Distilling Causal Metaknowledge from Massive Knowledge Graphs

Anonymous ACL submission

Abstract

In recent years, the growing information overload facilitates the access to billions of relational facts in the world, which are usually integrated in all manner of knowledge graphs. The metaknowledge, defined as the knowledge about knowledge, reveals the inner principle of arising these factual knowledge, and hence is of vital importance to be discovered for the understanding, exploiting and completion of knowledge. In this paper, we focus on capturing the causal component of metaknowledge, that is a metarule with causal semantic. For the propose, we devise an efficient causal rule discovery algorithm called CaRules that distills the causal rules between two knowledge graph schemata abstracted from instances from massive knowledge graphs. Extensive experiments demonstrate that the quality and interpretability of the causation-based rules outperform the correlation-based rules, especially in the out-of-distribution tasks.

1 Introduction

In the time of information explosion, knowledge graph (KG) is a powerful representation to integrate the billions of available relational facts implying the rich relationships of entities, which are the observational low-level knowledge in the world. Even though the massive knowledge can benefit various downstream applications, such as recommender system (Wang et al., 2019a,b), to better understand, exploit, and complete the knowledge, it is necessary to explore the inner principle of arising these factual knowledge. For this purpose, the concept of metaknowledge is proposed (Evans and Foster, 2011), which is defined as the *knowledge about knowledge*.

There are sevaral forms of metaknowledge (Burgin, 2016). A common one of them, the metarule, usually carries the causal semantic. For example, a metarule $A \rightarrow B$ means A *implies* B ("rain" *implies* "the ground gets wet"). In the meanwhile, (Fortunato et al., 2018) points that causality is necessary to identify the fundamental drivers of knowledge. Therefore, in this paper, we aim to distill causal rules from KG, which are the causal component of metaknowledge.

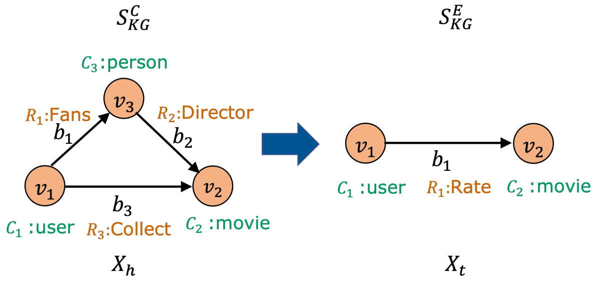

Figure 1: An illustrative example of causal rule, where the cause is a KGS defined on three concepts and three relations, and the effect is defined on two concepts and one relation. This example gives partial explanation of the cause of users' rate on the movie.

Identifying causality is a fundamental problem in many scientific fields. Based on the experimental or observational data, it aims to discover the causal relations between variables via statistical analysis methods (Imbens and Rubin, 2010; Pearl, 2010). In this work, we propose the concept of *KG schema* (KGS) which can be regarded as the variable in causal inference, and each causal rule represents the causal relationship between two KGSs. As shown in Fig 1, a KGS is a directed graph, which is composed of several relational triples defined at the concept level. In many tasks such as link prediction and KG completion, we are mostly interested in the causes leading to a target relational triple. Therefore, we focus on the triple-structured effect variables, while seeking for the cause variables represented by KGS.

One primary step in identifying causality is to form reasonable abstract on variables and samples. As the variables are defined with graph-structured KGS, it naturally induces a combinatorial complexity problem especially considering a massive KG with many concepts. Meanwhile, although it is straightforward to use a KGS instantiated with entities as a sample, how to define treated (or positive) and controlled (or negative) samples for the graph-structured variables with various types of relationships (e.g. binary, multi-level or continuous relationships) is still an open problem.

In this paper, we formulate the causal meta-knowledge learning problem into a local causal discovery problem with graph-structured variables. We first formally define the KGS, and for a targeted effect KGS, we subtly design a strategy to reduce down the space of candidate cause KGS without the loss of generality. We then propose a functional heuristic to form the treated and controlled sample space effectively and efficiently. Furthermore, we propose a causal rule discovery algorithm called CaRules, which is specifically designed for the target problem. It consists of an efficient path finding module and an effective PC-like process, which jointly conduct conditional independence tests to find causal structures. Extensive experiments demonstrate that the quality and interpretability of the causation-based rules outperform the correlation-based rules, especially in the out-of-distribution (OOD) tasks.

2 Related Work

Rule mining. In KG, compared with deductive knowledge, which is characterised by precise logical consequences, inductively acquiring knowledge involves generalising patterns from a given set of input observed facts, which can then be used to generate novel but potentially imprecise predictions. Therefore, inductive knowledge is very important for the fundamental tasks in KG, such as KG completion, KG reasoning. Even though, some techniques, such as KG representation learning, graph neural networks (Bordes et al., 2013; Yang et al., 2015; Wu et al., 2020), have achieve promising progress in the inductive knowledge learning in the past several years, such models often lack interpretability and suffer from the out-of-vocabulary problem, where they are unable to provide results for edges involving previously unseen nodes or edges. An alternative approach for inductive knowledge learning is *rule mining*, which refers to discovering meaningful patterns in the form of rules from large collections of background knowledge.

