Reasoning with imprecise probabilistic knowledge on enzymes for rapid screening of potential substrates or inhibitor structures

Weiru Liu, Anbu Yue
School of Electronics, Electrical Engineering
and Computer Science,
Queen's University Belfast,
Belfast BT7 1NN, UK
{w.liu, a.yue}@qub.ac.uk

David J Timson
School of Biological Science
Queen's University Belfast,
Belfast BT9 7BL, UK
d.timson@qub.ac.uk

Abstract

In many applications, there is a need to model and reason with imprecise probabilistic knowledge. In this paper, we discuss how to model imprecise probabilistic knowledge obtained from experiments in biological sciences on enzymes for rapid screening of potential substrate or inhibitor structures. Each imprecise probabilistic knowledge base is modelled as a probabilistic logic program (PLP). To predict a meaningful substrate structure, we have developed a framework (and a tool) in which a user (bioscientist) can query against a PLP (or a collection of PLPs), can examine how relevant a PLP is for answering a query, and can select a query result that is more satisfactory. This framework is implemented by integrating an optimizer in MatLab to solve the optimization problems subject to linear constraints. A preliminary version of the tool was demonstrated in the ECAI08 Demo session. Experimental results on evaluating the tool with probabilistic knowledge on enzymes for rapid screening of potential substrates or inhibitor structures demonstrate that this tool has a great potential to be used in many similar areas for the initial screening of compound structures in drug discovery.

Keywords. Imprecise probabilistic knowledge, prediction, substrate structure, enzymes, rapid screening.

1 Introduction

Most of the knowledge that is used, for example, for advanced knowledge base systems or for cognitive modeling is uncertain, incomplete, imprecise and subject to changes. Very often, this uncertainty and incompleteness is characterized by probabilities, especially, when the knowledge concerned is elicited from experiments. Therefore, there is a need to develop adequate theories and frameworks to model and reason with such probabilistic knowledge.

Probabilistic logic programming is a framework to represent and reason with imprecise (conditional) probabilistic knowledge. An agent's knowledge is represented by a *probabilistic logic program* (PLP) which is a set of (conditional) logical formulae with probability intervals. The impreciseness of the agent's knowledge is explicitly represented by assigning a probability interval (or a single probability) to every logical formula indicating that the probability of the formula shall be in the given interval. Probabilistic logic programming has been used to represent and reason with probabilistic knowledge in many real world applications, e.g., [2, 5, 9]. Among various types of probabilistic logic programming, conditional probabilistic logic programming (PLP for short) [7, 8] is particularly tailored to represent conditional events with probabilities of the form $(C|A_1 \wedge ... \wedge A_n)[l, u]$ where A_is are conditions, C is a conclusion. $(C|A_1 \wedge ... \wedge A_n)[l, u]$ is interpreted as the probability of conditional event $C|A_1 \wedge ... \wedge A_n$ falling in interval $[l, u]$.

To illustrate the use of conditional probabilistic logic programming, let us consider medical treatments for certain medical conditions (diseases), such as a patient is diagnosed with liver cancer. There are various types of treatments for cancers, such as, (A) surgery only to remove the organ; (B) surgery plus Radiotherapy; (C) Radiotherapy only, depending on the stage of the cancer, the health condition of the patient and possibly other factors. Then statistical summaries from clinical trials studied on the relationship between mortality and treatments can be represented as conditional events shown below.

$$Mortality(X, Year10)|LiverCancer(X, Year0) \wedge CancerStage(X, early) \wedge Surgery(X, Year0) \wedge RT(X; Year0)[0.223; 0.225]$$

This piece of imprecise probabilistic knowledge says that from this trial (or the meta-analysis of many trials), the probability of a patient's 10-year mortality, given that the patient is in his/her early liver cancer stage, undergoing a surgery plus Radiotherapy, is in between 0.223 to 0.225.

Conditional events like above cannot always be simply interpreted as cause-effect relationships. For the above example, it is not that the surgery and RT caused a patient to die in 10-years, rather, it says that if those actions are taken place (given that the patient's liver cancer stage is early), then what the probability of this patient being dead in 10-years could be. Of course, if no action was taken

place, the probability of the patient being dead in 10-years would be much greater than 22.5%. Therefore, conditional probabilistic logic programming offers a much more suitable framework for capturing imprecise probabilistic scientific knowledge of this kind than other approaches. Given a PLP and a query against the PLP, traditionally, a probability interval is returned as the answer. This interval implies that the true probability of the query shall be within the given interval. However, when this interval is too wide, it provides no useful information. For instance, if a PLP contains knowledge $\{(fly(X)|bird(X)[0.98, 1], (bird(X)|magpie(X))[1, 1]\}$, then the answer to the query *Can a magpie fly?* (i.e., $?(fly(t)|magpie(t))$) is a trivial bound $[0, 1]$.

One way to enhance the reasoning power of a PLP is to apply the maximum entropy principle [6]. From this principle, a single probability distribution is selected and it is assumed to be the most acceptable one for the query among all possible probability distributions. As a consequence, a precise probability is given for a query even when the agent's original knowledge is imprecise. In the above example, by applying the maximum entropy principle, 0.98 is returned as the answer for the query. Intuitively, accepting such a precise probability from (a prior) imprecise knowledge can be risky. When an agent's knowledge is rich enough then a single probability could be reliable, however, when an agent's knowledge is (very) imprecise, an interval is more appropriate than a single probability.

Therefore, how useful a probabilistic logic program (PLP) is to answering a given query? This question is important in two fold: first, it helps to analyze if a PLP is adequate to answer a query and second, if a PLP is sufficiently relevant to a query, then shall a single probability be obtained or shall a probability interval be more suitable? To answer these questions, in [18], we proposed two concepts, *the measure of ignorance* and *the measure of degree of satisfaction*, w.r.t. a PLP and a query. The former analyzes the impreciseness of the PLP w.r.t. the query, and the latter measures which (tighter) interval is sufficiently informative to answer the query.

In this paper, we present our investigation about how to use PLPs to represent and reason with imprecise probabilistic knowledge obtained from experiments, especially on substrates prediction in biomedical sciences. We first discuss the importance of evaluating the relevance of a knowledge base w.r.t a query, focusing on how reliable a query result returned from querying a PLP could be, knowing that the knowledge contained in the PLP is imprecise. To quantitatively measure the reliability of a query result, we introduce our formal analysis of ignorance and degree of satisfaction about a query result obtained from the PLP [18]. We then present our implementation of a probabilistic querying system which takes PLPs as input knowledge bases and produces probabilistic results for queries (against a chosen PLP). The results are either in the form of pure probabilistic terms (an interval or a maximum entropy), or the maximum entropy plus its ignorance, or an interval plus its degree of satisfaction. The first form of output is the traditional type of output from probabilistic logic programming, whilst the latter two are our extensions – adding extra information about a query result to tell a user how reliable this result could be when using this particular knowledge base.

