i = 1, \dots, m$, let $a_i : \Omega \to \mathbb{R}$ be a given function and let $h_i \in \mathbb{R}$ be its given expected value, so that the equation system (denoted by EQ) of linear constraints $\sum_{\omega \in \Omega} P(\omega) a_i(\omega) = h_i$, $i = 1, \dots, m$ induced by the functions and their expected values can be satisfied by a positive probability distribution.

Then the optimization problem OptGis(EQ) *is defined as:*

$$\begin{aligned} &\textit{minimize } K(P, Q) \\ &\textit{subject to } \textstyle\sum_{\omega \in \Omega} P(\omega) a_i(\omega) &&= h_i,\ i = 1, \dots, m \\ &\qquad\qquad \textstyle\sum_{\omega \in \Omega} P(\omega) &&= 1 \\ &\qquad\qquad P(\omega) &&> 0,\ \forall \omega \in \Omega \end{aligned} \tag{11}$$

If the preconditions of Definition 8 are met, the GIS algorithm can be applied to compute the solution to the optimization problem OptGis(EQ).

Since the constraints in (9) have been induced by a consistent set of soft probabilistic conditionals, they can be satisfied by a positive distribution according to Proposition 2. So in principle, the GIS algorithm can be applied to compute the solution to the optimization problem $\textsc{OptAgg}(\mathcal{R})$, since this matches the form of an optimization problem $\textsc{OptGis}(EQ)$.

5.2 Transforming OptGis(EQ) into a Normalized Form

The concrete application of the GIS algorithm to $\textsc{OptGis}(EQ)$ requires a transformation into a normalized form meeting some additional requirements.

Definition 9 (Optimization Problem OptGisNorm($\hat{EQ}$)). *Let $Q \in \mathcal{P}_\Omega$ be a given probability distribution. For $i = 1, \ldots, \hat{m}$, let $\hat{a}_i : \Omega \to \mathbb{R}$ be a given function and let $\hat{h}_i \in \mathbb{R}$ be its given expected value, so that the induced equation system (denoted by $\hat{EQ}$) of linear constrains $\sum_{\omega\in\Omega} P(\omega)\hat{a}_i(\omega) = \hat{h}_i, i = 1, \ldots, \hat{m}$, can be satisfied by a positive probability distribution. If*

$$\hat{a}_i(\omega) \geq 0, \forall\omega \in \Omega,\ i = 1, \ldots, \hat{m}, \tag{12}$$

$$\sum_{i=1}^{\hat{m}} \hat{a}_i(\omega) = 1, \qquad \forall\omega \in \Omega \tag{13}$$

$$\hat{h}_i > 0, \qquad i = 1, \ldots, \hat{m} \tag{14}$$

$$\sum_{i=1}^{\hat{m}} \hat{h}_i = 1 \tag{15}$$

hold, then the optimization problem $\textsc{OptGisNorm}(\hat{EQ})$ is defined as:

$$\begin{array}{ll} \textit{minimize} & K(P,Q) \\ \textit{subject to} & \sum_{\omega\in\Omega} P(\omega)\hat{a}_i(\omega) = \hat{h}_i,\ i = 1, \ldots, \hat{m} \\ & \sum_{\omega\in\Omega} P(\omega) \qquad = 1 \\ & P(\omega) \qquad\qquad > 0,\ \forall\omega \in \Omega \end{array} \tag{16}$$

In [6] it is shown in general that an optimization problem $\textsc{OptGis}(EQ)$ can always be transformed appropriately to meet the requirements (12) – (15) of $\textsc{OptGisNorm}(\hat{EQ})$. Note that in [6] the additional requirement is made that each function a_i in $\textsc{OptGis}(EQ)$ has at least one non-zero value, thereby assuring that (14) holds in $\textsc{OptGisNorm}(\hat{EQ})$ after the transformation. We will show later (in the proof of Proposition 6) that for our transformed problem (14) holds anyway, therefore we do not have to consider that additional requirement.

The normalized form ($\textsc{OptGisNorm}(\hat{EQ})$) can be reached by transforming the original constraints appropriately (to assure (12) and (14)) and by adding an additional correctional constraint (to assure (13) and (15)). Thus, in a preprocessing step to the GIS algorithm the original constraints of an optimization problem $\textsc{OptGis}(EQ)$ have to be transformed into an equivalent *normalized form* which mets the requirements (12) to (15) of an optimization problem $\textsc{OptGisNorm}(\hat{EQ})$. Then the GIS algorithm can be applied to $\textsc{OptGisNorm}(\hat{EQ})$ to compute a solution.

