# Learning to Assign Degrees of Belief in Relational Domains

Frédéric Koriche

LIRMM, Université Montpellier II
161 Rue Ada, 34392 Montpellier Cedex 5, France
Frederic.Koriche@lirmm.fr

**Abstract.** A recurrent question in the design of intelligent agents is how to assign degrees of beliefs, or subjective probabilities, to various events in a relational environment. In the standard knowledge representation approach, these probabilities are evaluated according to a knowledge base, such as a logical program or a Bayesian network. However, even for very restricted representation languages, the problem of evaluating probabilities from a knowledge base is computationally prohibitive. By contrast, this study embarks on the learning to reason (L2R) framework that aims at eliciting degrees of belief in an inductive manner. The agent is viewed as an anytime reasoner that iteratively improves its performance in light of the knowledge induced from its mistakes. By coupling exponentiated gradient strategies in online learning and weighted model counting techniques in reasoning, the L2R framework is shown to provide efficient solutions to relational probabilistic reasoning problems that are provably intractable in the classical framework.

## 1 Introduction

As uncertainty pervades the real world, it seems obvious that the decisions we make, the conclusions we reach, and the explanations we offer are usually based on our judgements of the probability of uncertain events such as success in a new medical treatment or the state of the market. For example, if an agent wishes to employ the expected-utility paradigm of decision theory in order to guide its actions, it must assign subjective probabilities to various assertions. Less obvious, however, is the question of *how* to elicit such degrees of beliefs.

The standard knowledge representation approach claims that the agent starts its life-cycle by acquiring a pool of knowledge expressing several constraints about its environment, such as properties of objects and relationships among them. This information is stored in some knowledge base using a logical representation language [1, 15, 16, 19] or a graphical representation language [8, 18]. After this period of knowledge preparation, the agent is expected to achieve optimal performance by evaluating any query with perfect accuracy. Indeed, according to the well-defined semantics of the representation language, a knowledge base provides a compact representation of a probability measure that can be used to evaluate queries. For example, if we select first-order logic as our representation language, the probability measure is induced by assigning equal likelihood to all models of the knowledge base; the degree of belief of any given query is thus the fraction of those models which are consistent with the query.

From a pragmatic perspective, the usefulness of a computational framework for assigning subjective probabilities depends both on the *accuracy* of the belief estimates and the *efficiency* of belief estimation. Unfortunately, in the standard knowledge representation approach, the task of assigning subjective probabilities can very much demanding from a computational point of view. In propositional logic, the problem of inferring the probability of any query from a knowledge base is complete for the class #P, and even the apparently easier question of approximating this probability in a very weak sense is NP-hard [20]. The problem is still more acute in the relational setting. Indeed, even if any function-free first-order theory defined over a finite domain can be transformed into a logically equivalent ground formula, the size of the resulting formula can grow exponentially with respect to the initial theory. As a consequence, relational probabilistic reasoning turns out to be #EXP-hard to evaluate and NEXP-hard to approximate. Similar results have been obtained for relational Bayesian networks [9].

In contrast, the *learning to reason* (L2R) framework has recently emerged as an active research field of ILP for dealing with the intractability of reasoning problems [10, 11, 23]. By incorporating a role of inductive learning within reasoning, this approach stresses the importance of combining the processes of knowledge acquisition and query evaluation together. The main departure from the classical approach is that knowledge is not ascribed *a priori*, in the purpose of describing an environment, but instead acquired *a posteriori*, by experience, in order to improve the agent's ability to reason efficiently in its environment.

Following the L2R paradigm, this study aims at eliciting degrees of beliefs in an inductive manner, using a computational model of learning. Namely, the world, or the domain in question, is modeled as a probability distribution $W$ on a space of relational interpretations. The reasoning agent starts its life-cycle with a simple set of ground atoms over some relational vocabulary, referred to the *background knowledge* [7]. To acquire additional knowledge from the world, the agent is given a "grace period" in which it can interact with its learning interface. The purpose of the learning interface is to help the agent in concentrating its effort toward finding a representation $KB$ of $W$ that is useful for evaluating queries in some target query language $\mathcal{Q}$. The reasoning performance is measured only after this period, when the agent is presented with new queries from $\mathcal{Q}$ and has to estimate their probability according to its representation $KB$. Thus, by contrast with the standard knowledge representation approach, the agent is not required to achieve optimal performance by evaluating *any* possible query with perfect precision. Instead, the performance is measured with regard to a restricted though expressive query language $\mathcal{Q}$.

Technically, our framework is based on the online mistake-driven learning model introduced by Littlestone [14]. In this setting, the L2R protocol is modeled as a repeated game between the reasoning agent and its learning interface. During each trial of the game, the agent receives a query $Q$ from $\mathcal{Q}$ and assigns a degree of belief $\mathbf{Pr}_{KB}(Q)$ to it. The agent is charged a mistake only if its prediction loss is not judged satisfactory for the task at hand. In this case, the agent is supplied the correct probability $\mathbf{Pr}_W(Q)$ and updates its knowledge base in light of this feedback information. In essence, the agent is an *anytime reasoner* which gradually improves its performance by interacting with its learning interface.

In the L2R framework, the requirements for efficient relational probabilistic reasoning are twofold. First, the length of the grace period needed to achieve full functionality must be polynomial in the size of the background knowledge. In other words, the agent's behavior must converge to yield accurate estimations after a polynomial number of interactions. Second, the computational cost needed to evaluate the degree of belief of any query from the language $\mathcal{Q}$ must also be polynomial in the cardinality of the background knowledge.

To satisfy these requirements, we develop an online L2R algorithm which combines techniques in regression learning and weighted model counting. The algorithm uses an exponentiated gradient strategy [3, 12] adapted for assigning probabilities to relational queries. The worst-case total number of mistakes made by the reasoner depends only *logarithmically* in the size of the target probability distribution, and hence *linearly* in the size of the background knowledge. Based on this property, the learning curve of the reasoner is guaranteed to converge to yield accurate estimations after a polynomial number of interactions.

The key idea behind efficient query evaluation lies in a representation of the "mistake-driven" knowledge that allows tractable forms of weighted model counting [21]. Namely, for various fragments of the so-called relational language $\mathcal{R}$ proposed by Cumby and Roth [4], the computational cost of assigning degrees of belief is polynomial in the number of mistakes made so far, and hence, the size of the background knowledge. This result highlights the interest of the L2R framework by providing efficient solutions to relational probabilistic reasoning problems that are provably intractable in the classical framework.

*Outline.* After introducing the learning to reason framework in section 2, we present the exponentiated gradient L2R algorithm in section 3. Tractable query languages are discussed in section 4. Finally, section 5 compares our framework with related work, and concludes with some perspectives of further research. For sake of clarity, proofs of technical results are given in the Appendix.

