CEUR-WS.org/Vol-2465/semex_paper4.pdf

Interactive Causal Discovery in Knowledge Graphs

Melanie MUNCH[1], Juliette DIBIE[1], Pierre-Henri WUILLEMIN[2], and Cristina MANFREDOTTI[1]

[1] UMR MIA-Paris, AgroParisTech, INRA, Paris-Saclay University, 75005 Paris, France
[2] Sorbonne University, UPMC, Univ Paris 06, CNRS UMR 7606, LIP6, 75005 Paris, France

Abstract. Being able to provide explanations about a domain is a hard task that requires from a probabilistic reasoning's viewpoint a causal knowledge about the domain variables, allowing one to predict how they can influence each others. However, causal discovery from data alone remains a challenging question. In this article, we introduce a way to tackle this question by presenting an interactive method to build a probabilistic relational model from any given relevant domain represented by a knowledge graph. Combining both ontological and expert knowledge, we define a set of constraints translated into a so-called relational schema. Such a relational schema can then be used to learn a probabilistic relational model, which allows causal discovery.

Keywords: Causal discovery, Probabilistic Relational Models, Knowledge Graph

1 Introduction

Probabilistic models such as Bayesian networks (BNs) are a good approach to represent complex domains, as they allow to express probabilistic links between variables. However, correlation does not imply causality, and thus these models lack explainability. Yet it could be useful when studying a disease to identify the cause (the actual illness) and the consequence (the symptoms). Uncovering causal relations from data alone is a difficult task: previous works have presented the use of interventions to construct causal models [21], but these interventions require to be able to change certain variables while keeping other constant, which is not always easily doable. Assessing for instance the impact of one's genotype and cigarettes smoking habits on lung cancer would theoretically require to intervene on both of these criteria. If controlling whether one is smoking or not

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

is possible (yet not really ethical), it is however impossible to directly control the genotype. As a consequence, for practical, ethical and economical reasons, direct interventions are often not available to learn causal relations. In this article, we present an interactive method that offers to introduce ontological and expert knowledge into the learning of a probabilistic model from a given knowledge graph (KG) [12], in order to discover causal knowledge. This causality helps to better explain a domains by allowing to reason on higher levels: a complete causal graph can answer causal questions such as ”If I take this drug, will I still be sick?”; or even answer counter factual questions as ”Had I not taken this drug, would I still be sick?”. We propose to achieve this by using probabilistic relational models (PRMs) [14]. PRMs are an object-oriented extension of BNs, thus allowing a better representation between the different attributes. However, their learning can be tricky due to this specificity. Using the semantic and structural information contained in a KG , it can be greatly eased and, thus, be guided toward a learned model close to the reality. However, many different probabilistic models can be deduced from a same KG depending on the user (a domain expert) expectation. We present in this paper an interactive method to help such a user to build a probabilistic reasoning model from a KG able to answer his/her questions. The first section of this paper presents the background and state of the art, especially on PRM and causal discovery. The second section presents our approach to learn a PRM guided by the ontology and the user's knowledge. The third section presents an application of our method on a portion of DBPedia. The last section concludes this paper.

2 Background and State of the Art

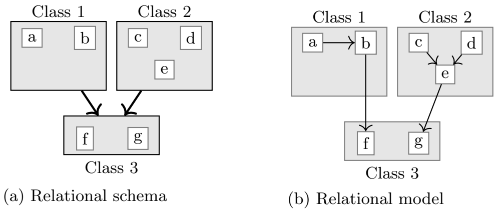
The main idea of our method is to learn a probabilistic model under causal constraints given both by a user and the ontology. From the learned model we then are able to extract causal knowledge.

2.1 Probabilistic Models: BN and PRM

A BN is the representation of a joint probability over a set of random variables that uses a directed acyclic graph (DAG) to encode probabilistic relations between variables. Learning a BN requires learning both its structure and parameters. In our case, since learning is done under causal constraints, we need to express the conditional independence of this BN, which could give us new insight on the causality of this graph. Indeed, even if a correlation found between two variables of a BN does not prevail on the arc's orientation (explaining why causal discovery from data alone is difficult to achieve), some of these arcs also indicates conditional independence and are necessary to ensure the probabilistic information encoded in the BN. An essential graph (EG) [16] is a semi-directed graph associated to a BN. They both share the same skeleton, but the EG's edges' orientation depends on the BN's Markov equivalence class. If the edge's orientation is the same for all the equivalent BNs, then it means that its orientation is necessary to keep the underlying probabilistic relations encoded in the graph: in this case, the edge is also oriented in the EG, and is called an **essential arc**. On the contrary, if the edge's orientation is not the same for all the equivalent BNs, then it means that its orientation can be both ways without changing the probabilistic relations, and it stays unoriented in the EG. Thus the EG expresses whether an orientation between two nodes can be reversed without modifying the probabilistic relations encoded in the graph: whenever the constraint given by an essential arc is violated, the conditional independence requirements are changed and the structure of the model itself has to be changed. With a BN learned under causal constraints such as in our method, the EG can then give us a new insight: if an arc is oriented, then it has to be kept if we want to conserve all the information we have provided during the learning.

