Learning Probabilistic Relational Models using an Ontology of Transformation Processes

Mélanie Munch, Pierre-Henri Wuillemin, Cristina Manfredotti, Juliette Dibie-Barthelemy, Stephane S. Dervaux

Learning Probabilistic Relational Models using an Ontology of Transformation Processes

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

[1] UMR MIA-Paris, AgroParisTech, INRA, Universite Paris-Saclay, 75005 Paris, France

[2] Sorbonne Universites, UPMC, Univ Paris 06, CNRS UMR 75005, LIP6, Paris, France

Abstract. Probabilistic Relational Models (PRMs) extend Bayesian networks (BNs) with the notion of class of relational databases. Because of their richness, learning them is a difficult task. In this paper, we propose a method that learns a PRM from data using the semantic knowledge of an ontology describing these data in order to make the learning easier. To present our approach, we describe an implementation based on an ontology of transformation processes and compare its performance to that of a method that learns a PRM directly from data. We show that, even with small datasets, our approach of learning a PRM using an ontology is more efficient.

Keywords: Probabilistic Relational Model, Ontology, Learning

1 Introduction

Probabilistic Relational Models (PRMs) extend Bayesian networks (BNs) with the notion of class of relational databases. Thanks to the addition of the oriented-object concepts (e.g. class, instantiation, reference) they offer a new expressivity to BNs: they provide a qualitative description of the structure of complex domains while representing the quantitative information provided by the probability distribution. However, because of this richness, learning PRMs from data is a difficult task. This is due, on the one hand, to the learning of both the **high level structure** (i.e. classes and relations between them) and the **low level structure** (i.e. attributes and their probabilistic dependences); that leads us to deal with a two layers learning problem. On the other hand, their expressivity allows the modelization of systems with a small amount of data which increases the complexity of the learning task. These difficulties explain the complexity of determining the best structure among all the possible ones.

Ontologies are nowadays used as a common and standardized vocabulary for representing a domain (e.g. in life-science, geography). They organize and structure the knowledge in terms of concepts, relations between these concepts and instances of these concepts [13]. The aim of this paper is to show that we can use the knowledge represented by an ontology to map the high level structure

of PRMs easing, in this way, the learning of their probability distribution. We choose to use ontologies as opposed, for example, to relational databases, because in the future we are interested in modeling non-stationary domains and the structure of an ontology is more adaptable to changes in the domain than that of a relational database.

We present, in this paper, our approach of learning a PRM using an ontology. We propose to use the knowledge of an ontology, first, to define the high level structure (i.e. the **relational schema**) of a PRM and, then, to learn this PRM from data. Using ontology helps us by integrating the experts' knowledge to ease the learning in complex domains.

To illustrate our approach of learning a PRM using an ontology, we propose to use an ontology of transformation processes where a transformation process can be represented as a sequence of operations, receiving different inputs and designed to obtain a specific output. Such an ontology allows the representation of the knowledge of a complex domain with several interesting characteristics:

- it is **complex**, multiple operations can occur at the same time and are linked together; inputs and outputs are characterized at multiple scales (i.e. environment, population, cellular and molecular) and studied with different types of measurement (e.g. physiological, biochemical, genetic);
- data is **scarce**, due to the difficulty to obtain results, this imposes to gather information from various sources;
- it presents problems of **missing data** (e.g. a parameter is not controlled) and **missing values** (e.g. the process' instructions are not precise);
- even with complete information, it is still characterized by **uncertainty**, instruments used to take measurements during a transformation process are able to return only an estimation of the quantity observed because their calibration cannot be entirely defined and repeated from an experiment to another and some internal and uncontrollable parameters (from both devices or outside the experiment) can influence the final result.

This paper is organized as follows. In Sect. 2 we present the ontology of transformation processes used, PRMs and their existing learning methods. In Sect. 3 we describe our approach of learning a PRM using an ontology. In Sect. 4 we present preliminary results where the efficiency of our approach is evaluated through a comparison of its performance to that of a method that learns probabilistic models without ontology. We conclude in Sect. 5.

