frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 19 February 2018
doi: 10.3389/frobt.2018.00008

Bridging Weighted Rules and Graph Random Walks for Statistical Relational Models

*Seyed Mehran Kazemi** and David Poole*

Laboratory of Computational Intelligence, Computer Science Department, University of British Columbia, Vancouver, BC, Canada

The aim of statistical relational learning is to learn statistical models from relational or graph-structured data. Three main statistical relational learning paradigms include weighted rule learning, random walks on graphs, and tensor factorization. These paradigms have been mostly developed and studied in isolation for many years, with few works attempting at understanding the relationship among them or combining them. In this article, we study the relationship between the path ranking algorithm (PRA), one of the most well-known relational learning methods in the graph random walk paradigm, and relational logistic regression (RLR), one of the recent developments in weighted rule learning. We provide a simple way to normalize relations and prove that relational logistic regression using normalized relations generalizes the path ranking algorithm. This result provides a better understanding of relational learning, especially for the weighted rule learning and graph random walk paradigms. It opens up the possibility of using the more flexible RLR rules within PRA models and even generalizing both by including normalized and unnormalized relations in the same model.

Keywords: statistical relational artificial intelligence, relational learning, weighted rule learning, graph random walk, relational logistic regression, path ranking algorithm

OPEN ACCESS

Edited by:
Sriraam Natarajan,
Indiana University System,
United States

Reviewed by:
Elena Bellodi,
University of Ferrara, Italy
Nicola Di Mauro,
Università degli studi di Bari Aldo Moro, Italy

****Correspondence:***
Seyed Mehran Kazemi
smkazemi@cs.ubc.ca

Specialty section:
This article was submitted to Computational Intelligence, a section of the journal Frontiers in Robotics and AI

Received: *17 October 2017*
Accepted: *18 January 2018*
Published: *19 February 2018*

Citation:
Kazemi SM and Poole D (2018) Bridging Weighted Rules and Graph Random Walks for Statistical Relational Models. Front. Robot. AI 5:8. doi: 10.3389/frobt.2018.00008

1. INTRODUCTION

Traditional machine learning algorithms learn mappings from a feature vector indicating categorical and numerical features to an output prediction of some form. Statistical relational learning (Getoor and Taskar, 2007), or statistical relational AI (StarAI) (De Raedt et al., 2016), aims at probabilistic reasoning and learning when there are (possibly various types of) relationships among the objects. The relational models developed in StarAI community have been successfully applied to several applications such as knowledge graph completion (Lao et al., 2011; Nickel et al., 2012; Bordes et al., 2013; Pujara et al., 2013; Trouillon et al., 2016), entity resolution (Singla and Domingos, 2006; Bhattacharya and Getoor, 2007; Pujara and Getoor, 2016; Fatemi, 2017), tasks in scientific literature (Lao and Cohen, 2010b), stance classification (Sridhar et al., 2015; Ebrahimi et al., 2016), question answering (Khot et al., 2015; Dries et al., 2017), etc.

During the past two decades, three paradigms of statistical relational models have appeared. The first paradigm is the weighted rule learning where first-order rules are learned from data and a weight is assigned to each rule indicating a score for the rule. The main difference among these models is in the types of rules they allow and their interpretation of the weights. The models in this paradigm include Problog (De Raedt et al., 2007), Markov logic (Domingos et al., 2008), probabilistic interaction logic (Hommersom and Lucas, 2011), probabilistic soft logic (Kimmig et al., 2012), and relational logistic regression (Hommersom and Lucas, 2011).

The second paradigm is the random walk on graphs, where several random walks are performed on a graph each starting at a random node and probabilistically transitioning to neighboring nodes. The probability of each node being the answer to a query is proportional to the probability of the random walks ending up at that node. The main difference among these models is in the way they walk on the graph and how they interpret obtained results from the walks. Examples of relational learning algorithms based on random walk on graphs include PageRank (Page et al., 1999), FactRank (Jain and Pantel, 2010), path ranking algorithm (Lao and Cohen, 2010b; Lao et al., 2011), and HeteRec (Yu et al., 2014).

The third paradigm is the tensor factorization paradigm, where for each object and relation an embedding is learned. The probability of two objects participating in a relation is a simple function of the objects' and relation's embeddings (e.g., the sum of the element-wise product of the three embeddings). The main difference among these models is in the type of embeddings and the function they use. Examples of models in this paradigm include YAGO (Nickel et al., 2012), TransE (Bordes et al., 2013), and ComplEx (Trouillon et al., 2016).

The models in each paradigm have their own advantages and disadvantages. Kimmig et al. (2015) survey the models based on weighted rule learning. Nickel et al. (2016) survey models in all paradigms for knowledge graph completion. Kazemi et al. (2017) compare several models in these paradigms for relational aggregation. None of these surveys, however, aims at understanding the relationship among these paradigms. In fact, these paradigms have been mostly developed and studied in isolation with few works aiming at understanding the relationship among them or combining them (Riedel et al., 2013; Nickel et al., 2014; Lin et al., 2015).

With several relational paradigms/models developed during the past decade and more, understanding the relationship among them and pruning the ones that either do not work well or are subsets of the other models is crucial. In this article, we study the relationship between two relational learning paradigms: graph random walk and weighted rule learning. In particular, we study the relationship among path ranking algorithm (PRA) (Lao and Cohen, 2010b) and relational logistic regression (RLR) (Kazemi et al., 2014). The former is one of the most well-known relational learning tools in graph random walk paradigm, and the latter is one of the recent developments in weighted rule learning paradigm. By imposing restrictions on the rules that can be included in models, we identify a subset of RLR models that we call RC-RLR. Then we provide a simple way to normalize relations and prove that PRA models correspond to RC-RLR models using normalized relations. Other strategies for walking randomly on the graph (e.g., data-driven path finding (Lao et al., 2011)) can then be viewed as structure learning methods for RC-RLR. Our result can be extended to several other weighted rule learning and graph random walk models.

The relationship between weighted rules and graph random walks has not been discovered before. For instance, Nickel et al. (2016) describe them as two separate classes of models for learning from relational data in their survey. Lao et al. (2011) compare their instance of PRA to a model based on weighted rules empirically, reporting their PRA model outperforms the weighted rule model, but not realizing that their PRA model could be a subset of the weighted rule model if they had normalized the relations.

Our result is beneficial for both graph random walk and weighted rule learning paradigms, as well as for researchers working on theory and applications of statistical relational learning. Below is a list of potential benefits that our results provide:

- It provides a clearer intuition and understanding on two relational learning paradigms, thus facilitating further improvements of both.
- It opens up the possibility of using the more flexible RLR rules within PRA models.
- It opens up the possibility of generalizing both PRA and RLR models by using normalized and unnormalized relations in the same model.
- It sheds light on the shortcomings of graph random walk algorithms and points out potential ways to improve them.
- One of the claimed advantages of models based on weighted rule learning compared to other relational models is that they can be easily explained to a broad range of people (Nickel et al., 2016). Our result improves the explainability of models learned through graph random walk, by providing a weighted rule interpretation for them.
- It identifies a subclass of weighted rules that can be evaluated efficiently and have a high modeling power as they have been successfully applied to several applications. The evaluation of these weighted rules can be even further improved using sampling techniques developed within graph random walk community (e.g., see Fogaras et al. (2005); Lao and Cohen (2010a); Lao et al. (2011)). Several structure learning algorithms (corresponding to random walk strategies) have been already developed for this subclass.
- It facilitates leveraging new insights and techniques developed within each paradigm (e.g., weighted rule models that leverage deep learning techniques (Šourek et al., 2015; Kazemi and Poole, 2018), or reinforcement learning-based approaches to graph walk (Das et al., 2017)) to the other paradigm.
- For those interested in the applications of relation learning, our result facilitates decision-making on selecting the paradigm or the relational model to be used in their application.