However, our method also requires to use ontology's classes to group attributes by specific causal relations in order to learn them, and BNs lack such notion of modularity. As a consequence we turn to PRMs, that extend BNs' representation with the oriented-object notion of classes and instantiations. PRMs [14] are defined by two parts: a high-level, qualitative description of the structure of the domain that defines the classes and their attributes (i.e. the **relational schema** RS as shown Fig. 1 (a)), and a low-level, quantitative information given by the probability distribution over the different attributes (i.e. its **relational model** RM as shown in Fig. 1 (b)). Classes in the RS are linked together by so-called **relational slots**, that indicates the direction of probabilistic links. For instance, Fig. 1 has two classes 1 and 2 with a relational slot toward Class 3: it means that probabilistic links can exist between the attributes of class 1 and 2 with class 3's, and that they have to be oriented from the attributes of class 1 and 2 towards those of class 3. Using the RS structural constraints, each class can then be learned like a BN (in our case, we use the classical statistical methods Greedy Hill Climbing). As a consequence, a system of instantiated classes linked together is equivalent to a bigger BN composed of small repeated BNs, and thus can be associated to an EG.

Numerous related works have established that using constraints while learning BNs brings more efficient and accurate results, for parameters learning [9] or structure learning [10]. In case of smaller databases, constraining the learning can also greatly improve the accuracy of the model [19]. In this article we define structural constraints as an ordering between the different variables. The K2 algorithm [7], for instance, requires a complete ordering of the attributes before learning a BN, allowing the introduction of precedence constraints between the attributes. This particular algorithm needs a complete knowledge over all the different attributes precedences; however problems of learning with partial order have also been tackled [20]. In our case we will likewise transcribe incomplete knowledge as partial structural organization for the PRM's RS in order to discover new causal relations.



(a) Relational schema (b) Relational model

Fig. 1. The high (a) and low (b) level structures of a PRM

2.2 Causal Discovery

Causal models are DAGs allowing one to express causality between its different attributes [21]. Their construction is complex and requires interventions or controlled randomized interventions, which are often difficult or impossible to test. As a consequence the task of discovering causal relations using data, known as causal discovery, has been researched in various fields over the last few years. There are two types of methods for structure learning from data: independence-based ones, such as the PC algorithm [22], and score-based ones, such as Greedy Equivalent Search (GES) [6]. Usually independence-based methods give a better outlook on causality between the attributes by finding the *"true"* arc orientation, while the score-based ones offer a structure that maximizes the likelihood considering the data. Finally, other algorithms such as MIIC [23] use independence-based algorithms to obtain information considered as partially causal and thus allowing to discover latent variables. In this article we propose to explore if combining ontological and user's knowledge with BN learning score-based algorithms allows causal discovery. Other works have already proposed the use of EG: [15] for instance proposes two optimal strategies for suggesting interventions in order to learn causal models with score-based methods and the EG. Integrating knowledge in the learning has also been considered: [8] uses ontological causal knowledge to learn a BN and discover new causal relations with the EG; [4] offers a method to iterative causal discovery by integrating knowledge from beforehand designed ontologies to causal BN learning; [2] proposes two new scores for score-based algorithms using experts knowledge and their reliability; and [5] presents a tool combining ontological and causal knowledge in order to generate different argument and counterarguments in favor of different facts by defining enriched causal knowledge.

2.3 Ontology and Probabilistic Models

Using ontological knowledge in order to build probabilistic models has already been presented in numerous works. [13] uses the structure of an ontology to build and modify a BN by addressing three main tasks: the determination of the relevant variables, the determination of relevant properties and the computing of the probabilities. The learned model can then be used to reason on the domain. [1] presents a method for autonomic decision making combining BNs and ontologies, using the framework BayesOWL [11]. This framework allow the expression of a BN using the OWL standardization, and offers a set of rules aiming to automate the translation from an ontology to a BN. [3] presents a method to generate Object Oriented Bayesian Networks from ontologies using a set of rules they have defined. While PRM offers a way to express and consider the expert knowledge in learning, to the best of our knowledge no causality learning method that combines ontological and user's knowledge has been proposed yet.

3 Causal Discovery Driven by an Ontology

In this article we present an interactive method aiming to build a RS from a KG relying on the ontological and user's knowledge. This RS presents the different PRM's classes, relational slots and attributes, and is used to learn a PRM under causal constraints, allowing the deduction of causal knowledge. This method is split into three parts: (1) building a first RS from the ontological knowledge; (2) helping the user improving the proposed RS; (3) learning a PRM from the RS from which causal knowledge can be deduced. In a previous work [17] we present a method to help the user to build the RS but without fully exploiting the ontological knowledge. In this article, we focus on the first and second parts.