2 Backgrounds

2.1 The ontology of transformation processes

To illustrate our approach, we propose to use the Process and Observation Ontology (PO^2) [9], written in OWL 2, designed to represent transformation processes. A transformation process is denoted as a sequence of steps (i.e. operations), receiving different participants (i.e. inputs) and designed to obtain a specific product (i.e. output).

An ontology is a representation of the knowledge of a domain and is composed of two main components: the conceptual component where the concepts, relations between these concepts and axioms are defined and the instance component which contains the facts. The conceptual component of PO^2 contains the following three main parts (see Fig. 1):

- **Step part**: contains the concepts *step*, *itinerary* and *process*
- **Participant part**: contains the concepts *method*, *mixture* and *device*
- **Observation part**: contains the concepts *observation*, *scale*, *measure*, *sensor output* and *computed observation*

In this ontology, a *process* is a whole operation: processes that are the same share the same goal. A variation in one *process* is called *itinerary*. An *itinerary* is defined as a succession of different *steps* linked to each other: each *step* is associated to the one(s) following it according to a chronological order. A *step* is defined both by its duration and its participants, that can be a *method*, a *mixture* or a *material*. Participants are characterized by inner attributes defined by experimental conditions; moreover, a *mixture* is composed of different *products* that represent its composition. Finally, during each *step*, one or more *observations* can take place to make measurements of one participant: they are made using specific participants (independently of the other step's participants), and at a specific *scale*. They have for result a *sensor output* and/or a *computed observation*, each of them can have for value a function or a simple measure. A *measure* is characterized by either a quantity and a unit of measure or a symbolic concept and a measurement scale.

Each *step* is defined as a concept to which a set of descriptor concepts is linked: participants (i.e. devices, mixtures and methods) are concepts whose parameters are set *a priori*; observations are concepts whose parameters are measured during the step. Therefore there exists for each step a compartmentalization between the different domain's objects. Moreover, the *time relation* linking steps gives information about their relative time (inside the process and with other steps). The instance component of PO^2 allows one to represent different transformation processes by a succession of instances of steps and instances of their associated descriptors.

We introduce an example of a domain ontology about the micro-organisms stabilization transformation process denoted by PO^2_{stab}. Fig. 2(a) gives an excerpt of the simplified conceptual component of PO^2_{stab} where there are 3 steps: *Fermentation*, *Culture* and *Stabilization* which are sub-concepts of the concepts *Step* and 2 attributes: *SugarQuantity* and *Temperature* which are sub-concepts of the concept *Attribute*. Fig. 2(b) gives an excerpt of the simplified instance component of PO^2_{stab}. In this example, there are three instances of steps linked by a linear temporal dependency *Fermentation_1* that is before *Culture_1* that is before *Stabilization_1*. The instance *Fermentation_1* of the concept *Fermentation* has for participant *Mixture_1* (an instance of the concept *Mixture*) which has for sugar quantity (the instance *SugarQuantity_1* of the concept *Attribute*) the value: $2g$. Moreover, an observation (the instance *Observation_1* of the concept

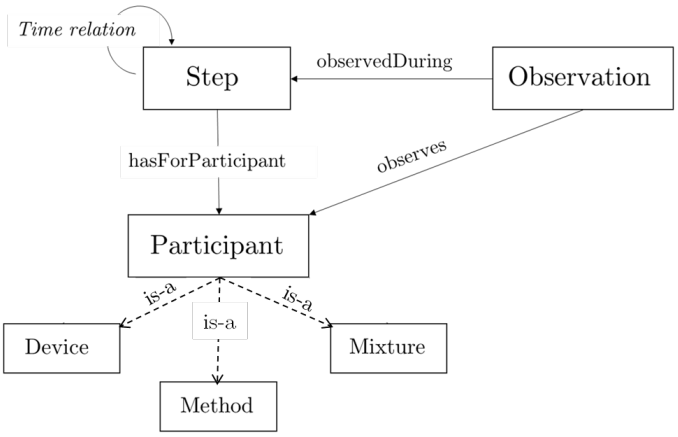


Fig. 1. Simplified schema of the conceptual component of PO^2. The ontology is divided in three main parts: **Step**, **Participant** and **Observation**. These parts interact to each other through semantic relations.

Observation) was made on the temperature (the instance *Temperature_1* of the concept *Attribute*) of *Mixture_1* which has for value: 5 celsius degree.

