Marrying Uncertainty and Time in Knowledge Graphs

Melisachew Wudage Chekol
Data and Web Science Group, University of Mannheim
mel@informatik.uni-mannheim.de

Giuseppe Pirrò
Rende(CS), Italy
pirro@icar.cnr.it

Joerg Schoenfisch
Data and Web Science Group
University of Mannheim
joerg@dwslab.de

Heiner Stuckenschmidt
Data and Web Science Group
University of Mannheim
heiner@informatik.uni-mannheim.de

Abstract

The management of uncertainty is crucial when harvesting structured content from unstructured and noisy sources. Knowledge Graphs (KGs) are a prominent example. KGs maintain both numerical and non-numerical facts, with the support of an underlying schema. These facts are usually accompanied by a confidence score that witnesses how likely is for them to hold. Despite their popularity, most of existing KGs focus on static data thus impeding the availability of timewise knowledge. What is missing is a comprehensive solution for the management of uncertain and temporal data in KGs. The goal of this paper is to fill this gap. We rely on two main ingredients. The first is a numerical extension of Markov Logic Networks (MLNs) that provide the necessary underpinning to formalize the syntax and semantics of uncertain temporal KGs. The second is a set of Datalog constraints with inequalities that extend the underlying schema of the KGs and help to detect inconsistencies. From a theoretical point of view, we discuss the complexity of two important classes of queries for uncertain temporal KGs: *maximum a-posteriori* and *conditional probability inference*. Due to the hardness of these problems and the fact that MLN solvers do not scale well, we also explore the usage of Probabilistic Soft Logics (PSL) as a practical tool to support our reasoning tasks. We report on an experimental evaluation comparing the MLN and PSL approaches.

Introduction

Open Information Extraction (OIE) or machine reading has been announced as a new paradigm for extracting domain independent knowledge from large Web corpora (Banko et al. 2007; Etzioni et al. 2008; Pujara et al. 2013). OIE is of particular interest for the creation of knowledge graphs (KGs) and enriching existing ones (Pirrò 2015). KGs like Google's knowledge graph (Dong et al. 2014), NELL (Mitchell et al. 2015), and ReVerb (Fader, Soderland, and Etzioni 2011) store probabilistic facts, that is, facts along with their confidence scores witnessing how likely they are to hold. Indeed, the automated construction of KGs often produces noisy and inaccurate facts and rules with errors that can propagate upon inference or knowledge base expansion (Chen and Wang 2014). The harvesting of KGs poses some key challenges. The first concerns the need to *clean* KGs *from noisy facts* to avoid maintenance costs and provide reliable content. A limitation of existing methods (e.g., (Schlobach et al. 2007; Sirin et al. 2007)) is the lack of capabilities to deal with probabilistic and temporal information. This leads to situations where statements that refer to objects at different points in time are assumed to be inconsistent. In addition, little has been done in terms of techniques to debug uncertain KGs, with the exception of the preliminary results in (Chekol et al. 2016; Huber, Meilicke, and Stuckenschmidt 2014; Chen and Wang 2014; Dylla, Sozio, and Theobald 2011). The second challenge is about *providing temporal information*. Most of existing approaches have focused on identifying static facts encoded as binary relations (Fionda, Gutierrez, and Pirrò 2016). However, the vast majority of facts are fluents (dynamic relations whose truth is a function of time), only holding during an interval of time. Facts like *(ClaudioRanieri, coach, Chelsea)* loose relevance without a temporal scope (*2000–2004* in this case). Inference or deduction rules and consistency checking constraints are useful to both derive implicit/new facts from existing ones and constrain or identify conflicting facts. As an example, the date of birth of a person is functional. The broad goal of this paper is to tackle the above challenges and study *uncertain temporal* KGs. Specifically, we tackle the following general problem:

Problem 1 *Given an uncertain temporal* KG $\mathcal{G}$*, a set of temporal inference rules* $\mathcal{F}$*, and a set of temporal constraints* $\mathcal{C}$*, what is the most probable and error free temporal* KG*?*

Related work. Preliminary results that highlight the use of Markov Logic Networks (MLNs) to debug temporal knowledge bases are presented in (Chekol et al. 2016; Huber, Meilicke, and Stuckenschmidt 2014). The idea is to use handcrafted temporal constraints to identify conflicts in knowledge bases containing date and time datatype values. However, this study: (i) does not provide a formal characterization in terms of syntax and semantics, (ii) only considers a subset of RDF(S) inference rules, and (iii) does not consider constraints for debugging numerical attributes.

Dylla et al (Dylla, Miliaraki, and Theobald 2013) extend probabilistic databases with a temporal dimension. Besides, in an earlier version (Dylla, Sozio, and Theobald 2011), authors proposed an approach for resolving temporal conflicts in RDF knowledge bases. The idea is to use first-order logic Horn formulas with temporal predicates to express temporal

and non-temporal constraints. However, these approaches are limited to a small set of temporal patterns and only allow for uncertainty in facts. Moreover, extending knowledge graphs using open domain information extraction, will often also lead to uncertainty about the correctness of schema information; a large variety of temporal inference rules and constraints, some of which will be domain specific, can also be the subject of uncertainty. Finally, Chen and Wang (Chen and Wang 2014) debug erroneous facts by using a set of functional constraints although they do not deal with numerical and temporal facts at the same time.