3.1 Relevant KGs

In theory, a PRM can be learned from any knowledge graph. However, not all are interesting to do so and some **selection criteria (SC)** must be fulfilled in order to learn a relevant probabilistic reasoning model. As an illustration we define a simple ontology dedicated to an university representation (Fig.2). It is composed of three main classes: the University class, the Student class and the Course class. An university is defined by its name and its fees; a student is defined by his/her name, sex, social standing, mean note and his/her subject of interest; a course is defined by its subject and its difficulty.

SC1. **The domain the KG is dedicated to must contain causal information to be deduced.** Our model can be used to simply discover simple probabilistic relations. However, it best shines when it encompasses causal knowledge, as it allows a far better explainability of the represented domain. Therefore, the user must have a causal question or at least an idea to search for causality information. In our university example, one might be interested in studying the influence of a student's social standing with his/her choice of courses and university.

SC2. **The KG contains datatype properties (DPs) whose values can be discretized.** The PRM's learning is based on classical BNs learning methods, which uses statistical analysis to learn the probabilistic relations. Therefore, our method needs data, which is given by DPs: they define our model's attributes. As a consequence, they must be relevant for the domain and their values discretizable for the learning: a DP indicating a student's ID is not interesting, as it is different for each student.

SC3. **The classes of the KG are instantiated enough and there are not too many missing DPs.** As stated before, the learning is based on statistical methods. As a consequence the studied KG must have enough instantiations in order to study their variability. Since all instances of the same class are compared together using their DPs, each instance's missing DP is considered as a missing value: as a consequence, each missing DP can decrease the precision of the model. For example, a single student's instance would not be enough to study the relations between a student and his/her courses; likewise, if we have multiple student's instances, but only one of them has a DP about his/her social standing, then we will not be able to study the influence of social standing over other parameters.

In order to deal with the causality, we consider in this article that the KG is **complete** and **verified**: all important variables are present (no confounding factor possible), and the distribution of the different values is balanced (none is arbitrarily prominent over others). Confounding factors occur when a correlation is found between two attributes, but with no direct causal link, and that the explanatory variable is missing. A classical example is the study of the correlation between one's reading ability and shoes' size: while both are indeed correlated, it is arguably not due to the fact that one causes the other. In this example, one's age is a confounding example, as it explains both: the older we are, the better we can read and the bigger our shoes are. As a consequence, confounding factors can lead to false causal reasoning, and must be avoided. In the rest of this article, we will consider that it is possible to learn from our data the true causal model of the domain (or at least a part of it). In the case where those criteria cannot be satisfied, then the causal learning could not be guarantied.

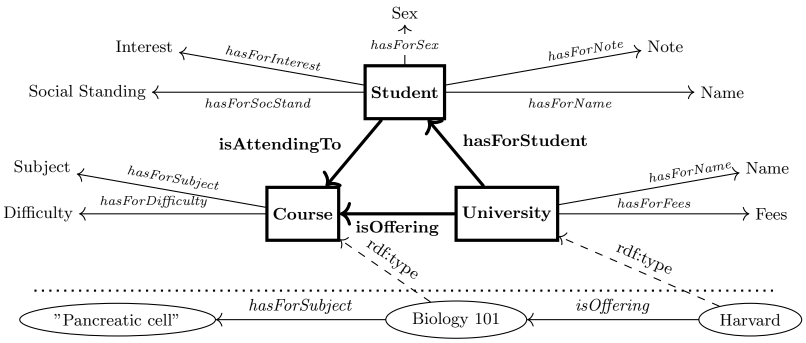


Fig. 2. Excerpt of a KG about students and universities

3.2 Interactive Building of a RS

From the ontological knowledge we automatically generate a first *RS* draft. The aim of this generation is to give the user a good preliminary overview of the

KG in order to help him/her build a probabilistic reasoning model. This transformation is done in three steps: (1) All ontology's classes become *RS* classes. With our university ontology we thus have three *RS* classes, University, Student and Course. (2) All ontology's DPs become attributes in the *RS* associated to their respective *RS* classes. In our example the University *RS* class owns two attributes, Name and Fees. (3) All ontology's object properties (OPs) become relational slots in the *RS*. In our example, the University *RS* class has two relational slots: one toward the Student *RS* class, and one toward the Course *RS* class. Before presenting to the user, we apply automatic **selection rules (SR)** based on the selection criteria presented above that directly modify the *RS*:

SR1. **The *RS* classes with too few instances are removed.**[3]

SR2. **The isolated *RS* classes are deleted.** If by applying **SR1** we break a path between two others *RS* classes, leading to the isolation of one of them (meaning there is no other relational slots linking this *RS* class), then the isolated *RS* class is also removed. We can illustrate this by adding a new OP in our example, *hasForTeacher*, taking for *domain* the Student class and for *range* a new Teacher class. In a regular situation, we would then be able to study the probabilistic relations between a teacher and a student, or a teacher and an university. However, if the student instances are not numerous enough to learn, then the Student *RS* class has to be removed, leaving the Teacher *RS* class isolated. As a consequence, it would not be possible anymore to study the probabilistic relations between a teacher and a university: the Teacher *RS* class has also to be removed.