## 2 The Framework

We consider problems of reasoning where the "world" is modeled as a probability distribution $W$ on the space of interpretations defined over some background knowledge $\mathcal{B}$. To this end, we assume a prefixed relational vocabulary, which consists in a finite set $R = \{\mathsf{r}_1, \cdots, \mathsf{r}_r\}$ of relation symbols and a finite set $C = \{\mathsf{c}_1, \cdots, \mathsf{c}_n\}$ of constants. As usual, a term is a variable or a constant, and an atom is a relation symbol followed by a bracketed $k$-tuple of terms, where $k$ is the arity of the relation. A ground atom is an atom with no occurrence of any variable. Based on these notions, the *background knowledge* is formalized by a set $\mathcal{B}$ of ground atoms generated from the relational vocabulary. Note that the cardinality of $\mathcal{B}$ is upper bounded by $rn^a$.

Given a first-order formula $Q$ defined over the vocabulary, the *size* of $Q$, denoted $|Q|$, is the number of occurrences of atoms in its description.

A relational interpretation $I$ over $\mathcal{B}$ is a subset of $\mathcal{B}$ with the underlying meaning that any ground atom $A \in \mathcal{B}$ is true in $I$ if $A \in I$, and false in $I$ if $A \notin I$. The space $\wp(\mathcal{B})$ of all possible interpretations generated from $\mathcal{B}$ is represented by an indexed set $\{I_1, \cdots, I_N\}$ where $N = 2^{|\mathcal{B}|}$. Given an interpretation $I$ and a closed formula $Q$ defined over the relational vocabulary, we say that $I$ is a *model* of $Q$, if $Q$ is true in $I$ according to the standard semantic rules. The *projection* of $Q$ onto the space $\wp(\mathcal{B})$ is a tuple $\mathcal{I}(Q) = (I_1(Q), \cdots, I_N(Q))$, where $I_i(Q) = 1$ if $I_i$ is a model of $Q$, and $I_i(Q) = 0$ otherwise.

*Example 1.* Our running example is a simple variant of the logistic domain [25]. The world includes two trucks $\mathsf{t}_1, \mathsf{t}_2$, four objects $\mathsf{o}_1, \cdots, \mathsf{o}_4$ and two cities $\mathsf{c}_1, \mathsf{c}_2$. We are given the predicate $\mathsf{At}(x, y)$ indicating the location of objects and vehicles, and the predicate $\mathsf{In}(x, y)$ for indicating that some objects are in some vehicle. The background knowledge thus consists in the 12 ground atoms $\{\mathsf{At}(\mathsf{t}_1, \mathsf{c}_1), \cdots, \mathsf{At}(\mathsf{o}_4, \mathsf{c}_2)\}$ and the 8 ground atoms $\{\mathsf{In}(\mathsf{o}_1, \mathsf{t}_1), \cdots, \mathsf{In}(\mathsf{o}_4, \mathsf{t}_2)\}$.

Any probability distribution $W$ over $\wp(\mathcal{B})$ is specified by a tuple $(w_1, \cdots, w_N)$ such that $w_i \in [0, 1]$ and $\sum_{i=1}^N w_i = 1$. The value $w_i$ is the probability of $I_i$ according to $W$. The probability of a formula $Q$ under $W$ is given by

$$\mathbf{Pr}_W(Q) = \sum_{i=1}^{N} w_i I_i(Q)$$

**Definition 1.** *A* belief assignment problem *with background knowledge $\mathcal{B}$ is a pair $(W, \mathcal{Q})$, where $W$ is a probability distribution on $\wp(\mathcal{B})$, and $\mathcal{Q}$ is a countable set of formulas over the relational vocabulary of $\mathcal{B}$. The distribution $W$ is called the* world *and the set $\mathcal{Q}$ is called the* target query language.

*Example 2.* Suppose that in the domain $W$ the location of the trucks and the objects are not known, but all objects are in the trucks. We can thus derive the following probabilities: $\mathbf{Pr}_W(\forall x\, \mathsf{In}(x, \mathsf{t}_1)) = \frac{1}{16}$, $\mathbf{Pr}_W(\exists x\, \mathsf{At}(\mathsf{t}_1, x)) = \frac{2}{3}$, $\mathbf{Pr}_W(\exists x\, \mathsf{At}(\mathsf{t}_1, x) \wedge \forall x\, \mathsf{In}(x, \mathsf{t}_1)) = \frac{1}{24}$ and $\mathbf{Pr}_W(\exists x\, \mathsf{At}(\mathsf{t}_1, x) \vee \exists y\, \mathsf{At}(\mathsf{t}_2, y)) = \frac{8}{9}$.

In the L2R framework, the goal of the reasoning agent is to acquire a representation $KB$ of the world $W$ for the target language $\mathcal{Q}$. In doing so, the agent is given access to a learning interface that governs the occurrences of queries drawn from $\mathcal{Q}$. The interface most appropriate in our setting is a variant of the reasoning query oracle [11] adapted for belief estimation. To assess regression discrepancies, we use the typical quadratic loss function $L(x, y) = (x - y)^2$.

**Definition 2.** *A* reasoning query oracle *for the problem $(W, \mathcal{Q})$, with respect to a tolerance parameter $\epsilon \in (0, 1]$, denoted* $\mathrm{RQ}_\epsilon(W, \mathcal{Q})$*, is an oracle that when accessed performs the following protocol with the agent $A$.* (1) *The oracle picks an arbitrary query $Q \in \mathcal{Q}$ and returns it to $A$.* (2) *The agent $A$ assigns a degree of belief $\mathbf{Pr}_{KB}(Q)$ to $Q$ according to its knowledge base $KB$.* (3) *The oracle responds by* "correct" *if $L(\mathbf{Pr}_{KB}(Q), \mathbf{Pr}_W(Q)) \leq \epsilon$, and* "incorrect" *otherwise. In case of mistake, the oracle supplies the correct probability $\mathbf{Pr}_W(Q)$ to $A$.*

The interaction protocol is modeled as a repeated game between the agent and its interface. During each trial $t$, the agent receives a query $Q_t$ supplied by the oracle, makes a prediction according to its knowledge base $KB_t$ and, in case of mistake, updates $KB_t$ in light of the correct response. The *mistake bound* for a reasoning agent $A$ on the problem $(W, \mathcal{Q})$, denoted $\mathrm{M}_{A,\epsilon}(W, \mathcal{Q})$, is the maximum number of mistakes that $A$ can make by interacting with $\mathrm{RQ}_\epsilon(W, \mathcal{Q})$ over every arbitrary infinite sequence of queries supplied by its oracle.

**Definition 3.** *An algorithm $A$ is an* efficient mistake bound L2R *algorithm for the problem $(W, \mathcal{Q})$ with background knowledge $\mathcal{B}$, if there are polynomials $p$ and $q$ such that the following conditions hold any $\epsilon \in (0, 1]$. (1) $\mathrm{M}_{A,\epsilon}(W, \mathcal{Q}) \leq p(|\mathcal{B}|, \frac{1}{\epsilon})$, (2) $A$ evaluates any query $Q \in \mathcal{Q}$ in $q(|Q|, |\mathcal{B}|, \frac{1}{\epsilon})$ time.*

The online learning to reason framework provides a natural way to make explicit the dependence of the reasoning performance on the knowledge acquired from the environment. After a "grace period" of interaction between the agent and its oracle, the agent is expected to predict the degree of belief of any query supplied by its interface without the help of the feedback response. Of course, we cannot force the agent to make all the mistakes within the grace period. However, using probabilistic tools developed in [6], the asymptotic behavior of the reasoner can be derived from its mistake bound. In particular, for any mistake bound L2R algorithm, the length of the grace period required to converge toward the desired reasoning behavior is polynomial in the input dimension.