We apply our theory and system to enzymes for rapid prediction of potential substrate or inhibitor structures. We conducted two sets of experiments, one is on the human enzyme galactokinase, which uses galactose as a substrate, and the other is on substrate prediction for NQO1. The experimental results demonstrate that using imprecise probabilistic knowledge as a first step in screening for substrates can be very useful and significant in many similar applications, since this initial prediction could allow bioscientists to selectively experiment on more hopeful candidates, saving both time and money in the whole process.

This paper is organized as follows. In Section 2, we briefly review probabilistic logic programming. In Section 3, we describe how to analyze the quality of knowledge in a PLP and in Section 4 we introduce a general theory and an instantiation on measuring the ignorance and the degree of satisfaction w.r.t. a PLP and a query. In Section 5, we describe our system architecture and efficient implementation. In Section 6, we illustrate our framework with two case studies in bioscience. Finally, we conclude the paper in Section 7.

2 Preliminary

We briefly review conditional probabilistic logic programming here [7, 8].

We use Φ to denote the finite set of predicate symbols, $\mathcal{V}$ to denote the set of *object variables*, and $\mathcal{B}$ to denote the set of *bound constants* which describe the bound of probabilities, and bound constants are in [0,1]. We use $a, b, \ldots$ to denote constants from Φ and $X, Y \ldots$ to denote object variables from $\mathcal{V}$. An *object term* t is a constant from Φ or an object variable from $\mathcal{V}$. An *atom* is of the form $p(t_1, \ldots, t_k)$, where p is a predicate symbol and $t_1, \ldots, t_k$ are object terms. We use Greek letters $\phi, \varphi, \psi, \ldots$ to denote *events* (or *formulae*) which are obtained from atoms by logic connectives $\wedge, \vee, \neg$ as usual. A *conditional event* is of the form $(\psi|\phi)$ where ψ and ϕ are events, and φ is called the *antecedent* and ψ is called the *consequent*. A *probabilistic formula*, denoted as $(\psi|\varphi)[l, u]$, means that the probability of conditional event $\psi|\varphi$ is between l and u, where l, u are bound constants. A set of probabilistic formulae is called a *conditional probabilistic logic program (PLP)*, a PLP is denoted as P in the rest of the paper.

A *ground* term, (resp. event, conditional event, probabilis-

tic formula, or PLP) is a term, (resp. event, conditional event, probabilistic formula, or PLP) that does not contain any object variables in $\mathcal{V}$.

All the constants in Φ form the Herbrand universe, denoted as HU_Φ, and the Herbrand base, denoted as HB_Φ, is the finite nonempty set of all events constructed from the predicate symbols in Φ and constants in HU_Φ. A subset I of HB_Φ is called a *possible world* and $\mathcal{I}_\Phi$ is used to denote the set of all possible worlds over Φ. A function σ that maps each object variable to a constant is called an *assignment*. It is extended to object terms by $\sigma(c) = c$ for all constant symbols from Φ. An event φ satisfied by I under σ, denoted by $I \models_\sigma \varphi$, is defined inductively as:

- $I \models_\sigma p(t_1, \ldots, t_n)$ iff $p(\sigma(t_1), \ldots, \sigma(t_n)) \in I$;
- $I \models_\sigma \phi_1 \wedge \phi_2$ iff $I \models_\sigma \phi_1$ and $I \models_\sigma \phi_2$;
- $I \models_\sigma \phi_1 \vee \phi_2$ iff $I \models_\sigma \phi_1$ or $I \models_\sigma \phi_2$;
- $I \models_\sigma \neg\phi$ iff $I \not\models_\sigma \phi$

An event φ is satisfied by a possible world I, denoted by $I \models_{cl} \varphi$, iff $I \models_\sigma \varphi$ for all assignments σ. An event φ is a *logical consequence* of event ϕ, denoted as $\phi \models_{cl} \varphi$, iff all possible worlds that satisfy ϕ also satisfy φ.

In this paper, we use $\top$ to represent (ground) tautology, and we have that $I \models_{cl} \top$ for all I and all assignments σ. And we use $\bot$ to denote $\neg\top$.

If Pr is a function (or distribution) on $\mathcal{I}_\Phi$ (i.e., as $\mathcal{I}_\Phi$ is finite, Pr is a mapping from $\mathcal{I}_\Phi$ to the unit interval [0,1] such that $\sum_{I\in\mathcal{I}_\Phi} Pr(I) = 1$), then Pr is called a *probabilistic interpretation*. For an assignment σ, the probability assigned to an event φ by Pr, is denoted as $Pr_\sigma(\varphi)$ where $Pr_\sigma(\varphi) = \sum_{I\in\mathcal{I}_\Phi, I\models_\sigma\varphi} Pr(I)$. When φ is ground, we simply written it as $Pr(\varphi)$. When $Pr_\sigma(\phi) > 0$, the conditional probability, $Pr_\sigma(\psi|\phi)$, is defined as $Pr_\sigma(\psi|\phi) = Pr_\sigma(\psi \wedge \phi)/Pr_\sigma(\phi)$. When $Pr_\sigma(\phi) = 0$, $Pr_\sigma(\psi|\phi)$ is undefined. Also, when $(\psi|\phi)$ is ground, we simply write $Pr(\psi|\phi)$.

A probabilistic interpretation Pr satisfies or is a *probabilistic model* of a probabilistic formula $(\psi|\phi)[l, u]$ under assignment σ, denoted as $Pr \models_\sigma (\psi|\phi)[l, u]$, iff $l \leq Pr_\sigma(\psi|\phi) \leq u$ or $Pr_\sigma(\phi) = 0$. A probabilistic interpretation Pr satisfies or is a *probabilistic model* of a probabilistic formula $(\psi|\phi)[l, u]$ iff Pr satisfies $(\psi|\phi)[l, u]$ under all assignments. A probabilistic interpretation Pr satisfies or is a *probabilistic model* of a PLP P iff for all assignment σ, $\forall(\psi|\phi)[l, u] \in P, Pr \models_\sigma (\psi|\phi)[l, u]$. A probabilistic formula $(\psi|\varphi)[l, u]$ is a *consequence* of PLP P, denoted by $P \models (\psi|\varphi)[l, u]$, iff all probabilistic models of P satisfy $(\psi|\varphi)[l, u]$. A probabilistic formula $(\psi|\varphi)[l, u]$ is a *tight consequence* of P, denoted by $P \models_{tight} (\psi|\varphi)[l, u]$, iff $P \models (\psi|\varphi)[l, u]$, $P \not\models (\psi|\varphi)[l, u']$, $P \not\models (\psi|\varphi)[l', u]$ for all $l' > l$ and $u' < u$ ($l', u' \in [0, 1]$). It is worth noting that if $P \models (\phi|\top)[0, 0]$ then $P \models_{tight} (\psi|\phi)[1, 0]$ where $[1, 0]$ stand for the empty set.