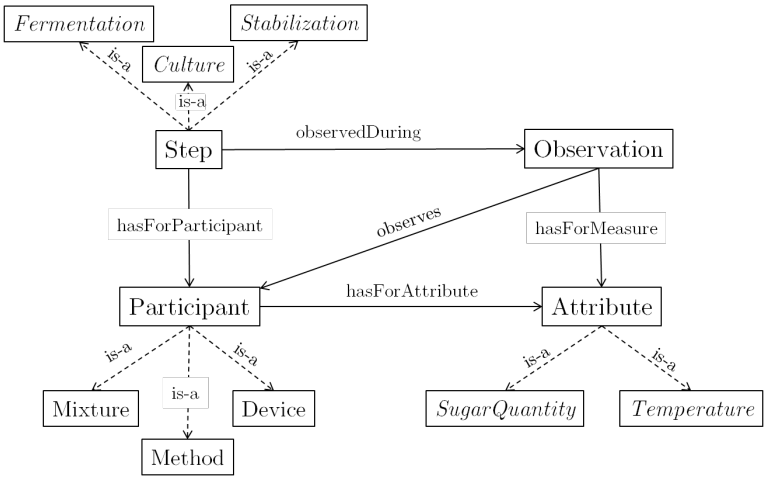
2.2 Probabilistic Relational Models

Probabilistic Relational Models (PRMs) extend Bayesian networks (BNs) with the notion of class of relational databases. 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 (see Fig. 3(a)). However, in the case of numerous random variables with repetitive patterns (for instance different steps in the same transformation process), it cannot efficiently represent every probabilistic link.

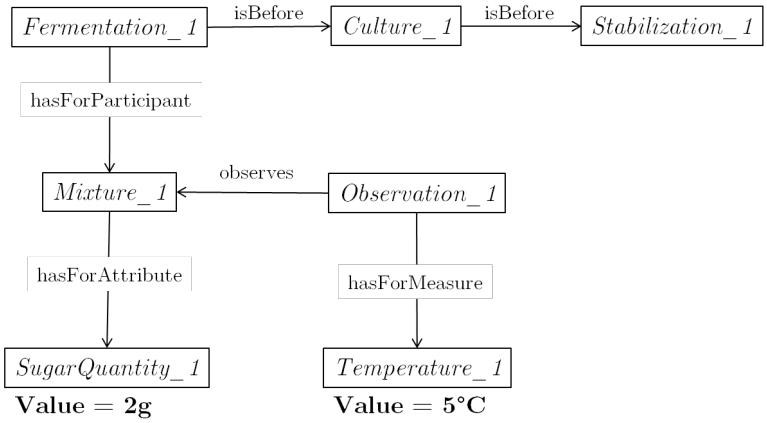
PRMs extend the BN representation with a relational structure between potentially repeated fragments of BN called classes [15]. A **class** is defined as a DAG over a set of attributes. These attributes can be inner attributes or attributes from other classes referenced by so-called **reference slots**. The analysis of the BNs in Fig. 3(a) reveals two recurrent patterns, that can be translated into two interconnected classes $\mathscr{E}$ and $\mathscr{F}$, as presented in Fig. 3(b).

The high level structure of a PRM (i.e. its **relational schema**, see Fig. 3(b)) describes a set of classes C, associated with attributes $A(C)$ and reference slots $R(C)$. A slot chain is defined as a sequence of reference slots that allows one to put in relation attributes of objects that are indirectly related.

The probabilistic models are defined on the low level structure (i.e. at the class level) over the set of inner attributes, conditionally to the set of outer attributes and represent generic probabilistic relations inside the classes. This is the **relational model** of the PRM (see Fig. 3(c)).