**Observation 1.** *Let $A$ be an efficient mistake bound L2R algorithm for the problem $(W, \mathcal{Q})$. Then for any probability distribution $D$ over the target query language $\mathcal{Q}$ and any $\delta, \epsilon \in (0, 1]$, with probability $1 - \delta$ the algorithm $A$ will make a correct prediction at any trial $t \geq \frac{1}{\delta} \mathrm{M}_{A,\epsilon}(W, \mathcal{Q})$, provided that every query supplied by $\mathrm{RQ}_\epsilon(W, \mathcal{Q})$ is drawn independently at random according to $D$.*

## 3 Exponentiated Gradient Learning to Reason

The L2R algorithm suggested in this study aims at combining exponentiated gradient strategies in online learning and weighted model counting techniques in probabilistic reasoning. Namely, the backbone of the algorithm is formed by "implicitly" maintaining a probability distribution $\hat{W}$ over the space $\mathcal{I}$ of relational interpretations that approximates the world $W$. In the spirit of online exponentiated gradient learning algorithms [3, 12], the reasoning agent predicts with its hypothesis $\hat{W}$ and, in case of mistake, adjusts the probabilities in $\hat{W}$ according to a multiplicative weight update rule.

In relational probabilistic reasoning, a direct implementation of $\hat{W}$ is clearly infeasible, as the agent would need to maintain $2^{|\mathcal{B}|}$ distinct probabilities. So, in order to alleviate this issue, the key idea behind the algorithm is to encode the probability distribution $\hat{W}$ into a weighted knowledge base $KB$, whose size is polynomial in the number of mistakes made so far. In this representation, the prediction task is thus translated into a weighted model counting problem [21].

**Input:**

background knowledge $\mathcal{B}$

learning rate $\eta > 0$

**Initialization:**

set $KB_1 = \varnothing$

**Trials:** in each trial $t \geq 1$

receive a query $Q_t$

predict $\hat{y}_t = \mathbf{Pr}_{KB_t}(Q_t)$

if the prediction is *correct* then

$KB_{t+1} = KB_t$

else

receive $y_t = \mathbf{Pr}_W(Q_t)$

set $KB_{t+1} = KB_t \wedge (Q_t \leftrightarrow \mathsf{q}_t)$ where $\rho(\mathsf{q}_t) = e^{\eta(y_t - \hat{y}_t)}$

**Algorithm 1**: The EG-L2R algorithm

To this end, we shall assume that our relational vocabulary is extended with a countable set $\{\mathsf{q}_1, \mathsf{q}_2, \ldots\}$ of *weighted* relations of arity zero. Any ground atom $A$ defined over the extended vocabulary is assigned a weight $\rho(A)$ with the condition that $\rho(A) = 1$ whenever the predicate of $A$ is a standard relation in $\{\mathsf{r}_1, \cdots, \mathsf{r}_r\}$. An *extended interpretation* is a set of ground atoms defined over the extended vocabulary. The *weight* of an extended interpretation $I$, denoted $\rho(I)$, is the product of weights of all ground atoms occurring in $I$. Given a formula $KB$ defined over the extended vocabulary, an extended interpretation $I$ is a *minimal model* of $KB$ if 1) $KB$ is true in $I$ and 2) $KB$ is false in each proper subset $I'$ of $I$ obtained by removing from it any weighted relation $\mathsf{q}_i$. The *weight* of $KB$, denoted $\rho(KB)$, is the sum of weights of the extended interpretations that are minimal models of $KB$. Based on these notions, the *degree of belief* in a query $Q$ according to a weighted formula $KB$ is given by

$$\mathbf{Pr}_{KB}(Q) = \frac{\rho(KB \wedge Q)}{\rho(KB)}$$

We are now in position to examine the algorithm. During each trial $t$, if the agent makes a mistake on a query $Q_t$, then $KB_t$ is expanded with a rule $Q_t \leftrightarrow \mathsf{q}_t$ which indicates that every minimal model of $Q_t$ must satisfy the goal $\mathsf{q}_t$ and conversely. The update value is memorized by appropriately adjusting the weight of $\mathsf{q}_t$.

The *entropy* of a probability distribution $W$ over the space $\wp(\mathcal{B})$ is given by $H(W) = \sum_{i=1}^{N} w_i \log \frac{1}{w_i}$. Based on this measure, the reasoning performance of the exponentiated gradient L2R algorithm is captured by the following theorem.

**Theorem 1.** *For any belief assignment problem* $(W, \mathcal{Q})$ *with background knowledge* $\mathcal{B}$*, on input* $\epsilon > 0$*, when* $\eta = 4$*, the exponentiated gradient L2R algorithm has the following mistake bound*

$$\mathrm{M}_\epsilon(W, \mathcal{Q}) \leq \frac{1}{2\epsilon}\big(|\mathcal{B}| - H(W)\big)$$

In a nutshell, the exponentiated gradient L2R algorithm is characterized by two important features. First, the mistake bound is *logarithmic* in the number of possible interpretations, and hence, *linear* in the size of the background knowledge. Interestingly, since $H(W)$ is always nonnegative, the reasoning performance of the algorithm improves with the entropy of its environment. Second, the size of the hypothesis $KB$ maintained by the reasoner is also linear in the number of ground atoms. This follows from the fact the algorithm is conservative and updates its knowledge base only if it makes a mistake.

## 4 Tractable Query Languages

This section highlights the utility of the framework by demonstrating that belief assignment problems defined over various fragments of the relational language $\mathcal{R}$ developed by Cumby and Roth [4] are "mistake bound learnable". In the following, the cardinality of the background knowledge $\mathcal{B}$ is denoted $b$, and the largest size of any query in the target language is denoted $l$.

Formulas in $\mathcal{R}$ are defined to be restricted relational expressions in which there is only a single predicate in the scope of each variable. A *quantified atom* is an atomic expression where each variable occurs in the scope of a quantifier $\forall$ or $\exists$. A *quantified literal* is a quantified atom or its negation. A *quantified conjunctive (disjunctive) query* is a conjunction (disjunction) of quantified literals.

To obtain tractable forms of model counting, we need to introduce an additional restriction on the language. Given a quantified expression $Q$, let $\mathcal{B}(Q)$ be the set of ground atoms defined as follows: $A \in \mathcal{B}(Q)$ if there is a ground substitution $\theta$ such that $A \in \mathcal{B} \cap Q\theta$. Then, two expressions $Q$ and $Q'$ are called *independent* if $\mathcal{B}(Q) \cap \mathcal{B}(Q') = \varnothing$. A *decomposable conjunctive (disjunctive) query* is a conjunction (disjunction) of pairwise independent quantified literals.