Despite the general complexity of MLNs, it has been shown that it can be used to reason about facts extracted at Web scale using a combination of hand-crafted (Schoenmackers, Etzioni, and Weld 2008) and extracted inference rules (Schoenmackers et al. 2010). MLNs can be used to deal with temporal relations in open information extraction (Ling and Weld 2010) or check the consistency of knowledge bases (Chen and Wang 2014). Building upon this experience, we make use of an extension of MLNs to provide a formal characterization of uncertain temporal KGs. Our contributions are the following: (i) a formal syntax and semantics, based on a numerical extension of MLN, for uncertain temporal KGs along with a set of temporal inference rules, (ii) a formalization of the maximum a-posteriori and conditional probability inference problems in uncertain temporal KGs along with a study that shows how these problems remain NP-hard and #P-hard respectively, and (iii) a set of constraints to clean erroneous facts in KGs. To support the theoretical results, we carry out a set of experiments using state-of-the-art MLN solvers and their scalable variants.

Preliminaries

We now briefly outline KGs and MLNs along with their temporal and numerical extensions, respectively. We also discuss probabilistic soft logic (PSL).

Knowledge Graphs

For ease of exposition we assume KGs to be encoded in the W3C standard RDF data model (Hayes 2004). Let $\mathcal{I}$ and $\mathcal{L}$ be two disjoint infinite sets denoting the set of IRIs (identifying resources) and literals (character strings or some other type of data), respectively. We abbreviate the union of these sets ($\mathcal{I} \cup \mathcal{L}$) as $\mathcal{IL}$. A triple of the form $(s, p, o) \in \mathcal{I} \times \mathcal{I} \times \mathcal{IL}$ is called an *RDF triple*[1]; s is the *subject*, p is the *predicate*, and o is the *object* of the triple. Each triple can be thought of as an edge between the subject and the object labeled by the predicate; hence a set of RDF triples is referred to as an *RDF graph*. We use the term *knowledge graph* loosely to refer to an RDF graph.

Temporal Knowledge Graphs. In (Motik 2012; Gutierrez, Hurtado, and Vaisman 2005), it has been shown that an RDF graph can be extended with temporal information by labeling each triple in the graph with a temporal element. The temporal element represents the time period in which the triple is valid, i.e., the *valid time* of the triple. We consider a discrete time domain $\mathcal{T}$ as a linearly ordered finite sequence of *time points*; for instance, days, minutes, or milliseconds. The finite domain assumption ensures that there are finitely many possible worlds in MLNs. A *time interval* is an ordered pair $[t_1, t_2]$ of time points, with $t_1 \leq t_2$ and $t_1, t_2 \in \mathcal{T}$, which denotes the closed interval from t_1 to t_2. We will work with the interval-based temporal domain for defining our data model. Note that point-based temporal domains can be converted into interval-based domains by using for every time point t an interval $[t, t]$.

Definition 1 (Temporal KG) *A temporal* KG *is a* KG *where each fact (s, p, o) in the graph has a valid time $[t_1, t_2]$, i.e., $\mathtt{f} = (s, p, o, [t_1, t_2])$. We refer to* $\mathtt{f}$ *as a temporal fact.*

For a temporal KG G, its *snapshot* at time t is the graph $G(t)$ (the non-temporal KG): $G(t) = \{(s, p, o) \mid (s, p, o, [t, t]) \in G\}$. The KG associated with a temporal KG, denoted $u(G)$, is $\bigcup_t G(t)$, the union of the graphs $G(t)$. We define *temporal entailment* as follows: for temporal KGs G_1 and G_2, $G_1 \models_t G_2$ if $G_1(t) \models G_2(t)$ for each t; $\models_t$ denotes temporal entailment (Gutierrez, Hurtado, and Vaisman 2005) and $\models$ is the standard RDF entailment (Hayes 2004). We use MLNs to extend temporal KGs with uncertainty.

Markov Logic Networks

Markov Logic Networks (MLNs) combine Markov networks and first-order logic (FOL) by attaching weights to first-order formulas and treating them as templates for features of Markov networks (Richardson and Domingos 2006). MLNs have been extended with numerical (Chekol et al. 2016) and continuous (Wang and Domingos 2008) constraints. In this paper, we will use the numerical extension, specifically *MLN with numerical constraints*, which is useful for reasoning in uncertain temporal KGs.

Definition 2 *A numerical constraint* NC *is composed of numerical constants (such as elements of natural numbers $\mathbb{N}$, integers $\mathbb{I}$, and so on), variables, elementary operators or functions (such as, $+$, $*$, $-$, $\div$, $\%$, $\sqrt{\ }$), standard relations ($>$, $<$, $=$, $\neq$, $\geq$, $\leq$), and boolean operators ($\wedge$, $\vee$, $\neg$). An MLN L with numerical constraints (simply MLN) is a set of pairs (FC_i, w_i) where FC_i is a FOL formula that may contain a* NC *and w_i is a real number representing the weight of formula FC_i.*

Together with a finite set of constants C, a MLN with numerical constraints defines a Markov Network $M_{L,C}$, where $M_{L,C}$ contains one node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground predicate is true, and 0 otherwise. The probability distribution over possible worlds x, specified by the ground Markov network $M_{L,C}$, is:

$$P(X = x) = \frac{1}{Z} \exp\Big(\sum_{i=1}^{F} w_i n_i(x)\Big)$$

where F is the number of formulas in the MLN and $n_i(x)$ is the number of true groundings of FC_i in x. The groundings of a formula are formed simply by replacing its variables with constants in all possible ways.

[1] We do not consider blank nodes.

Example 1 *Using MLN it is possible to represent the hard constraint: footballers born before 1850 are not alive:* $footballer(a) \wedge bdate(a, y) \wedge \mathsf{NC}(y) \rightarrow dead(y)$, $\mathsf{NC}(y) = y < 1850$.

A common inference task with MLNs is finding *the most probable state* of the world, i.e., finding a complete assignment to all ground atoms which maximizes the probability. This is known as maximum a-posteriori inference (MAP). Finding a most likely world of an MLN is a generalization of the (NP-hard) MaxSAT problem. Another equally important inference problem is conditional probability inference. This is the task of computing the probability of a set of variables given evidence. The complexity of this problem is known to be #P-hard (Richardson and Domingos 2006).