(a) Excerpt of the simplified conceptual component of PO^2_{stab}



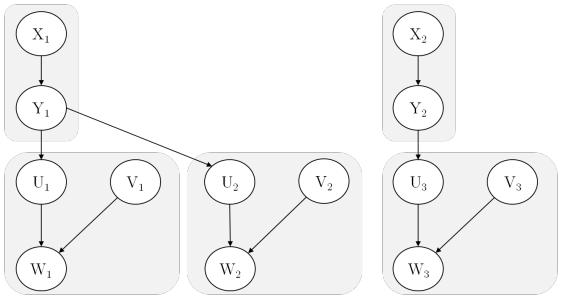
(b) Excerpt of the simplified instance component of PO^2_{stab}

Fig. 2. An example of a domain ontology about the micro-organisms stabilization transformation process: PO^2_{stab}

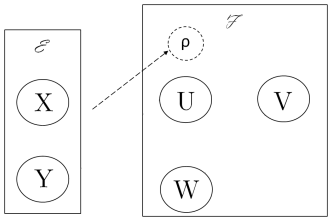
Classes can be **instantiated** for each specific situation (see Fig. 3 (d)). A system in a PRM provides a probability distribution over a set of instances of a relational schema [16]. PRMs define the high-level, qualitative description of the structure of the domain and the quantitative information given by the probability distribution [5].

2.3 Learning PRMs

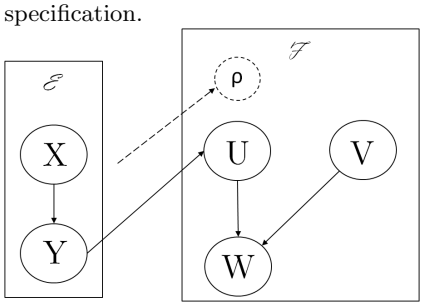
PRM learning is composed of two different parts: **structure selection** and **parameter estimation**. Structure selection can be decomposed in two layers: a



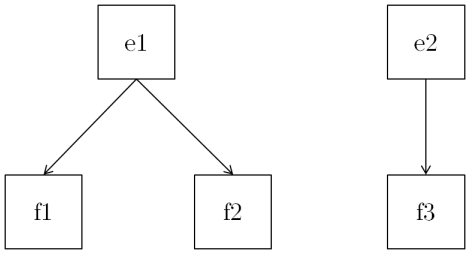
(a) An example of two BNs. The gray areas represent the repetitive patterns, but are not part of the BN specification.



(b) The **relational schema** of the PRM. It is composed of two connected classes $\mathscr{E}$ and $\mathscr{F}$.



(c) The PRM **relational model**. Relational links between attributes were added to the relational schema in (b).



(d) A system for the PRM in (c). Instantiation of the classes of the PRM representing the BN in (a).

Fig. 3. BNs and PRMs: the analysis of the BN in (a) reveals two recurrent patterns, that can be translated into two interconnected classes $\mathscr{E}$ and $\mathscr{F}$ of a PRM (b) and (c). An equivalent system can, thus, be constructed through the instantiation of twice the class $\mathscr{E}$ and three times the class $\mathscr{F}$ (d).

high level layer that organizes the knowledge under an entity-relation pattern (using classes and references); and a lower level layer that employs a graphical language to represent the probability distribution in a compact way by exploiting the probabilistic dependencies between the attributes. The **relational schema** is learned with the high level layer while the **relational model** is the final result of structure selection. Due to these multiple layers, the number of free parameters is high, and the target model is not unique: selecting one requires making subjective choices. Moreover, the richness of this tool allows us to represent new and complex systems where data can be scarce or incomplete. This can be another obstacle while learning PRMs, for example, in life science.

In [5] an algorithm based on an **heuristic search**, such as a greedy algorithm, is proposed to select the legal structure (i.e. a structure representing a coherent probability model) with the highest score. The score proposed has a decomposability property that helps to analyze small parts of structures, easing the search. Other score-based approaches have been equally proposed based on a relational extension [6].

On the contrary of heuristic search, **dependency analysis** tries to discover dependency relations from the data itself and then attempts to learn the structure. This constraint guided approach was exploited in [10] that extends to the relational context, or in [4] that proposes an exact approach to learn PRMs.

In this paper we propose to learn a PRM starting from its relational schema. This relational schema can be deduced from a relational database, however ontologies can also be used to define it. In fact, the notions of class in PRMs and of concepts in ontologies are very similar. We therefore propose to deduce the relational schema of a PRM from the concepts' structuration defined in the ontology's conceptual component.

The use of ontology has already been proposed for learning BNs [3] [8] and Object Oriented Bayesian Networks [1]. An approach to define a relational schema from an ontology has been proposed in [11], but the task of learning PRMs using an ontology has not been addressed yet.