**Theorem 2.** *For any decomposable conjunctive (disjunctive) query $Q$, the problem of counting the number of models of $Q$ can be evaluated in $O(|\mathcal{B}||Q|)$ time.*

*Example 3.* Let us return to our logistic domain. Initially, the knowledge base $KB$ of the agent is empty, so $\rho(KB) = 2^{20}$. Now, suppose that the agent is given the decomposable conjunctive query $Q = \forall x \, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y \, \mathsf{At}(\mathsf{t}_1, y)$. Let $\overline{\mathcal{B}}(Q)$ be the set of all possible ground atoms that do not occur in $\mathcal{B}(Q)$. The number of models of $Q$ can be determined using a simple decomposition method that factorizes the number of models of each component of $Q$. Specifically, we have $\rho(Q) = \rho(\forall x \, \mathsf{In}(x, \mathsf{t}_1)) \cdot \rho(\exists y \, \mathsf{At}(\mathsf{t}_1, y)) \cdot 2^{|\overline{\mathcal{B}}(Q)|} = 1 \cdot 3 \cdot 2^{14} = 3 \cdot 2^{14}$. It follows that the degree of belief $\mathbf{Pr}_{KB}(Q)$ in the query $Q$ according to $KB$ is $\frac{3}{64}$.

### 4.1 Hitting Query Languages

In propositional logic, an important class of formulas for which model counting can be determined in polynomial time is the class of *hitting formulas* [2]. Based on the notion of decomposable queries, we extend this class to the relational setting and show that any belief assignment problem defined over a hitting query language is mistake-bound learnable.

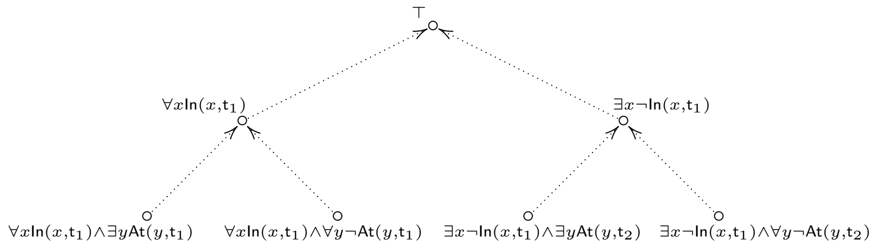


**Fig. 1.** A hitting query language

Given two queries $Q_1$ and $Q_2$, we say that $Q_1$ *entails* $Q_2$, denoted $Q_1 \models Q_2$, if and only if for every interpretation $I \in \wp(\mathcal{B})$, if $I$ is a model $Q_1$ then $I$ is a model of $Q_2$. Two queries $Q_1$ and $Q_2$ are called *comparable* if either $Q_1 \models Q_2$ or $Q_2 \models Q_1$. Moreover, two queries $Q_1$ and $Q_2$ are called *mutually unsatisfiable* if there is no interpretation in $\wp(\mathcal{B})$ satisfying $Q_1 \wedge Q_2$.

**Definition 4.** *A* hitting *conjunctive (disjunctive) query language is a set $\mathcal{Q}$ of decomposable conjunctive (disjunctive) queries such that for any $Q_1, Q_2 \in \mathcal{Q}$, $Q_1$ and $Q_2$ are either comparable or mutually unsatisfiable.*

*Example 4.* Interestingly, any hitting query language that does not contain a pair of mutually equivalent queries can be represented by a directed tree, where each vertex $Q_i$ is a query and each edge $(Q_i, Q_j)$ indicates that $Q_1$ entails $Q_j$. The tree must satisfy the property that any pair $(Q_1, Q_j)$ of vertices that are not joined by a path forms a pair of mutually unsatisfiable queries. Consider for example the tree $\mathcal{Q}$ illustrated in figure 1. Then the corresponding set of vertices forms a hitting conjunctive query language.

**Theorem 3.** *Let $\mathcal{Q}$ be a hitting conjunctive (disjunctive) query language, and $KB = \bigwedge_{i=1}^{m}(Q_i \leftrightarrow \mathsf{q}_i)$ be a weighted knowledge base such that $Q_i \in \mathcal{Q}$ for $1 \leq i \leq m$. Then for any $Q$ in $\mathcal{Q}$, $\mathbf{Pr}_{KB}(Q)$ can be evaluated in $O(bl^2m^2)$ time.*

By theorem 1, we know that the worst-case number $m$ of mistakes of the EG-L2R algorithm is linear in the size of the background knowledge. Thus, by coupling this property with theorem 3, we can derive the following result.

**Corollary 1.** *There exists an efficient mistake-bound L2R algorithm for any belief assignment problem $(W, \mathcal{Q})$ with background knowledge $\mathcal{B}$ where $\mathcal{Q}$ is a hitting conjunctive (disjunctive) query language.*

### 4.2 Cluster Query Languages

The expressiveness of tractable query languages can be further increased by forming clusters of hitting languages. Intuitively, a cluster formula is a union of independent hitting formulas [17]. This notion can be ported to the relational setting in the following way.

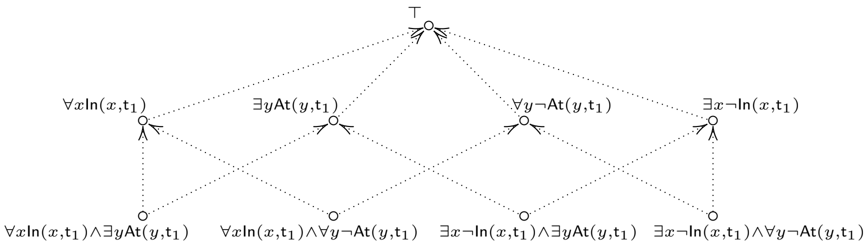


**Fig. 2.** A cluster query language

**Definition 5.** *A* cluster *conjunctive (disjunctive) query language is a set $\mathcal{Q}$ of decomposable conjunctive (disjunctive) queries such that for any $Q_1, Q_2 \in \mathcal{Q}$, $Q_1$ and $Q_2$ are either independent, comparable, or mutually unsatisfiable.*

*Example 5.* Any cluster query language can be represented by a directed graph where each vertex $Q_i$ is a query and each edge $(Q_i, Q_j)$ indicates that $Q_i$ entails $Q_j$. The graph must satisfy the property that any pair $(Q_i, Q_j)$ of vertices that are not joined by a path denotes a pair of independent or mutually unsatisfiable queries. For instance, the set of vertices of the directed graph in figure 2 forms a cluster conjunctive query language.

**Theorem 4.** *Let $\mathcal{Q}$ be a cluster conjunctive (disjunctive) query language, and $KB = \bigwedge_{i=1}^{m}(Q_i \leftrightarrow \mathsf{q}_i)$ be a weighted knowledge base such that $Q_i \in \mathcal{Q}$ for $1 \leq i \leq m$. Then for any $Q$ in $\mathcal{Q}$, $\mathbf{Pr}_{KB}(Q)$ can be evaluated in $O(bl^2m^2)$ time.*

**Corollary 2.** *There exists an efficient mistake-bound L2R algorithm for any belief assignment problem $(W, \mathcal{Q})$ with background knowledge $\mathcal{B}$ where $\mathcal{Q}$ is a cluster conjunctive (disjunctive) query language.*

*Example 6.* Consider the learning interface $\mathrm{RQ}_\epsilon(W, \mathcal{Q})$ for the truck domain, where $\mathcal{Q}$ is the cluster query language specified in example 5, and $\epsilon = \frac{1}{1000}$. In the following scenario, we suppose that the agent is specified with $\eta = 4$.