SR3 **The attributes must be useful.** Since the learning of the PRM is based on statistical methods, problematic variables such as ones with too many missing data, values that do not repeat (for example IDs different for each instance) or that are not different (if we study a single university, its name is useless) are to be removed from the *RS*. In our example, if we had 50 students but only 3 with a DP about their social standing, then this corresponding attribute cannot be used to learn and is removed from the *RS*.

SR4 **The symmetric relational slots are deleted.** The PRM does not support cyclic relations, symmetric OPs cannot therefore be kept: as a consequence one of the corresponding relational slot in the RS must be discarded. In a first time, we automatically keep if possible the relational slot that corresponds to the most instantiated OP; if not, we randomly select one.

Once defined the *RS* is presented to the user who can intervene on different points. These **user modifications (UMs)** also directly modifies the *RS*:

UM1. **The choice of attributes.** Despite being instantiated enough, some selected DPs may be irrelevant according to the user, and thus their corresponding attributes need to be removed.

UM2. **The choice of relational slots.** The orientation of the relational slots has a great influence on the causal learning: if there is a relational slot from a

[3] The accepted missing values ratio is determined with the user.

class A to a class B, then all probabilistic links learned between attributes of class A and B have to be identically oriented. Broadly speaking it means that class A's attributes can explain class's B attributes, but not the contrary. However, not all ontology's OP are causal by default: as a consequence we need the user to validate when possible the orientation of the relational slots, or reverse it to express causality. He is also able to remove or add relations slots between classes if necessary.

UM3. **The choice of *RS* classes.** The orientation of the relational slots have a great influence over the learning of the causal knowledge. However, some *RS* classes' attributes might be intricate, meaning that two *RS* classes can be both explaining of and explained by a same other *RS* class. In our example, we can consider the relation between a student and his/her courses: the student's interest might explain his/her courses' subject; however, the courses' difficulty might explain the student's note. Fig. 3 (a) shows a first *RS*, in which both the interest and the note can explain the course's subject and difficulty. This is inconsistent with the idea that, on the contrary, the course's difficulty should explained the student's note. As a consequence, we offer the user a tool to split the *RS* classes in order to reflect this causal information. In Fig. 3 (b), the Student *RS* class has been split in two: a first *RS* class above with the interest attribute that can still explain the course's attributes, and a second below with the note attribute that can be explained by both the student's interest and the course's subject and difficulty.

UM4. **The choice of attributes.** As mentioned before the user can choose whether a DP can be kept or not in the *RS*. By default, a DP is directly translated into an attribute. However, when multiple identical DPs are involved it requires an intervention of the user: it can be the case when a single instance has several time the same DP (such as a Student who has multiple interests), or when a same *RS* class's instance can be explained (through a relational slot) by multiple instances of another *RS* class (e.g. a single course instance can be attended by many students). Here, the repeated DPs cannot be distinguished given the ontology: in these particular cases, we need to aggregate the given DP in order to allow a statistical learning. The aggregation can take many forms, depending on what the user wants (e.g. the mean value, the maximum value, if a certain value is present or not). For example, if we consider that a single course can have a variable number of different students, then it is not possible to learn a statistical model: some course will have 5 students, other 30, 12... No comparison is possible, and even if two courses had the same number of students, there is no way to distinguish one from another. As a consequence we need to transform these possible multiple attributes in the *RS* in an unique one, which is what aggregation allows us to do. For instance, instead of considering all the student's notes, we calculate the mean value: each course now have one attribute for the note, whether they had 1 or 100 students in the beginning. Aggregator must be defined by the user. If no aggregator can be found to characterize an aggregated attribute corresponding to a group of DPs, then this group of corresponding DPs attributes must be removed from the *RS*.

UM5. **The choice of instances in the KG.** Sometimes the user wants to be able to study only a particular part of the KG (e.g. students that are registered in at least one course). This UM allows some conditions to be defined in order to select the instances that are consistent with the building of the *RS*: if we have a relational slot from the University *RS* class to the Student *RS* class, then all student instances in the KG must be registered in an university.

Once the user has done all the modifications he deemed necessary, we can learn the probabilistic model using the *RS*.