A query is of the form $?(\psi|\phi)$ or $?(\psi|\phi)[l, u]$, where ψ and ϕ are ground events and $l, u \in [0, 1]$. For query $?(\psi|\phi)$, by the tight consequence relation, a bound $[l, u]$ is given as the answer, such that $P \models_{tight} (\psi|\phi)[l, u]$. For query $?(\psi|\phi)[l, u]$, a bound $[l, u]$ is given by the user. A PLP returns *True (or Yes)* if $P \models (\psi|\phi)[l, u]$ and *False (or No)* if $P \not\models (\psi|\phi)[l, u]$ [8].

The principle of maximum entropy is a well known techniques to represent probabilistic knowledge. Entropy quantifies the indeterminateness inherent to a distribution Pr by $H(Pr) = -\Sigma_{I\in\mathcal{I}_\Phi} Pr(I) \log Pr(I)$. Given a logic program P, the *principle of maximum entropy model* (or *me-model*), denoted by $me[P]$, is defined as: $H(me[P]) = \max H(Pr) = \max_{Pr\models P} -\Sigma_{I\in\mathcal{I}_\Phi} Pr(I) \log Pr(I)$

$me[P]$ is the unique probabilistic interpretation Pr that is a probabilistic model of P and that has the greatest entropy among all the probabilistic models of P.

Let P be a ground PLP, we say that $(\psi|\varphi)[l, u]$ is a *me-consequence* of P, denoted by $P \models^{me} (\psi|\varphi)[l, u]$, iff P is unsatisfiable, or $me[P] \models (\psi|\varphi)[l, u]$.

We say that $(\psi|\varphi)[l, u]$ is a *tight me-consequence* of P, denoted by $P \models^{me}_{tight} (\psi|\varphi)[l, u]$, iff either P is unsatisfiable, $l = 1$, $u = 0$, or $P \models \bot \leftarrow \varphi$, $l = 1$, $u = 0$, or $me[P](\varphi) > 0$ and $me[P](\psi|\varphi) = l = u$.

3 A Formal Analysis of PLPs

In information theory, the information entropy is a measure of the uncertainty associated with a random variable. Entropy quantifies information in a piece of data. Informally, $-\log p(X = x_i)$ means the degree of surprise[1] when one observes that the random variable turns out to be x_i. In another word, $-\log p(X = x_i)$ reflects the information one receives from the observation. The entropy is an expectation of the information one may receive from a random domain by observing random events. Inspired by this, we define a knowledge entropy, which reflects how much an agent knows the truth value of ψ given ϕ prior any observations. Informally, more surprised an agent is by the observation, more knowledge it learns from the observation, and thus, less knowledge it has about ψ given ϕ before observing ψ or $\neg\psi$ given ϕ.

Definition 1 *Let P be a PLP, and $(\psi|\phi)$ be a conditional event. Suppose that Pr is a probabilistic model for P, then the* knowledge entropy *of inferring ψ from ϕ under Pr, denoted as $K_{Pr}(\psi|\phi)$, is defined as $K_{Pr}(\psi|\phi)$*

$$= 1 + \frac{1}{2}(Pr(\psi|\phi)\log Pr(\psi|\phi) + Pr(\neg\psi|\phi)\log Pr(\neg\psi|\phi))$$

It is obvious that $K_{Pr}(\psi|\phi) = K_{Pr}(\neg\psi|\phi)$ and $K_{Pr}(\psi|\phi) \in [0, 1]$. Trivially, $K_{Pr}(\phi|\phi) = 1$ and $K_{Pr}(\neg\phi|\phi) = 1$, since from Pr, the truth values of an event and its negation are known, when the event is given.

[1] http://en.wikipedia.org/wiki/Self-information

By extending the above definition, we can define a knowledge measurement for a PLP.

Definition 2 *Let P be a PLP, and $(\psi|\phi)$ be a conditional event. Suppose that Pr is a probabilistic model for P and $Pr(\phi) > 0$, then the* knowledge measurement $K_P(\psi|\phi)$ *is defined by:*

$$\begin{array}{lcl} minK_P(\psi|\phi) & = & \min_{Pr\models P} K_{Pr}(\psi|\phi) \\ maxK_P(\psi|\phi) & = & \max_{Pr\models P} K_{Pr}(\psi|\phi) \\ K_P(\psi|\phi) & = & [minK_P(\psi|\phi), maxK_P(\psi|\phi)] \end{array}$$

The measurement $K_P(\psi|\phi)$ is used to characterize the usefulness of knowledge contained in PLP P for inferring ψ when knowing or observing ϕ. When ψ or $\neg\psi$ can be inferred from ϕ under P, P contains all the necessary knowledge for inferring ψ given ϕ, and so we have $minK_P(\psi|\phi) = 1$. When knowledge in P excludes the possibility that the probability of ψ (or $\neg\psi$) may be 1 given ϕ, i.e., $P \cup \{(\psi|\phi)[1,1]\}$ (or $P \cup \{(\psi|\phi)[0,0]\}$) is unsatisfiable, then the knowledge contained in P can not fully support ψ given ϕ, so $maxK_P(\psi|\phi) < 1$. Specifically, if it can not be inferred that ψ is more (or less) likely to be true than $\neg\psi$ (i.e. the probability of ψ given ϕ is bigger (or smaller) than $\neg\psi$ given ϕ), then $minK_P(\psi|\phi) = 0$.

We can define a partial order $\preceq$ over the set $\{[x,y]|x,y \in [0,1]\}$ as $[a,b] \preceq [c,d]$ iff $a \geq c, b \leq d$, and $[a,b] \prec [c,d]$ iff $[a,b] \preceq [c,d]$ and $a > c$ or $b < d$. We say a PLP P is more precise than P' about $\psi|\phi$, if $K_P(\psi|\phi) \preceq K_{P'}(\psi|\phi)$, denoted as $P \preceq^k_{(\psi|\phi)} P'$.

If $minK_P(\psi|\phi) \neq maxK_P(\psi|\phi)$ given P, then the knowledge contained in P is not sufficient to decide the probability of ψ given ϕ, that is, the knowledge contained in P about inferring ψ given ϕ is imprecise. In order to infer the actual probability of ψ given ϕ under P, we need more knowledge.

Proposition 1 *Let P and P' be two PLPs. If $P \models P'$ then $P \preceq^k_{(\psi|\phi)} P'$ for any conditional event $(\psi|\phi)$.*

This proposition suggests that the consequence relation $\models$ considers all statements in the PLP while the knowledge measurement focuses only on the knowledge about ψ given ϕ.

In the view of knowledge entropy, reasoning under the maximum entropy principle implicitly introduces some extra knowledge to enhance the reasoning power of PLP. We should be aware that although this assumption seems intuitive, it may be wrong, as shown below.