There are two main categories in rule mining studies. (Galárraga et al., 2013, 2015; Omran et al., 2019) use predefined metrics confidence and support, to find rules satisfying the given thresholds of the metrics, based in a top-down fashion. Due to the frequent open world assumption (OWA) (Hogan et al., 2021) in KG, these methods mainly learn the monotonic rules. Based on the previous techniques, (Gad-Elrab et al., 2016; Ho et al., 2018; Tanon et al., 2017) investigate the methods to learn non-monotonic rules, which with negated edges in the body. Different to the co-occurrence metrics-based methods, the other line of research is called differentiable rule mining, which allows end-to-end learning. The core idea is that the joined relations in rule bodies can be represented as matrix multiplication. Neural-LP (Yang et al., 2017) adopts an attention mechanism to select a variable-length sequence as the body of rules for which confidences are learnt. DRUM (Sadeghian et al., 2019) uses bidirectional recurrent neural networks to learn sequences of relations, which are the body of rules, and their confidences.

Our work can be seen as one of solutions for rule mining. Different to past the association-based rule mining methods, our work aims to discover deeper relationship, causation, since correlation does not imply causation, via a more rigorous statistical inference system. This is the fist attempt to study rule mining problem under causal perspective, as far as we know.

Causal Discovery. There are two main frameworks for causal discovery, called Rubin causal models (RCMs) (Imbens and Rubin, 2010) and structural causal models (SCMs) (Pearl, 2010). The former mainly investigates the causal effect between the treatment and the effect based on the potential outcome model. While the latter mainly focuses on discovering the causal structure between variable via the bayesian network. Our problem is more similar to the traditional causal discovery problem in SCM.

There are two types of causal discovery algorithms, *constraint-based* and *score-based* (Spirtes et al., 2000). The constraint-based algorithms construct the causal structure based on conditional independence constraints, while the score-based algorithms generate a number of candidate causal graphs, assign a score to each, and select a final graph based on the scores. Typical constraint-based algorithms include PC (Tsagris, 2019) and Fast Causal Inference (Glymour et al., 2019). Such approaches have widely applicability because they can handle various types of data distributions and causal relations, given reliable conditional independence testing methods.

For our problem, the intermediate conditional independence testing results can provide rich insights about the causal rules, so we design a causal rule discovery method based on the PC algorithm.

3 Preliminaries

3.1 Knowledge Graphs

A knowledge graph (KG) $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{S})$ can be regarded as a type of heterogeneous information network, where $\mathcal{E}$ is the *entity* set, $\mathcal{R}$ is the *relation* set, and $\mathcal{S} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ is the *triple* set. A triple can be denoted as $< h, R, t >$ (or $R(h,t)$[1]), where h is the head entity, t is the tail entity, and these two entities are connected by a relation R to form a *fact* in $\mathcal{G}$, An entity usually responses to multiple *concepts*, such as *apple* can be regarded as a fruit or a brand. Different relations may focus on different concepts of entities, for example, CEOof (Tim Cook, apple) and Contain (apple, glucose). The relation R usually represents a binary predicate in KG which is used to describe the existence of the relation between two entities, such as fact LocatedIn (Statue of Liberty, New York). Here we define this type of relation as *state relation*. In the more general KG, there may be numerical property of one relation, such as Rate(Lily, Titanic), the value of the relation Rate provide the quantitative opinion of the user Lity to the movie Titanic. This type of relation is denoted as *quantitative relation*.

3.2 Problem Definition

The fundamental tasks in KG, such as knowledge graph completion and knowledge graph reasoning, normally concern the specific relation. Therefore, the causes of the specific relation are essential for the KG. We define the effect of the causal rule as one relation R with the corresponding concepts C_1 and C_2 of the relation's head entities and tail entities. To get the effective causal rules, we assume the treatment of the effect should also be related with C_1 and C_2, therefore, we think the treatment of the causal rule should be the other relations between C_1 and C_2. Here we give the definition of KG Schema, based on which, we will formally define the causal rule.

Definition 3.1. ***KG Schema.*** *A KG schema (KGS) is a meta template for a KG $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{S})$, defined on a concept set $\mathcal{C}_S$ and a relation set $\mathcal{R}_S$, with $\mathcal{R}_S \in \mathcal{R}$. A KGS is a directed graph, denoted as $S_{KG} = (\mathcal{C}_S, \mathcal{R}_S, \mathcal{V}, \mathcal{B})$, with an node mapping function $\phi : \mathcal{V} \to \mathcal{C}_S$ and an edge mapping function $\psi : \mathcal{B} \to \mathcal{R}_S$, where each node v_i in node set $\mathcal{V}$ corresponds to one particular concept $\phi(v_i) \in \mathcal{C}_S$, and each edge b_i in edge set $\mathcal{B}$ corresponds to one particular relation $\psi(b_i) \in \mathcal{R}_S$.*

An instance of the KGS is a subgraph of KG, where each node of KGS is assigned one entity according to the concept $\phi(v_i)$.

Definition 3.2. ***Causal Rule.*** *A causal rule is defined on a pair of KGSs and two concepts C_1 and C_2, which correspond to two specific nodes v_1 and v_2 in each KGS. The causal rule can be represented as the following form:*

$$\underbrace{\mathcal{S}^C_{KG}(v^C_1, v^C_2)}_{body} \to \underbrace{\mathcal{S}^E_{KG}(v^E_1, v^E_2)}_{head} : \alpha, \tag{1}$$

where $\mathcal{S}^C_{KG}(v^C_1, v^C_2)$ is the body of the rule, $\mathcal{S}^E_{KG}(v^E_1, v^E_2)$ is the head of the rule, and α is the weight of the rule, representing the strength of the causal relationship between body and head. The head of the rule $\mathcal{S}^E_{KG}$ only includes one relation. Without loss of generality, we define $\phi^C(v^C_1) = \phi^E(v^E_1) = C_1$ and $\phi^C(v^C_2) = \phi^E(v^E_2) = C_2$.