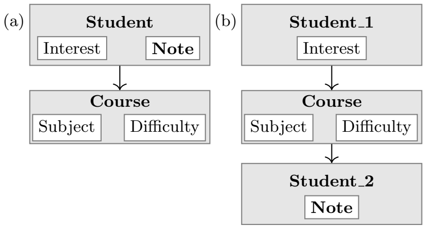


Fig. 3. Example of a *RS* class splitting in a PRM as detailed in **UM3**. The Student *RS* class of (a) has been split in two *RS* classes Student_1 and Student_2 in (b).

3.3 Model Validation

The *RS* has been defined using constraints from both the ontological and the user's knowledge. As a consequence the PRM learned using this *RS* has been learned under causal constraints, and then can be used to deduce causal knowledge. However, the *RS* are not good enough to discover new causal relations. Since it is easier for a user to criticize when confronted to mistakes, we have devised a method to validate the learned model [17].

First, the inter *RS* classes relations are presented to the user. Those relations flow directly from the relational slots defined during the *RS* building: their orientation has been fixed either by the ontology or by the user. They are thus easier to criticize for the user than if they have been built from scratch: if their orientation contradicts a piece of knowledge the user has about the domain, then the *RS* has been badly constructed, and has to be reconsidered. Then, the intra *RS* classes relations are presented. Their orientation is not ruled by the *RS*, so in order to criticize them we need to look at the EG. If this arc is not an essential arc, then it can be reversed without consequences; however, if it is not, then the *RS* has to be modified in order to reflect this change. Finally, if the user challenges a learned relation that should not exist (for instance, between two attributes he knows are independent), then it means that the KG is not balanced enough: for example, scientists that might have tested too much of an hypothesis and not enough of an other. In this case, we cannot continue, as our data is not robust enough to deduce causality.

3.4 Causal Knowledge Discovery

Once the *RS* has been built using the ontological and user's knowledge (Sec. 3.2) and the learned model validated by the user (Sec. 3.3), we can use it to discover causal knowledge. Causal knowledge can be validated by three means:

- **the Ontology**: the orientation of a learned relation between attributes from two different *RS* classes defined by the ontology (e.g. between a student and his/her university) has been constrained by its causal information.
- **the User**: During the *RS*'s interactive building, the user was able to inject causal knowledge with UMs. If a relation is learned between two attributes from two *RS* classes (or whose relational slot has been) defined by the user (e.g. between the classes Student_1 and Course in Fig. 3), then the learning has been constraint by the user who validates the causal knowledge discovery.
- **the EG**: Since the model has been learned under causal constraints given by the ontology and the user, the EG's essential arcs can give causal information. Indeed, an oriented arc in the EG is oriented for all of Markov's equivalence's classes of the learned BN, meaning that, if our model has been learned under right conditions (i.e. complete data set, good given constraints), then it is highly probably causal, allowing the discovery of causal knowledge between attributes of a same *RS* class (e.g. a student's Interest and his/her Note).

The interest of this discovery has two goals: first, it can help a user validate his/her hypothesis on a domain; second, it can suggest new experiments to conduct to test new hypothesis. For instance, using this method, [18] suggests a strong link between plausible control variables and some parameters of the studied cheese, whereas it also indicates that some other experiments had to be conducted to understand the whole process.

4 Application on DBpedia

We illustrate our method with a part of the DBpedia[4] KG dedicated to writers.

4.1 Dataset Presentation

The DBpedia database collects and organize all available information from the Wikipedia[5] encyclopedia. Since it describes 4.58 million things (including persons, places, ...), we have decided for our test to only study a small part of it, on a subject simple enough where we could easily play the role of an expert. As a consequence, we have restrained our study to a much smaller KG[6], dedicated to writers. During this first pre-selection, we have selected four classes to represent our domain: Writer, University, Country and Book. The selected KG is presented in Fig. 4. Considering all possible DPs for every instances of these classes, and also all OPs between them, we have a dataset of 2,966,073 triples.

[4] https://wiki.dbpedia.org/

[5] https://www.wikipedia.org/

[6] https://bit.ly/2X0eeCw

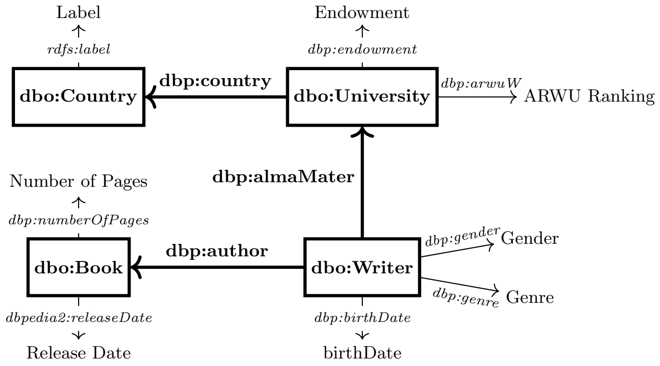


Fig. 4. Schema of the used excerpt of DBPedia with the DPs kept in the final *RS*.