Our experimental findings indicate that MLN solvers do not scale well. This comes as no surprise due to the complexity of inference in MLN; thus we exchange their expressiveness for scalability and choose to use our extended probabilistic soft logic (PSL) solver in the experiments.

Probabilistic Soft Logic

Probabilistic Soft Logic (PSL) uses first-order logic to specify templates for probabilistic graphical models. MLNs are defined over Boolean variables whereas PSL is defined over random variables with soft truth values in the interval $[0, 1]$. In addition, PSL formulas are restricted to rules with conjunctive bodies. PSL is a template language for HR-MRFs (Hinge-Loss Markov Random Fields) that are defined over continuous variables (Bach et al. 2015). Unlike MLNs, PSL does not support negative weights. We chose PSL over Tractable Markov Logic (TML) (Domingos and Webb 2012) because it retains most of the rich expressiveness of MLN while being scalable. On the other hand, TML imposes heavy restrictions on the structure of the KG to achieve tractability. Due to this, most of the constraints and rules that we use for experimentation are not applicable to TML and its variants. By using probabilistic graphical models (for instance, MLN and PSL), it is possible to represent uncertainty in temporal KGs.

Reasoning in Uncertain Temporal KGs

Uncertain temporal knowledge graphs (UTKGs) are extensions of temporal KGs with probabilistic graphical models that are capable of representing uncertainties and reasoning over temporal knowledge bases. A UTKG is a temporal knowledge graph where each fact has a valid-time and an associated weight or confidence. In other words, each temporal fact has a confidence value.

Syntax. A UTKG graph $\mathcal{G} = (\mathsf{D}, \mathsf{U})$ consists of a deterministic (*hard*) temporal KG D and an uncertain (*soft*) temporal KG U with $\mathsf{D} \cap \mathsf{U} = \emptyset$. $\mathsf{U} = \{\langle \mathtt{f}_i, w_{\mathtt{f}_i} \rangle\}$ where $\mathtt{f}_i$ is a temporal fact and $w_{\mathtt{f}_i}$ is a real-valued weight assigned to $\mathtt{f}_i$. The syntax of an uncertain temporal fact is similar to the underlying temporal RDF; besides, each fact has an associated weight, written as $\{(\mathtt{f}_i, w_{\mathtt{f}_i})\}$.

Example 2 *Consider the following* UTKG *which represents sport's personality Claudio Raineri's (CR) career:*

```
(1) (CR, coach, Chelsea, [2000,2004])      0.9
(2) (CR, coach, Leicester, [2015,2016])    0.7
(3) (CR, playsFor, Palermo, [1984,1986])   0.5
(4) (CR, bdate,1951)                       1.0
(5) (CR, coach, Napoli, [2001,2003])       0.6
```

Before providing semantics to UTKGs, we need to extend the use of membership ($\in$) and subset ($\subseteq$) relations as follows: given a UTKG $\mathcal{G}$, a temporal fact $(s, p, o, [t_1, t_1'])$, and a UTKG $\mathcal{G}'$, we say that $(s, p, o, [t_1, t_1']) \in \mathcal{G}$ if $\exists (s, p, o, [t_2, t_2']) \in \mathcal{G}$ such that $t_2 \leq t_1$ and $t_1' \leq t_2'$; we also say that $\mathcal{G}' \subseteq \mathcal{G}$ if for all $\mathtt{f} \in \mathcal{G}'$, then $\mathtt{f} \in \mathcal{G}$.

Semantics. The semantics of a UTKG is based on a joint probability distribution over the uncertain part of the UTKG. In particular, the weights of the facts in U determine a log-linear probability distribution. As mentioned earlier, we assume that the time domain, in which the validity of facts is expressed, is finite as well as discrete; hence, the set of possible worlds is finite. Formally, for a given UTKG $\mathcal{G} = (\mathsf{D}, \mathsf{U})$ and some $\mathcal{G}'$ over the same set of IRIs and literals $\mathcal{IL}$, the probability of $\mathcal{G}'$ is defined as:

$$P(\mathcal{G}') = \begin{cases} \frac{1}{Z} \exp\left(\sum_{\{(\mathtt{f}_i, w_{\mathtt{f}_i}) \in \mathsf{U} : \mathcal{G}' \models_t \mathtt{f}_i\}} w_{\mathtt{f}_i} \right) & \text{if } \mathcal{G}' \models_t \mathsf{D} \\ 0 \qquad \text{otherwise} \end{cases}$$

where $\models_t$ is a temporal entailment relation, and Z is the normalization constant of the log-linear probability distribution P. Note that in MAP inference, which gives the most probable temporal KG, Z is not computed. A UTKG can be mapped into a first-order knowledge base by transforming every temporal fact into a quad atom as shown in Definition 3.

Herbrand Models. Temporal KG inference rules[2] $\mathcal{F}$ are listed in Figure 1. Let $\mathcal{C}$ be the set of IRIs and Literals that appear in some UTKG $\mathcal{G}$, the Herbrand base of $\mathcal{F}$ can be constructed by instantiating all the variables in $\mathcal{F}$ using the constants in $\mathcal{C}$. The function $\boldsymbol{\theta}$, given a finite set $\mathcal{C}$ and a set of time points $\mathcal{T}$, maps each fact in some UTKG into a subset of the Herbrand base HB of $\mathcal{F}$ with respect to $\mathcal{C}$ and $\mathcal{T}$. Each subset of the Herbrand base is a Herbrand interpretation specifying which ground atoms are true. A Herbrand interpretation H is a Herbrand model of $\mathcal{F}$, denoted as $\models_H \mathcal{F}$, iff it satisfies all groundings of the formulas in $\mathcal{F}$.