In the following, we will demonstrate how the constraints from (9) can be transformed into a set of normalized constraints meeting the normalization requirements (12) to (15).

So we consider again the consistent set $\mathcal{R}$ of soft constraint from Definition 7. Let

$$G_i^{\#} := |\mathcal{H}^{\boldsymbol{x}(A_i,B_i)}|$$

be the number of groundings of a conditional $r_i \in \mathcal{R}$, $1 \leq i \leq m$, and let

$$\mathcal{G}^{\#} := \sum_{i=1}^{m} G_i^{\#} \tag{17}$$

denote the total number of groundings of all conditionals in $\mathcal{R}$.

Definition 10 (Non-Negative Feature Function). *For each feature function σ_i of a conditional $r_i \in \mathcal{R}$ of the optimization problem* $\textsc{OptAgg}(\mathcal{R})$*, let the* non-negative feature function $\sigma'_i : \mathit{\Omega} \to \mathbb{R}_0^+$ *be defined as*

$$\sigma'_i(\omega) := \sigma_i(\omega) + d_i G_i^{\#},\ \forall \omega \in \mathit{\Omega} \tag{18}$$

and the expected value of σ'_i, denoted by ε'_i, is set to

$$\varepsilon'_i := d_i G_i^{\#} \tag{19}$$

Proposition 5. *For a feature function σ'_i and its expected value ε'_i according to Definition 10 the following holds:*

$$0 \leq \sigma'_i(\omega) \leq G_i^{\#},\ \forall \omega \in \mathit{\Omega} \tag{20}$$

$$\begin{array}{c} \sum_{\omega \in \mathit{\Omega}} P(\omega)\sigma_i(\omega) = 0 \\ \Leftrightarrow \sum_{\omega \in \mathit{\Omega}} P(\omega)\sigma'_i(\omega) = \varepsilon'_i \end{array} \tag{21}$$

Proof. From Definition 5 it follows directly that

$$0 \leq v_i^{\#}(\omega) \leq G_i^{\#} \text{ and } 0 \leq f_i^{\#}(\omega) \leq G_i^{\#}$$

and therefore

$$-G_i^{\#} d_i \leq \sigma_i(\omega) \leq G_i^{\#}(1 - d_i) \tag{22}$$

holds. Together with the definition (18) of σ'_i, it follows that (20) holds. The constraints in (21) are equivalent because:

$$\begin{aligned} &\textstyle\sum_{\omega \in \mathit{\Omega}} P(\omega)\sigma_i(\omega) &&= 0 \\ \Leftrightarrow &\textstyle\sum_{\omega \in \mathit{\Omega}} P(\omega) d_i G_i^{\#} + P(\omega)\sigma_i(\omega) &&= d_i G_i^{\#} \\ \Leftrightarrow &\textstyle\sum_{\omega \in \mathit{\Omega}} P(\omega) \left(d_i G_i^{\#} + \sigma_i(\omega)\right) &&= d_i G_i^{\#} \\ \Leftrightarrow &\textstyle\sum_{\omega \in \mathit{\Omega}} P(\omega)\sigma'_i(\omega) &&= \varepsilon'_i \end{aligned}$$

Definition 11 (Normalized Feature Function). *For each feature function σ_i of a conditional $r_i \in \mathcal{R}$ of the optimization problem* $\textsc{OptAgg}(\mathcal{R})$*, let the* normalized feature function $\hat{\sigma}_i : \Omega \to [0,1]$ *be defined as*

$$\hat{\sigma}_i(\omega) := \frac{\sigma'_i(\omega)}{\mathcal{G}^\#} = \frac{\sigma_i(\omega) + d_i G_i^\#}{\mathcal{G}^\#},\ \forall \omega \in \Omega \tag{23}$$

and the expected value of $\hat{\sigma}_i$, denoted by $\hat{\varepsilon}_i$, is set to

$$\hat{\varepsilon}_i := \frac{\varepsilon'_i}{\mathcal{G}^\#} = \frac{d_i G_i^\#}{\mathcal{G}^\#} \tag{24}$$