4.2 Interactive Construction of the *RS*

First, we translate all the selected classes as new *RS*'s classes, and all DPs as new *RS*'s attributes. In our case, there is no symmetric OPs, so we keep the original ones present in DBpedia (as depicted in Fig. 4) to define the direction of the relational slots. By applying the selection rule **SR3**, a first automatic selection removes all attributes that correspond to DPs that are not represented enough: for instance, over the 32,511 instances of writers, only 12,188 have the DP *occupation*. This selection is coupled with the expert selection using **UM1** which removes attributes that correspond to uninteresting DPs. We also apply **UM5**, which filters some instances: for example, in our case, we want to study writers that have written books. However, on the whole database, only 6,028 writers instances are linked to at least one book instance. As a consequence, we remove authors with no books since they are out of the scope of our study. Then as a user we apply **UM2**. Since we consider that a country can explains the values of an university's variables, and not the contrary: we reverse the relational slot corresponding to the OP *dbp:country*. One country can have multiple universities, but one university can only have one country: reversing the relational slot removes the aggregation of universities and creates a simple linear relation, since now one university can be explained by at most one country. Moreover, we want to study the possible influence of an university over a writer's work, so we need to reverse the relational slot corresponding to the OP *dbp:almaMatter*. Since a person can register in one or more universities, then his/her attributes can be explained by a combination of his/her universities'. We apply **UM4** and create an aggregation from universities to writers. For each writer, we create two aggregated variables: the highest rank and the highest endowment among all of the universities he/she went to. But doing so break the relation between the Country *RS* class and the Writer *RS* class, since they are linked trough the University *RS* class. The only way to keep a relational slot between the country and the writer is to also aggregate the country's attributes. However, the only available country's attribute is the label, and there is no way of intelligently aggregating it. As a consequence, with the aggregation of universities, we loose the information about countries for the writers and their books. In the end thanks to the rule **SR3**, only interesting attributes which have no missing values and are easily discretizable are kept. For each class, we keep the following attributes:

- **dbo:Country**: each country is only represented by its **label**. Since the majority of our writers are Anglo-Saxon, we distinguish five categories: USA, Canada, Great Britain, Europa and Asia.
- **dbo:University**: each university is represented by its **Academic Ranking of World Universities** (ARWU), and its **endowment**. The endowment is split by its median value. The ARWU ranking is split between the first hundred universities, and the rest.
- **dbo:Writer**: each writer is represented by his/her **gender**, his/her **genre** and his/her **birth date**. Genders are split between male and female, while genres are split between fiction and non-fiction. Birth dates are separated by their median, 1950. Two aggregated attributes have been also added: the **highest rank** among all universities he/she went to, and the **highest endowment** he/she went to, with the same discretization used before.
- **dbo:Book**: each book is represented by its **number of pages** and its **release date**. The number of pages is split between books with 250 pages or less and the others; the release date attribute is split between books published before 1980 and those published after.

In the end we have drastically dropped the number of instances to 6,908 triples and 185 writers. The final *RS* defined both by ontological and user's knowledge is presented in Fig. 5. The direction of relational slots indicates how the considered variables can influence each other: for instance, a writer's genre or highest university rank can influence the number of pages of his/her books.

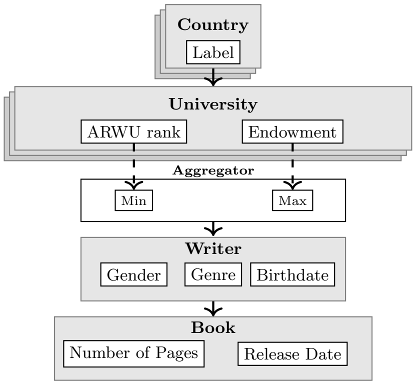
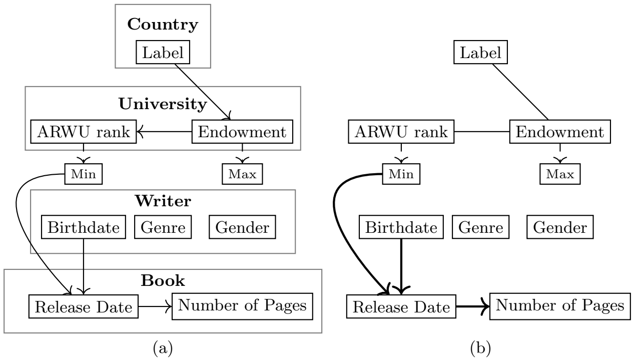


Fig. 5. Relation Schema defined from ontological and user's knowledge. Since a writer can have multiple universities, we introduced an aggregation between the two classes.