At trial (1), the agent receives the query $Q_1 = \exists y\, \mathsf{At}(\mathsf{t}_1, y)$. As $KB$ is initially empty, the reasoner predicts $\mathbf{Pr}_{KB}(Q_1) = \frac{3}{4}$. However, since $\mathbf{Pr}_W(Q_1) = \frac{2}{3}$, the resulting quadratic loss is $\frac{1}{144}$. Hence, the agent has made a mistake. Its knowledge base $KB$ is expanded with the rule $Q_1 \leftrightarrow \mathsf{q}_1$ where $\rho(\mathsf{q}_1) = e^{-\frac{1}{3}}$.

At trial (2), the agent is supplied the query $Q_2 = \forall x\, \mathsf{In}(x, \mathsf{t}_1)$. We thus have $\rho(Q_2 \wedge KB) = \rho(\forall x\, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y\, \mathsf{At}(\mathsf{t}_1, y) \wedge \mathsf{q}_1) + \rho(\forall x\, \mathsf{In}(x, \mathsf{t}_1) \wedge \forall y\, \neg \mathsf{At}(\mathsf{t}_1, y))$. Hence, $\rho(Q_2 \wedge KB) = (1 + 3e^{-\frac{1}{3}})2^{14}$. As $\rho(KB) = (1 + 3e^{-\frac{1}{3}})2^{18}$, it follows that $\mathbf{Pr}_{KB}(Q_2) = \frac{1}{16}$. Since $\mathbf{Pr}_W(Q_2) = \frac{1}{16}$ the prediction is therefore correct.

At trial (3), the agent sees $Q_3 = \forall x\, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y\, \mathsf{At}(\mathsf{t}_1, y)$. Since $Q_3 \models Q_1$, we have $\rho(Q_2 \wedge KB) = \rho(\forall x\, \mathsf{In}(x, \mathsf{t}_1) \wedge \exists y\, \mathsf{At}(\mathsf{t}_1, y) \wedge \mathsf{q}_1) = 3e^{-\frac{1}{3}} \cdot 2^{14}$. Thus $\mathbf{Pr}_{KB}(Q_3) \sim 0.01422$. As $\mathbf{Pr}_W(Q_3) = \frac{1}{24}$, the prediction is still accurate.

## 5 Discussion

Along the lines of making relational probabilities applicable under tractable conditions, this study has stressed the importance of combining learning and reasoning processes together. By coupling exponentiated gradient strategies in learning and weighted model counting techniques in reasoning, our results have shown that some restricted though expressive classes of probabilistic reasoning problems are mistake-bound learnable in the strong sense.

*Related Work.* The framework described here has natural connections with other approaches on learning to reason. The foundations of this paradigm have been established by Khardon and Roth [10, 11] in the setting of propositional deduction. In a similar spirit, the neuroidal architecture developed by Valiant in [22, 23] handles the relational setting by characterizing a variant of the language $\mathcal{R}$ for which learning and deduction are polynomially bounded. Our framework attempts to move one step further by showing that expressive fragments of $\mathcal{R}$ are also tractable for relational probabilistic reasoning.

More generally, the problem of inducing a representation of a probability measure on a space of relational interpretations has been a subject of intensive research in statistical relational learning. Although most approaches are devoted to classificatory purposes, several authors have suggested to learn Bayesian networks [5, 13, 24] in order to facilitate probabilistic reasoning. Yet, from a computational viewpoint, the tractability of query evaluation in Bayesian networks is only guaranteed for very restricted languages, such as Markov-blanket queries [5]. In most cases, including even atomic formulas, the evaluation problem is #P-hard [20]. By contrast, our framework seems to offer a gain of expressiveness by efficiently handling nontrivial fragments of relational logic.

*Perspectives.* Clearly, there are many directions in which one might attempt extensions. For example, we have restricted the study to degrees of belief, or unconditional subjective probabilities. This assumption is justifiable in many situations where the learner is allowed to evaluate the context of its queries. Namely, based on a hitting language, the learner can estimate the value of $\mathbf{Pr}_W(Q|C)$ using $\mathbf{Pr}_{KB}(Q \wedge C)$ and $\mathbf{Pr}_{KB}(C)$. Yet, in some situations the learner is not necessarily informed about the probability of the context, and further formal steps need to be done in the direction of conditional reasoning. An other important research avenue is to extend the expressiveness of query languages. One possible direction is to extend the scope of quantifiers, while maintaining computational efficiency. For example a query such as $\forall x, y\,(\mathsf{In}(y, x) \rightarrow \mathsf{Truck}(x))$ can be transformed into a linear number of decomposable disjunctions. From an orthogonal perspective, we can relax the assumption of strict cluster languages and explore the larger setting of query languages with bounded "cluster-width". To this point, recent results on backdoors in model counting [17] look promising.

## References

1. F. Bacchus, A. J. Grove, J. Y. Halpern, and D. Koller. From statistical knowledge bases to degrees of belief. *Artificial Intelligence*, 87(1-2):75–143, 1996.

2. H. Kleine Büning and X. Zhao. Satisfiable formulas closed under replacement. *Electronic Notes in Discrete Mathematics*, 9:48–58, 2001.
3. T. Bylander. Worst-case analysis of the perceptron and exponentiated update algorithms. *Artificial Intelligence*, 106(2):335–352, 1998.
4. C. M. Cumby and D. Roth. Relational representations that facilitate learning. In *17th Int. Conf. on Knowledge Representation and Reasoning*, pages 425–434, 2000.
5. R. Greiner, A. J. Grove, and D. Schuurmans. Learning bayesian nets that perform well. In *13th Conference on Uncertainty in AI*, pages 198–207, 1997.
6. D. Haussler, N. Littlestone, and M. K. Warmuth. Predicting $\{0,1\}$-functions on randomly drawn points. *Information and Computation*, 115(2):248–292, 1994.
7. T. Horváth and G. Turán. Learning logic programms with structured background knowledge. *Artificial Intelligence*, 128:31–97, 2001.
8. M. Jaeger. Relational bayesian networks. In *13th Int. Conference on Uncertainty in AI*, pages 266–273, 1997.
9. M. Jaeger. On the complexity of inference about probabilistic relational models. *Artificial Intelligence*, 117(2):297–308, 2000.
10. R. Khardon and D. Roth. Learning to reason. *ACM Journal*, 44(5):697–725, 1997.
11. R. Khardon and D. Roth. Learning to reason with a restricted view. *Machine Learning*, 35(2):95–116, 1999.
12. J. Kivinen and M. K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.
13. D. Koller and A. Pfeffer. Learning probabilities for noisy first-order rules. In *Proc. of the 15th Int. Joint Conference on Artificial Intelligence*, pages 1316–1323, 1997.
14. N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.
15. R. Ng and V. S. Subrahmanian. Probabilistic logic programming. *Information and Computation*, 101(2):150–201, 1992.
16. L. Ngo and P. Haddawy. Answering queries from context-sensitive probabilistic knowledge bases. *Theoretical Computer Science*, 171(1-2):147–177, 1997.
17. N. Nishimura, P. Ragde, and S. Szeider. Solving #SAT using vertex covers. In *SAT: 9th International Conference*, pages 396–409, 2006.
18. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
19. D. Poole. The independent choice logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 94(1-2):7–56, 1997.
20. D. Roth. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302, 1996.
21. T. Sang, P. Beame, and H. A. Kautz. Performing Bayesian inference by weighted model counting. In *20h National Conference on Artificial Intelligence (AAAI)*, pages 475–482, 2005.
22. L. G. Valiant. A neuroidal architecture for cognitive computation. *Journal of the ACM*, 47(5):854–882, 2000.
23. L. G. Valiant. Robust logics. *Artificial Intelligence*, 117(2):231–253, 2000.
24. T. Van Allen, R. Greiner, and P. Hooper. Bayesian error-bars for belief net inference. In *17th Conference on Uncertainty in AI*, pages 522–529, 2001.
25. M. Veloso. *Learning by Analogical Reasoning in General Problem Solving*. Ph.D thesis, Carnegie Mellon University, Pittsburgh, PA, 1992.