For an instance of causal rule, v^C_1 and v^E_1 (or v^C_2 and v^E_2) must be assigned with the same entity. For example, the causal rule in Fig 1 is defined on the concepts *user* and *movie*, which reveals one of the causes of the users' rate.

In this paper, we mainly focus on discovering the causal rule and the corresponding weight in KG.

4 Causal Discovery in Knowledge Graphs

Informally, causation is defined as a relationship between two variables X and Y such that changes in X lead to changes in Y. The key difference between association and causation lies in the potential of confounding. Suppose that no direct causal relationship exists between X and Y but rather a third variable Z causes both X and Y. In this case, even though X and Y are strongly associated, altering X will not lead to changes in Y. Z is called a confounder. More formally, causation is a relationship between variables A and B that remains

[1] We will mainly use this expression in this paper, since it is often used in the rule mining literature (Galárraga et al., [n.d.]; Galárraga et al., 2015; Sadeghian et al., 2019)

after adjusting for confounders. Confounders can be observed or unobserved (latent).

The causal structure can be represented by a set of causal relationships among a set of variables, and the causal discovery is normally regarded as a the problem of learning the *whole* causal structure from observational data in the prior works. However, for the fundamental tasks in knowledge graph, such as knowledge graph completion and knowledge graph reasoning, only a specific subset of whole causal structure is concerned. If we have the knowledge that what information has effect on the queried relations, the original problem will be simplified based on the causation. Consequently, we only need to solve a *local* causal discovery problem, which is to find the all the KGS which has an impact on a given KGS. Since the *causal rule* (Definition 3.2) is defined on two concepts C_1 and C_2, the *variable* can be any KGS whose concept set $\mathcal{C}_S$ includes C_1 and C_2. Here we give the assumption 4.1 about the candidate causes for a specific KGS. When we examine the causation between the given KGS and one of its candidate causes, the rest of the candidate KGSs are valid confounders.

Assumption 4.1. ***Candidate Causes of KGS.*** *Given a KGS $\mathcal{S}_{KG} = (\mathcal{C}_S^E, \mathcal{R}_S^E, \mathcal{V}^E, \mathcal{B}^E)$ and two concepts $C_1 = \phi(v_1^E)$ and $C_2 = \phi(v_2^E)$, the candidate causes of KGS $\mathcal{S}_{KG}^E(v_1^E, v_2^E)$ is any KGS $\mathcal{S}_{KG}^{Ca} = (\mathcal{C}_S^{Ca}, \mathcal{R}_S^{Ca}, \mathcal{V}^{Ca}, \mathcal{B}^{Ca})$, with $C_1, C_2 \in \mathcal{C}_S^{Ca}$.*

4.1 Causal Variables and Samples

Due to there is no concept of ‘variables’ and corresponding ‘samples’ in KG, we first need to give the definitions of these two concepts in knowledge graph scenario, thus we can apply the traditional causal discovery methods to our problem. The causal rule is defined between two KGSs. For a given KGS, it is natural to treat this KGS and its candidate causes as the investigated *causal variables*, on which the independent tests are conducted. In the next, we will discuss the sampling mechanism of the causal variables, which is based on the following function defined on the instances of KGS.

Definition 4.1. ***KGS Function.*** *A KGS function $f_{\mathcal{S}_{KG}}$ is a set of mapping rules defined on a KGS $\mathcal{S}_{KG} = (\mathcal{C}_S, \mathcal{R}_S, \mathcal{V}, \mathcal{B})$ and a knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{S})$, and can be formulated as:*

$$f_{\mathcal{S}_{KG}} : E_1 \times E_2 \times \cdots \times E_n \to \mathbf{Y}, \tag{2}$$

where E_i is the entity set corresponding to the node v_i's concept $\phi(v_i)$ and $\mathbf{Y}$ is a m-tuple. Given an ordered edge list $[b_1, \ldots, b_m]$ of a $\mathcal{S}_{KG}$ and an instance of $\mathcal{S}_{KG}$ with $[e_1, \ldots, e_n]$, the mapping rule can be formulated as when relation $R_j = \psi(e_j)$ is a state relation:

$$\begin{cases} \mathbf{Y}_i = True \quad if \quad R_j(e_j^h, e_j^t) \in \mathcal{S}, \\ \mathbf{Y}_i = False \quad otherwise. \end{cases} \tag{3}$$

If R_j is a quantitative relation, the mapping rule should be:

$$\begin{cases} \mathbf{Y}_i = m \quad if \quad R_j(e_j^h, e_j^t) \in S, \\ \mathbf{Y}_i = None \quad otherwise, \end{cases} \tag{4}$$

where m is the value of the facts $R_j(e_j^h, e_j^t)$.

With $f_{\mathcal{S}_{KG}}$, we adopt language bias to define the sampling mechanism for the causal variables in knowledge graph. For a causal variable $X = \mathcal{S}_{KG}(v_1, v_2)$, an entity pair $(e_1 \in E_1, e_2 \in E_2)$ corresponds to a set of samples $\mathbf{x}(e_1, e_2)$ with different instances of $E_3, \ldots, E_n$. To remove the redundant and nonstandard results of KGS function, we define the following mapping rules: given the ith instance $(e_1, e_2, e_3^i \ldots, e_n^i)$,

(1) the relations of $\mathcal{S}_{KG}$ are all state relations:

- $x^i(e_1, e_2) = 1$, if every element of $f_{\mathcal{S}_{KG}}(e_1, e_2, e_3^i \ldots, e_n^i)$ is true. This is the *positive sample* for the variable $X = \mathcal{S}_{KG}(v_1, v_2)$.