2. BACKGROUND AND NOTATIONS

In this section, first we define some basic terminology. Then we introduce a running example, which will be used throughout the article. Then we describe relational logistic regression and path ranking algorithm for relational learning. While semantically identical, our descriptions of these two models may be slightly different from the descriptions in the original articles as we aim at describing the two algorithms in a way that simplifies our proofs.

2.1. Terminologies

Throughout the article, we assume True is represented by 1 and False is represented by 0.

A **population** is a finite set of objects (or individuals). A **logical variable (logvar)** is typed with a population. We represent logvars with lower case letters. The population associated with a logvar

x is Δ_x. The cardinality of Δ_x is $|\Delta_x|$. For every object, we assume that there exists a unique *constant* denoting that object. A lower case letter in bold represents a tuple of logvars, and an upper case letter in bold represents a tuple of constants. An **atom** is of the form $\mathsf{V}(t_1, \ldots, t_k)$, where V is a functor, and each t_i is a logvar or a constant. When *range*$(\mathsf{V}) \in \{0,1\}$, V is a predicate. A **unary** atom contains exactly one logvar, and a **binary** atom contains exactly two logvars. We write a **substitution** as $\theta = \{\langle x_1, \ldots, x_k\rangle / \langle t_1, \ldots, t_k\rangle\}$, where each x_i is a different logvar and each t_i is a logvar or a constant in Δ_{x_i}. A **grounding** of an atom $\mathsf{V}(x_1, \ldots, x_k)$ is a substitution $\theta = \{\langle x_1, \ldots, x_k\rangle / \langle X_1, \ldots, X_k\rangle\}$ mapping each of its logvars x_i to an object in Δ_{x_i}. Given a set $\mathcal{A}$ of atoms, we denote by $\mathcal{G}(\mathcal{A})$ the set of all possible groundings for the atoms in $\mathcal{A}$. A **value assignment** for a set of groundings $\mathcal{G}(\mathcal{A})$ maps each grounding $\mathsf{V}(X) \in \mathcal{G}(\mathcal{A})$ to a value in *range*(V).

A **literal** is an atom or its negation. A **formula** φ is a literal, a disjunction $\varphi_1 \vee \varphi_2$ of formulae or a conjunction $\varphi_1 \wedge \varphi_2$ of formulae. Our formulae correspond to open formulae in negation normal form in logic. An **instance** of a formula φ is obtained by replacing each logvar x in φ by one of the objects in Δ_x. Applying a **substitution** $\theta = \{\langle x_1, \ldots, x_k\rangle / \langle t_1, \ldots, t_k\rangle\}$ on a formula φ (written as $\varphi\theta$) replaces each x_i in φ with t_i. A *weighted formula (WF)* is a pair $\langle w, \varphi\rangle$ where w is a weight and φ is a formula.

A binary predicate $\mathsf{S}(x, y)$ can be viewed as a function whose domain is Δ_x and whose range is 2^{Δ_y}: each $X \in \Delta_x$ is mapped to $\{Y : \mathsf{S}(X, Y)\}$. Following Lao and Cohen (2010b), we consider S^{-1} as the inverse of S whose domain is Δ_y and whose range is $2^{\Delta x}$, such that S^{-1} (x, y) holds iff $\mathsf{S}(y, x)$ holds. A **path relation** $\mathcal{PR}$ is of the form $x_0 \xrightarrow{\mathsf{R}_1} x_1 \xrightarrow{\mathsf{R}_2} \ldots \xrightarrow{\mathsf{R}_l} x_l$, where R_1, R_2, ... R_l are predicates, $x_0, \ldots, x_l$ are different logvars, *domain*$(\mathsf{R}_i) = \Delta_{x_{i-1}}$ and *range*$(\mathsf{R}_i) = \Delta_{x_i}$. We define *domain*$(\mathcal{PR}) = \Delta_{x_0}$ and *range*$(\mathcal{PR}) = \Delta_{x_l}$. Applying a substitution $\theta = \{\langle x_1, \ldots, x_k\rangle / \langle t_1, \ldots, t_k\rangle\}$ on a path relation $\mathcal{PR}$ (written as $\mathcal{PR}\theta$) replaces each x_i in $\mathcal{PR}$ with t_i. A **weighted path relation (WPR)** is a pair $\langle w, \mathcal{PR}\rangle$, where w is a weight and $\mathcal{PR}$ is a path relation.

2.2. Running Example

As a running example, we use the *reference recommendation* problem: finding relevant citations for a new paper. We consider three populations: the population of new papers for which relevant citations are to be found, the population of existing papers whose citations are known, and the population of publication years. The atoms used for this problem throughout the article are the following. $\mathsf{WillCite}(q, p)$ is the atom to be predicted and indicates whether a query/new paper q will cite an existing paper p. Cited (p_1, p_2) shows whether an existing paper p_1 has cited another existing paper p_2. $\mathsf{PubIn}(p, y)$ shows that p has been published in year y. $\mathsf{ImBef}(y_1, y_2)$ indicates that y_2 is the year immediately before y_1. The reference recommendation problem can be viewed as follows: given a query paper Q, find a subset of existing papers that Q will cite (i.e., find any paper P such that $\mathsf{WillCite}(Q, P)$ holds).

2.3. Relational Logistic Regression

Relational logistic regression (Kazemi et al., 2014) defines conditional probabilities based on weighted rules. It can be viewed as the directed analog of logistic regression and as the directed analog of Markov logic (Domingos et al., 2008).

Let $\mathsf{V}(\mathbf{x})$ be an atom whose probability depends on a set $\mathcal{A}$ of atoms, ψ be a set of WFs containing only atoms from $\mathcal{A}$, $\hat{I}$ be a value assignment for the groundings in $\mathcal{G}(\mathcal{A})$, $\mathbf{X}$ be an assignment of objects to x, and $\{\mathbf{x}/\mathbf{X}\}$ be a substitution mapping logvars $\mathbf{x}$ to objects $\mathbf{X}$.

Relational logistic regression (RLR) defines the probability of $\mathsf{V}(\mathbf{X})$ given $\hat{I}$ as follows:

$$Prob_{\psi}(\mathsf{V}(\mathbf{X}) = True \mid \hat{I}) = \sigma\left(\sum_{\langle w, \varphi\rangle \in \psi} w * \eta(\varphi\{\mathbf{x}/\mathbf{X}\}, \hat{I})\right) \quad (1)$$

where $\eta(\varphi\{\mathbf{x}/\mathbf{X}\}, \hat{I})$ is the number of instances of $\varphi\{\mathbf{x}/\mathbf{X}\}$ that are True with respect to $\hat{I}$ and σ is the sigmoid function. RLR makes the closed-world assumption: any ground atom that has not been observed to be True is False. Note that $\eta(\mathsf{True}, \hat{I}) = 1$.