Example 1 *Let $P_1 = \{(headUp(X)|toss(X))[0.5,0.5]\}$, $P_2 = \{(headUp(X)|toss(X))\ [0,1]\}$ be two PLPs. Here, P_1 states that tossing a fair coin may result in head up with probability 0.5, however, in P_2, we do not know whether the coin is fair.*

In this example, the knowledge in P_1 is richer than that in P_2 since from P_1 we know the coin is fair. With the maximum entropy principle, we get that $P_1 \models^{me} (headUp(coin)|toss(coin))[0.5,0.5]$, $P_2 \models^{me} (headUp(coin)|toss(coin))[0.5,0.5]$. This result suggests that the difference between P_1 and P_2 is ignored under the maximum entropy reasoning. By calculating the knowledge entropy of P_1 and P_2, we have $K_{P_1}(headUp(coin)|toss(coin)) = [0,0]$ and $K_{P_2}(headUp(coin)|toss(coin)) = [0,1]$. Thus we know that P_1 is more precise than P_2. Obviously, the conclusion $(headUp(coin)|toss(coin))[0.5,0.5]$ is more acceptable under P_1 than under P_2.

4 Ignorance and Degree of Satisfaction

The knowledge measurement defined above is not sufficient either. Intuitively, the knowledge measurement $K_P(\psi|\phi)$ indicates the ignorance about the conditional event $(\psi|\phi)$. Unfortunately, such an interval cannot sufficiently reflex the ignorance about $(\psi|\phi)$. This is not surprising, since $K_P(\psi|\phi)$ is determined only by the tight probability bound of the conditional event $(\psi|\phi)$, and other knowledge is not considered in $K_P(\psi|\phi)$.

Example 2 *Let a PLP P be defined as*

$$P = \left\{ \begin{array}{l} (fly(X)|bird(X))[0.9,1], \\ (bird(X)|magpie(X))[1,1] \\ (sickMagpie(X)|magpie(X))[0,0.1], \\ (magpie(X)|sickMagpie(X))[1,1] \end{array} \right\}$$

From P, we can infer that

$$\begin{array}{l} P \models_{tight} (fly(t)|magpie(t))[0,1], \\ P \models_{tight} (fly(t)|sickmagpie(t))[0,1], \\ P \models^{me}_{tight} (fly(t)|magpie(t))[0.9,0.9], \\ P \models^{me}_{tight} (fly(t)|sickMagpie(t))[0.9,0.9]. \end{array}$$

Here, we have $K_P(fly(t)|sickmagpie(t)) = K_P(fly(t)|magpie(t))$. However, since the proportion of sick magpies in birds is smaller than the proportion of magpies in birds, the knowledge about birds can fly should be cautiously applied to sick magpies than magpies. In another word, *more than 90% birds can fly* is more about magpies than sick magpies. Therefore, accepting that 90% magpies can fly is more rational than accepting that 90% sick magpies can fly. However, knowledge measurement cannot differentiate this. Below, we introduce two measures to overcome this weakness. These two measures are the instantiated measures from the general framework for analyzing and reasoning with imprecise PLPs proposed in [18].

In probabilistic theory and information theory, how to measure the *distance* between probability distributions is a major topic. One of the most common measures for comparing probability distributions is the KL-divergence.

Definition 3 *Let Pr and Pr' be two probability distributions over the same set $\mathcal{I}_\Phi$. The KL-divergence between Pr and Pr' is defined as:*

$$KL(Pr\|Pr') = -\Sigma_{I\in\mathcal{I}_\Phi} Pr(I)\log\frac{Pr'(I)}{Pr(I)}$$

It is worth noting that KL-divergence is asymmetric. KL-divergence is also called *relative entropy*.[2] An important conclusion is that $H(Pr) = KL(Pr||Pr_{unif})$, where Pr_{unif} is the uniform distribution.

From the KL-divergence, we can measure the amounts of the information that should be received to believing lower and upper bounds for $(\psi|\phi)$ other than the probability given by maximum entropy.

$$\begin{aligned}\nu^{pos}_{P,(\psi|\phi)}(v) &= \min_{Pr\models P, Pr(\psi|\phi)=v} KL(Pr||me[P]),\\ &\quad \text{where } v \geq me[P]\\ \nu^{neg}_{P,(\psi|\phi)}(v) &= \min_{Pr\models P, Pr(\psi|\phi)=v} KL(Pr||me[P]),\\ &\quad \text{where } v \leq me[P]\end{aligned}$$

$$\begin{aligned}dis^{pos}_{P,(\psi|\phi)}(u,v) &= |\nu^{pos}_{P,(\psi|\phi)}(u) - \nu^{pos}_{P,(\psi|\phi)}(v)|\\ dis^{neg}_{P,(\psi|\phi)}(u,v) &= |\nu^{neg}_{P,(\psi|\phi)}(u) - \nu^{neg}_{P,(\psi|\phi)}(v)|\end{aligned}$$

Definition 4 *Let P be a PLP and $(\psi|\phi)$ be a conditional event. Suppose that $P \models_{tight} (\psi|\phi)[l,u]$ and $P \models^{me}_{tight} (\psi|\phi)[p_{me}, p_{me}]$, then we have that:*

$$\mathsf{SAT}^{KL}_P((\psi|\phi)[a,b]) = \begin{cases} 0.5 * (\frac{dis^{pos}_{P,(\psi|\phi)}(p_{me},\min(u,b))}{dis^{pos}_{P,(\psi|\phi)}(p_{me},u)} + \frac{dis^{neg}_{P,(\psi|\phi)}(p_{me},\max(a,l))}{dis^{neg}_{P,(\psi|\phi)}(p_{me},l)}), \\ \quad \textit{if } p_{me}\in[a,b] \\ 0, \quad \textit{otherwise}\end{cases}$$

It is proved in [18] that $\mathsf{SAT}^{KL}_P(\psi|\phi)[a,b]$ can be interpreted as the *second order* probability that the actual probability of $(\psi|\phi)$ falls in the interval $[a,b]$.

Example 3 *Let P be a PLP:*

$$P = \left\{\begin{array}{l}(fly(X)|bird(X))[0.9,1]\\(bird(X)|magpie(X))[1,1]\\(magpie(X)|sickmagpie(X))[1,1]\end{array}\right\}$$

Let two queries be $?(fly(t)|magpie(t))$ and $?(fly(t)|sickmagpie(t))$. The we have (c.f. [18])

$P \models_{tight} (fly(t)|magpie(t))[0,1]$,
$P \models^{me}_{tight} (fly(t)|magpie(t))[0.9,0.9]$ *and*
$P \models_{tight} (fly(t)|sickmagpie(t))[0,1]$,
$P \models^{me}_{tight} (fly(t)|sickMagpie(t))[0.9,0.9]$.