Proposition 6. *For a feature function $\hat{\sigma}_i$ and its expected value $\hat{\varepsilon}_i$ according to Definition 11 the following holds:*

$$0 \le \hat{\sigma}_i(\omega) \le 1,\ \forall \omega \in \Omega \tag{25}$$

$$0 \le \sum_{i=1}^{m} \hat{\sigma}_i(\omega) \le 1,\ \forall \omega \in \Omega \tag{26}$$

$$\begin{aligned} &\textstyle\sum_{\omega\in\Omega} P(\omega)\sigma_i(\omega) = 0 \\ \Leftrightarrow &\textstyle\sum_{\omega\in\Omega} P(\omega)\hat{\sigma}_i(\omega) = \hat{\varepsilon}_i \end{aligned} \tag{27}$$

$$\hat{\varepsilon}_i > 0 \tag{28}$$

$$0 < \sum_{i=1}^{m} \hat{\varepsilon}_i < 1 \tag{29}$$

Proof. Equation (25) follows directly from (20) and the definition $\hat{\sigma}_i$. According to Proposition 5, $0 \le \sigma'_i(\omega) \le G_i^\#$ holds for every $1 \le i \le m$, and therefore $0 \le \sum_{i=1}^{m} \sigma'_i(\omega) \le \sum_{i=1}^{m} G_i^\#$ must hold as well. Using the definitions from (17) and (23) it becomes obvious that (26) must hold:

$$0 \le \sum_{i=1}^{m} \hat{\sigma}_i(\omega) = \sum_{i=1}^{m} \frac{\sigma'_i(\omega)}{\sum_{j=1}^{m} G_j^\#} = \frac{\sum_{i=1}^{m} \sigma'_i(\omega)}{\sum_{i=1}^{m} G_i^\#} \le 1$$

Using (21) and Definition 11, the constraints in (27) are equivalent because:

$$\begin{aligned} &\textstyle\sum_{\omega\in\Omega} P(\omega)\sigma_i(\omega) = 0 \\ \Leftrightarrow &\textstyle\sum_{\omega\in\Omega} P(\omega)\sigma'_i(\omega) = \varepsilon'_i \\ \Leftrightarrow &\textstyle\sum_{\omega\in\Omega} P(\omega)\frac{\sigma'_i(\omega)}{\mathcal{G}^\#} = \frac{\varepsilon'_i}{\mathcal{G}^\#} \\ \Leftrightarrow &\textstyle\sum_{\omega\in\Omega} P(\omega)\hat{\sigma}_i(\omega) = \hat{\varepsilon}_i \end{aligned}$$

Since $d_i \in (0,1)$, it holds $0 < d_i G_i^\# \le \sum_{j=1}^{m} G_j^\#$. So together with (17) and (24) it follows that (28) must hold:

$$\hat{\varepsilon}_i = \frac{d_i G_i^\#}{\mathcal{G}^\#} = \frac{d_i G_i^\#}{\sum_{j=1}^{m} G_j^\#} > 0$$

Finally (29) holds, since from $d_i \in (0,1)$ we get:

$$0 < \sum_{i=1}^{m} \hat{\varepsilon}_i = \sum_{i=1}^{m} \frac{d_i G_i^{\#}}{\mathcal{G}^{\#}} = \frac{\sum_{i=1}^{m} G_i^{\#} d_i}{\sum_{i=1}^{m} G_i^{\#}} < 1$$

Definition 12 (Correctional Feature Function). *Let* $\hat{\sigma}_1, \ldots, \hat{\sigma}_m$ *and* $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_m$ *be as in Definition 11. Then the additional* correctional feature function $\hat{\sigma}_{\hat{m}}$ *with* $\hat{m} := m + 1$ *is defined as*

$$\hat{\sigma}_{\hat{m}}(\omega) := 1 - \sum_{i=1}^{m} \hat{\sigma}_i(\omega) \tag{30}$$

and the corresponding additional correctional expected value $\hat{\varepsilon}_{\hat{m}}$ *is set to*

$$\hat{\varepsilon}_{\hat{m}} := 1 - \sum_{i=1}^{m} \hat{\varepsilon}_i \tag{31}$$

Proposition 7. *For the additional correctional feature function* $\hat{\sigma}_{\hat{m}}$ *and expected value* $\hat{\varepsilon}_{\hat{m}}$ *from Definition 12 it holds:*

$$0 \leq \hat{\sigma}_{\hat{m}}(\omega) \leq 1, \; \forall \omega \in \Omega \tag{32}$$

$$\sum_{i=1}^{\hat{m}} \hat{\sigma}_i(\omega) = 1, \; \forall \omega \in \Omega \tag{33}$$

$$\hat{\varepsilon}_{\hat{m}} > 0 \tag{34}$$

$$\sum_{i=1}^{\hat{m}} \hat{\varepsilon}_i = 1 \tag{35}$$

Proof. Equations (33) and (35) follow directly from the definition of $\hat{\sigma}_{\hat{m}}$ and $\hat{\varepsilon}_{\hat{m}}$, respectively. Equation (32) holds, because $0 \leq \sum_{i=1}^{m} \hat{\sigma}_i(\omega) \leq 1$ holds due to (26). Equation (34) holds due to $0 < \sum_{i=1}^{m} \hat{\varepsilon}_i < 1$ in (29).