Following Kazemi et al. (2014) and Fatemi et al. (2016), we assume that formulae in WFs have no disjunction and replace conjunction with multiplication. Then atoms whose functors have a continuous range can be also allowed in formulae. For instance, if a value assignment maps $\mathsf{R}(X)$ to 1, $\mathsf{S}(X)$ to 0.9 and $\mathsf{T}(X)$ to 0.3, then the formula $\mathsf{R}(X) * \mathsf{S}(X) * \mathsf{T}(X)$ evaluates to $1 * 0.9 * 0.3 = 0.27$.

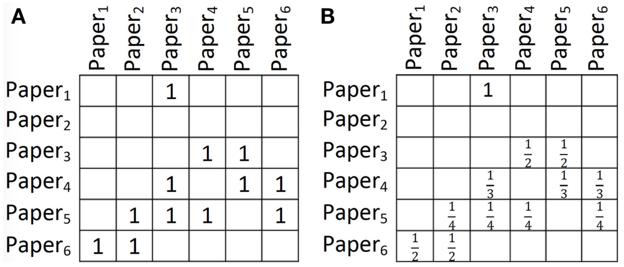
Example 1: An RLR model may use the following WFs to define the conditional probability of $\mathsf{WillCite}(q, p)$ in our running example:

$$WF_0 : \langle w_0, \mathsf{True}\rangle$$
$$WF_1 : \left\langle w_1, \mathsf{PubIn}(q, y) * \mathsf{ImBef}(y, y') * \mathsf{PubIn}(p, y')\right\rangle$$
$$WF_2 : \left\langle w_2, \mathsf{PubIn}(q, y) * \mathsf{PubIn}(p', y) * \mathsf{Cited}(p', p)\right\rangle$$
$$WF_3 : \langle w_3, \mathsf{Cited}(p_1, p_2) * \mathsf{Cited}(p_2, p)\rangle$$

WF_0 is a bias. WF_1 considers existing papers that have been published a year before the query paper. A positive weight for this WF indicates that papers published a year before the query paper are more likely to be cited. WF_2 considers existing papers cited by the other papers published in the same year as the query paper. A positive weight for this WF indicates that as the number of times a paper has been cited by the other papers published in the same year as the query paper grows, the chances of the query paper citing that paper increases. WF_3 considers existing papers that have been cited by other papers that have been themselves cited by other papers. Note that the score of the last WF depends only on the paper being cited not on the paper citing.

Consider the citations among existing papers in **Figure 1A**, and let the publication year for all the six papers be 2017. Suppose we have a query paper Q that is to be published in 2017 and we want to find the probability of $\mathsf{WillCite}(Q, Paper_2)$ according to the WFs above. Applying the substitution $\{\langle q, p\rangle / \langle Q, Paper_2\rangle\}$ to the above four WFs gives the following four WFs, respectively:

$$WF_0 : \langle w_0, \mathsf{True}\rangle$$
$$WF_1 : \left\langle w_1, \mathsf{PubIn}(Q, y) * \mathsf{ImBef}(y, y') * \mathsf{PubIn}(Paper_2, y')\right\rangle$$
$$WF_2 : \left\langle w_2, \mathsf{PubIn}(Q, y) * \mathsf{PubIn}(p', y) * \mathsf{Cited}(p', Paper_2)\right\rangle$$
$$WF_3 : \left\langle w_3, \mathsf{Cited}(p_1, p_2) * \mathsf{Cited}(p_2, Paper_2)\right\rangle .$$



A

	$Paper_1$	$Paper_2$	$Paper_3$	$Paper_4$	$Paper_5$	$Paper_6$
$Paper_1$			1			
$Paper_2$						
$Paper_3$				1	1	
$Paper_4$			1		1	1
$Paper_5$		1	1	1		1
$Paper_6$	1	1				

B

	$Paper_1$	$Paper_2$	$Paper_3$	$Paper_4$	$Paper_5$	$Paper_6$
$Paper_1$			1			
$Paper_2$						
$Paper_3$				$\frac{1}{2}$	$\frac{1}{2}$	
$Paper_4$			$\frac{1}{3}$		$\frac{1}{3}$	$\frac{1}{3}$
$Paper_5$		$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$		$\frac{1}{4}$
$Paper_6$	$\frac{1}{2}$	$\frac{1}{2}$				

FIGURE 1 | **(A)** A relation showing citations among papers (papers on the *Y* axis cite papers on the *X* axis). **(B)** The relation in part **(A)** after row-wise count normalization.

Then we evaluate each WF. The first one evaluates to w_0. The second evaluates to 0 as Q is being published in 2017 and $Paper_2$ has also been published in 2017. The third WF evaluates to $w_2 * 2$ as there are 2 papers that have been published in the same year as Q and cite $Paper_2$. The last WF evaluates to $w_3 * 4$ as $Paper_5$ and $Paper_6$ (that cite $Paper_2$) are each cited by two other papers. Therefore, the conditional probability of WillCite(Q, $Paper_2$) is as follows:

$$\sigma\left(w_0 + w_2 * 2 + w_3 * 4\right).$$

2.4. Path Ranking Algorithm

Let $\mathsf{V}(s, e)$ be a target binary predicate, i.e., for a query object $S \in \Delta_s$, we would like to find the probability of any $E \in e$ having the relation V with S. **Path ranking algorithm (PRA)** (Lao and Cohen, 2010b) defines this probability using a set of WPRs ψ. The first logvar of each path relation in ψ is either s or a logvar other than s and e, the last logvar is always e, and the middle logvars are neither s nor e.

In PRA, each path relation $\mathcal{PR} = x_0 \xrightarrow{\mathsf{R}_1} x_1 \xrightarrow{\mathsf{R}_2} \ldots \xrightarrow{\mathsf{R}_l} e$ defines a distribution over the objects in Δ_e. This distribution corresponds to the probability of following $\mathcal{PR}$ and landing at each of the objects in Δ_e and is computed as follows. First, a uniform distribution D_0 is considered on the objects in Δ_{x_0}, corresponding to the probability of landing at each of these objects if the object is selected randomly. For instance, if there are α objects in Δ_{x_0}, D_0 for all objects is $\frac{1}{\alpha}$. Then, the distribution D_1 over the objects in Δ_{x_1} is calculated by marginalizing over the variables in D_0 and following a random step on R_1. For instance, for an object $X_1 \in \Delta_{x_1}$, assume R_1 (x_0, X_1) holds only for two objects X_0 and X'_0 in Δ_{x_0}. Also assume X_0 and X'_0 have the R_1 relation with β and γ objects in x_1, respectively. Then the probability of landing at X_1 is $\frac{1}{\alpha} * \frac{1}{\beta} + \frac{1}{\alpha} * \frac{1}{\gamma}$. The following distributions $D_2, \ldots, D_l$ can be computed similarly. D_l gives the probability of landing at any object in Δ_e.