- $x^1(e_1, e_2) = 0$, if there is no instance, which can make every element of $f_{\mathcal{S}_{KG}}$ to be true. This is the *negative sample* for the variable $X = \mathcal{S}_{KG}(v_1, v_2)$.

(2) some relations of $\mathcal{S}_{KG}$ are quantitative relations:

- $x^i(e_1, e_2) = \mathbf{m}$, where $\mathbf{m}$ is a k-d vector and every dimension of $\mathbf{m}$ corresponds to the value of quantitative relation in $f_{\mathcal{S}_{KG}}(e_1, e_2, e_3^i, \ldots, e_n^i)$ following the edges order defined by $\mathcal{S}_{KG}$.

- The instance result is removed if one element of $f_{\mathcal{M}_{KG}}(e_1^i, \ldots, e_n^i)$ is None.

Based on the above mapping rules, we can see for an entity pair, there may be several samples for a KGS-based variable, which depends on the instances of other entity nodes. So based on the entity pair, the one-to-one sample mapping for KGS-based variables is impossible. Therefore, we take the many-to-many mapping. Given a KGS $\mathcal{S}_{KG}(v_1, v_2)$ and its $k-1$ candidate causes, for an entity pair $(e_1 \in E_1, e_2 \in E_2)$, the causal samples of k variables will be the following cartesian product: $\mathbf{x}_1(e_1, e_2) \times \cdots \times \mathbf{x}_k(e_1, e_2)$.

4.2 Causal Rule Discovery

Our goal is to discover all the KGS, which have causal relationship with the given KGS. Here we propose an efficient causal rule discovery method *CaRules* which performs the following steps:

Step-1: Data-Driven Path Finding. According to assumption 4.1, any KGS, which can support any entity pair $(e_1, e_2), e_1 \in E_1, e_2 \in E_2$, is a valid candidate cause for $\mathcal{S}^E_{KG}(v_1, v_2)$. So we find all the candidate causes by searching all the paths between entity pair (e_1, e_2) of $\mathcal{S}^E_{KG}(v_1, v_2)$ based on the following considerations:

1) Any graph contains two specific nodes can be represented as a path between them (duplicate nodes are permitted). For example, as shown in Fig.2, the structure between C_1 and C_2 can be uniquely inferred by the path $C_1 \xrightarrow{R_1} C_2 \xrightarrow{R_3} C_4 \xleftarrow{R_3} C_2 \xrightarrow{R_2} C_3$.

2) There are many well-studied path finding algorithms, which can search the paths under different types of constraints, such as Dijkstra's algorithm (Lanning et al., 2014), A* search (Cui and Shi, 2011), best-first search (Heusner et al., 2018), etc. These off-the-shelf methods can be directly adopted to our framework. In the experiments, we adopt the best-first search algorithm.

3) Lots of existing rule mining methods (Sadeghian et al., 2019; Yang et al., 2017; Ho et al., 2018) designed for the *closed rules*, meaning that each entity set appears in at least two edges of the rule. It renders the path-like graph structure in most cases.

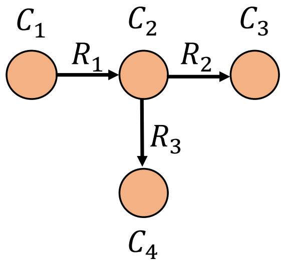

Figure 2: An illustrative KGS, which includes four concepts and three relations.

Since the number of candidate KGS can be the power level of the number of relation types, we require that any KGS created during path finding need to be supported by at least a_{sup} entity pair (e_1, e_2) in the training KG, and the length of the path is no more than ℓ, where a_{sup} and ℓ are the hyper-parameters.

Step-2: Refinement of the Identical Relations. In KG, there may be some identical relations, even though they have different relation names. For example, Wife(A,B) $\leftrightarrow$ Husband(B,A), if A is the wife of B, then B must be the husband of A. However, they will lead the invalid independence test in the following causal discovery step, even though these two relations have very strong causal relationship with each other. In particular, based on the causal variable and sample definitions in KG (Sec 4.1), these two relations are the same variables for the causal discovery method, since the values of their samples are the same all the time. When one relation is treated as the conditional variable in the independent test of the other one, the conditional independent (CI) test $CI(X, Y|X)$ will be judged as independent. So for an analyzed KGS $\mathcal{S}^E_{KG}$, we search all the identical KGSs in the input KG and temporarily remove them from the candidate cause set in the independent test period. The causal rules which include the identical KGSs will have the highest weight, when they are applied into the downstream tasks.

Step-3: PC-like Causal Rule Discovery. PC algorithm (Tsagris, 2019) is a prototypical constraint-based algorithm for learning Bayesian networks. This algorithm starts with the fully connected network and uses the conditional independence test to decide whether an edge will be removed or retained. This feature makes the PC algorithm efficient in the sparse true underlying graphs.

In our problem, we assume the causal relationships between the KGSs are sparse and propose a PC-like causal rule discovery method (Algo. 1) in KG. Given a KGS $\mathcal{S}^E_{KG}$ (denoted as variable Y in this algorithm), for each candidate cause $\mathcal{S}^{Ca}_{KG}$ (denoted as variable X_i), the proposed algorithm decides whether X should be retained in candidate causes set S^{Ca} by testing the independence of X_i and Y conditioning on a subset Z of $S^{Ca} \backslash \{X_i\}$. The CI tests are organised by levels (based on the size d of the conditioning sets). At the first level ($d = 0$), all pairs of variables are tested conditioning on the empty set. Some of the candidate causes would be removed and the algorithm only tests the remaining candidate causes in the next level ($d = 1$). The size of the conditioning set, d, is progressively increased (by one) at each new level until d is greater than $|S^{Ca}| - 1$.