5.3 Normalized Optimization Problem

The above definitions of normalized feature functions $\hat{\sigma}_1, \ldots, \hat{\sigma}_m$ and a correctional feature function $\hat{\sigma}_{\hat{m}}$ (and their expected values $\varepsilon_1, \ldots, \varepsilon_m$ and $\hat{\varepsilon}_{\hat{m}}$) allows us to define the following optimization problem, which represents the optimization problem OptAgg($\mathcal{R}$) in a normalized form, meeting all requirements to apply the GIS algorithm technique:

Definition 13 (Optimization Problem OptAggNorm($\mathcal{R}$)). *Let*

$$\hat{\sigma}_i(\omega) = \frac{\sigma_i(\omega) + d_i G_i^{\#}}{\mathcal{G}^{\#}}, \; \forall \omega \in \Omega, \; 1 \leq i \leq m, \quad \textit{and} \tag{36}$$

$$\hat{\varepsilon}_i = \frac{d_i G_i^{\#}}{\mathcal{G}^{\#}},\ 1 \leq i \leq m \tag{37}$$

be the normalized feature functions and their expected values constructed (according to Definition 11) from the set $\mathcal{R}$ *of the optimization problem* $\textsc{OptAgg}(\mathcal{R})$*.*

Define $\hat{m} := m + 1$ *and let*

$$\hat{\sigma}_{\hat{m}}(\omega) = 1 - \sum_{i=1}^{m} \hat{\sigma}_i(\omega),\ \forall \omega \in \Omega, \quad \textit{and} \tag{38}$$

$$\hat{\varepsilon}_{\hat{m}} = 1 - \sum_{i=1}^{m} \hat{\varepsilon}_i \tag{39}$$

be the corresponding correctional feature function and its expected value (according to Definition 12). Let P_U *be the uniform distribution. Then the optimization problem* $\textsc{OptAggNorm}(\mathcal{R})$ *is defined as:*

$$\begin{array}{ll} \textit{minimize} \ K(P, P_U) & \\ \textit{subject to} \ \sum_{\omega \in \Omega} P(\omega)\hat{\sigma}_i(\omega) & = \hat{\varepsilon}_i,\ 1 \leq i \leq \hat{m} \\ \phantom{\textit{subject to}} \ \sum_{\omega \in \Omega} P(\omega) & = 1 \\ \phantom{\textit{subject to}} \ P(\omega) & > 0,\ \forall \omega \in \Omega \end{array} \tag{40}$$

Proposition 8. *The optimization problems* $\textsc{OptAgg}(\mathcal{R})$ *and* $\textsc{OptAggNorm}(\mathcal{R})$ *have the same solution.*

Proof. Due to (27) in Proposition 6, each constraint of $\textsc{OptAgg}(\mathcal{R})$ has equivalently been transformed into a constraint of $\textsc{OptAggNorm}(\mathcal{R})$. The additionally introduced correctional constraint of $\textsc{OptAggNorm}(\mathcal{R})$ is merely a combination of all other linear constraints, therefore it does not constrain the optimization problem any further. Proposition 4 ensures that under equivalent sets of constraints, minimizing the relative entropy (with respect to the uniform distribution P_U) yields the same solution as maximizing the entropy, therefore $\textsc{OptAgg}(\mathcal{R})$ and $\textsc{OptAggNorm}(\mathcal{R})$ have the same solution.