Definition 3 (Mapping UTKG **into FOL)** *Given a* UTKG $\mathcal{G}$ *over a finite set of IRIs and literals $\mathcal{C}$, a time domain $\mathcal{T}$, and HB the Herbrand base of $\mathcal{F}$ with respect to $\mathcal{C}$ and $\mathcal{T}$, $\boldsymbol{\theta} : \mathcal{P}(\mathcal{G}) \rightarrow \mathcal{P}(HB)$ maps $\mathcal{G}$ into subsets of HB as follows:*

$$\boldsymbol{\theta}(\mathcal{G}) = \bigcup_{\mathtt{f} \in \mathcal{G}} \boldsymbol{\theta}(\mathtt{f}), \text{ where } \boldsymbol{\theta}((s, p, o, T)) = \mathtt{quad}(s, p, o, T).$$

The predicate quad is typed, i.e., $s, p \in \mathcal{I}$, $o \in \mathcal{IL}$, and $T = [t_1, t_1']$ where $t_1, t_1' \in \mathcal{T}$. At this point, we need to show that the function $\boldsymbol{\theta}$ is bijective, i.e., it induces a one-to-one correspondence between the Herbrand models of $\mathcal{F}$

[2] In Figure 1, we abbreviate RDF/S vocabulary names as follows: *sp* for `rdfs:subPropertyOf`, *type* for `rdf:type`, *property* for `rdf:Property`, *sc* for `rdfs:subClassOf`, *class* for `rdfs:Class`, *dom* for `rdfs:domain`, and *ran* for `rdfs:range`.

(r_1) $\mathtt{q}(a, \alpha, property, T_1) \rightarrow \mathtt{q}(a, sp, a, T_1)$

(r_2) $\mathtt{q}(a, sp, b, T_1) \wedge \mathtt{q}(b, sp, c, T_2) \wedge check(T_1, T_2) \rightarrow \mathtt{q}(a, sp, c, T_3)$

(r_3) $\mathtt{q}(a, sp, b, T_1) \wedge \mathtt{q}(x, a, y, T_2) \wedge check(T_1, T_2) \rightarrow \mathtt{q}(x, b, y, T_3)$

(r_4) $\mathtt{q}(a, \alpha, class, T_1) \rightarrow \mathtt{q}(a, sc, a, T_1)$

(r_5) $\mathtt{q}(a, sc, b, T_1) \wedge \mathtt{q}(b, sc, c, T_2) \wedge check(T_1, T_2) \rightarrow \mathtt{q}(a, sc, c, T_3)$

(r_6) $\mathtt{q}(a, sc, b, T_1) \wedge \mathtt{q}(x, \alpha, a, T_2) \wedge check(T_1, T_2) \rightarrow \mathtt{q}(x, \alpha, b, T_3)$

(r_7) $\mathtt{q}(a, dom, c, T_1) \wedge \mathtt{q}(x, a, y, T_2) \wedge check(T_1, T_2) \rightarrow \mathtt{q}(x, \alpha, c, T_3)$

(r_8) $\mathtt{q}(a, ran, d, T_1) \wedge \mathtt{q}(x, a, y, T_2) \wedge check(T_1, T_2) \rightarrow \mathtt{q}(y, \alpha, d, T_3)$

$$T_3 = [t_1, t_1'] \bowtie [t_2, t_2'] = \begin{cases} [t_1, t_1'] & \text{if } t_1 = t_2 \wedge t_1' = t_2' \\ [t_1', t_2] & \text{if } t_1' = t_2 \\ [t_2, t_1'] & \text{if } t_1 < t_2 \wedge t_2 < t_1' \wedge t_1' < t_2' \\ [t_1, t_1'] & \text{if } t_1 < t_2 \wedge t_1' < t_2' \\ [t_1, t_1'] & \text{if } t_1 = t_2 \wedge t_1' < t_2' \\ [t_2, t_2'] & \text{if } t_1' < t_1 \wedge t_2 = t_2' \\ \emptyset & \text{if } t_1' < t_2 \end{cases}$$

Figure 1: A set of temporal RDF inference rules that we denote by $\mathcal{F}$. $check(T_1, T_2) = false$ if $T_1 \bowtie T_2 = \emptyset$ and $true$ otherwise. α denotes the RDF *type* relation and q is a shorthand for quad. Moreover, all of the formulas are universally quantified over all the variables.

and expanded KGs. Applying $\mathcal{F}$ repeatedly on an UTKG may generate a set of new facts; this results in an *expanded* KG.

Theorem 1 *Let $\mathcal{C} \subseteq \mathcal{IL}$ be a set of IRIs and literals and let $\mathcal{T}$ be a set of time points. In addition, let $\mathcal{G}$ be a* UTKG *over $\mathcal{C}$ and let HB be the Herbrand base of $\mathcal{F}$ with respect to $\mathcal{C}$. Then, for any $\mathcal{G}' \subseteq \mathcal{G}$, $\mathcal{G} \models_t \mathcal{G}' \Rightarrow \boldsymbol{\theta}(\mathcal{G}') \models_H \mathcal{F}$ and for any $H \subseteq HB$, $H \models_H \mathcal{F} \Rightarrow \boldsymbol{\theta}^{-1}(H) \models \mathcal{G}''$ and $\mathcal{G} \models_t \mathcal{G}''$.*

Relying on the above theorem, we can introduce MAP inference in UTKGs.