4.3 Results

Using the dataset and the RS, it is now possible to learn a PRM and study its EG (respectfully Fig. 6 (a) and (b)). We apply the discretization presented in Sec.4.2, and consider any missing data as a new category "Unknown".

- **Inter RS classes relations.** We have three inter RS classes relations: one between Label and Endowment, one between the highest ARWU rank and the book's release date, and another one between the author's birth date and the book's release date. Since the RS classes was built from the ontology, and the relational slot's direction decided by the user, then we have a causal discovery validated by both the ontological and user's knowledge.
- **Intra RS class relations.** Three relations are oriented in the EG (see Fig. 1 (b)), but only one is an intra RS class relation: from Release Date toward Number of Pages. Thus, the causality of this relation is validated by the EG. There is another intra RS class relation (between ARWU Rank and the Endowment), but it is not oriented in the EG: the given RS and dataset are not enough to assume the causality between those two attributes.



(a)

(b)

Fig. 6. (a) PRM learned. Plain arrows indicates probabilistic relations. (b) Associated EG. Plain arrows indicates essential arcs, unoriented ones indicate the edges. Dashed arrows only serve as a visual cue to indicate aggregation.

4.4 Discussion

Despite not being experts of the domain, most of our results appears to agree with common sense. For instance, it seems logical that an university's ARWU rank and its endowment are correlated, itself explained by its university's country. However our KG's representativeness casts doubts on other results. For instance, we find that a book's release date can be explained by both the highest rank of the university its author went to, and this author's birth date (the joint probability is presented in Table. 1). Basically, authors born before 1950 tend to publish more before 1980 when they are from a top-tiers school. On an other hand, youngest authors tend to publish after 1980, which at first seems logical: writers born after 1980 would hardly be able to publish books prior to their birth. However, we have no instance in our dataset of books published before 1980 written by persons born after 1950, which explain why we learned this relation. This underlines the importance of a complete and verified KG: if our dataset is representative, then we acknowledge the fact youngest authors cannot publish before 1980. On another hand, if our dataset is not representative, then it means that our learned relation cannot be causal, as we are missing arguments. In the end, the main point of this example is to illustrate our method:

1. The *RS* construction from the KG is simplified thanks to selection rules that preemptively remove *RS* classes, attributes... that are not learnable. In our case, numerous attributes corresponding to DPs with not enough instantiations were removed (such as *dbp:occupation* for the writer).
2. The user introduced causal knowledge in the *RS* with UMs: **UM1** to remove attributes irrelevant for the problem (e.g. the wikipedia page ID), **UM2** to reverse relational slot to express causality (e.g. between a writer and his/her universities), **UM4** to formulate aggregations (e.g. since writers had a variable number of universities, we had to aggregate the universities' attributes), and **UM5** to remove instances that did not have certain properties (e.g. all writers with no book or no birth date).

writer.birthDate	**writer.min_arwu**	**releaseDate**	
		before_1980	*after_1980*
before_1950	*100_or_less*	0.58	0.42
after_1950	*100_or_less*	0.01	0.99
before_1950	*101_or_more*	0.44	0.56
after_1950	*101_or_more*	0.01	0.99

Table 1. Joint probability of the attribute **releaseDate** depending on the attributes **writer.birthDate** and **writer.min_arwu**. The low values 0.01 are an artefact of learning, and indicates that these combinations are not encountered in the dataset.

UM3 was not used here. However, should we have had a variable about an author's success, it would then have been possible to study the impact of an author's books on his/her success. To do so, we would have split the author *RS* class in two, to see how an aggregation of the books' attributes would have influenced this variable. Fig.7 presents the corresponding *RS*: we can see that since it is the same *RS* class split in two, both the writer's other attributes (genre, gender, birth date) and the aggregated book attributes (mean number of pages, oldest release date) can explain the writer's success.

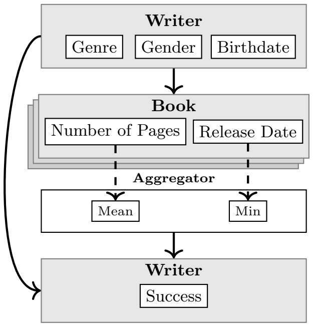


Fig. 7. Example of a *RS* class split with creation of an aggregation.