Proposition 9. *The optimization problem* $\textsc{OptAggNorm}(\mathcal{R})$ *yields an instance of the optimization problem* $\textsc{OptGisNorm}(\hat{EQ})$*, i. e. in particular, the feature functions and expected values of* $\textsc{OptAggNorm}(\mathcal{R})$ *satisfy the corresponding requirements (12) – (15) in Definition 9.*

Proof. Requirement (12) of $\textsc{OptGisNorm}(\hat{EQ})$ is satisfied by all feature functions $\hat{\sigma}_1, \ldots, \hat{\sigma}_m, \hat{\sigma}_{\hat{m}}$ of $\textsc{OptAggNorm}(\mathcal{R})$ due to (25) in Proposition 6 and (32) in Proposition 7. Requirement (13) is assured by (33) in Proposition 7. Requirement (14) is assured for $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_m$ by (28) in Proposition 6 and for $\hat{\varepsilon}_{\hat{m}}$ by (34) in Proposition 7. Requirement (15) is assured by (35) in Proposition 7.

The following proposition is a direct consequence of Propositions 8 and 9:

Proposition 10. *For any consistent set $\mathcal{R}$ of soft conditionals, the GIS algorithm technique can directly be applied to the optimization problem* $\textsc{OptAggNorm}(\mathcal{R})$ *to compute its solution P^*. Since P^* is also the solution to the optimization problem* $\textsc{OptAgg}(\mathcal{R})$*, the computation delivers the inference operator* $\mathrm{ME}_{\odot}(\mathcal{R}) = P^*$.

5.4 GIS Algorithm for Aggregation Semantics

Based on the basic template for a GIS algorithm in [6], we present a practical GIS algorithm which computes the solution P^* of the optimization problem $\textsc{OptAggNorm}(\mathcal{R})$. So according to Proposition 10, the algorithm delivers $\mathrm{ME}_{\odot}(\mathcal{R})$ as result. The pseudo-code of the GIS algorithm for aggregation semantics is depicted in Fig. 1.

```
GIS Algorithm for Aggregation Semantics

Input: a consistent set $\mathcal{R}$ of $m$ soft probabilistic conditionals
Output: inference operator $\mathrm{ME}_{\odot}(\mathcal{R}) := P^*$
Algorithm

– Let $P_{(0)} := P_U$
– Let $\hat{\sigma}_1, \ldots, \hat{\sigma}_m, \hat{\sigma}_{\hat{m}}$ be normalized feature functions and
  $\hat{\varepsilon}_1, \ldots, \hat{\varepsilon}_m, \hat{\varepsilon}_{\hat{m}}$ be expected values constructed from $\mathcal{R}$ (according to Def. 13)
– Initialize iteration counter: $k := 0$
– Repeat until an abortion condition holds:
    • Increase iteration counter: $k := k + 1$
    • Calculate the scaling factor $\beta_{(k),i}$ of each feature function $\hat{\sigma}_i$:
        $\beta_{(k),i} := \frac{\hat{\varepsilon}_i}{\sum_{\omega \in \Omega} P_{(k-1)}(\omega)\hat{\sigma}_i(\omega)}, \quad 1 \leq i \leq \hat{m}$
    • Scale all probabilities appropriately:
        $P'_{(k)}(\omega) := P_{(k-1)}(\omega) \prod_{i=1}^{\hat{m}} \left(\beta_{(k),i}\right)^{\hat{\sigma}_i(\omega)}, \qquad \forall \omega \in \Omega$
    • Normalize the probability values:
        $P_{(k)}(\omega) := \frac{P'_{(k)}(\omega)}{\sum_{\omega \in \Omega} P'_{(k)}(\omega)}, \quad \forall \omega \in \Omega$
– Let $P^* := P_{(k)}$ denote the final distribution of the iteration.
```

Fig. 1. GIS Algorithm for Aggregation Semantics

The algorithm starts with the uniform distribution as initial distribution. In the k-th iteration step, for each feature function $\hat{\sigma}_i$ the current ratio $\beta_{(k),i}$ between its given expected value $\hat{\varepsilon}_i$ and its current expected value

$\sum_{\omega \in \Omega} P_{(k-1)}(\omega)\hat{\sigma}_i(\omega)$ under the current distribution $P_{(k-1)}$ is determined. So $\beta_{(k),i}$ is the factor required to scale $P_{(k-1)}$ appropriately so that the expected value $\hat{\varepsilon}_i$ of $\hat{\sigma}_i$ would be met exactly. Since the actual scaling of $P_{(k-1)}$ has to be performed with respect to all scaling factors $\beta_{(k),1}, \ldots, \beta_{(k),\hat{m}}$, the scaled distribution $P_{(k)}$ cannot fit all expected values immediately, but it is guaranteed by the GIS approach that a distribution iteratively computed that way converges to the correct solution.