MAP Inference

MAP inference in UTKG corresponds to obtaining the most probable, consistent, and non-probabilistic temporal KG. Given a UTKG $\mathcal{G}$, a set of inference rules $\mathcal{F}$, and a translation function $\boldsymbol{\theta}$, we denote the MAP problem by $map(\boldsymbol{\theta}(\mathcal{G}), \mathcal{F})$. Computing $map(\boldsymbol{\theta}(\mathcal{G}), \mathcal{F})$ requires to translate $\mathcal{G}$ with the function $\boldsymbol{\theta}$ into an equivalent Markov logic formalization. Then, the inference rules $\mathcal{F}$ are added to this translation. The MAP state is computed with the help of a cutting planes algorithm (Chekol et al. 2016) applied to this input data. To do so, the evidence clauses $\boldsymbol{\theta}(\mathcal{G})$ and the grounding of $\mathcal{F}$ with respect to $\boldsymbol{\theta}(\mathcal{G})$ are given as input. Applying the inverse translation function $\boldsymbol{\theta}^{-1}$ to the MAP state, yields the most probable temporal KG. The MAP problem in MLN can be turned into an integer linear program (Noessner, Niepert, and Stuckenschmidt 2013), which allows to integrate external functions (e.g., to check the conditions in Figure 1).

Theorem 2 *Given the following:*

- *a* UTKG $\mathcal{G} = (\mathsf{D}, \mathsf{U})$ *over a finite set $\mathcal{IL}$ of IRIs and literals, and a finite set of time points $\mathcal{T}$,*
- *the Herbrand base HB of the formulas $\mathcal{F}$ with respect to $\mathcal{IL}$ and $\mathcal{T}$,*
- *the set of ground formulas $\mathcal{G}_1$ constructed from* D*, and*
- *the set of ground formulas $\mathcal{G}_2$ constructed from* U.

The most probable, expanded and consistent *temporal* KG *is obtained with:*

$$\boldsymbol{\theta}^{-1}(H) = \underset{HB \supseteq H \models \mathcal{G}_1 \cup \mathcal{F}}{\arg\max} \left(\sum_{(\mathtt{f}, w_j) \in \mathcal{G}_2 : H \models_{\mathit{it}} \mathtt{f}_j} w_j \right)$$

From Theorem 1 and the results in (Chekol et al. 2016) it follows that the problem of computing the most probable temporal KG is NP-hard.

Example 3 (MAP state) *Given a* UTKG*, which contains the uncertain temporal facts (1)–(5) of Example 2 and the hard temporal constraints (6) and (7) below, its most probable and consistent temporal* KG *contains the facts (1)–(4).*

- *A person cannot be a coach of two clubs at the same time.*
  (6) $\mathtt{quad}(x, coach, y, T_1) \wedge \mathtt{quad}(x, coach, z, T_2) \wedge y \neq z \rightarrow disjoint(T_1, T_2)$
- *A person cannot be a coach before s/he was born.*
  (7) $\mathtt{quad}(x, bdate, y, T_1) \wedge \mathtt{quad}(x, coach, z, T_2) \rightarrow before(T_1, T_2)$

The predicates disjoint *and* before *are Allen's interval relations (Allen 1983). Below, we introduce expressive constraints that allow to identify erroneous facts.*

Conditional Probability Inference

Given a UTKG $\mathcal{G}$, the conditional probability of a temporal fact f is the sum of the probabilities of the consistent temporal KGs containing f. In general, a *conditional probability query* is a conjunction of a set of temporal facts given some UTKG. Given a query q and a UTKG $\mathcal{G}$, the conditional probability of q is given by:

$$P_q(q \mid \mathcal{G}) = \sum_{\mathcal{G}' : q \subseteq \mathcal{G}'} P(\mathcal{G}')$$

$\mathcal{G}'$ is a possible world over the same signature $\mathcal{IL}$ and $\mathcal{T}$ as $\mathcal{G}$. In order to sum over all $\mathcal{G}'$, if the valid time ranges of the temporal facts in q does not appear in $\mathcal{G}$, we need to compare time intervals in the facts of q with those of $\mathcal{G}$. To do so, we rewrite the query q as follows: for each temporal fact $\mathtt{f} \in q$ if $\exists \mathtt{f}' \in \mathcal{G}$ and that $\mathtt{f} \subseteq^+ \mathtt{f}'$, then we replace f in q with f′. The relation $\subseteq^+$ is defined as follows: for two temporal facts $\mathtt{f}{=}(s, p, o, [t_1, t_1'])$ and $\mathtt{f}'{=}(s', p', o', [t_2, t_2'])$, $\mathtt{f} \subseteq^+ \mathtt{f}'$ if $s{=}s'$, $p{=}p'$, $o{=}o'$, $t_2 \leq t_1$ and $t_1' \leq t_2'$. This allows us to compute conditional probabilities on top of current solvers such as MC-SAT (Poon and Vanderwende 2010). The rewriting can be done in polynomial time in the size of the UTKG in the worst case. For instance, given the KG $\mathcal{G}$ in Example 2, the conditional query: $\mathtt{q}(CR, coach, Chelsea, [2001, 2003] \mid \mathcal{G})$ is rewritten as: $P_q(\mathtt{q}(CR, coach, Chelsea, [2000, 2004]) \mid \mathcal{G})$. Since no additional computation is required, the complexity of conditional probability inference remains #P-hard for UTKGs. Since conditional inference is intractable, computing exact probabilities is hard. Thus, it is customary to approximate inference via sampling. The state of the art marginal

inference algorithm is MC-SAT, which is based on Monte Carlo sampling and samples consistent or conflict-free temporal KGs according to the distribution P_q. This is very difficult for three reasons: (i) the complexity of reasoning in MLN; (ii) the size of uncertain KGs (such as NELL, ReVerb), and; (iii) the presence of deterministic dependencies in the UTKGs. Because of these reasons, emerging lifted inference techniques should be used for marginal inference (Singla and Domingos 2008). We leave this as a future work.