Let $\theta = \{\langle s, e\rangle / \langle S, E\rangle\}$. To find $Prob(\mathsf{V}(S, E))$, for each path relation $\mathcal{PR} \in \psi$, PRA calculates the probability of landing at E according to $\mathcal{PR}\theta$ (denoted by $h(\mathcal{PR}\theta)$) and calculates $Prob(\mathsf{V}(S, E))$ by taking the sigmoid of the weighted sum of these probabilities as follows:

$$Prob(\mathsf{V}(S, E)) = \sigma\left(\sum_{\langle w, \mathcal{PR}\rangle \in \Psi} w \cdot h(\mathcal{PR}\theta)\right) \tag{2}$$

ALGORITHM 1 | $h(\mathcal{PR})$.

```
Input: Relation path $\mathcal{PR} = x_0 \xrightarrow{R_1} x_1 \xrightarrow{R_2} \ldots \xrightarrow{R_l} x_l$
Output: Probability of landing at any object in $\Delta_{x_l}$ when starting randomly at any object in $\Delta_{x0}$ and walking on $\mathcal{PR}$.
if $l = 0$ then
    return $uniform(\Delta_{x0})$
$\mathcal{PR}' = x_0 \xrightarrow{R_1} x_1 \xrightarrow{R_2} \ldots \xrightarrow{R_{l-1}} x_{l-1}$
$pLand_{l-1} = h(\mathcal{PR}')$
for $E \in range(\mathcal{PR})$do
    $pLand_l\ (E) = 0$
for $E^{'} \in range(\mathcal{PR}')$ do
    $C_{R_l}(E') = \#E \in range(\mathcal{PR})$ s.t. $R_l(E', E) = 1$
    for $E \in range(\mathcal{PR})$ do
        $pWalk(E', E) = \frac{R_l(E', E)}{C_{R_l}(E')}$
        $pLand_l(E) \mathrel{+}= pLand_{l-1}(E') * pWalk(E', E)$
return $pLand_l$
```

Algorithm 1 shows a recursive algorithm for calculating $h(\mathcal{PR})$ for a path relation $\mathcal{PR}$. The first if statement specifies that the walk starts randomly at any object in Δ_{x_0}. *uniform* (Δ_{x_0}) indicates a uniform probability over the objects in Δ_{x_0}. This is the termination criterion of the recursion. When $\mathcal{PR} = x_0 \xrightarrow{\mathsf{R}_1} x_1 \xrightarrow{\mathsf{R}_2} \ldots \xrightarrow{\mathsf{R}_l} x_l$ is not empty ($l \neq 0$), first the probability of landing at any object E' in the range of $\mathcal{PR}_l = x_0 \xrightarrow{\mathsf{R}_1} x_1 \xrightarrow{\mathsf{R}_2} \ldots \xrightarrow{\mathsf{R}_{l-1}} x_{l-1}$ is calculated using a recursive call to $h(\mathcal{PR}')$ and stored in $pLand_{l-1}$. The probability of landing at any object E in range of $\mathcal{PR}$ by randomly walking on $\mathcal{PR}$ can then be calculated as the sum of the probabilities of landing at each object $E^{'}$ by randomly walking on $\mathcal{PR}^{'}$ multiplied by the probability of reaching E from $E^{'}$ by a random walk according to the predicate R_l. The two nested for loops calculate the probability of landing at any object $E \in range(\mathcal{PR})$ according to R_l. R_l (E', E) indicates whether there is a link from E' to E (otherwise the probability of transitioning from E' to E according to R_l is 0), and C_{R_l} is a normalization constant indicating the number of possible transitions from E' according to R_l. $pWalk(E', E)$ indicates the probability of walking from E' to E if one of the objects connected to E' through R_l is selected uniformly at random, which equals $\frac{\mathsf{R}_l(E', E)}{C_{\mathsf{R}_l}}$. $pLand_l$ stores the probability of landing at any object E in the range of $(\mathcal{PR})$ following $\mathcal{PR}$ and is returned as the output of the function.

Example 2: A PRA model may use the following WPRs to define the conditional probability of $\mathsf{WillCite}(q, p)$ in our running example:

$$WPR_0 : \langle w_0, p\rangle$$

$$WPR_1 : \left\langle w_1, q \xrightarrow{\mathsf{PubIn}} y \xrightarrow{\mathsf{ImBef}} y' \xrightarrow{\mathsf{PubIn}^{-1}} p \right\rangle$$

$$WPR_2 : \left\langle w_2, q \xrightarrow{\mathsf{PubIn}} y \xrightarrow{\mathsf{PubIn}^{-1}} p' \xrightarrow{\mathsf{Cited}} p \right\rangle$$

$$WPR_3 : \left\langle w_3, p_1 \xrightarrow{\mathsf{Cited}} p_2 \xrightarrow{\mathsf{Cited}} p \right\rangle$$

WPR_0 is a bias, WPR_1 considers the papers published a year before the query paper, WPR_2 considers papers cited by other papers published in the same year as the query paper, and WPR_3

mimics PageRank algorithm for finding important papers in terms of citations (cf. (Lao and Cohen, 2010b) for more detail). Consider the citations among existing papers in **Figure 1A,** and let the publication year for all the six papers be 2017. Suppose we have a query paper Q, which is to be published in 2017 and we want to find the probability of WillCite(Q, $Paper_2$) according to the PRA model above. Applying the substitution $\{\langle q, p\rangle / \langle Q, Paper_2\rangle\}$ to the above WPRs gives the following WPRs, respectively:

$$WPR_0 : \langle w_0, Paper_2 \rangle$$

$$WPR_1 : \left\langle w_1, Q \xrightarrow{\mathsf{PubIn}} y \xrightarrow{\mathsf{ImBef}} y' \xrightarrow{\mathsf{PubIn}^{-1}} Paper_2 \right\rangle$$

$$WPR_2 : \left\langle w_2, Q \xrightarrow{\mathsf{PubIn}} y \xrightarrow{\mathsf{PubIn}^{-1}} p' \xrightarrow{\mathsf{Cited}} Paper_2 \right\rangle$$

$$WPR_3 : \left\langle w_3, p_1 \xrightarrow{\mathsf{Cited}} p_2 \xrightarrow{\mathsf{Cited}} Paper_2 \right\rangle$$

WPR_0 evaluates to w_0. WPR_1 evaluates to 0. WPR_2 evaluates to $w_2 * (\frac{1}{6} * \frac{1}{4} + \frac{1}{6} * \frac{1}{2}) = w_2 * 0.125$ as for the path $y \xrightarrow{\mathsf{PubIn}^{-1}} p'$, there is $\frac{1}{6}$ probability for randomly walking to either $Paper_5$ or $Paper_6$ and then there is $\frac{1}{4}$ probability to walk randomly from $Paper_5$ to $Paper_2$ and $\frac{1}{2}$ probability to walk randomly from $Paper_6$ to $Paper_2$ according to Cited relation. WPR_3 evaluates to $w_3 * \frac{1}{6} * (\frac{1}{2} * \frac{1}{4} + \frac{1}{3} * (\frac{1}{4} + \frac{1}{2}) + \frac{1}{4} * \frac{1}{2}) \approx w_3 * 0.083$. The $\frac{1}{6}$ outside parenthesis is the probability of randomly starting at any paper, $\frac{1}{2} * \frac{1}{4}$ is the probability of transitioning from $Paper_3$ to $Paper_5$ and then to $Paper_2$, and so forth. Therefore, the conditional probability of WillCite (Q, $Paper_2$) is as follows:

$$\sigma(w_0 + w_2 * 0.125 + w_3 * 0.083)$$

3. RLR WITH NORMALIZED RELATIONS GENERALIZES PRA

To prove that RLR with normalized relations generalizes PRA, we first define relation chains and describe some of their properties.

3.1. Relations Chain

Definition 1: We define a **relations chain** as a list of binary atoms $\mathsf{V}_1(x_0,x_1),\ldots, \mathsf{V}_m(x_{m-1}, x_m)$ such that for each V_i and V_{i+1}, the second logvar of V_i is the same as the first logvar of V_{i+1}, $x_0,\ldots,x_m$ are different logvars, and V_i and V_j can be the same or different predicates.