So, we cannot differentiate magpies from sick magpies in their ability of flying, although sick magpies are more a special kind of magpies, and therefore they are less likely to be able to fly than magpies. In contract, in our framework, we have $\mathsf{SAT}^{KL}_P((fly(t)|magpie(t))[0.8,1]) = 0.58$, and $\mathsf{SAT}^{KL}_P((fly(t)|sickmagpie(t))[0.8,1]) = 0.53$ for two queries $?(fly(t)|magpie(t))[0.8,1]$ and $?(fly(t)|sickmagpie(t))[0.8,1]$. By comparing their KL degrees of satisfaction, it is clear that magpies are more likely able to fly than sick magpies.

[2] It should be noted that $KL(Pr\|Pr')$ is undefined if $Pr'(I) = 0$ and $Pr(I) \neq 0$. This means that Pr has to be absolutely continuous w.r.t. Pr' for $KL(Pr\|Pr')$ to be defined.

5 A System for Answering Queries

5.1 Efficient implementation

To efficiently return a query result given a PLP, we implemented the efficient algorithms proposed in [6, 8]. Using these algorithms, a PLP can be translated into a liner or nonlinear optimization problem. We implemented these algorithms in Java and solved the underlying optimization problem using a component in Matlab. In addition, we also implemented the calculation of ignorance and degree of satisfaction with the algorithms given below. These algorithms rely on the algorithms provided in [6, 8] as well as the software Matlab to optimize a PLP.

Algorithm 1 (KLIgnorance)
Input*: PLP P and a ground query $Q = ?(\psi|\phi)$*
Output*: Ignorance value for Q*

1. **IF** *P is unsatisfiable* **THEN return** 1
2. **IF** $P \models_{tight} (\phi|\top)[0,0]$ **THEN return** 1
3. *Compute the tight bound $[l,u]$ for $(\psi|\phi)$ by Algorithm Tight_0_Consequence in Fig. 5. in [6].*
4. *Compute the simplified PLP D index sets R and associate numbers a_r and optimal solution $y^\star_r$ ($r \in R$) by Algorithm Tight_me_Consequence in Fig. 7. in [6].*
5. *Compute the optimal value ig_{neg} of the optimization problem:*

$$ig_{neg} = \max\left(-\sum_{r\in R} y^l_r(\log y^l_r - \log a_r)\right)$$

subject to: y^l_r satisfies $LC(\top, D^l, R)$, where $D^l = D \cup \{(\psi|\phi)[l,l]\}$

6. *Compute the optimal value ig_{pos} of the optimization problem:*

$$ig_{pos} = \max\left(-\sum_{r\in R} y^u_r(\log y^u_r - \log a_r)\right)$$

subject to: y^u_r satisfies $LC(\top, D^u, R)$, where $D^u = D \cup \{(\psi|\phi)[u,u]\}$.

7. *Compute optimal solution y'_r ($r \in R$) for $P' = \emptyset$ by Algorithm Tight_me_Consequence in Fig. 7. in [6]. $p_{me} := me[P'](\psi|\phi)$.*
8. *Compute the optimal value ig'_{neg} of the optimization problem:*

$$ig'_{neg} = \max\left(-\sum_{r\in R} y^l_r(\log y^l_r - \log a_r)\right)$$

subject to: y^l_r satisfies $LC(\top, D^l_0, R)$, where $D^l_0 = \{(\psi|\phi)[l,l]\}$

9. *Compute the optimal value ig'_{pos} of the optimization problem:*

$$ig'_{pos} \quad = \quad \max\left(-\sum_{r\in R} y_r^u(\log y_r^u - \log a_r)\right)$$

subject to: y_r^u satisfies $LC(\top, D_0^u, R)$, where $D_0^u = \{(\psi|\phi)[u,u]\}$.

10. **IF** $p_{me} < u$ **THEN** $s_1 := 1$ **ELSE** $s_1 := -1$
11. **IF** $p_{me} > l$ **THEN** $s_2 := 1$ **ELSE** $s_2 := -1$
12. $ig := (s_1 * ig_{pos} + s_2 * ig_{neg})/(ig'_{pos} + ig'_{neg})$
13. **RETURN** ig

Algorithm 2 (KLDivergence)
Input*: PLP P, $me[P]$, a conditional event $(\psi|\phi)$, and a probability value v.*
Output*: $kl = \min_{Pr\models P, Pr(\psi|\phi)=v} KL(Pr||me[P])$*

1. *$me[P]$ is obtained from Algorithm 1 and is represented as y^{me}.*
2. *Compute the tight bound $[l', u']$ for $(\psi|\phi)$ by Algorithm Tight_0_Consequence in Fig. 5. in [6].*
3. **IF** $v \notin [l', u']$ **THEN return** *ERROR.*
4. *Compute the optimal value kl of the optimization problem:*

$$kl \quad = \quad \min\left(\sum_{r\in R} y_r \log y_r - \sum_{r\in R} y^r \log y^{me}\right)$$

subject to: y_r satisfies $LC(\top, D^V, R)$, where $D^V = D \cup \{(\psi|\phi)[v,v]\}$.

5. **return** *kl*

Algorithm 3 (KLSatisfaction)
Input*: PLP P and a ground query $Q = ?(\psi|\phi)[l,u]$*
Output*: KL degree of satisfaction for Q*

1. **IF** $P \models_{tight} (\phi|\top)[0,0]$ **THEN return** 1.
2. **IF** $l \geq u$**THEN return** *0.*
3. *Compute the tight bound $[l', u']$ for $(\psi|\phi)$ by Algorithm Tight_0_Consequence in Fig. 5. in [6].*
4. **IF** $l < l'$ **THEN** $l := l'$.
5. **IF** $u > u'$ **THEN** $u := u'$.
6. *Compute $s_p = \nu^{pos}_{P,(\psi|\phi)}(u')$ by Algorithm 2.*
7. *Compute $s_n = \nu^{neg}_{P,(\psi|\phi)}(u')$ by Algorithm 2.*
8. *Compute $s'_p = \nu^{pos}_{P,(\psi|\phi)}(l)$ by Algorithm 2.*
9. *Compute $s'_n = \nu^{neg}_{P,(\psi|\phi)}(u)$ by Algorithm 2.*
10. $sat := 0.5 * (s'_p/s_p + s'_n/s_n)$
11. **return** sat

In our querying system, shown in Figure 1, we have observations, background knowledge, as well as the knowledge obtained from sources (e.g. experts). Background knowledge and the knowledge from different sources are merged to obtain a PLP. Observations are used when constructing queries. Each PLP can be analyzed with the measures defined/introduced previously. The details on other components (like merging and revision) are omitted here due to space limitation.