Algorithm 1 PC-like Causal Rule Discovery

```
Input: $Y$ and $\{y_i\}, i = 1, \dots, N$ :variable and samples of analyzed KGS $\mathcal{S}^{E}_{KG}$ ; $S^{Ca} = \{X_j\}, j = 1, \dots, M$ and $\{\{x_i\}_j\}, i = 1, \dots, N$: variables and samples of candidate causes;
Output: causes $S^C$ of $Y$
  Let level $d = 0$
  repeat
      for each $X \in S^{Ca}$ do
          for each subset $Z \in S^{Ca} \backslash \{X\}$ and $|Z| = d$ do
              Test CI(X,Y|Z)
              if CI(X,Y|Z) then
                  Remove $X$ from $S^{Ca}$
                  Break
              end if
          end for
      end for
      $d = d + 1$
  until $d > |S^{Ca}| - 1$
  $S^C = S^{Ca}$
```

For the CI test part in Algo. 1, we adopt SCI algorithm (Marx and Vreeken, 2019) in the experiments, which can works well on limited samples and multiple conditional variables.

5 Experiment

In this section, we evaluate CaRules on three scenarios: simulated link prediction under closed world assumption (CWA)[2] (Hogan et al., 2021), kinship prediction, and movie rating prediction. We also empirically assess the quality and interpretability of the learned causal rules. The sampling size of each KGS-based variable a_{sup} is set to 50 in our experiments. Statistics about each data set are shown in Table 1. As the closest bunch of related works with us is rule mining, we select two popular and state-of-the-art methods for rule mining Neural LP (Yang et al., 2017) and DRUM (Sadeghian et al., 2019) as our baselines.

5.1 Link Prediction under CWA

In a real knowledge graph, the CWA assumption can hardly be strictly satisfied, due to the incredible negative edges in graph. However, the quality of non-monotonic rules is hard to evaluate without CWA. Consequently, we conduct a simulated experiment under CWA, where training and testing use the disjoint set of entities. We generate experimental KGs, which include three concepts and four state relations, by a causal graph defined on three KGSs (shown in Fig. 3). We use the performance of link prediction task on R_3 to evaluate all methods. The facts of the KG are splited into three parts:*train*, *test info*, and *test*. And the entities in test parts and the train part are disjoint. The *test info* part includes facts of new entities on R_1, R_2, R_4, and *test* part includes the queried facts on R_3. We design two settings with different causal mechanisms, corresponding to the monotonic and non-monotonic rules respectively (shown in Table 2). In particular, the conditional probability distributions of X_2 and X_3 given X_1 are Bernoulli distributions, whose probability mass function is $f_X(x) = p^x(1-p)^{1-x}$. The corresponding parameters in these two settings are listed in Table 2. To evaluate the quality of learned rules from biased data, we generate training and I.I.D testing samples from a Bernoulli distribution of X_1 with $p = 0.5$. For OOD testing setting, we use $p = 0.1$ and $p = 0.9$.

Table 1: Dataset statistics for all the experiments

	#Triplets	#Relations	#Entities
Simulation	6095	4	1590
Family	28356	12	3007
Recommendation	174941	20	32056

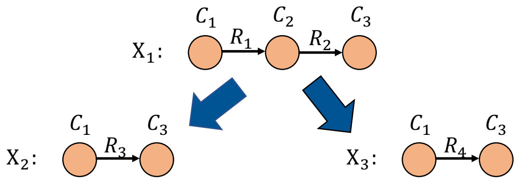

Figure 3: The causal graph of KGSs, based on which the simulated KGs are generated .

Table 2: The parameters of conditional distributions in different settings.

settings	$X_2\|X_1 = 1$	$X_2\|X_1 = 0$	$X_3\|X_1 = 1$	$X_3\|X_1 = 0$
mono.	p=0.8	p=0.1	p=0.8	p=0.1
non-mono.	p=0.2	p=0.9	p=0.2	p=0.9

In the inference process of Neural LP (Yang et al., 2017) and DRUM (Sadeghian et al., 2019), given an entity e_h, the score of each valid entity e_t

[2] CWA assumes a knowledge representation is a complete description of the world. A statement that is true is also known to be true, and the unknown statement is false.

Table 3: Top 3 rules obtained by each system learned on family dataset. Results of DRUM and Neural-LP are taken from (Sadeghian et al., 2019). The strikethroughs indicate the wrong results. The rules whose head can be inferred by the body uniquely are in bold.

Neural-LP	brother(A, B) ← sister(B,A)
	brother(A,C) ← sister(B,A), sister(C,B)
	brother(A,C) ← sister(B,A), brother(C,B)
		~~wife(A,C) ← husband(B, A), husband(C, B)~~
		wife(A,B) ← husband(B,A)
		~~wife(A,C) ← husband(B,A), daughter(B,C)~~
			son(A,C) ← brother(A,B), son(B,C)
			~~son(A,B) ← brother(A,B)~~
			~~son(A,C) ← mother(B,A), son(B,C)~~

DRUM	brother(A,C) ← uncle(B,A), nephew(C,B)
	brother(A,C) ← nephew(A,B), nephew(C,B)
	brother(A,C) ← sister(B,A), bother(C,B)
		wife(A,B)← husband(B,A)
		wife(A,C)← mother(A,B), father(C,B)
		wife(A,C) ← son(B,A), father(C,B)
			son(A,C) ← nephew(A,B), brother(B,C)
			son(A,C) ← brother(A,B), mother(C,B)
			son(A,C) ← brother(A,B), daughter(B,C)

CaRules	**brother(A,C) ← son(A,B), father(B,C)**
	brother(A,C) ← father(B,A), daughter(C,B)
	brother(A,C) ← brother(A,B), sister(B,C)
		wife(A,B) ← husband(B,A)
		wife(A,C) ← son(B,A), son(B,C)
		wife(A,C) ← mother(A,B), father(C,B)
			son(A,B) ← father(B,A)
			son(A,C) ← son(A,B),husband(B,C)
			son(A,C) ← sister(B,A),daughter(B,C)

is defined as sum of the weight of rules that imply query(e_h,e_t), and a ranked list of entities will be returned, where higher the score implies higher the ranking. This method works well for the monotonic rules, but it will fail for the non-monotonic rules.