Example 3: $\mathsf{V}_1(x, y)$, $\mathsf{V}_2(y, z)$ is a relations chain, and $\mathsf{V}_1(x, y)$, $\mathsf{V}_2(z, y)$ and $\mathsf{V}_1(x, y)$, $\mathsf{V}_2(y, z)$, $\mathsf{V}_3(z, x)$ are not relations chains.

Definition 2: A first-order formula corresponds to a relations chain if all its literals are binary predicates and non-negated, and there exists an ordering of the literals, i.e, a relations chain.

Example 4: The first-order formula $\mathsf{V}_1(x_1, x_2) * \mathsf{V}_2(x_3, x_1)$ corresponds to a relations chain as the order $\mathsf{V}_2(x_3, x_1)$, $\mathsf{V}_1(x_1, x_2)$ is a relations chain.

It follows from RLR definition that re-ordering the literals in each of its WFs does not change the distribution. For any WF whose formula corresponds to a relations chain, we assume hereafter that its literals have been re-ordered to match the order of the corresponding relations chain.

Definition 3: Let $\mathsf{V}(x, y)$ be a target atom. *Relations chain RLR (RC-RLR)* is a subset of RLR for defining a conditional probability distribution for $\mathsf{V}(x, y)$, where:

- formulae of WFs correspond to relations chains,
- for each WF, the second logvar of the last atom is y,
- x may only appear as the first logvar of the first atom,
- y may only appear as the second logvar of the last atom.

ALGORITHM 2 | *Eval*(φ).

```
Input: Formula $\varphi = \mathsf{R}_1(x_0, x_1) * \mathsf{R}_2(x_1, x_2) * \ldots * \mathsf{R}_l\,(x_{l-1}, x_l)$.
Output: Evaluation of $\varphi$.
 1: if $l = 0$ then
 2:     return $ones(|\Delta_{x_0}|)$
 3: $\varphi\,' = \mathsf{R}_1(x_0, x_1) * \mathsf{R}_2(x_1, x_2) * \ldots * \mathsf{R}_{l-1}(x_{l-2}, x_{l-1})$
 4: $eval_{l-1} = Eval_{\mathrm{l}}\,(\varphi')$
 5: for $E \in \Delta_{x_l}$ do
 6:     $eval_l(E) = 0$
 7: for $E' \in \Delta_{x_{l-1}}$ do
 8:     for $E \in \Delta_{x_l}$ do
 9:         $canWalk(E', E) = R_l(E', E)$
10:         $eval_l(E) \mathrel{+}= eval_{l-1}(E') * canwalk(E', E)$
11: return $eval_l$
```

For RLR models, to evaluate a formula, one may have nested loops over logvars of the formula that do not appear in the target atom or conjoin all literals one by one and then count. WFs of RC-RLR, however, can be evaluated in a special way. To evaluate a formula in RC-RLR, starting from the end (or beginning), the effect of each literal can be calculated and then the literal can be removed from the formula. **Algorithm 2** indicates how a formula corresponding to a relations chain can be evaluated. This evaluation grows with the product of the number of literals in the formula and the number of observed data, which makes it highly scalable.

When $l = 0$, the formula corresponds to True and evaluates to 1 for any $X_0 \in x_0$. Therefore, in this case, the algorithm returns a vector of ones of size $|\Delta_{x_0}|$. Otherwise, the algorithm first evaluates $\varphi' = \mathsf{R}_1(x_0, x_1) * \mathsf{R}_2(x_1, x_2) * \ldots * \mathsf{R}_{l-1}(x_{l-2}, x_{l-1})$ using a recursive call to the *Eval* function. The resulting vector is stored in $eval_{l-1,}$ such that for a $E' \in \Delta_{x_{l-1}}$, $eval_{l-1}[E']$ indicates the result of evaluating $\varphi' = \mathsf{R}_1(x_0, x_1) * \mathsf{R}_2(x_1, x_2) * \ldots * \mathsf{R}_{l-1}(x_{l-2}, E')$. Then to evaluate φ for some $E \in \Delta_{x_l}$, we sum $eval_{l-1}[E']$ s for any $E' \in \Delta_{x_{l-1}}$ such that $R_l(E', E)$ holds. *canWalk* in the algorithm is 1 if $R_l(E', E)$ holds and 0 otherwise, and $eval_l(E) \mathrel{+}= eval_{l-1}(E') * canwalk(E', E)$ $\mathrel{+}=$ adds $eval_{l-1}[E']$ to $eval_1[E]$ if *canWalk* is 1.

Proposition 1: ***Algorithm 2** is correct.*

Proof: Let $\varphi = \mathsf{R}_1(x_0, x_1) * \mathsf{R}_2(x_1, x_2) * \ldots * \mathsf{R}_l\,(x_{l-1}, x_l) * \mathsf{eval}_l(x_l)$ ($\mathsf{eval}_l(x_l)$ can be initialized to a vector of ones at the beginning of the algorithm. Since by definition of relations chain x_l only appears in R_l and $\mathsf{eval}_l\,(x_l)$, for any $X_{l-1} \in \Delta x_{l-1}$ we can evaluate $eval_{l-1}(X_{l-1}) = \sum_{X_l \in \Delta_{x_l}} \mathsf{R}_l(X_{l-1}, X_l) {}^{*} \mathsf{eval}_l(X_l)$ separately and replace $\mathrm{R}_1(x_{l-1}, x_1) * \mathsf{eval}_l(x_l)$ with $\mathsf{eval}_{l-1}\,(x_{l-1})$, thus getting $\varphi' = \mathsf{R}_1(x_0, x_1) * \mathsf{R}_2(x_1, x_2) * \ldots * \mathsf{R}_{l-1}(x_{l-2}, x_{l-1}) * \mathsf{eval}_{l-1}\,(x_{l-1})$. The same procedure can compute φ'.

3.2. From PRA to Relation Chains

Proposition 2: *A path relation corresponds to a relations chain.*

Proof: Let $\mathcal{PR} = x_0 \xrightarrow{\mathsf{R}_1} x_1 \xrightarrow{\mathsf{R}_2} \ldots \xrightarrow{\mathsf{R}_l} x_l$ be a path relation. We create a relation atom $\mathsf{R}_i(x_{i-1}, x_i)$ for any subpath $x_{i-1} \xrightarrow{\mathsf{R}_i} x_i$ resulting in relations $\mathsf{R}_1(x_0, x_1)$, $\mathsf{R}_2(x_1, x_2), \ldots, \mathsf{R}_l(x_{l-1}, x_l)$. By definition of path relations, the second logvar of any relation R_i is the same as the first logvar of the next relation. Since by definition the logvars in a path relation are different, the second logvar of any relation R_i is only equivalent to the first logvar of the next relation.

Example 5: Consider the path relation $q \xrightarrow{\mathsf{PubIn}} y \xrightarrow{\mathsf{PubIn}^{-1}} p' \xrightarrow{\mathsf{Cited}} p$ from Example 2. This path relation corresponds to a relations chain with atoms $\mathsf{PubIn}(q, y)$, $\mathsf{PubIn}^{-1}(y, p')$, and $\mathsf{Cited}(p', p)$.