5 Conclusion

While causal knowledge can be useful for explaining a domain, causal discovery is a hard task, especially from data alone. In this paper, we present an interactive method aiming to allow a user to combine his/her knowledge with that of a KG in order to learn a probabilistic model from a KG able to help him/her uncover new causal explanations. The main idea is to combine the knowledge of both of these sources in order to interactively build a *RS* able to guide and causally constraint the learning of a PRM. This method is split into three parts: (1) automatic design of a first *RS* from the KG; (2) modification of this *RS* by the user; (3) learning of the PRM using the *RS*. This method is **interactive** (i.e. the user can interact with the algorithm to give his/her inputs and influence the learning) and **generic** (i.e. it can be applied on any KG as long as it is relevant for causal discovery). It is also dependant on the quality of the dataset: it has to be checked (i.e. no errors) and complete (i.e. no missing attributes or incomplete data). Our future work will focus on the explanation of the discovered causal relations in order to help the user to improve his/her knowledge (e.g. by enriching the ontology) and clarify his/her reasoning needs.

References

1. Aguilar, J., Torres, J., Aguilar, K.: Autonomie decision making based on bayesian networks and ontologies. pp. 3825–3832 (07 2016)
2. Amirkhani, H., Rahmati, M., Lucas, P.J.F., Hommersom, A.: Exploiting experts knowledge for structure learning of bayesian networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(11), 2154–2170 (Nov 2017)
3. Ben Ishak, M., Leray, P., Ben Amor, N.: Ontology-based generation of object oriented bayesian networks. vol. 818, pp. 9–17 (01 2011)
4. Ben Messaoud, M., Leray, P., Ben Amor, N.: Integrating ontological knowledge for iterative causal discovery and visualization. In: Sossai, C., Chemello, G. (eds.) Symbolic and Quantitative Approaches to Reasoning with Uncertainty. pp. 168–179 (2009)

5. Besnard, P., Cordier, M., Moinard, Y.: Arguments using ontological and causal knowledge. In: Foundations of Information and Knowledge Systems - 8th International Symposium, FoIKS 2014, Bordeaux, France, March 3-7, 2014. Proceedings. pp. 79–96 (2014)
6. Chickering, D.M.: Optimal structure identification with greedy search. J. Mach. Learn. Res. 3, 507–554 (Mar 2003)
7. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine Learning 9(4), 309–347 (Oct 1992)
8. Ćutić, D., Gini, G.: Creating causal representations from ontologies and bayesian networks (2014)
9. De Campos, C.P., Ji, Q.: Improving bayesian network parameter learning using constraints. In: 2008 19th International Conference on Pattern Recognition. pp. 1–4 (Dec 2008)
10. De Campos, C., Zhi, Z., Ji, Q.: Structure learning of bayesian networks using constraints. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 113–120. ICML '09, ACM, New York, USA (2009)
11. Ding, Z., Peng, Y., Pan, R.: BayesOWL: Uncertainty Modeling in Semantic Web Ontologies, pp. 3–29. Springer Berlin Heidelberg, Berlin, Heidelberg (2006)
12. Ehrlinger, L., W, W.: Towards a definition of knowledge graphs (09 2016)
13. Fenz, S.: Exploiting experts knowledge for structure learning of bayesian networks. Data & Knowledge Engineering 73, 73 – 88 (2012)
14. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages. pp. 1300–1309 (1999)
15. Hauser, A., Bühlmann, P.: Two optimal strategies for active learning of causal models from interventional data. Int. J. Approx. Reasoning 55, 926–939 (2014)
16. Madigan, D., Andersson, S.A., Perlman, M.D., Volinsky, C.T.: Bayesian model averaging and model selection for markov equivalence classes of acyclic digraphs. Communications in Statistics–Theory and Methods 25(11), 2493–2519 (1996)
17. Munch, M., Dibie, J., Wuillemin, P., Manfredotti, C.E.: Towards interactive causal relation discovery driven by an ontology. In: Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019. [17], pp. 504–508
18. Munch, M., Wuillemin, P., Dibie, J., Manfredotti, C.E., Allard, T., Buchin, S., Guichard, E.: Identifying control parameters in cheese fabrication process using precedence constraints. In: Discovery Science - 21st International Conference, DS 2018, Limassol, Cyprus, October 29-31, 2018, Proceedings. pp. 421–434 (2018)
19. Munch, M., Wuillemin, P.H., Manfredotti, C., Dibie, J., Dervaux, S.: Learning probabilistic relational models using an ontology of transformation processes. In: On the Move to Meaningful Internet Systems. OTM 2017 Conferences. pp. 198–215 (2017)
20. Parviainen, P., Koivisto, M.: Finding optimal bayesian networks using precedence constraints. Journal of Machine Learning Research 14, 1387–1415 (2013)
21. Pearl, J.: Causality: Models, Reasoning and Inference. Cambridge University Press, New York, USA, 2nd edn. (2009)
22. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. MIT press, 2nd edn. (2000)
23. Verny, L., Sella, N., Affeldt, S., Singh, P.P., Isambert, H.: Learning causal networks with latent variables from multivariate information in genomic data. PLOS Computational Biology 13(10), e1005662 (2017)