## Appendix

**Theorem 1.** *For any belief assignment problem* $(W, \mathcal{Q})$ *with background knowledge* $\mathcal{B}$*, on input* $\epsilon > 0$*, when* $\eta = 4$*, the exponentiated gradient L2R algorithm has the following mistake bound* $\mathrm{M}_\epsilon(W, \mathcal{Q}) \leq \frac{1}{2\epsilon}\big(|\mathcal{B}| - H(W)\big)$.

*Proof.* The proof is divided into two parts. First, we present a *direct* EG-L2R algorithm that explicitly uses a distribution $\hat{W}$ on the space $\wp(\mathcal{B})$, and we show that the total number of mistakes of the direct algorithm is bounded by the above result. Second, we show that our *indirect* EG-LR2 algorithm, which maintains a weighted knowledge base, is an exact simulation of the direct algorithm.

The direct EG-L2R algorithm is specified as follows. Initially, $\hat{W}_1$ is given by the uniform distribution over $\wp(\mathcal{B})$. In each trial $t$, the algorithm is first supplied a query $Q_t$ from the oracle. Next, the algorithm predicts the belief $\hat{y}_t$ according to its distribution $\hat{W}_t$, that is, $\hat{y}_t = \mathbf{Pr}_{\hat{W}_t}(Q_t)$. Finally, let $y_t = \mathbf{Pr}_{\hat{W}}(Q_t)$ be the correct probability. If $L(\hat{y}_t, y_t) \leq \epsilon$ then $\hat{W}_{t+1} = \hat{W}_t$. Otherwise, the algorithm updates its probabilities $w_{t,i}$ $(1 \leq i \leq N)$ according to the multiplicative rule

$$\hat{w}_{t+1,i} = \frac{\hat{w}_{t,i}\, e^{\eta(y_t - \hat{y}_t)I_i(Q_t)}}{\sum_{j=1}^{N} \hat{w}_{t,j}\, e^{\eta(y_t - \hat{y}_t)I_j(Q_t)}}$$

The mistake-bound analysis follows [12] with a notable difference in the derived bounds. Let $S = \{(Q_1, \cdots, Q_T)\}$ be a sequence of queries by the oracle. We proceed by showing that the divergence between $\hat{W}_t$ and $W$ decreases after each mistake. The divergence function employed for the analysis is the relative entropy $D(W \| \hat{W}_t) = \sum_{i=1}^{N} w_i \log(w_i / \hat{w}_{t,i})$. We have

$$\sum_{t=1}^{T} D(W\|\hat{W}_t) - D(W\|\hat{W}_{t+1}) = D(W\|\hat{W}_1) - D(W\|\hat{W}_{T+1}) \leq |\mathcal{B}| - H(W) \quad (1)$$

We therefore need to prove that the total number of mistakes is bounded by this sum. In doing so, we show that there exists a positive coefficient $a$ satisfying

$$a(y_t - \hat{y}_t)^2 \leq D(W\|\hat{W}_t) - D(W\|\hat{W}_{t+1}) \quad (2)$$

By developing and approximating the right-hand side, we can derive the following bounds. The first inequality is obtained by using the property that the exponentiated term is a convex function $e^\eta > 1$ and $(y_t - \hat{y}_t)I_i(Q_t) \in [0, 1]$. The second inequality is derived from the logarithmic bound $\log(1 - z(1 - e^x)) \leq xz + x^2/8$.

$$\begin{aligned} D(W\|\hat{W}_t) - D(W\|\hat{W}_{t+1}) &= \eta y_t(y_t - \hat{y}_t) - \log\left(\sum_{i=1}^{N} w_{t,i} e^{\eta(y_t - \hat{y}_t)I_i(Q_t)}\right) \\ &\geq \eta y_t(y_t - \hat{y}_t) - \log\Big(1 - \hat{y}_t\big(1 - e^{\eta(y_t - \hat{y}_t)}\big)\Big) \\ &\geq \eta y_t(y_t - \hat{y}_t) - \eta\hat{y}_t(y_t - \hat{y}_t) - \frac{\eta^2(y_t - \hat{y}_t)^2}{8} \end{aligned}$$

A sufficient condition to show (2) is $a(y_t - \hat{y}_t)^2 - \eta(y_t - \hat{y}_t)^2 + \eta^2(y_t - \hat{y}_t)^2/8 \leq 0$. When the algorithm incurs a mistake we must have $y_t \neq \hat{y}_t$. Hence $a \leq \eta\left(1 - \frac{\eta}{8}\right)$. The maximum is obtained when $\eta = 4$ and, for this value, $a = 2$. Summing over all trials and using the upper bound (1) we have $\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \leq \frac{1}{2}(|\mathcal{B}| - H(W))$. Finally, we know that a mistake arises only when $(y_t - \hat{y}_t)^2 > \epsilon$. By reporting this condition into the last inequality we obtain the desired bound.