Note that the constraint $\sum_{\omega \in \Omega} P(\omega) = 1$, which is contained in each of the above optimization problems, is not explicitly encoded as a constraint in the GIS algorithm in Fig. 1. Instead, the scaled probability values $P^{'}_{(k)}(\omega)$ are normalized in each iteration step, so that $P_{(k)}$ is a proper probability distribution (which is important to determine the correct $\beta_{(k+1),i}$ with respect to $P_{(k)}$).

The GIS algorithm iteratively calculates a sequence of distributions which converges to the solution of the optimization problem. So in practice, an abortion condition must be defined which allows to stop the iteration if the solution has been approximated with a sufficient accuracy. A practical abortion condition is, e. g. to stop after iteration step k if $|1-\beta_{(k),i}| < \delta_\beta$ holds for all $1 \leq i \leq \hat{m}$, with δ_β being an appropriate accuracy threshold, i. e. if there is no more need to scale any values because all scaling factors are almost 1 within accuracy δ_β. Alternatively, the iteration could stop after step k if $|P_{(k)}(r_i) - d_i| < \delta_r, 1 \leq i \leq m$ holds (with $P_{(k)}(r_i)$ denoting the probability of conditional r_i under distribution $P_{(k)}$), i. e. if the probability of each conditional under the current distribution $P_{(k)}$ matches its prescribed probability d_i within accuracy δ_r.

6 Implementation

The GIS algorithm for aggregation semantics has successfully been implemented as a plugin for the KReator system. KReator [2] is an integrated development environment for representing, reasoning, and learning with relational probabilistic knowledge which aims at providing a versatile toolbox for researchers and knowledge engineers in the field of statistical relational learning. [1] Since KReator features a flexible plugin architecture for knowledge representation formalisms, the implementation of a new plugin for knowledge bases with aggregation semantics was quite easy. Implementing aggregation semantics as a KReator plugin saved a lot of time compared to the from scratch development of a stand-alone implementation, because KReator offers direct support for central concepts like e. g. knowledge bases, first order logic, and model-based inference. Since KReator is written in Java, the aggregation semantics plugin and its core component, the GIS algorithm that we developed, have been implemented in Java as well. Our first implementation of the GIS algorithm is a straight-forward implementation of the pseudo-code from Fig. 1, with further optimizations being referred to further refinements.

[1] The development of KReator is part of the KReate project, cf. `www.fernuni-hagen.de/wbs/research/kreate/`

The following example will be used to examine the runtime behavior of the implementation:

Example 2. Suppose we have a zoo with a population of monkeys. The predicate $\mathit{Feeds}(X, Y)$ expresses that a monkey X feeds another monkey Y and $\mathit{Hungry}(X)$ says that a monkey X is hungry. The knowledge base $\mathcal{R}_{\mathrm{mky}}$ contains conditionals which express generic knowledge as well as one conditional stating exceptional knowledge about a monkey *Charly*:

$$
\begin{aligned}
r_1 &: (\mathit{Feeds}(X, Y) \mid \neg \mathit{Hungry}(X) \wedge \mathit{Hungry}(Y))\ [0.80] \\
r_2 &: (\mathit{Feeds}(X, Y) \mid \mathit{Hungry}(X))\ [0.001] \\
r_3 &: (\mathit{Feeds}(X, Y) \mid \neg \mathit{Hungry}(X) \wedge \neg \mathit{Hungry}(Y))\ [0.10] \\
r_4 &: (\mathit{Feeds}(X, \mathit{charly}) \mid \neg \mathit{Hungry}(X))\ [0.95] \\
r_5 &: (\mathit{Feeds}(X, X) \mid \top)\ [0.001]
\end{aligned}
$$

Considering the above example together with set $\mathit{Const} = \{\mathit{andy}, \mathit{bobby}, \mathit{charly}\}$ of constants, the corresponding Herbrand base contains 12 ground atoms. Therefore $|\Omega| = 2^{12} = 4,096$ elementary probabilities have to be computed in every iteration step of the algorithm. Using the values of all scaling factors as abortion condition (as suggested in Sec. 5.4) with an accuracy threshold of $\delta_\beta = 0.001$, the GIS algorithm requires 20,303 steps to compute a solution with sufficient accuracy. On a computer with an Intel Core i5-2500K CPU (4 cores, 3.3 Ghz) the computation of the ME-distribution P^* takes 14 seconds. To additionally check the accuracy of the calculated distribution P^*, the probabilities of the conditionals from $\mathcal{R}_{\mathrm{mky}}$ have been recalculated under P^*. Comparing these probabilities with the prescribed probabilities of the conditionals reveals an deviation of $\delta_r = 0.0017$ at most. Performing the same computation with an improved accuracy of $\delta_\beta = 0.0001$ results in 81,726 iteration steps taking 59 seconds and revealing an improved deviation of $\delta_r = 0.00017$ as well. Once the distribution P^* has been computed, it can be (re-)used for probabilistic inference. That way, arbitrary queries like e. g. $q := (\mathit{Feeds}(\mathit{andy}, \mathit{bobby}) \mid \mathit{Hungry}(\mathit{charly}))$ can be addressed to the knowledge base and the answer (i. e. the queries's probability under P^*) is determined immediately, e. g. $P^*(q) = 0.21$.