Here we design a new inference method, which works for both monotonic and non-monotonic rules. Given an entity (e_h,e_t), the formula of calculating the probability of the query (e_h,e_t) to be true is as follows:

$$y_p = \sum_{i}^{K} \tilde{w}_i \big(Q_i \bar{Y}_{X_i=1} + (1 - Q_i)\bar{Y}_{X_i=0}\big) \quad (5)$$

where K is the number of causal rules for the queried meta KG, $\tilde{w}_i$ is the normalized weight for the ith result rule. $\bar{Y}_{X_i=1}$ denotes the proportion of the queried relation to be true when the body of the ith causal rule is true in the training data and $\bar{Y}_{X_i=0}$ denotes the proportion of the queried relation to be true when the body of the ith causal rule is false. $Q_i = 1$ when the body of the ith causal rule holds for the entity pair (e_h,e_t), otherwise $Q_i = 0$. The results will be ranked by y_p of each valid e_t.

The evaluation metrics we used are Hits@k and MRR. In particular, MRR is the average of the reciprocal ranks of the desired entities, while Hits@k computes the percentage of how many desired entities are ranked among top k. We compare the performances of the proposed CaRules with the rule mining algorithms Neural-LP (Yang et al., 2017) and DRUM (Sadeghian et al., 2019).

The results demonstrate that CaRules empirically outperforms DRUM and Neural-LP in both monotonic and non-monotonic settings. Note that the results of DRUM in all settings and Neural-LP in the OOD setting($p_{X_1} = 0.9$) show the clear degradation for the non-monotonic rules. Compared with the baseline methods, our method suffers less performance degradation under the OOD setting, especially with the non-monotonic link prediction method. Our method provides a path to find the high level knowledge, which can explain the generation mechanism. Moreover the results show the CaRules can learn both the monotonic and non-monotonic rules. With the our proposed inference method, CaRules achieves the highest performance in all settings.

Table 4: The results of link prediction task under CWA. CaRules denotes the results of our rule mining algorithm with the link prediction method used in DRUM and Neural-LP. CaRules (non-moto.) denotes the results of our rule mining algorithm with the non-monotonic link prediction function (Eq. 5).

	p_{X_1}	Method	monotonic setting				non-monotonic setting			
			MRR	k=10	k=3	k=1	MRR	k=10	k=3	k=1
I.I.D	0.5	Neural-LP	0.71	0.73	0.69	0.69	0.77	0.78	0.76	0.76
		DRUM	0.76	0.87	0.83	0.70	0.48	0.92	0.75	0.24
		CaRules	**1.00**	**1.00**	**1.00**	**1.00**	0.91	0.90	0.90	0.90
		CaRules (non-moto.)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
OOD	0.1	Neural-LP	0.71	0.73	0.69	0.69	0.89	0.90	0.88	0.88
		DRUM	0.45	0.54	0.34	0.17	0.40	0.90	0.66	0.14
		CaRules	0.71	0.70	**0.70**	**0.70**	0.19	0.15	0.15	0.15
		CaRules (non-moto.)	**0.74**	**0.93**	**0.70**	**0.70**	**1.00**	**1.00**	**1.00**	**1.00**
OOD	0.9	Neural-LP	0.79	0.81	0.78	0.78	0.44	0.41	0.41	0.41
		DRUM	0.88	0.99	0.93	0.82	0.32	0.76	0.39	0.32
		CaRules	**1.00**	**1.00**	**1.00**	**1.00**	**0.97**	**0.97**	**0.97**	**0.97**
		CaRules (non-moto.)	**1.00**	**1.00**	**1.00**	**1.00**	0.84	0.92	0.82	0.82

Table 5: Experimental results of link prediction task on family data set. Results of DRUM and Neural-LP are taken from (Sadeghian et al., 2019).

Method	MRR	Hits@10	Hits@3	Hits@1
Neural-LP	0.91	0.99	0.96	0.86
DRUM	0.92	**1.00**	**0.99**	0.86
CaRules	**0.96**	0.99	0.98	**0.94**

5.2 Kinship Prediction

Following DRUM (Sadeghian et al., 2019), we conducted experiments on the kinship prediction task based on a family dataset (Kok and Domingos, 2007), which contains the bloodline relationships between individuals of multiple families. Compared with the open KGs, the relationships of families on the dataset (Kok and Domingos, 2007) are

Table 6: Experimental results of link prediction task on family data set under limited rules number.