3.3. Row-Wise Count Normalization

Having a binary predicate $\mathsf{V}(x, y)$ and a set of pairs of objects for which V holds, one may consider the importance of these pairs to be different. For instance, if a paper has cited only 20 papers, the importance of these citations may be more than the importance of citations for a paper citing 100 papers. One way to take the importance of the pairs into account is to normalize the relations. A simple way to normalize a relation is to normalize it by row-wise counts. For some $X \in \Delta_x$, let α represent the number of $Y' \in \Delta_y$, such that $\mathsf{V}(X, Y')$ holds. When $\alpha \neq 0$, instead of considering $\mathsf{V}(X, Y) = 1$ for a pair $\langle X, \mathrm{Y} \rangle$, we normalize it to $\mathsf{V}(X, Y) = \frac{1}{\alpha}$. After this normalization, the citations of a paper with 20 citations are 5 times more important than the citations of a paper with 100 citations overall. Note that when $\alpha = 0$, we do not change any values. We refer to this normalization method as *row-wise count (RWC) normalization*. **Figure 1B** shows the result of applying RWC normalization to the relation in **Figure 1A**. Note that there may be several other ways to normalize a relation; here, we introduced RWC because, as we will see in the upcoming sections, it is the normalization method used in PRA.

3.4. Main Theorem

Theorem 1: *Any PRA model is equivalent to an RC-RLR model with RWC normalization.*

Proof: Let $\Psi = \{\langle w_0, \mathcal{PR}_0 \rangle, \ldots, \langle w_k, \mathcal{PR}_k \rangle\}$ represent a set of WPRs used by a PRA model. We proved in Proposition 2 that any path relation $\mathcal{PR}_i$ in Ψ corresponds to a relations chain. By multiplying the relations in the relation chain, one gets a formula φ_i for each $\mathcal{PR}_i$, and this formula is by construction guaranteed to correspond to a relations chain. We construct an RC-RLR model whose WFs are $\psi = \{\langle v_0, \varphi_0 \rangle, \ldots, \langle v_k, \varphi_k \rangle\}$. Given that the relations (and their order) used in $\mathcal{PR}_i$ and φ_i are the same for any i, the only differences between the evaluation of $\mathcal{PR}_i$ and φ_i according to **Algorithm 1** and **Algorithm 2** are: (1) **Algorithm 1** divides $R_l(E', E)$ by $C_{R_l}(E')$,while **Algorithm 2** does not, and (2) in the termination condition, **Algorithm 1** returns a uniform distribution over objects in Δ_{x_0}, while **Algorithm 2** returns a vector of ones of size $|\Delta_{x_0}|$. Dividing $R_l(E', E)$ by $C_{R_l}(E')$ is equivalent to RWC normalization, and the difference in the constant value of the function in the termination condition gets absorbed in the weights that are multiplied to each path relation or formula. Therefore, the RC-RLR model with WFs ψ is identical to the PRA model with WPRs Ψ after normalizing the relations using RWC.

Example 6: Consider the PRA model in Example 2. For the four WPRs in that model, we create the following corresponding WFs for an RC-RLR model by multiplying the relations in the path relations:

$$\langle v_0, \mathsf{True} \rangle$$
$$\Big\langle v_1, \mathsf{PubIn}(q, y_1) * \mathsf{ImBef}(y_1, y_2) * \mathsf{PubIn}^{-1}(y_2, p) \Big\rangle$$
$$\Big\langle v_2, \mathsf{PubIn}(q, y_1) * \mathsf{PubIn}^{-1}(y_1, p') * \mathsf{Cited}(y_1, p) \Big\rangle$$
$$\langle v_3, \mathsf{Cited}(p_1, p_2) * \mathsf{Cited}(p_2, p) \rangle$$

Consider computing $\mathsf{WillCite}$ (Q, $Paper_2$) according to an RC-RLR model with the above WFs, where all existing papers and Q have been published in 2017 and the relations have been normalized using RWC normalization (e.g., as in **Figure 1B** for relation Cited). Then the first formula evaluates to v_0. The second WF evaluates to 0. The third WF evaluates to $v_2 * \frac{1}{6} * (\frac{1}{4} + \frac{1}{2})$ as the values in relation PubIn^{-1} have been normalized to $\frac{1}{6}$ for year 2017 and the values in relation Cited have been normalized to $\frac{1}{4}$ and $\frac{1}{2}$ for $Paper_5$ and $Paper_6$ as in **Figure 1B**. The last WF evaluates to $v_3 * (\frac{1}{2} * \frac{1}{4} + \frac{1}{3} * (\frac{1}{4} + \frac{1}{2}) + \frac{1}{4} * \frac{1}{2})$. The $\frac{1}{2} * \frac{1}{4}$ comes from $\mathsf{Cited}(Paper_3, Paper_5) * \mathsf{Cited}(Paper_5, Paper_2)$, $\frac{1}{3} * (\frac{1}{4} + \frac{1}{2})$ comes from $\mathsf{Cited}(Paper_4, Paper_5) * \mathsf{Cited}(Paper_5, Paper_2)$ and $\mathsf{Cited}(Paper_4, Paper_6) * \mathsf{Cited}(Paper_6, Paper_2)$, and $\frac{1}{4} * \frac{1}{2}$ comes from $\mathsf{Cited}(Paper_5, Paper_6) * \mathsf{Cited}(Paper_6, Paper_2)$. As it can be viewed from Example 2, after creating the equivalent RC-RLR model and normalizing the relations using RWC normalization, all WPRs evaluate to the same value as their corresponding WF, except the last WF. The $\frac{1}{6}$ before the parenthesis in Example 2 is missing when evaluating the last WF. This $\frac{1}{6}$, however, is a constant independent of the query (it is the constant value of the uniform distribution in the if statement corresponding to the termination criteria in **Algorithm 1**). Assuming $v_3 = w_3 * \frac{1}{6}$ and all other v_is are the same as w_is, the conditional probability of $\mathsf{Cited}(Q, Paper_2)$ according to the RC-RLR model above will be the same as the PRA model in Example 2.

3.5. From Random Walk Strategies to Structure Learning

The restrictions imposed on the formulae by path relations in PRA reduce the number of possible formulae to be considered in a model compared to RLR models. However, there may still be many possible path relations, and considering all possible path relations for a PRA model may not be practical.

Lao and Cohen (2010b) allow the random walk to follow any path, but restrict the maximum number of steps. In particular, they only allow for path relations whose length is less than some l. The value of l can be selected based on the number of objects, relations, available hardware, and the amount of time one can afford for learning/inference. This strategy automatically gives a (very simple) structure learning algorithm for RC-RLR by considering only formulae whose number of relations are less than l.

Lao et al. (2011) follow a more sophisticated approach for limiting the number of path relations. Besides limiting the maximum length of the path relations to l, Lao et al. (2011) impose two more restrictions: for any path relation to be included, (1) the probability of reaching the target objects must be non-zero for at least a fraction α of the training query objects, and (2) it should at least retrieve one target object in the training set. During parameter learning, they impose a Laplacian prior on their weights to further reduce the number of path relations. In an experiment on knowledge completion for NELL (Carlson et al., 2010), they show that these two restrictions plus the Laplacian prior reduce the number of possible path relations by almost 99.6 and 99.99% when $l = 3$ and $l = 4$, respectively. Therefore, their random walk strategy is capable of taking more steps (i.e., selecting a larger value for l) and capture features that require longer chains of relations. This random walk strategy is called *data-driven path finding*.