7 Conclusion and Future Work

In this paper, we investigated the aggregation semantics for first-order probabilistic logic and its ME-based inference operator $\mathrm{ME}_\odot$. We illustrated how the convex optimization problem induced by $\mathrm{ME}_\odot$ can be expressed in terms of feature functions.

We developed an approach allowing us to use a GIS algorithm technique for solving this optimization problem and presented the pseudo-code of a concrete algorithm which employs GIS to calculate $\mathrm{ME}_\odot(\mathcal{R})$, i. e. the ME-optimal probability distribution which satisfies all conditionals in $\mathcal{R}$. A realization of this algorithm has been integrated in a plugin for aggregation semantics in the KReator system, being the first practical implementation of $\mathrm{ME}_{\odot}$ inference.

The application of the algorithm presented in this paper to a consistent set of probabilistic conditionals requires that all conditional have a soft probability to assure that they can be satisfied by a positive probability distribution. In [5], it is noted this requirement of the original GIS algorithm is not always necessary. We will further investigate this topic to clarify if the requirement of soft probabilities can be relaxed under certain circumstances.

In this paper, we focused on the development of an algorithm which is capable of calculating $\mathrm{ME}_{\odot}\ (\mathcal{R})$ in principle, so we did not discuss any performance issues. The size of Ω is exponential in the number of ground atoms, which directly depends on the number of constants and predicates. Therefore, any algorithm working on a complete representation of a distribution $P \in \mathcal{P}_{\Omega}$ has exponential running-time concerning the size of the input, i. e. concerning conditionals over *Const* and *Pred*. But for some smaller examples, working on a complete representation of P can still be feasible. It is also shown in [6], that the solution to an optimization problem of the form (16) can always be presented in product form, i. e. there exist $\hat{\alpha}_i \in \mathbb{R}$ so that $P^*(\omega) = \hat{\alpha}_0 \prod_{i=1}^{\hat{m}} \hat{\alpha}_i^{\hat{\sigma}_i(\omega)}$ holds. These α_i values can directly be calculated within the iteration of GIS algorithm, therefore a compact representation of P^* is available.

In future work, we will study to what extend it is possible to work on decomposed distributions by employing junction trees and sophisticated propagation techniques, as proposed in [1, 19]. Similar techniques have already been successfully adopted in [14] to an ME-inference operator for propositional languages and have been implemented in the expert system Spirit [18]. As shown in [12], this propositional ME-inference operator is just a special case of the $\mathrm{ME}_{\odot}$ inference operator, because the operators coincide for ground conditionals. We also plan on investigating the application of the *Improved Iterative Scaling* (IIS) algorithm from [3].

References

1. Badsberg, J.H., Malvestuto, F.M.: An implementation of the iterative proportional fitting procedure by propagation trees. Computational Statistics & Data Analysis 37(3), 297–322 (September 2001)
2. Beierle, C., Finthammer, M., Kern-Isberner, G., Thimm, M.: Automated reasoning for relational probabilistic knowledge representation. In: Giesl, J., Hähnle, R. (eds.) Automated Reasoning: Fifth International Joint Conference (IJCAR'10). No. 6173 in Lecture Notes in Computer Science (July 2010)
3. Berger, A.L., Della Pietra, S., Della Pietra, V.J.: A maximum entropy approach to natural language processing. Computational Linguistics 22(1), 39–71 (1996)
4. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York, NY, USA (2004)
5. Csiszar, I.: A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling. Annals of Statistics 17(3), 1409–1413 (1989)