Conflict Detection in Uncertain KGs

Often uncertain knowledge graphs may contain a large number of numerical data like dates, times, latitudes/longitudes, numerical values measured in different units, and so on. For instance, the fact that Claudio Ranieri is 1.82 meters tall can be expressed as $(CR, height, 1.82)$ with a numeric data (1.82). Uncertain facts that contain numerical data can be conflicting. One way of resolving such errors is to use a set of (probabilistic) constraints and compute a MAP state of a given KG, which basically throws out facts that have inferior weights or confidences. However, this is not enough. Consider, for instance, an uncertain KG that contains two facts: (1) $\langle (CR, height, 1.80), 0.3\rangle$ and (2) $\langle (CR, height, 3.5), 0.9\rangle$. Assume that these facts are translated into an MLN framework along with the constraint that the property 'height' is functional, i.e., $t(x, height, y) \wedge t(x, height, y') \rightarrow y = y'$. In this setting, performing MAP inference results in a KG containing the certain fact $(CR, height, 3.5)$. However, the correct output should contain only the first triple because normally people are not taller than 2.5 meters. In order to rule out such conflicts, we can add another constraint as discussed below. Constraints are used in description logics and database systems to ensure data validity. In the following, we introduce constraints to ensure validity of numerical attributes in uncertain KGs. The constraints will also serve to extend the schema of the underlying KG.

Constraints A Datalog constraint is an expression of the form $body \rightarrow head$, where the *head* is an atom (i.e., an expression of the form $p(x_1, \ldots, x_n)$ in which each x_i is either a constant or a variable) and *body* is a set of atoms, such that each variable occurring in the *head* also occurs in some atom in the *body* (Abiteboul and Vianu 1991). Since our choice of MLN with numerical constraints allows to use external functions, whose truth values are computed outside the MLN setting, we can extend Datalog constraints (specifically, *inclusion dependencies*, *equality generating dependencies* and *negative constraints* (Abiteboul and Vianu 1991)) with numerical constraints. To debug uncertain KGs we can introduce a set of Datalog-inspired constraints that become *hard* (deterministic) or *soft* (uncertain) formulas in MLNs. For instance, if we want to state that "a person cannot be taller that 2.5 meters", then we can introduce a rule of the form: $t(x, type, person) \wedge t(x, height, y) \rightarrow y < 2.5$. We introduce three different kinds of constraints.

Inclusion dependencies with inequalities (IDIs). IDIs are first-order logic formulas of the form $\forall \mathbf{x}, \mathbf{y} : \Phi(\mathbf{x}, \mathbf{y}) \wedge \mathsf{NC}(\mathbf{x}_i, \mathbf{y}_j) \rightarrow \Psi(\mathbf{y})$, where $\Phi(\mathbf{x}, \mathbf{y})$ is the body of the formula, it is a conjunction of atoms, $\Psi(\mathbf{y})$ is the head of the formula, $\mathbf{x}, \mathbf{y}$ are sets of variables, and $\mathbf{x}_i \subseteq \mathbf{x}$ and $\mathbf{y}_j \subseteq \mathbf{y}$. In addition, $\mathsf{NC}(\mathbf{x}_i, \mathbf{y}_j)$ denotes a numerical constraint which is an arithmetic expression (see Definition 2).

Example 4 *Those who are above the age of 40 are probably retired footballers:* $t(x, type, Footballer) \wedge t(x, age, y) \wedge \mathsf{NC}(y) \rightarrow t(x, type, RFootballer)$, $\mathsf{NC}(y) = y > 40$.

(In)equality generating dependencies (IGDs). IGDs are first-order formulas of the form $\forall \mathbf{x}{:}\Phi(\mathbf{x}) \rightarrow \mathsf{NC}(\mathbf{x}_i)$, where $\Phi(\mathbf{x})$ is a conjunction of atoms ($\mathbf{x}$ and $\mathbf{x}_i$ are defined as above).

Example 5 *Temperature Celsius tc can be converted into an equivalent Fahrenheit scale tf using the formula* $tf = 1.8tc + 32$*:* $t(x, tempc, tc) \wedge t(x, tempf, tf) \rightarrow \mathsf{NC}(tc, tf)$, $\mathsf{NC}(tc, tf) = 1.8tc + 32$. *From a practical viewpoint, this rule can be used for checking if two facts (e.g., extracted from Wikipedia), one containing temperature in Celsius format and the other in Fahrenheit, are conflicting.*

Disjointness constraints (DCs). DCs are first-order formulas of the form $\forall \mathbf{x} : \Phi(\mathbf{x}) \wedge \mathsf{NC}(\mathbf{x}_i) \rightarrow \bot$.

Example 6 *Using DCs we can formulate the constraint "a valid life span of a person is less than 150 years" as follows:* $t(x, bdate, bd) \wedge t(x, ddate, dd) \wedge \mathsf{NC}(bd, dd) \rightarrow \bot$, $\mathsf{NC}(bd, dd) = (dd - bd) > 0 \wedge (dd - bd) < 150$.

These constraints are more expressive than RDF schema constraints because they allow to express disjointness, functionality of properties, and inverse properties, among the others. Once an uncertain KG is translated into an equivalent Markov logic formalism using the formula $\boldsymbol{\theta}$, and sets of IDIs, IGDs, and DCs constraints over the KG have been constructed, we can apply MAP inference in order to retrieve the most probable and conflict-free KG using $map(\boldsymbol{\theta}(\mathcal{G}), \mathcal{F}, \mathcal{C})$.

Experiment

We conducted two different kinds of experiments: (i) performance test in terms of running times for MAP inference comparing three state-of-the-art solvers, and (ii) conflict detection in a noisy setting. We ran the experiments on a 2GHz 24-core processor with 386GB of RAM running Debian 8.