Now, we show that the indirect EG-L2R algorithm is an exact simulation of the direct EG-L2R algorithm. To this end, we assume that both algorithms are run on the same parameter $\eta$ and the same sequence $S = (Q_1, \cdots, Q_T)$ of queries supplied by the same oracle. We must show that $\mathbf{Pr}_{\hat{W}_t}(Q_t) = \mathbf{Pr}_{KB_t}(Q_t)$ on each trial $t$. Given an interpretation $I$, let $\tilde{I}$ be the conjunction of all ground literals that are true in $I$. Given a query $Q$, let $\|Q\|$ denote the set of interpretations in $\wp(\mathcal{B})$ that are models of $Q$. Finally, given a standard interpretation $I$ and an extended interpretation $I'$, we say that $I'$ is an *extension* of $I$ if $I \subseteq I'$ and $I' - I$ is a set of weighted relations. Now, consider the following invariant: for any trial $t$ and any interpretation $I \in \wp(\mathcal{B})$, (1) there exists exactly one extension $I^*$ which is a minimal model of $KB_t$ and (2) $\hat{W}_t(I) = \mathbf{Pr}_{KB_t}(\tilde{I})$. This property is sufficient to prove that $\mathbf{Pr}_{\hat{W}_t}(Q_t) = \mathbf{Pr}_{KB_t}(Q_t)$. Indeed, we have

$$\begin{aligned}\mathbf{Pr}_{\hat{W}_t}(Q_t) &= \sum_{I \in \|Q_t\|} \hat{W}_t(I) = \sum_{I \in \|Q_t\|} \mathbf{Pr}_{KB_t}(\tilde{I}) \\ &= \sum_{I \in \|Q_t\|} \frac{\rho(KB_t \wedge \tilde{I})}{\rho(KB_t)} = \frac{\rho(KB_t \wedge Q_t)}{\rho(KB_t)} = \mathbf{Pr}_{KB_t}(Q_t)\end{aligned}$$

The existence of this invariant is proven by induction on the number of trials. Let $t = 1$. Here, the knowledge base is empty. So $I = I^*$ and we have

$$\hat{W}_1(I) = \frac{1}{N} = \frac{\rho(I)}{\sum_{i=1}^{N}\rho(I_i)} = \frac{\rho(KB_1 \wedge \tilde{I})}{\rho(KB_1)} = \mathbf{Pr}_{KB_1}(\tilde{I})$$

Now, consider that $t > 1$ and assume by induction hypothesis that the property holds at the beginning of the trial. We have $\hat{y}_t = \mathbf{Pr}_{\hat{W}_t}(Q_t) = \mathbf{Pr}_{KB_t}(Q_t)$. On receiving the oracle's response, two cases are possible. First, suppose that $L(y_t, \hat{y}_t) \leq \epsilon$. In this case, the property trivially holds at the end of the trial. Now suppose that $L(y_t, \hat{y}_t) > \epsilon$. Before applying the update rule, we know that any interpretation $I$ has exactly one extension $I_t^*$ which is a minimal model of $KB_t$. After the update step, if $I$ is a model of $Q_t$, then $I^* = I_t^* \cup \{\mathsf{q}_t\}$ is the unique minimal extension satisfying $KB_t \wedge (Q_t \leftrightarrow \mathsf{q}_t)$. Dually, if $I$ is not a model of $Q_t$, then $I^* = I_t^*$ is the unique minimal extension satisfying $KB_t \wedge (Q_t \leftrightarrow \mathsf{q}_t)$. In both situations, condition (1) is validated at the end of the trial. Finally, condition (2) is validated according to the following equalities

$$\begin{aligned}\hat{W}_{t+1}(I_i) &= \frac{\hat{w}_{t,i} e^{\eta(y_t - \hat{y}_t) I_i(Q_t)}}{\sum_{j=1}^{N} \hat{w}_{t,j} e^{\eta(y_t - \hat{y}_t) I_j(Q_t)}} = \frac{\rho(KB_t \wedge \tilde{I}_i) \cdot \rho(\mathsf{q}_t)^{I_i(Q_t)}}{\sum_{j=1}^{N} \rho(KB_t \wedge \tilde{I}_j) \cdot \rho(\mathsf{q}_t)^{I_j(Q_t)}} \\ &= \frac{\rho(I_i^*)}{\sum_{j=1}^{N}\rho(I_j^*)} = \frac{\rho(KB_{t+1} \wedge \tilde{I}_i)}{\rho(KB_{t+1})} = \mathbf{Pr}_{KB_{t+1}}(\tilde{I}_i)\end{aligned}$$

**Theorem 2.** *For any decomposable conjunctive (disjunctive) query $Q$, the problem of counting the number of models of $Q$ can be evaluated in $O(|\mathcal{B}||Q|)$ time.*

*Proof.* Consider a decomposable query $Q$ containing up to $l$ quantified literals. Let $\overline{\mathcal{B}}(Q)$ be the set of all possible ground atoms that do not occur in $\mathcal{B}(Q)$. If $Q$ is a conjunction, then $\rho(Q) = 2^{|\overline{\mathcal{B}}(Q)|} \cdot \prod_{i=1}^{l} \rho(L_i)$. Dually, if $Q$ is a disjunction, then $\rho(Q) = 2^{|\mathcal{B}|} - \rho(\neg Q)$. Now, let $L$ be a quantified literal. Then $\rho(L)$ can be obtained inductively by the following rules: (1) if $L$ is a ground atom, then $\rho(L) = 1$, (2) if $L$ is of the form $\neg A$ then $\rho(L) = 2^{\mathcal{B}(Q)} - \rho(A)$, (3) if $L$ is of the form $\forall x\, A(x)$ then $\rho(L) = \prod_{i=1}^{n} \rho(A[x/\mathsf{c}_i])$, and (4) if $L$ is of the form $\exists x\, A(x)$ then $\rho(L) = \sum_{i=1}^{n} 2^{n-i} \rho(A[x/\mathsf{c}_i])$. Based on these decomposition rules, the number of models of $\rho(Q)$ can be determined in $O(|\mathcal{B}||Q|)$ time. □

**Lemma 1.** *Let $Q_1$ and $Q_2$ be two decomposable conjunctive (disjunctive) queries of the target language $\mathcal{Q}$. Then the problem of deciding whether $Q_1$ and $Q_2$ are either comparable or mutually unsatisfiable can be determined in $O(bl^2)$ time.*

*Proof.* Consider two quantified literals $A$ and $\neg A'$. Deciding whether $A \wedge \neg A'$ is unsatisfiable can be determined in $O(b)$ time by enumerating all ground atoms that are models of $A$ and checking whether they are models of $\neg A'$. In any decomposable conjunction (disjunction), quantified literals are pairwise independent. So, checking whether $Q_1 \wedge Q_2$ is unsatisfiable can be decided in $O(bl^2)$ time. Finally, as entailment reduces to unsatisfiability, the result follows □

Two weighted formulas $F$ and $G$ are *equivalent*, denoted $F \equiv G$, if every minimal model of $F$ is a minimal model of $G$ and conversely. Now, given a weighted knowledge base $KB = \bigwedge_{i=1}^{t}(Q_i \leftrightarrow \mathsf{q}_i)$, the *canonical form* of $KB$ is

$$\begin{aligned} can(KB) &= rem(KB) \vee \bigvee\nolimits_{1 \leq i \leq t} ext(Q_i, KB) \text{ where} \\ ext(Q_i, KB) &= Q_i \wedge \bigwedge\nolimits_{j; Q_i \models Q_j} \mathsf{q}_j \wedge \bigwedge\nolimits_{j \neq i; Q_j \models Q_i} \neg Q_j \text{ and} \\ rem(KB) &= \bigwedge\nolimits_{i; \forall j\, Q_i \not\models Q_j} \neg Q_i \end{aligned}$$

**Lemma 2.** *Let $\mathcal{Q}$ be a hitting conjunctive query language. Then any weighted knowledge base $KB = \bigwedge_{i=1}^{t}(Q_i \leftrightarrow \mathsf{q}_i)$ where $Q_i \in \mathcal{Q}$ for $1 \leq i \leq t$ is equivalent to its canonical form.*