6. Darroch, J.N., Ratcliff, D.: Generalized iterative scaling for log-linear models. In: Annals of Mathematical Statistics, vol. 43, pp. 1470–1480. Institute of Mathematical Statistics (1972)
7. Delgrande, J.: On first-order conditional logics. Artificial Intelligence 105, 105–137 (1998)
8. Domingos, P., Lowd, D.: Markov Logic: An Interface Layer for Artificial Intelligence. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers (2009)
9. Fisseler, J.: Learning and Modeling with Probabilistic Conditional Logic, Dissertations in Artificial Intelligence, vol. 328. IOS Press, Amsterdam (2010)
10. Kern-Isberner, G.: Conditionals in nonmonotonic reasoning and belief revision. Springer, Lecture Notes in Artificial Intelligence LNAI 2087 (2001)
11. Kern-Isberner, G., Lukasiewicz, T.: Combining probabilistic logic programming with the power of maximum entropy. Artificial Intelligence, Special Issue on Nonmonotonic Reasoning 157(1-2), 139–202 (2004)
12. Kern-Isberner, G., Thimm, M.: Novel semantical approaches to relational probabilistic conditionals. In: Lin, F., Sattler, U., Truszczyński, M. (eds.) Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR'10). pp. 382–392. AAAI Press (May 2010)
13. Kersting, K., De Raedt, L.: Bayesian Logic Programming: Theory and Tool. In: Getoor, L., Taskar, B. (eds.) An Introduction to Statistical Relational Learning. MIT Press (2007)
14. Meyer, C.H.: Korrektes Schließen bei unvollständiger Information. Ph.D. thesis, FernUniversität Hagen (1998)
15. Paris, J.: The uncertain reasoner's companion – A mathematical perspective. Cambridge University Press (1994)
16. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1998)
17. Rödder, W.: Conditional logic and the principle of entropy. Artificial Intelligence 117, 83–106 (2000)
18. Rödder, W., Meyer, C.H.: Coherent Knowledge Processing at Maximum Entropy by SPIRIT. In: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence (UAI 1996). pp. 470–476 (1996)
19. Teh, Y., Welling, M.: On improving the efficiency of the iterative proportional fitting procedure. In: Proc. of the 9th Int'l. Workshop on AI and Statistics (AISTATS-03) (2003)
20. Thimm, M.: Probabilistic Reasoning with Incomplete and Inconsistent Beliefs. Ph.D. thesis, Technische Universität Dortmund (2011)

Verzeichnis der zuletzt erschienenen Informatik-Bericht

[346] vor der Brück, T.:
Application of Machine Learning Algorithms for Automatic Knowledge Acquisition and Readability Analysis Technical Report

[347] Fechner, B.:
Dynamische Fehlererkennungs- und –behebungsmechanismen für verlässliche Mikroprozessoren

[348] Brattka, V., Dillhage, R., Grubba, T., Klutsch, A.:
CCA 2008 - Fifth International Conference on Computability and Complexity in Analysis

[349] Osterloh, A.:
A Lower Bound for Oblivious Dimensional Routing

[350] Osterloh, A., Keller, J.:
Das GCA-Modell im Vergleich zum PRAM-Modell

[351] Fechner, B.:
GPUs for Dependability

[352] Güting, R. H., Behr, T., Xu, J.:
Efficient *k*-Nearest Neighbor Search on Moving Object Trajectories

[353] Bauer, A., Dillhage, R., Hertling, P., Ko K.I., Rettinger, R.:
CCA 2009 Sixth International Conference on Computability and Complexity in Analysis

[354] Beierle, C., Kern-Isberner, G.
Relational Approaches to Knowledge Representation and Learning

[355] Sakr, M.A., Güting, R.H.
Spatiotemporal Pattern Queries

[356] Güting, R. H., Behr, T., Düntgen, C.:
SECONDO: A Platform for Moving Objects Database Research and for Publishing and Integrating Research Implementations

[357] Düntgen, C., Behr, T., Güting, R.H.:
Assessing Representations for Moving Object Histories

[358] Sakr, M.A., Güting, R.H.
Group Spatiotemporal Pattern Queries

[359] Hartrumpf, S., Helbig, H., vor der Brück, T. , Eichhorn, C.
SemDupl: Semantic Based Duplicate Identification

[360] Xu, J., Güting, R.H.
A Generic Data Model for Moving Objects

[361] Beierle, C., Kern-Isberner, G.
Evolving Knowledge in Theory and Application: 3rd Workshop on Dynamics of Knowledge and Belief, DKB 2011

[362] Xu, J., Güting, R.H.:
GMOBench: A Benchmark for Generic Moving Objects