Data: At present, uncertain temporal datasets are not available. Thus, we prepared datasets by extracting temporal facts from **footballdb.com** and **wikidata.org**. Therefore, we can test the efficiency and scalability of the proposed approach. In addition, we experimented with YAGO (Galárraga et al. 2015) to mine temporal rules as discussed below.

- **Footballdb**: Table data often contain numeric and temporal data. Recently, table data extraction has attracted considerable attention from the data mining community (e.g., (Ritze et al. 2016)). Inspired by this, we extracted temporal facts about American football players from footballdb.com, that contains two important relations: *playsFor* and *birthdate*. We extracted >13K temporal facts for the *playsFor* relation and >6K facts for the birthdate relation.

- **Wikidata**: Wikidata contains structured temporal information obtained from various sources using OIE. At the time of writing, we extracted over 6.3 million temporal facts from Wikidata. We extracted temporal facts for various relations including: *playsFor* (>4 million facts), *educatedAt* (>6K), *memberOf* (>23K), *occupation* (>4.5K), *spouse* (>20K), and so on.

Constraints: The extraction of temporal rules/constraints is a well-known problem (Galárraga et al. 2015). Using one of the most famous rule mining tools (i.e., AIME (Galárraga et al. 2015)), we mined rules from the YAGO dataset (see below). However these rules are not sufficient to capture many conflicts (e.g., valid life span of a person, or a footballer cannot play for two clubs at the same time); thus we handcrafted more complex constraints that are used to identify conflicts in UTKGs. We used several of these constraints in order to detect conflicts in footballdb and wikidata KGs.

- **Mined**: With a workaround to AIME, we were able to learn rules of the following form from the YAGO dataset.

```
(a) A person's birth date is before his
death date.
?e bd ?a  ?e dd ?b => ?a before ?b 0.968
(b) Birth and death date are functional
?e bd ?a  ?e bd ?b => ?a equal ?b  0.734
?e dd ?a  ?e dd ?b => ?a equal ?b  0.686
```

- **Hand-crafted**: We use the rules in Figure 1 and designed 20 different constraints including the ones already discussed in the examples (see for instance Example 7).

Tools: We used the following tools to conduct the experiments: (i) two state-of-the-art MLN solvers, namely *Tuffy* (Niu et al. 2011) and *nRockIt* (Chekol et al. 2016), and PSL solver (Bach et al. 2015); (ii) we *implemented* a numerical extension, that we call *nPSL*, on top of PSL for temporal reasoning. In our experiments, we found out that Tuffy and nRockIt hardly scale for predicates of arity 4 (we stopped the execution after a 24h timeout). In one occasion, while running Tuffy on footballdb, we noticed that its grounded database is 400GB large, thereby our execution eventually ran out of memory. To overcome this shortcoming, we resort to PSL. While MLN is more expressive than PSL, PSL scales well since it computes a soft approximation of the discrete MAP state. Note that, using PSL, we compute the most probable explanation (MPE) which is the same as MAP inference in MLN (Bach et al. 2015).

Performance: MAP Inference

We computed the running times of nRockIt, Tuffy and nPSL for performing MAP inference on an uncertain temporal footballdb data and obtained 12181ms, *did not terminate after several hours*, 6129.2ms respectively. The running times are averaged over 10 runs. Independently, we performed MPE inference using nPSL on varying sizes of uncertain temporal Wikidata and obtained the following results:

Data size:	50K	100K	200K	400K	500K
Time (m):	0.33	1.44	3.85	8.72	11.25

Injection		0%	10%	25%	50%	75%	100%
nPSL	P	1.00	0.93	0.85	0.70	0.63	0.53
	R	1.00	0.94	0.86	0.81	0.72	0.70
Time (m)		0.77	0.86	0.93	1.15	1.37	1.56
nRockIt	P	1.00	0.91	0.83	0.67	0.55	0.50
	R	1.00	0.95	0.94	0.94	0.94	0.93
Time (m)		3.05	4.4	5.74	6.53	6.92	11.1

Table 1: Precision (P) and recall (R) scores for computing the MPE and MAP state with increasing percentage of wrong temporal facts injection.

Conflict Detection in UTKGs

In this experiment we used nPSL and nRockIt for conflict detection on a subset of Wikidata containing 50K uncertain temporal facts. Due to space constraints, we only report results on this data size. However, we run both tools on various data sizes. We excluded Tuffy due to scalability after attempting repeatedly with varying configurations were not successful. We generated erroneous temporal facts specifying the playsFor, birthdate, and deathdate relations. We injected a fraction of 10%, 25%, 50%, 75%, and 100% incorrect facts to the Wikidata dataset. For instance, injecting 25% erroneous facts means that we added 25% additional wrong facts for each of the three relations. We randomly assigned weights in the range [0.5, 1.0] to the newly added facts and [0.8, 1.0] to each of the original temporal facts.

The results of our experiments are shown in Table 1. Since we randomly assigned weights, we repeated each experiment 10 times and present average scores. We were able to compute meaningful results in highly inconsistent settings. Even in a setting where we added 100% incorrect temporal facts, we are still able to achieve a precision of 53% for nPSL and 50% for nRockIt. Note that in the case of nRockIT, the runtime does not increase linearly with respect to the size of the input data. This is due to each added incorrect temporal fact might be involved in a conflict resulting in a non trivial optimization problem.

Conclusions and Future Work

We have presented an MLN based approach for reasoning over UTKGs. We proposed a formal syntax and semantics for UTKGs and formalized the MAP and conditional probability inference problems; we showed that these problems remain NP-hard and #P-hard, respectively. We used Datalog constraints to detect erroneous facts in UTKGs. Then, we applied MAP inference to obtain a most probable and conflict-free temporal KG from an uncertain one. We carried out experiments on state-of-the-art tools and datasets and reported their performances.