Indeed, once the structure of the relational schema is known, learning the relational model of a PRM can be compared to selecting the structure of a BN [6]. The main difference is that probabilistic dependences between attributes in the same class are forced to be identical: the PRM relational schema and the ontology's semantic knowledge give us patterns on which to learn.

3 Learning a PRM using an ontology

We present, in this section, our approach to learn PRMs using ontologies. We first present the relational schema mapping from the PO^2's conceptual component and then our **ON2PRM** algorithm.

3.1 Relational Schema mapping

We briefly present our relational schema mapping from the PO^2's conceptual component that relies on the one proposed in [11]. Our mapping was motivated both by the description of transformation processes in PO^2 and the definition of *state* as explained in the theory of control and expert systems.

In the theory of control, a system can be described as a succession of *states* through time [14]. A state contains a set of every attributes that enables to describe the system. Observations can be made to evaluate these attributes: however, the act of observing is independent of the state itself. These definitions and the semantic representation of transformation processes defined in PO^2 allow us to define the following temporal dependences properties:

- **Observations can be longer in time than the states they are observing.** For instance, some measurement methods in biology are based on time dependent reactions; in this case, the result of observations can be physically obtained even if the step linked to these has ended before;
- **States influence the result of observations, but observations do not influence states' values.** From this property, we can deduce that observations cannot influence other observations.

In the relational schema, we therefore propose to define two classes built from the ontology's concepts defined in its conceptual component:

- The **Participant Class**, $\mathscr{P}$. It groups every *a priori* attributes: the attributes of the participant concepts mixtures, devices and methods (Fig. 1).
- The **Observation Class**, $\mathscr{O}$. It groups every measured attributes.

At each time step t, we instantiate these two classes: $\mathscr{P}_t$ and $\mathscr{O}_t$. We call *Step*, denoted by $\mathscr{S}_t$, the couple $\mathscr{P}_t$ and $\mathscr{O}_t$.

The temporal dependences properties introduced above can be formalized between the two classes $\mathscr{P}_t$ and $\mathscr{O}_t$ as the following **temporal dependences constraints**: $\mathscr{P}_t$ can have none or multiple $\mathscr{P}$ parents at time *t-1* (that we call altogether $\mathscr{P}_{t-1}$), but always maximum one child at time *t+1* ($\mathscr{P}_{t+1}$). $\mathscr{O}_t$ only depends on $\mathscr{P}_t$. To each $\mathscr{P}$ class an $\mathscr{O}$ class is linked. Through slot chain, each $\mathscr{P}_T$ class has access to every attributes of $\mathscr{P}_t$ with t<T, and each $\mathscr{O}_t$ has only access to the attributes of $\mathscr{P}_t$.

The relational schema mapped from the PO^2 ontology is represented in Fig. 4: the arrows represent the reference slots; given two classes $\mathscr{P}_t$ and $\mathscr{P}_{t-1}$, $\mathscr{P}_{t-1}$ o$\rightarrow$ $\mathscr{P}_t$ means that $\mathscr{P}_t$'attributes can depend on $\mathscr{P}_{t-1}$'s, attributes of $\mathscr{P}_{t-1}$ can be parents of attributes of $\mathscr{P}_t$. However, according to the temporal dependences constraints, attributes of $\mathscr{O}_{t-1}$ cannot be parents of attributes of $\mathscr{O}_t$.

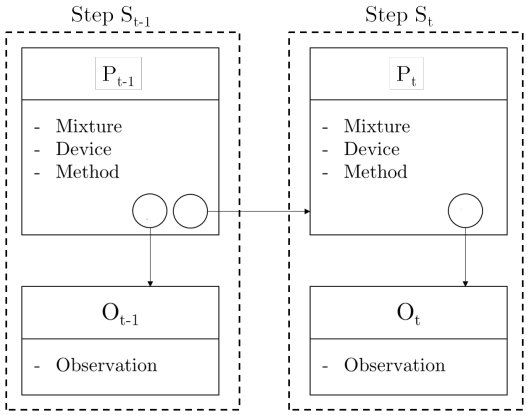


Fig. 4. Relational Schema mapped from the PO^2 ontology for two steps.