	MRR	Hits@10	Hits@3	Hits@1
Neural-LP(Top1)	0.43	0.44	0.43	0.39
CaRules(Top1)	**0.61**	**0.66**	**0.62**	**0.49**
Neural-LP(Top2)	0.68	0.68	0.67	0.64
CaRules(Top2)	**0.78**	**0.89**	**0.84**	**0.68**
Neural-LP(Top3)	0.81	0.82	0.81	**0.76**
CaRules(Top3)	**0.83**	**0.93**	**0.88**	0.75

more strong and compact. This data set is convenient to evaluate the quality and interpretability of the rules. In the link prediction experiments of family data set in DRUM, the facts are split into three parts: facts, train and test, where facts are used to construct the relation adjacency matrix and train are used to learn the parameters of the model. Since facts and train are all visible in the training period for the baseline, we use facts and train to learn the rules and use the test data to examine the method. The maximum path length ℓ is 2, which is a best choice based on DRUM's results. The rules in family data set are monotonic, so we adopt the same inference method with DRUM to achieve the fair comparison.

Table 5 shows the results of the proposed method and baseline methods. Particularly, for Hits@10 and Hits@3, CaRules performs slightly worse than DRUM by 1%, while it obtains significantly improvement on MRR and Hits@3 by 4% and 8% respectively, which demonstrates the effectiveness of the proposed method. Besides, we further evaluate the quality of the mined rules using the top-k explainable rules given by each algorithm. According to the results shown in Table 6, we can find the our method significantly outperform the baseline method on all metrics, especially when only one rule is used in the link prediction. In the meanwhile, we also show the top-3 interpretable results for three relations in Table 3 for reference, and we can find the following two observations: (1) the proposed methods gives more accurate prediction than Neural-LP; (2) the proposed method gives more stable prediction than DRUM. For the three relations, our method can give at least one rule, whose body's relations can infer head's relation uniquely.

Table 7: Movie Rating Prediction

Method	Neural-LP	DRUM	CaRules
Accuracy	72.86	74.17	**77.39**

5.3 Movie Rating Prediction

To demonstrate the effectiveness of CaRules for mining quantitative causal rules, we conduct an experiment on a typical recommendation application, movie rate prediction. We use the real data set collected from Douban and construct a KG, whose statistics are shown in Table 1. Douban[3] is a famous Chinese website of movie information and reviews, where the user can rate any movies and reviews about them. The rates fall in the range of 1 to 5, where higher ratings mean users like the movies while lower rates mean users' negative feedbacks to the movies. Normally, a movie is identified as a user's favorite movie if the user rates the movie with a score of 4 or 5. So we transform the original 5-level rating to 2-level rating with the threshold 4. Based on the data set, we first mine the causal rules which may effect the rating of the users for movies. Then we evaluate the discovered rules, by the rating prediction task.

Although this experiment also is a binary prediction task, it is not the same with the state link prediction, like the kinship prediction. Because given a user-movie pair, the corresponding rating relation actually has three possibilities: high, low and not existent. In this experiment, we take the prediction accuracy for the Rating of test queries as the evaluation metric. Due to the space limitation, we give the experimental details in Appendix. Besides, Table 7 shows the proposed method also gives the best recommendation result, surpassing the runner up by 3.22%.

6 Conclusion

In this paper, we investigate the problem of distilling the causal component of metaknowledge from knowledge graph, that is the causal rules generating the factual knowledge. For this purpose, we define the concept of KG schema and learn the causal relationship between two KGSs on the basis of a generic mining paradigm. To effectively capture the explainable causal rules, we propose a novel algorithm called CaRules, consisting of the candidate causes search via the advanced path finding method and an efficient PC-like process that conducts reduced conditional independence tests to find causal structure. The extensive experiments can well support our claims from a variety of aspects.

[3] https://www.douban.com/

References

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. Advances in neural information processing systems 26 (2013).

Mark Burgin. 2016. Theory of knowledge: structures and processes. Vol. 5. World scientific.

Xiao Cui and Hao Shi. 2011. A*-based pathfinding in modern computer games. International Journal of Computer Science and Network Security 11, 1 (2011), 125–130.

James A Evans and Jacob G Foster. 2011. Metaknowledge. Science 331, 6018 (2011), 721–725.

Santo Fortunato, Carl T Bergstrom, Katy Börner, James A Evans, Dirk Helbing, Staša Milojević, Alexander M Petersen, Filippo Radicchi, Roberta Sinatra, Brian Uzzi, et al. 2018. Science of science. Science 359, 6379 (2018).

Mohamed H Gad-Elrab, Daria Stepanova, Jacopo Urbani, and Gerhard Weikum. 2016. Exception-enriched rule learning from knowledge graphs. In International Semantic Web Conference. Springer, 234–251.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. 2015. Fast rule mining in ontological knowledge bases with AMIE+. The VLDB Journal 24, 6 (2015), 707–730.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd international conference on World Wide Web. 413–422.

Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. [n.d.]. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In Proceedings of the 22nd international conference on World Wide Web - WWW '13 (Rio de Janeiro, Brazil, 2013). ACM Press, 413–422. https://doi.org/10.1145/2488388.2488425

Clark Glymour, Kun Zhang, and Peter Spirtes. 2019. Review of causal discovery methods based on graphical models. Frontiers in genetics 10 (2019), 524.

Manuel Heusner, Thomas Keller, and Malte Helmert. 2018. Best-case and worst-case behavior of greedy best-first search. International Joint Conferences on Artificial Intelligence.

Vinh Thinh Ho, Daria Stepanova, Mohamed H Gad-Elrab, Evgeny Kharlamov, and Gerhard Weikum. 2018. Rule learning from knowledge graphs guided by embedding models. In International Semantic Web Conference. Springer, 72–90.

Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia d'Amato, Gerard de Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, et al. 2021. Knowledge graphs. Synthesis Lectures on Data, Semantics, and Knowledge 12, 2 (2021), 1–257.

Guido W Imbens and Donald B Rubin. 2010. Rubin causal model. In Microeconometrics. Springer, 229–241.