5.2 Additional information used in querying PLPs

Observation vs. a priori facts: In PLPs, ground formulae of the form $(\phi(t)|\top)[1,1]$ are used to state a priori facts from statistics, i.e., something must be true (statistically) is regarded as a fact. From $(\phi(t)|\top)[1,1]$, we know that object t must possess property ϕ even before we observe it. This is different from observing t having property ϕ. Observing an event (such as a test result) about an individual does not infer that the event would happen for sure (for another individual). So, observations cannot be represented as formulae of the form $(\psi(a)|\top)[1,1]$ in a PLP. Doing so implies that we know $\psi(a)$ being true even before it is observed. In another word, taking $\psi(a)$ as a probabilistic event, we cannot predict if $\psi(a)$ is true or false before we observe it. In our System, all observations are stored in a separate database (named OBS). When querying $(\psi|\phi)[l,u]$ on PLP P, this observation database OBS is automatically called, so querying $(\psi|\phi)[l,u]$ is equivalent to querying $(\psi|\phi \wedge \bigwedge OBS)[l,u]$ on P.

Background knowledge: In practice, source knowledge bases (PLPs) can be obtained from experts, from some experiments, or are elicited from data in published papers. However, given an application, there is richer knowledge that is normally not included in a paper or stated in an experiment, but this knowledge may have been implicitly used. When such knowledge is present, we include it in a PLP when appropriate. For example, knowledge about some general population statistics should be treated as background knowledge, while the effectiveness of a new drug should be treated as specialized knowledge.

6 Application to Substrates Prediction

Considerable investment has been made into the in silico prediction of substrates, and especially, inhibitors of enzymes. This investment has been driven by a fundamental desire to understand more about how biomolecules recognize their ligands and by the commercial imperative to develop new drugs. Almost all pharmaceutical companies include an element of target-based approaches in their drug discovery programmes. The aim of our analyzing/querying system is to provide a very rapid screening for likely ligands (either substrates or inhibitors, depending on the context). It will be particularly useful in situations where a number of similar compounds have been screened experimentally, but information is not available for all possible members of that group of compounds. By providing a simple means to encode existing experimental knowledge and return results within minutes we see this as a valuable addition to initial computational screening approaches.

6.1 Case study I: Rapid sugar kinase enzymes prediction

Our first example is from biochemistry on the human enzyme galactokinase, which uses galactose as a substrate.

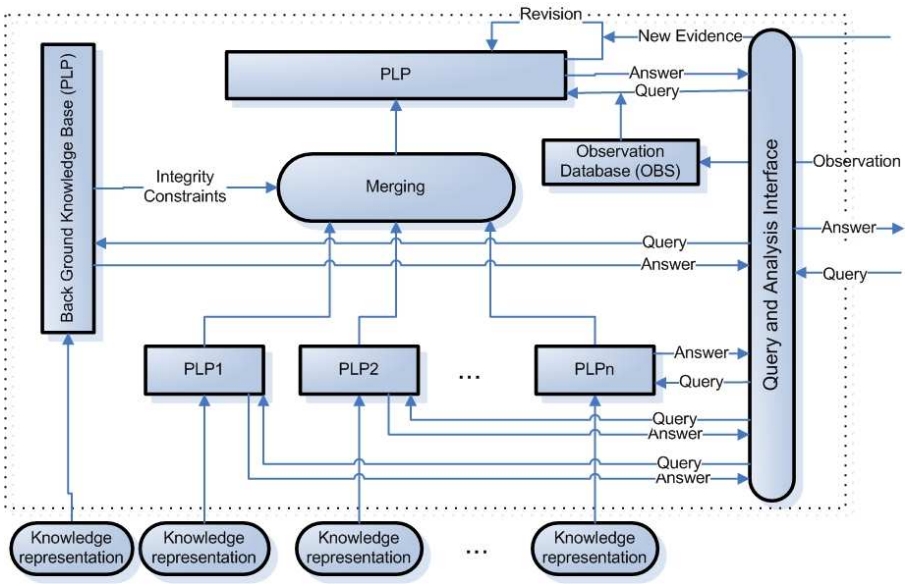


Figure 1: System Architecture

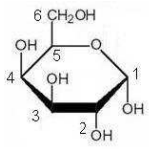


Figure 2: The α-D-Galactose molecule

Galactose has the molecular formula $C_6H_{12}O_6$, but other compounds have the same or similar formula. Since not all possible substrates for the enzyme have been tested, the information regarding this enzyme and its substrates is incomplete, can we then predict which will be the substrates for the human enzyme galactokinase based on incomplete and imperfect information? Many factors lead to the information being imperfect including different research laboratories using different criteria for scoring a compound as a substrate and some information is based on galactokinases from other species, so we cannot be certain that substrate specificity is conserved for humans.

Each galactose molecule is arranged as a hexagonal ring (e.g., the α-D-Galactose molecule in Figure 2). There are six carbon atoms in a galactose molecule and one oxygen atom. These six carbon atoms are numbered from 1 to 6 with the right-most carbon atom numbered 1, and then the remaining carbons are numbered clockwise round the ring. The oxygen atom is not numbered. The other atoms can be regarded as coming off these carbon atoms. The first four of the carbon atoms each has an OH molecule attached to it, and the fifth one has the sixth carbon atom attached to it from outside the ring, forming a CH_2OH group. The OH group can either be up or down (i.e. they are chiral). The combination of ups and downs gives a specific form of the molecule (in effect, each form of the molecule is a different compound), and the actual combination can significantly affect the biochemical behavior of the molecule. Therefore, for the OH groups attached to these atoms, we need to know if they are *up, down* or *absent*. The sixth carbon is not chiral, and so the OH is neither up nor down. Hence, the OH for the sixth carbon is marked as either *present* or *absent*. Current experimental results published in the literature provide a set of conditional events with probabilities suggesting how likely a particular structure is a substrate for the enzyme. Table 1 contains this knowledge collected from papers [14, 15, 16, 17], which is then translated into a PLP. For instance, the 5th row of the table (Talose) defines a probabilistic formula as

$$(sub(X)|c1(X,d) \wedge c2(X,u) \wedge c3(X,u) \\ \wedge c4(X,u) \wedge c5(X,u) \wedge c6(X,p))[0.4, 0.6]$$

Initially probabilities were estimated using experimental data and an element of intuition. Where a particular substrate had been demonstrated experimentally to be a substrate of human galactokinase it was assigned a probability of 1.0. Where there was experimental data indicating that a substrate was not phosphorylated by human galactokinase, a value of 0 was assigned. Compounds which had been shown to be substrates of galactokinase from other species were assigned probabilities between 0 and 1. However, not all substrates are equally good. Therefore a second measure, the *product* was calculated. To calculate this value, the specificity constant k_{cat}/K_m was used [4], scaled such that the product value with galactose (which is expected to be the best substrate) was equal to 1.0.