While we think that this work opens several research directions; our primary objective is to tackle scalability. This can be investigated, for instance, (1) by parallelizing MLN solvers, or (2) by using relational database model for SQL-based inference. Taking account preferences (Fionda and Pirrò 2013) is another line of future work.

References

Abiteboul, S., and Vianu, V. 1991. Datalog extensions for database queries and updates. *Journal of Computer and System Sciences* 43(1):62–124.

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843.

Bach, S. H.; Broecheler, M.; Huang, B.; and Getoor, L. 2015. Hinge-loss markov random fields and probabilistic soft logic. arXiv:1505.04406 [cs.LG].

Banko, M.; Cafarella, M. J.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open information extraction for the web. In *IJCAI*, volume 7, 2670–2676.

Chekol, M. W.; Huber, J.; Meilicke, C.; and Stuckenschmidt, H. 2016. Markov logic networks with numerical constraints. In *ECAI 2016*, 1017–1025.

Chen, Y., and Wang, D. Z. 2014. Knowledge expansion over probabilistic knowledge bases. In *SIGMOD*, 649–660. ACM.

Domingos, P. M., and Webb, W. A. 2012. A Tractable First-Order Probabilistic Logic. In *AAAI*.

Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion. In *SIGKDD*, 601–610.

Dylla, M.; Miliaraki, I.; and Theobald, M. 2013. A temporal-probabilistic database model for information extraction. *Proc. of the VLDB Endowment* 6(14):1810–1821.

Dylla, M.; Sozio, M.; and Theobald, M. 2011. Resolving Temporal Conflicts in Inconsistent RDF Knowledge Bases. In *BTW*, 474–493.

Etzioni, O.; Banko, M.; Soderland, S.; and Weld, D. S. 2008. Open Information Extraction from the Web. *Communications of the ACM* 51(12):68–74.

Fader, A.; Soderland, S.; and Etzioni, O. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*.

Fionda, V., and Pirrò, G. 2013. Querying Graphs with Preferences. In *Proc. of International Conference on Information and Knowledge Management*, 929–938. ACM.

Fionda, V.; Gutierrez, C.; and Pirrò, G. 2016. Building Knowledge Maps of Web Graphs. *Artificial Intelligence* 239:143–167.

Galárraga, L.; Teflioudi, C.; Hose, K.; and Suchanek, F. M. 2015. Fast Rule Mining in Ontological Knowledge Bases with AMIE+. *The VLDB Journal* 24(6):707–730.

Gutierrez, C.; Hurtado, C.; and Vaisman, A. 2005. Temporal RDF. In *Proc. of European Semantic Web Conference*, 93–107.

Hayes, P. 2004. RDF Semantics. W3C Recommendation.

Huber, J.; Meilicke, C.; and Stuckenschmidt, H. 2014. Applying Markov Logic for Debugging Probabilistic Temporal Knowledge Bases. In *Proceedings of the 4th Workshop on Automated Knowledge Base Construction (AKBC)*.

Ling, X., and Weld, D. S. 2010. Temporal information extraction. In *AAAI*, volume 10, 1385–1390.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; Krishnamurthy, J.; Lao, N.; Mazaitis, K.; Mohamed, T.; Nakashole, N.; Platanios, E.; Ritter, A.; Samadi, M.; Settles, B.; Wang, R.; Wijaya, D.; Gupta, A.; Chen, X.; Saparov, A.; Greaves, M.; and Welling, J. 2015. Never-Ending Learning. In *AAAI*.

Motik, B. 2012. Representing and querying validity time in rdf and owl: A logic-based approach. *J. Web Semantics* 12:3–21.

Niu, F.; Ré, C.; Doan, A.; and Shavlik, J. 2011. Tuffy: Scaling up statistical inference in markov logic networks using an rdbms. *Proc. of the VLDB Endowment* 4(6):373–384.

Noessner, J.; Niepert, M.; and Stuckenschmidt, H. 2013. Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models. In *AAAI*.

Pirrò, G. 2015. Explaining and Suggesting Relatedness in Knowledge Graphs. In *Proc. of International Semantic Web Conference*, 622–639.

Poon, H., and Vanderwende, L. 2010. Joint inference for knowledge extraction from biomedical literature. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 813–821. Association for Computational Linguistics.

Pujara, J.; Miao, H.; Getoor, L.; and Cohen, W. 2013. Knowledge graph identification. In *Proc. of International Semantic Web Conference*.

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine learning* 62(1-2):107–136.

Ritze, D.; Lehmberg, O.; Oulabi, Y.; and Bizer, C. 2016. Profiling the Potential of Web Tables for Augmenting Cross-domain Knowledge Bases. In *Proc. of International World Wide Web Conference*, 251–261.

Schlobach, S.; Huang, Z.; Cornet, R.; and Van Harmelen, F. 2007. Debugging incoherent terminologies. *Journal of Automated Reasoning* 39(3):317–349.

Schoenmackers, S.; Etzioni, O.; Weld, D. S.; and Davis, J. 2010. Learning first-order horn clauses from web text. In *EMNLP*, 1088–1098.

Schoenmackers, S.; Etzioni, O.; and Weld, D. S. 2008. Scaling Textual Inference to the Web. In *EMNLP*, 79–88.

Singla, P., and Domingos, P. M. 2008. Lifted first-order belief propagation. In *AAAI*, volume 8, 1094–1099.

Sirin, E.; Parsia, B.; Grau, B. C.; Kalyanpur, A.; and Katz, Y. 2007. Pellet: A Practical OWL-DL Reasoner. *J. Web Semantics* 5(2):51–53.

Wang, J., and Domingos, P. M. 2008. Hybrid markov logic networks. In *AAAI*, volume 8, 1106–1111.