Stanley Kok and Pedro Domingos. 2007. Statistical predicate invention. In Proceedings of the 24th international conference on Machine learning. 433–440.

Daniel R Lanning, Gregory K Harrell, and Jin Wang. 2014. Dijkstra's algorithm and Google maps. In Proceedings of the 2014 ACM Southeast Regional Conference. 1–3.

Alexander Marx and Jilles Vreeken. 2019. Testing conditional independence on discrete data using stochastic complexity. In The 22nd International Conference on Artificial Intelligence and Statistics. PMLR, 496–505.

Pouya Ghiasnezhad Omran, Kewen Wang, and Zhe Wang. 2019. An embedding-based approach to rule learning in knowledge graphs. IEEE Transactions on Knowledge and Data Engineering (2019).

Judea Pearl. 2010. Causal inference. Causality: Objectives and Assessment (2010), 39–58.

Ali Sadeghian, Mohammadreza Armandpour, Patrick Ding, and Daisy Zhe Wang. 2019. DRUM: End-To-End Differentiable Rule Mining On Knowledge Graphs. Advances in Neural Information Processing Systems 32 (2019), 15347–15357.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. 2000. Causation, prediction, and search. MIT press.

Thomas Pellissier Tanon, Daria Stepanova, Simon Razniewski, Paramita Mirza, and Gerhard Weikum. 2017. Completeness-aware rule learning from knowledge graphs. In International Semantic Web Conference. Springer, 507–525.

Michail Tsagris. 2019. Bayesian network learning with the PC algorithm: an improved and correct variation. Applied Artificial Intelligence 33, 2 (2019), 101–123.

Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019b. Multi-task feature learning for knowledge graph enhanced recommendation. In The World Wide Web Conference. 2000–2010.

Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019a. Kgat: Knowledge graph attention network for recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 950–958.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems 32, 1 (2020), 4–24.

Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the International Conference on Learning Representations (ICLR) 2015 (proceedings of the international conference on learning representations (iclr) 2015 ed.).

Fan Yang, Zhilin Yang, and William W Cohen. 2017. Differentiable learning of logical rules for knowledge base reasoning. In Proceedings of the 31st International Conference on Neural Information Processing Systems. 2316–2325.

Appendix

We supply the details of the movie rating prediction task here. This task aims to prove the effectiveness of CaRules for mining quantitative causal rules. Although we transform the original 5-level rating to 2-level rating to fit other baseline models, it is worth noting that it is not the same with the state link prediction, like the kinship prediction, because the corresponding rating relation actually has three possibilities: high, low and not existent. Therefore, for the traditional state rule mining method, we transform the rating relation into two relations high rating and low rating and conduct the rule mining for them, respectively. For a query Rating(user1, movie1), we conduct the link prediction task for both highrating(user1,?) and lowrating(user1,?). The relation, which give the higher rank for movie1, will be treated as the final result. For the quantitative relation, our method only consider existent samples in rule mining period. In the prediction period, the queried entity pair (user1, movie1) corresponds to the specific information of quantitative causal rules in the training data. Taking the first mined causal rules in Table 8 as an example, in the training data, we can access all the ratings of user1 for a movie, whose editor is the same with movie1's. Therefore in this experiment, we give the rating prediction results for the query Rating(user1,movie1) based on the following formula:

$$y_v = \sum_i^K \tilde{w}_i \big(Q_i \bar{Y}^s_{X_i=1} + (1-Q_i)\bar{Y}^s_{X_i=0}\big) + \sum_i^H \tilde{w}_i \bar{Y}^q_i, \tag{6}$$

where K and H is the number of state and quantitative causal rules, respectively, $\tilde{w}_i$ is the normalized weight, $\bar{Y}^s_{X_i=1}$ denotes the mean of the queried relation when the body of the ith state causal rule is true in the training data and $\bar{Y}^s_{X_i=0}$ denotes the mean of the queried relation to be true when the body of the ith causal rule is false. $Q_i = 1$ when the body of the ith causal rule holds for the entity pair (user1,movie1), otherwise $Q_i = 0$. And $\bar{Y}^q_i$ denotes the mean of the instances' rating, which satisfied the ith quantitative causal rule. For a query Rating(user1,movie1), the result will be high if y_v is over than 0.5, otherwise it is low.

The results in Table 8 suggests that the rating from the uses is mainly based on the rated movies which shares the same staff, such as writer, actors, director, etc. According to the rating results, we can find a stronger causal relationship between the rating of the movie and its writer than other pairs.

Table 8: Rules for recommendation dataset

Rank	Rules
1	Rating(User,Movie) ← Rating(User,Movie),Writer(Person,Movie),Writer(Person,Movie)
2	Rating(User,Movie) ← Rating(User,Movie),Actress(Person,Movie),Actress(Person,Movie)
3	Rating(User,Movie) ← Rating(User,Movie),Director(Person,Movie),Actor(Person,Movie)
4	Rating(User,Movie) ← Rating(User,Movie),Director(Person,Movie),Director(Person,Movie)
5	Rating(User,Movie) ← Rating(User,Movie),Composer(Person,Movie),Composer(Person,Movie)
6	Rating(User,Movie) ← Rating(User,Movie),Director(Person,Movie),Writer(Person,Movie)
7	Rating(User,Movie) ← Rating(User,Movie),Producer(Person,Movie),Producer(Person,Movie)
8	Rating(User,Movie) ← Rating(User,Movie),Writer(Person,Movie),Director(Person,Movie)
9	Rating(User,Movie) ← Rating(User,Movie),Writer(Person,Movie),Actor(Person,Movie)