Therefore, in Table 1, we have a column representing their probabilities (or intervals) and another column representing their products of the corresponding compounds to be

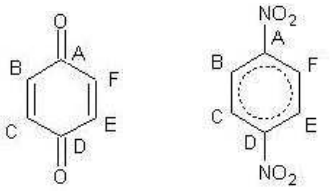
Sugar	C1 -OH	C2 -OH	C3 -OH	C4 -OH	C5 -CH_2OH	C6 -OH	P(substrate)	Product	Source
Galactose	D	D	U	U	U	P	1.0	1	[15]
Glucose	D	D	U	D	U	P	0.0	0	[15]
2-Deoxygalactose	D	A	U	U	U	P	1.0	0.47	[15]
Fucose	D	D	U	U	U	A	0.0	0	[15]
Talose	D	U	U	U	U	P	[0.4, 0.6]	[0.056,0.084]	[17, 14]
4-deoxygluocse	D	D	U	A	U	P	[0, 0.5]	[0,0.021]	[16]
3-deoxygalactose	D	D	A	D	U	P	[0.6, 0.9]	[0.036,0.054]	[16]

Table 1: The compounds and their probabilities and products to be substrates, obtained from published papers.

(good) substrates. Column *Source* indicates from which published paper this knowledge is obtained. Based on the probabilistic knowledge in the probabilistic logic program, we can predict the probability for any combination of these six carbons. Twenty-six queries detailed in Table 2 were executed against this PLP and the query results are presented in Table 2. Below we analysis these query results.

By querying the tight bounds for the twenty-six queries detailed in Table 6.1, we can only obtain a trivial interval $[0, 1]$ for all of these queries. This trivial interval indicates that we know nothing about the probability of a compound being a substrate. Reasoning under the maximum entropy principle, we can get probabilities as listed in Table 6.1. One question is that how reliable these values are to guide us finding most possible substrate from these possible compounds. With the analysis of knowledge entropy in Section 3, we have $K_P(sub(s)|\phi_s) = [0, 1]$ for any compound s where the compound structure is stated by ϕ_s. So this measurement does not provide us with useful information either. Now we look into the ignorance measures of these query results. The ignorance values of the query results of the twenty-six compounds w.r.t. this PLP are around 0.005, a very low value. These ignorance values suggest that the probabilities obtained by applying the maximum entropy principle are acceptable and can serve as good indicators about how likely a compound can be a substrate. Since in substrate prediction, the comparisons of probabilities are much more important than actual probability values, we do not need to calculate the degrees of satisfaction for these queries with intervals.

Overall, the predictions appear to over-estimate the probabilities for each possible substrate. For example, given that the *Fucose* (which has the OH group attached to the sixth carbon atom absent) has been shown experimentally not to be a substrate, it is surprising to see compounds which also lack this OH group predicted as having high probabilities as substrates. Of course in compiling the data in Table 1, all the information was weighted equally - for example the presence or absence of the OH group at position 6 was considered of equal worth to the information about the OH at position 2. In fact it is likely that some positions are more important than others in determining substrate specificity. However, in implementing screens such as these, the amount of knowledge to be included will always be a balance between including enough to enable valid predictions, but not so much that the initial knowledge collection and tabulation becomes unreasonably time consuming.



(a) (b)

Figure 3: Examples of NAD(H)-quinone oxidoreductase 1 (NQO1) substrates.

Despite these limitations, the predictions do appear to have some value in that the ranking of the compounds in terms of their probability of being a substrate seems mostly reasonable and in line with chemical intuition. Ultimately for such a system to be useful to bioscientists, it is this ranking which must be reliable. The most likely use of such a system is to act as a preliminary screen for potential substrates or inhibitors followed by experimental testing of those compounds. Time and expense can be saved if those compounds most likely to be good substrates (or inhibitors) appear at the top of the list and are, therefore, prioritized in the experimental work. Thus the absolute values of the predicted probabilities are less important than the rank order of the compounds.

6.2 Case Study II: Substrate prediction for NQO1

NAD(H)-quinone oxidoreductase 1 (NQO1) is a broad specificity enzyme which catalyses the reduction of a range of aromatic compounds. It was chosen for the second case study as a large variety of different compounds (including quinones, nitroaromatics and benzimidazoles) have been tested as substrates. In contrast to Case study I, the chemical diversity of the known substrates is wider leading to a greater number of variables to consider.

Two of the many compounds which have been tested experimentally as substrates for NQO1 are a quinone com-

Sugar	$C1$ -OH	$C2$ -OH	$C3$ -OH	$C4$ -OH	$C5$ -CH_2OH	$C6$ -OH	P(substrate)	Product
2dAll	D	A	D	D	U	P	0.6529	0.4611
2dGlc	D	A	U	D	U	P	0.6154	0.3939
2dGul	D	A	D	U	U	P	0.6694	0.5000
I	D	A	A	D	U	P	0.5869	0.4083
II	D	A	A	U	U	P	0.6676	0.5376
2,3,4d	D	A	A	A	U	P	0.5509	0.4721
3dAll	D	D	A	D	U	P	0.6003	0.1138
3dMan	D	U	A	D	U	P	0.5539	0.5000
3dTal	D	U	A	U	U	P	0.5636	0.4282
III	D	D	A	A	U	P	0.5321	0.3503
IV	D	U	A	A	U	P	0.5134	0.4785
4dAll	D	D	D	A	U	P	0.5314	0.4611
4dMan	D	U	U	A	U	P	0.4706	0.4282
V	D	A	D	D	U	A	0.5463	0.4811
VI	D	A	U	D	U	A	0.5481	0.4514
VII	D	A	D	U	U	A	0.5481	0.5000
VIII	D	A	A	D	U	A	0.5703	0.4572
IX	D	A	A	U	U	A	0.5682	0.5020
X	D	A	A	A	U	A	0.5233	0.4814
XI	D	D	A	D	U	A	0.5451	0.3518
XII	D	U	A	D	U	A	0.5234	0.5000
XIII	D	U	A	U	U	A	0.5278	0.4670
XIV	D	D	A	A	U	A	0.5146	0.4179
XV	D	U	A	A	U	A	0.5064	0.4895
XVI	D	D	D	A	U	A	0.5144	0.4811
XVIII	D	U	U	A	U	A	0.4879	0.4670

Table 2: The probabilities and products of some compounds being a substrate by querying on the PLP.

pound, benzo-1,4-quinone (Figure 3(a)) and a nitroaromatic compound 1,4-dinitrobenzene (Figure 3(b)). Representing these compounds in tabular form required assigning each position in the six-membered ring a letter descriptor from A to F. For each molecule, the most oxidised substituent was placed at the top of the structural representation and designated A. Positions B through F were then defined by moving round the ring sequentially in an anticlockwise fashion. In these initial studies we concentrated on six membered rings substituted with ketone, methyl and nitro groups.