Both restrictions in data-driven path finding can be easily verified for RC-RLR formulae and the set of possible formulae can be restricted accordingly. Furthermore, during parameter learning, a Laplacian prior can be imposed on the weights of the weighted formulae. RC-RLR models learned in this way correspond to PRA models learned using data-driven path finding. Therefore, data-driven path finding can be also considered as a structure learning algorithm for RC-RLR. With the same reasoning, several other random walk strategies can be considered as structure learning algorithms for RC-RLR, and *vice versa*. This allows for faster development of the two paradigms by leveraging ideas developed in each community in the other.

4. PRA VS. RLR

An advantage of PRA models over RLR models is their efficiency: there is a smaller search space for WFs, and all WFs can be evaluated efficiently. Such efficiency makes PRA scale to larger domains where models based on the weighted rule learning such as RLR often have scalability issues. It also allows PRA models to scale to and capture features that require longer chains of relations. However, the efficiency comes at the cost of losing modeling power. In the following subsections, we discuss such costs.

4.1. Shortcomings of Relations Chains

Since PRA models restrict themselves to relations chains of a certain type, they lose the chance to leverage many other WFs. As an example, to predict $\mathsf{Cites}(p_1,p_2)$ for the reference recommendation task, suppose we would like to recommend papers published a year before the target paper that have been cited by the papers published in the same year as the target paper. Such a feature requires the following formula: $\mathsf{PubIn}\ (p_1, y) * \mathsf{Before}(y, y') * \mathsf{PubIn}\ (p_2, y') * \mathsf{Cites}(p', p_2) * \mathsf{PubIn}(p', y)$. It is straightforward to verify that this formula cannot be included in RC-RLR (and consequently in PRA) as p_2 (the second logvar of the target atom) is appearing twice in the formula, thus violating the last condition in Definition 3. While restricting the formulae to the ones that correspond to relations chain may speed up learning and reasoning, it reduces the space of features that can be included in a relational learning model, thus potentially decreasing accuracy.

4.2. Non-Binary Atoms

One issue with PRA models is the difficulty in including unary atoms in such models. As an example, suppose in Example 2, we would like to treat conference papers and journal papers differently. For an RLR model, this can be easily done by including Conference(p) or Journal(p) as an extra atom in the formulae. For PRA, however, this cannot be done. The way unary atoms are currently handled in PRA models is through isA and isA^{-1} relations (Lao et al., 2011). For instance, a path relation may contain $paper \xrightarrow{\text{isA}} type$, but the only next predicate that can be applied to this path is isA^{-1} giving the other papers with the same type as the paper in the left of the arrow. However, this is limiting and does not allow for, e.g., treating conference and journal papers differently.

Atoms with more than two logvars are another issue for PRA models because they restrict their models to binary atoms. While any relation with more than two arguments can be converted into several binary atoms, the random walk strategies used for PRA models (and the probabilities for making these random steps) make it unclear how atoms with more than two logvars can be leveraged in PRA models.

4.3. Continuous Atoms

For any subpath $x \xrightarrow{\mathsf{R}} y$ in a path relation of a PRA model, R typically has a range {0, 1}: for any object $X \in \Delta_x$, this subpath gives the objects in Δ_y participating in relation R with X. PRA models can be extended to handle some forms of continuous atoms. For instance for the reference recommendation problem, suppose we have an atom $\mathsf{Sim}\ (p, p')$ indicating a measure of similarity between the titles of two papers. The higher the $\mathsf{Sim}\ (p, p')$, the more similar the titles of the two papers. A sensible WF for an RLR model predicting $\mathsf{Cites}\ (p_1, p_2)$ may be $\mathsf{Sim}\ (p_1, p') * \mathsf{Cites}(\text{p}', \text{p}_2)$. To extend PRA models to be able to leverage such continuous atoms, one has to change line 8 in **Algorithm 1** to sum the values of $\mathsf{R}_l(E', E)$ instead of counting how many times the relation holds.

For many types of continuous atoms, however, it is not straightforward to extend PRA models to leverage them. As an example, suppose we have an atom Temperature(r, d) showing the temperature of a region in a specific date. It is not clear how a random walk step can be made based on this atom as the temperature can, e.g., be positive or negative.

4.4. Relational Normalization

Normalizing the relations is often ignored in models based on weighted rule learning. For the most part, this ignorance may be because several of these models cannot handle continuous atoms. Given that PRA is a special form of weighted rule learning models such as RLR with RWC normalization, not normalizing the relations may be the reason why in Lao et al.'s (Lao et al., 2011) experiments, PRA outperforms the weighted rule learning method FOIL (Quinlan, 1990) for link prediction in NELL (Carlson et al., 2010).

The type of normalization used in PRA (RWC) may not be the best option in many applications. As an example, suppose for the reference recommendation task we want to find papers similar to the query paper in terms of the words they use. Let $\mathsf{Contains}^{-1}$ (w, p) show the relation for words in each paper. It is well known in information retrieval that words do not have equal importance and a normalization of $\mathsf{Contains}^{-1}\ (w, p)$ is necessary to take such importance into account. PRA models consider the importance of each word W as $Score_1(W) = \frac{1}{f(W)}$, where *f(W)* is the number of papers containing the word W (see, e.g., Lao and Cohen (2010b)). However, it has been well known in information retrieval community for several decades, and information theoretically justified more than a decade ago (Robertson, 2004), which $Score_2(W) = log(\frac{\#papers}{f(W)})$ provides a better importance score. Most TF-IDF based information retrieval algorithms (Salton and Buckley, 1988) currently rely on $Score_2$. It is straightforward to include the latter score in an RLR model: one only has to multiply the formulae using word information by $Score_2(W)$, without normalizing the $\mathsf{Contains}^{-1}$ (w, p) relation [see, e.g., Fatemi (2017)]. However, it is not straightforward how such a score can be incorporated into PRA models as they do not include unary or continuous atoms.

4.5. Evaluating Formulae

Evaluating the formulae in models based on weighted rule learning is known to be expensive, especially for relations with lower sparsities and for longer formulae. In practice, approximations are typically used for scaling the evaluations. Since formulae in RC-RLR correspond to path relations, these formulae can be approximated efficiently using sampling techniques developed within graph random walk community such as fingerprinting (Fogaras et al., 2005; Lao and Cohen, 2010a), weighted particle filtering (Lao and Cohen, 2010a), and low-variance sampling (Lao et al., 2011), without noticeably affecting the accuracy. Extending sampling ideas to other formulae is an interesting future direction.

5. CONCLUSION

With abundance of relational and graph data, statistical relational learning has gained great amounts of attention. Three main relational learning paradigms have been developed during the past decade and more: weighted rule learning, graph random walk, and tensor factorization. These paradigms have been mostly developed and studied in isolation with few works aiming at understanding the relationship among them or combining them. In this article, we studied the relationship between two relational learning paradigms: weighted rule learning and graph random walk. In particular, we studied the relationship between relational logistic regression (RLR), one of the recent developments in weighted rule learning paradigm, and path ranking algorithm (PRA), one of the most well-known algorithms in graph random walk paradigm. Our main contribution was to prove that PRA models correspond to a subset of RLR models after row-wise count normalization. We discussed the advantages that this proof provides for both paradigms and for statistical relational AI community in general. Our result sheds light on several issues with both paradigms and possible ways to improve them.

AUTHOR CONTRIBUTIONS

SK did this work under supervision of DP.

FUNDING

This work is funded by an NSERC discovery grant to DP.