This relational schema has two interesting properties we use in the learning. First, it preserves the **compartmentalization** between the different steps and between participants in the process and observations about the process. In the relational schema the attributes of an observation class only depend on the attributes of the participant class it is associated with. This will allow us to consider, while learning the relational model, only meaningful attributes, which are defined in the conceptual component of the ontology. In example of Fig. 2,

we can deduce from the instance component of PO^2_{stab} that the *SugarQuantity* is an attribute of the mixture in the Participant class and the *Temperature* is an attribute of the Observation class. Moreover, we can deduce that these two attributes are specific to the Fermentation step.

Second, it preserves the integrity of the steps through time: a choice made at time t (i.e. the value of an attribute of $\mathscr{P}_t$) cannot influence an observation at time *t-1*. This leads us to define the **direction learning constraint** used in the learning presented below: if attributes are dependent in the instance component of PO^2, the learnt links between them can only have one direction. In example of Fig. 2, we can deduce from the instance component of PO^2_{stab} that the sugar quantity, an attribute of the mixture, can have an influence on the temperature, an observation attribute of the mixture. Moreover, considering that the fermentation step is before the culture step, the sugar quantity can also have an influence on the values of the attributes associated with the culture step.

In the next subsection, we present our algorithm for learning PRMs' relational models using its relational schema and the ontology PO^2.

3.2 Our ON2PRM algorithm

The learning approach we propose for learning a PRM given its relational schema is very similar to a classic approach for learning BNs. However, we propose to use the semantic knowledge of the ontology: the concepts' structure defined in its conceptual component and the links between concepts defined in its instance component, as presented in the mapping defined above.

Let us consider a database D about a transformation process, where each attribute using concepts defined in the conceptual component of the ontology is represented (e.g. fermentation is a step according to the concepts' hierarchy of PO^2_{stab} as presented in Fig. 2). Following the **compartmentalization** property introduced in the relational schema, several sub-databases are created, during the learning from D, each containing the data of one step: only the attributes of this step (i.e. attributes from the $\mathscr{P}_t$ and the $\mathscr{O}_t$ classes) and their parents (i.e. attributes from the $\mathscr{P}_{t-1}$ class) are considered. This ensures that the organization between participant and observation is preserved. Afterwards, using the **direction** learning constraint, we force a learning order over the attributes of the same sub-database. This ensures that the temporal order between steps is preserved. However preserving organization and temporal order does not imply links existence but only that, if they exist, the orientation of the links is defined by the direction and the organization given. In example of Fig. 2, we can deduce from the instance component of PO^2_{stab} that the attribute quantity of sugar is included both in the fermentation $\mathscr{P}_t$ and the culture $\mathscr{P}_{t-1}$, while the temperature is only included in the fermentation $\mathscr{O}_t$ class.

We call ON2PRM(M) our algorithm of learning a PRM's relational model from an ontology where M is a learning method for Bayesian Networks that can be used to draw probabilistic dependencies between attributes from a database. For each step (e.g. the steps fermentation, culture and stabilization in Fig. 2), the ON2PRM(M) algorithm uses M over the attributes (e.g. the attribute quantity

of sugar in the fermentation $\mathscr{P}_t$ and the culture $\mathscr{P}_{t-1}$ and the attribute temperature in the fermentation $\mathscr{O}_t$ class) following the established learning order to learn a small BN for each identified class of the PRM. Once every class has been learnt, the PRM relational model is defined and can be instantiated.

Input: ontology PO^2 + relational schema + database D + learning method M
Result: a PRM relational model
//the **for** loop is justified by the compartmentalization property of the relational schema
//the identification of the steps relies on the concepts and concepts' hierarchy defined in the conceptual component of PO^2
for *each step at time t* **do**
  //the identification of the attributes relies on the concepts and concepts' hierarchy defined in the conceptual component of PO^2 ;
  identify attributes for $\mathscr{P}_t$;
  identify attributes for $\mathscr{P}_{t-1}$;
  identify attributes for $\mathscr{O}_t$;
  create a **sub-database** from D from the identified attributes;
  //the **learning order** is defined from the instance component of PO^2 as defined in the direction constraint ;
  define the **learning order** ;
  learn a BN of a PRM class from **sub-database** + **learning order** + **method** M;
end
//the PRM relational model is the set of the PRM classes generated above, linked to each other following the PRM relational schema ;
create the PRM relational model ;