A	B	C	D	E	F	Probability
NO2	H	H	H	H	H	[0,0]
NO2	H	NO2	H	H	H	[0,0]
NO2	H	H	CHO	H	H	[0,0]
NO2	NO2	H	H	H	H	[0,0]
NO2	H	H	NO2	H	H	[0,0]
O	H	H	O	H	H	[0.20,0.28]
O	CH3	H	O	H	H	[0.17,0.31]
O	CH3	H	O	CH3	H	[0.19,0.33]
O	CH3	CH3	O	CH3	H	[0.20,0.28]

Table 3: The compounds and their probability intervals, obtained from published papers [1, 3].

In this initial case study, knowledge was collected from a limited number of papers [1, 3] which described the activity of the enzyme towards a number of structurally related compounds (Table 3). Probabilities were derived from published data in these papers on specificity constants in which the error in the experimental determination was used to define the range of values. When used to make predictions about unknown compounds (Table 4), the results were broadly similar to those seen in Case Study I. Table 4 gives the summary of sixteen queries based on the probabilistic knowledge given in Table 3. There appeared to be a tendency to over-estimate probabilities (especially for compounds closely related in structure to those with low, or zero, experimentally determined activity). Nevertheless, if these compounds are excluded the rank order of the remaining ones appears sensible.

7 Related Work and Conclusion

Some systems are provided for modeling and querying on probabilistic knowledge, for example, SPIRIT [12] and PIT [13]. These two systems work on propositional probabilistic logics while our system works on PLPs. The main advantage of our framework is its ability to analyze the knowledge contained in PLPs, especially w.r.t. queries. For analyzing probabilistic knowledge bases, in [11, 10], the authors provided a second order uncertainty to measure the reliability of accepting the precise prob-

A	B	C	D	E	F	Probability
NO2	H	H	H	NO2	H	0.0000
NO2	H	H	NO2	CH3	H	0.3194
NO2	H	H	CHO	CH3	H	0.3194
NO2	H	H	O	CH3	H	0.3294
NO2	H	NO2	H	CH3	H	0.3217
NO2	H	NO2	NO2	H	H	0.1949
NO2	H	NO2	O	H	H	0.2235
NO2	NO2	H	O	H	H	0.2172
O	H	H	H	NO2	H	0.2949
O	H	H	NO2	CH3	H	0.3917
O	H	H	CHO	CH3	H	0.3197
O	H	H	O	CH3	H	0.3629
O	H	NO2	H	CH3	H	0.4000
O	H	NO2	NO2	H	H	0.3612
O	H	NO2	O	H	H	0.3477
O	NO2	H	O	H	H	0.3338

Table 4: The Predictions for some compounds.

ability obtained by applying maximum entropy principle as the answer to a query in propositional probabilistic logic. Their second order uncertainty is directly computed from the probability interval of the query inferred from P, and therefore is independent of the knowledge base. In contrast, our ignorance provides more information about the underlying knowledge base and is more accurate in terms of reflecting the knowledge in a PLP. In Example 3, the second order probabilities of $(fly(t)|magpie(t))$ and $(fly(t)|sickMagpie(t))$ are the same. However the ignorance values for the two queries are different.

In this paper, we provided a framework and a tool for reasoning with imprecise probabilistic logic programs. In our framework, background knowledge and application-specific knowledge are combined to create a PLP (or maybe multiple PLPs), and observations are represented in a separate set. In this tool, a user has an option to analyze the quality of a PLP by retrieving the ignorance values with respect to application-specific queries. Also, the reasoning power is enhanced because reliable informative bounds can be extracted for any query. Two case studies are deployed to demonstrate how this framework and the tool can be used in real world applications. Our system can also perform merging when multiple PLPs concerning the same application are available, and perform revision (of a PLP) when some sure new evidence is collected.

References

[1] Z Anusevicius, J Sarlauskas, and N Cenas. Two-electron reduction of quinones by rat liver nad(p)h:quinone oxidoreductase: quantitative structure-activity relationships. *Arch Biochem Biophys*, 404(2):254–262, 2002.

[2] C Baral and M Hunsaker. Using the probabilistic logic programming language p-log for causal and counterfactual reasoning and non-naive conditioning. In *Proc. of IJCAI*:243-249, 2007.

[3] N Cenas, A Nemeikaite-Ceniene, E Sergediene, H Nivinskas, Z Anusevicius, and J Sarlauskas. Quantitative structure-activity relationships in enzymatic single-electron reduction of nitroaromatic explosives: implications for their cytotoxicity. *Biochim Biophys Acta*, 1528(1):31–38, 2001.

[4] A Cornish-Bowden. *Fundamentals of Enzyme Kinetics (3rd Edition)*. Portland Press, London, UK, 2004.

[5] N Fuhr. Probabilistic datalog: Implementing logical information retrieval for advanced applications. *JASIS*, 51(2):95–110, 2000.

[6] G Kern-Isberner and T Lukasiewicz. Combining probabilistic logic programming with the power of maximum entropy. *Artificial Intelligence*, 157(1-2):139–202, 2004.

[7] T Lukasiewicz. Probabilistic logic programming. In *Proc. of ECAI*:388–392, 1998.

[8] T Lukasiewicz. Probabilistic logic programming with conditional constraints. *ACM Trans. Comput. Log.*, 2(3):289–339, 2001.

[9] L De Raedt, A Kimmig, and H Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *Proc. of IJCAI*:2462–2467, 2007.

[10] W Rödder. On the measurability of knowledge acquisition, query processing. *Int. J. Approx. Reasoning*, 33(2):203–218, 2003.

[11] W Rödder and G Kern-Isberner. From information to probability: An axiomatic approach - inference isinformation processing. *Int. J. Intell. Syst.*, 18(4):383–403, 2003.

[12] W Rödder, E Reucher, and F Kulmann. Features of the expert-system-shell spirit. *Logic Journal of the IGPL*, 14(3):483–500, 2006.

[13] M Schramm and V Fischer. Probabilistic reasoning with maximum entropy - the system pit (system description). In *WLP*, 1997.

[14] C Sellick and R Reece. Contribution of Amino Acid Side Chains to Sugar Binding Specificity in a Galactokinase, Gal1p, and a Transcriptional Inducer, Gal3p. *J. Biol. Chem.*, 281(25):17150–17155, 2006.

[15] DJ Timson and RJ Reece. Sugar recognition by human galactokinase. *BMC Biochemistry*, 4:16, 2003.

[16] J Yang, X Fu, Q Jia, J Shen, J Biggins, J Jiang, J Zhao, J Schmidt, P Wang, and J Thorson. Studies on the substrate specificity of escherichia coli galactokinase. *Organic Letters*, 5(13):2223–2226, 2003.

[17] J Yang, L Liu, and J Thorson. Structure-basedenhancement of the first anomeric glucokinase. *ChemBioChem*, 5(7):992–996, 2004.

[18] A Yue, W Liu, and A Hunter. Measuring the ignorance and degree of satisfaction for answering queries in imprecise probabilistic logic programs. In *Proc. of SUM*:386-400, 2008.