REFERENCES

Bhattacharya, I., and Getoor, L. (2007). Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data* 1, 5. doi:10.1145/1217299.1217304

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). “Translating embeddings for modeling multi-relational data,” in *NIPS*, Lake Tahoe, 2787–2795.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B. Jr., Hruschka, E. R., and Mitchell, T. M. (2010). “Toward an architecture for never-ending language learning,” in *AAAI*, Atlanta, Vol. 5, 3.

Das, R., Dhuliawala, S., Zaheer, M., Vilnis, L., Durugkar, I., Krishnamurthy, A., et al. (2017). “Go for a Walk and Arrive at the Answer: Reasoning Over Paths in Knowledge Bases with Reinforcement Learning,” in *NIPS-17 Workshop on Automatic Knowledge-Base Construction*, Long Beach.

De Raedt, L., Kersting, K., Natarajan, S., and Poole, D. (2016). Statistical relational artificial intelligence: logic, probability, and computation. *Synth. Lect. Artif. Intell. Mach. Learn.* 10, 1–189. doi:10.2200/S00692ED1V01Y201601AIM032

De Raedt, L., Kimmig, A., and Toivonen, H. (2007). “Problog: a probabilistic prolog and its application in link discovery,” in *IJCAI*, Hyderabad, 7.

Domingos, P., Kok, S., Lowd, D., Poon, H., Richardson, M., and Singla, P. (2008). “Markov logic,” in *Probabilistic Inductive Logic Programming*, eds L. D. Raedt, P. Frasconi, K. Kersting, and S. Muggleton (New York: Springer), 92–117.

Dries, A., Kimmig, A., Davis, J., Belle, V., and De Raedt, L. (2017). “Solving probability problems in natural language,” in *IJCAI*, Melbourne.

Ebrahimi, J., Dou, D., and Lowd, D. (2016). “Weakly supervised tweet stance classification by relational bootstrapping,” in *EMNLP*, Austin, 1012–1017.

Fatemi, B. (2017). *Finding a Record in a Database*. Vancouver: Master’s thesis, University of British Columbia.

Fatemi, B., Kazemi, S. M., and Poole, D. (2016). “A Learning Algorithm for Relational Logistic Regression: Preliminary Results,” in *IJCAI-16 Workshop on Statistical Relational AI*, New York.

Fogaras, D., Rácz, B., Csalogány, K., and Sarlós, T. (2005). Towards scaling fully personalized pagerank: algorithms, lower bounds, and experiments. *Internet Math.* 2, 333–358. doi:10.1080/15427951.2005.10129104

Getoor, L., and Taskar, B. (2007). *Introduction to Statistical Relational Learning*. Cambridge: MIT Press.

Hommersom, A., and Lucas, P. J. (2011). “Generalising the interaction rules in probabilistic logic,” in *IJCAI*, Barcelona.

Jain, A., and Pantel, P. (2010). “Factrank: random walks on a web of facts,” in *Proceedings of the 23rd International Conference on Computational Linguistics*, Beijing, 501–509.

Kazemi, S. M., Buchman, D., Kersting, K., Natarajan, S., and Poole, D. (2014). “Relational logistic regression,” in *KR*, Vienna.

Kazemi, S. M., Fatemi, B., Kim, A., Peng, Z., Tora, M. R., Zeng, X., et al. (2017). “Comparing Aggregators for Relational Probabilistic Models,” in *UAI Workshop on Statistical Relational AI*, Sydney.

Kazemi, S. M., and Poole, D. (2018). “Relnn: a deep neural model for relational learning,” in *AAAI*, New Orleans.

Khot, T., Balasubramanian, N., Gribkoff, E., Sabharwal, A., Clark, P., and Etzioni, O. (2015). “Markov Logic Networks for Natural Language Question Answering,” in *Empirical Methods in Natural Language Processing*, Lisbon.

Kimmig, A., Bach, S., Broecheler, M., Huang, B., and Getoor, L. (2012). “A short introduction to probabilistic soft logic,” in *Proceedings of the NIPS Workshop on Probabilistic Programming: Foundations and Applications*, Lake Tahoe, 1–4.

Kimmig, A., Mihalkova, L., and Getoor, L. (2015). Lifted graphical models: a survey. *Mach. Learn.* 99, 1–45. doi:10.1007/s10994-014-5443-2

Lao, N., and Cohen, W. W. (2010a). “Fast query execution for retrieval models based on path-constrained random walks,” in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Washington, DC: ACM), 881–888.

Lao, N., and Cohen, W. W. (2010b). Relational retrieval using a combination of path-constrained random walks. *Mach. Learn.* 81, 53–67. doi:10.1007/s10994-010-5205-8

Lao, N., Mitchell, T., and Cohen, W. W. (2011). “Random walk inference and learning in a large scale knowledge base,” in *EMNLP*, Edinburgh, UK, 529–539.

Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., and Liu, S. (2015). “Modeling Relation Paths for Representation Learning of Knowledge Bases,” *Empirical methods in natural language processing*, Lisbon.

Nickel, M., Jiang, X., and Tresp, V. (2014). “Reducing the rank in relational factorization models by including observable patterns,” in *NIPS*, Montreal, 1179–1187.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proc. IEEE* 104, 11–33. doi:10.1109/JPROC.2015.2483592

Nickel, M., Tresp, V., and Kriegel, H.-P. (2012). “Factorizing yago: scalable machine learning for linked data,” in *Proceedings of the 21st International Conference on World Wide Web* (Inria: ACM), 271–280.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). *The Pagerank Citation Ranking: Bringing Order to the Web. Technical Report*. Stanford: Stanford InfoLab.

Pujara, J., Miao, H., Getoor, L., and Cohen, W. W. (2013). “Knowledge graph identification,” in *International Semantic Web Conference (ISWC)* (Sydney: Springer).

Pujara, J. and Getoor, L. (2016). “Generic Statistical Relational Entity Resolution in Knowledge Graphs,” in *IJCAI-16 Workshop on Statistical Relational AI*, New York.

Quinlan, J. R. (1990). Learning logical definitions from relations. *Mach. Learn.* 5, 239–266. doi:10.1023/A:1022699322624

Riedel, S., Yao, L., McCallum, A., and Marlin, B. M. (2013). “Relation extraction with matrix factorization and universal schemas,” in *HLT-NAACL*, Atlanta, 74–84.

Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *J. Doc.* 60, 503–520. doi:10.1108/00220410410560582

Salton, G., and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Info. Process. Manag.* 24, 513–523. doi:10.1016/0306-4573(88)90021-0

Singla, P., and Domingos, P. (2006). “Entity resolution with Markov logic,” in *Data Mining, 2006. ICDM’06. Sixth International Conference on* (Hong Kong: IEEE), 572–582.

Šourek, G., Aschenbrenner, V., Železny, F., and Kuželka, O. (2015). “Lifted relational neural networks,” in *Proceedings of the 2015th International Conference on Cognitive Computation: Integrating Neural and Symbolic Approaches-Volume 1583* (Montreal: CEUR-WS.org), 52–60.

Sridhar, D., Foulds, J. R., Huang, B., Getoor, L., and Walker, M. A. (2015). “Joint models of disagreement and stance in online debate,” in *ACL*, Beijing, Vol. 1, 116–125.

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., and Bouchard, G. (2016). “Complex embeddings for simple link prediction,” in *ICML*, New York, 2071–2080.

Yu, X., Ren, X., Sun, Y., Gu, Q., Sturt, B., Khandelwal, U., et al. (2014). “Personalized entity recommendation: a heterogeneous information network approach,” in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining* (New York: ACM), 283–292.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.