*Proof.* This can be made by induction on the number $t$ of rules. The case where $t = 1$ is immediate. So, let $t > 1$ and suppose by induction hypothesis that $KB_{t-1} \equiv can(KB_{t-1})$. We have $KB_t \equiv can(KB_{t-1}) \wedge (Q_t \leftrightarrow \mathsf{q}_t)$. By applying distribution, four cases must be considered. First, observe each conjunction of the form $ext(Q_i, KB_{t-1}) \wedge (Q_t \wedge \mathsf{q}_t)$. If $Q_i \models Q_t$, the conjunction reduces to $ext(Q_i, KB_t)$. If $Q_t \models Q_i$, the conjunction reduces to $ext(Q_t, KB_t)$. Otherwise, $Q_i$ and $Q_t$ are conflicting and the conjunction is hence unsatisfiable. Second, observe each conjunction of the form $ext(Q_i, KB_{t-1}) \wedge (\neg Q_t \wedge \neg \mathsf{q}_t)$. If $Q_i \models Q_t$ then the conjunction is unsatisfiable. Otherwise, either $Q_t \models Q_i$ or $Q_i \wedge Q_t$ is unsatisfiable; in both situations the conjunction reduces to $ext(Q_i, KB_t)$. Third, examine the expression $rem(KB_{t-1}) \wedge (Q_t \wedge \mathsf{q}_t)$. If there is no query $Q_i$ in $KB_{t-1}$ such that $Q_t \models Q_i$ then the conjunction reduces to $ext(Q_t, KB_t)$. Otherwise, it leads to an inconsistency. Finally, the expression $rem(KB_{t-1}) \wedge (\neg Q_t \wedge \neg \mathsf{q}_t)$ reduces to $rem(KB_t)$. By eliminating inconsistencies, $KB_t \equiv can(KB_t)$.

**Theorem 5.** *Let $\mathcal{Q}$ be a hitting conjunctive (disjunctive) query language, and let $KB = \bigwedge_{i=1}^{m}(Q_i \leftrightarrow \mathsf{q}_i)$ be a weighted knowledge base such that $Q_i \in \mathcal{Q}$ for $1 \leq i \leq m$. Then for any $Q \in \mathcal{Q}$, $\mathbf{Pr}_{KB}(Q)$ can be evaluated in $O(bl^2m^2)$ time.*

*Proof.* We first examine conjunctive queries. Consider an arbitrary trial and let $Q$ be the supplied query. By assuming that $KB$ is in canonical form and that $\rho(KB)$ was memorized during the last mistake, we only need to evaluate $\rho(KB \wedge Q)$. Consider each extension $ext(Q_i, KB)$. If $Q \wedge Q_i$ is unsatisfiable, then $\rho(ext(Q_i, KB) \wedge Q) = 0$. Otherwise, $Q \wedge Q_i$ can be reduced to $Q_i$ or $Q$. Let $Q_i^*$ denote the resulting query. Now, let $Q_j$ be a query such that $Q_j \models Q_i$ and $\neg Q_j^*$ be the formula obtained by removing each literal in $\neg Q_j$ which is conflicting with a literal in $Q_i^*$. Finally, let $\beta_i = |\mathcal{B}(\bigwedge_{j \neq i; Q_j \models Q_i} \neg Q_j^*)|$. We thus have

$$\rho(ext(Q_i, KB) \wedge Q) = \rho(Q_i^*) \cdot \prod_{j : Q_i \models Q_j} \rho(\mathsf{q}_j) \cdot \Big(2^{\beta_i} - \sum_{j \neq i : Q_j \models Q_i} \rho(Q_j^*)\Big)$$

As there are at most $m$ queries $Q_j$ in the extension of $Q_i$, the evaluation takes $O(bl^2m)$ time. Now, consider the reminder of $KB$. If $Q \models Q_i$ for some $Q_i$ in $KB$, then $\rho(rem(KB) \wedge Q) = 0$. Otherwise, let $\neg Q_i^*$ be the formula obtained by removing each literal in $\neg Q_i$ which is conflicting with a literal in $Q$. We have

$$\rho(rem(KB) \wedge Q) = 2^{|\mathcal{B}(KB)|} - \sum_{Q_i \in rem(KB)} \rho(Q_i^*)$$

The evaluation requires $O(bl^2m)$ time. By taking the weighted sum of the different components, the prediction step takes $O(bl^2m^2)$ time. Now, let us turn to the update step. In case of mistake, $\rho(KB_{t+1}) = \rho(KB_t) + \rho(KB_t \wedge Q_t)(e^{\eta(y_t - \hat{y}_t)} - 1)$. The transformation of $can(KB_t)$ into $can(KB_{t+1})$ takes $O(bl^2m)$ time. Now, let us turn to disjunctive queries. For any $Q$, we have $\mathbf{Pr}_{KB}(Q) = 1 - \mathbf{Pr}_{KB}(\neg Q)$. As $\neg Q$ is a conjunction, the prediction step takes $O(bl^2m^2)$ time, provided that $KB$ is defined over a conjunctive language. The condition is fulfilled using the dual update policy: if the learner has made a mistake on trial $t$, then expand $KB_t$ with the rule $Q_t \leftrightarrow \mathsf{q}_t$ where $\rho(\mathsf{q}_t) = e^{-\eta(y_t - \hat{y}_t)}$.

**Theorem 6.** *Let $\mathcal{Q}$ be a cluster conjunctive (disjunctive) query language, and let $KB = \bigwedge_{i=1}^{m}(Q_i \leftrightarrow \mathsf{q}_i)$ be a weighted knowledge base such that $Q_i \in \mathcal{Q}$ for $1 \leq i \leq m$. Then for any $Q \in \mathcal{Q}$, $\mathbf{Pr}_{KB}(Q)$ can be evaluated in $O(bl^2m^2)$ time.*

*Proof.* Any cluster query language $\mathcal{Q}$ is the union $\mathcal{Q}_1 \cup \cdots \cup \mathcal{Q}_b$ of up to $b$ pairwise independent hitting query sets. Not that some of these sets can be empty. Thus any knowledge base $KB$ over $\mathcal{Q}$ is a conjunction $KB_1 \wedge \cdots \wedge KB_b$ of weighted formulas such that the set of queries occurring in each formula $KB_i$ is included in the cluster $\mathcal{Q}_i$. By lemma 2, each $KB_i$ can be rewritten into its canonical form. Now, consider an arbitrary trial and suppose that the supplied query $Q$ belongs to $\mathcal{Q}_i$. The task of finding $KB_i$ takes $O(blm)$ time, as there are at most $m$ nonempty sub-bases. If $KB_i$ is empty, then $\mathbf{Pr}_{KB}(Q) = \rho(Q)/N$, which takes $O(bl)$ time. Otherwise, $\mathbf{Pr}_{KB}(Q) = \rho(can(KB_i \wedge Q))/\rho(can(KB_i))$. By application of theorem 3, this takes $O(bl^2m^2)$ time. Additionally, the cost of updating $KB_i$ simply takes $O(bl^2m^2)$ time.