Algorithm 1: ON2PRM(M): Learning a PRM using an ontology

As explained in section 2.2, the PRM relational model can be instantiated with data in D providing the system of the PRM. In the following we use the instantiated PRM to compare the performance of our approach to that of a method that learns BNs directly from data. We demonstrate that, thanks to the use of the semantic knowledge represented in an ontology, learning a PRM with an ontology is more efficient than learning without an ontology. We compare the performance of learning with our algorithm ON2PRM(M) to the performance of learning only with the method M.

4 Experiments

In order to validate our approach we propose to compare the performance of learning with and without ontology implementing two learning methods[3]:

- *Greedy Hill Climbing* algorithm with BIC score, denoted by $M1$;
- *Local Search with Tabu List* algorithm with BDeu score, denoted by $M2$.

The proposed experiment consists in comparing the instantiated PRM, learnt with ON2PRM from a database D of transformation processes and PO^2, with a BN learnt from D, using both methods $M1$ and $M2$. All our experiments were implemented using the PyAgrum Python library [7].

In order to have an experiment as generic as possible, we perform our learnings from several randomly generated databases D_i. We first present the generation of our test databases and then our results.

4.1 Databases generation

The databases generation (1) generates PRM relational models representing transformation processes using the PRM relation schema of Fig. 4 and (2) builds the domain ontologies corresponding to the generated PRM relational models. We first present the generation of the PRM relational models, then the construction of the corresponding domain ontologies, finally the databases generation.

The PRM relational models generation One of our motivations to study transformation processes is their complexity (of which we presented the main characteristics in the introduction section. One process cannot therefore encompass alone the entire diversity spectrum of processes. We define five **process complexity degrees criteria** to randomly generate PRM relational models representing transformation processes as much as possible generic :

1. the number s of steps in a process;
2. the maximal number p of parallel steps, representing how many parents a step can have;
3. the number n of attributes in a class;
4. the number m of modalities for the attributes;
5. the number d of probabilistic dependencies an attribute may have.

The higher the process complexity degrees criteria are, the harder to learn the corresponding PRM relational models are. As a matter of fact, during the learning phase of a PRM relational model:

- a high number of steps induces more PRMs' classes to learn;
- a high number of parallel steps, attributes and probabilistic dependencies induces more possible links to draw;

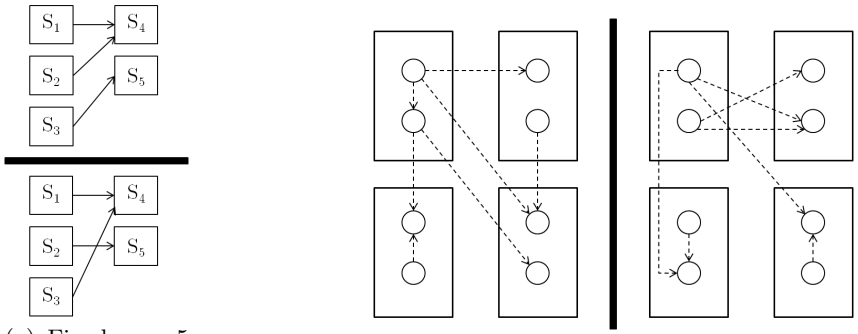
[3] These are two standard well known methods for learning BN. These and others methods can be found in [12]

– a high number of modalities induces a more difficult learning.

In the following, we assume that the process complexity degrees criteria are better addressed by ON2PRM where the ontology semantic knowledge reduces the learning's complexity. Therefore, we argue that if the results of our approach outperforms that of a standard method for simple processes, it will have better results in learning more complex processes. Considering this assumption and to be as close as possible to the modelization of real transformation processes,

we decided to fix two process complexity degrees criteria: $m = 2$ (i.e. binary attributes) and $d = 3$, and to have three criteria that vary: $s \in \{3, 5, 8\}$, $p \in \{1, 2, 3\}$ and $n \in \{2, 4\}$. This leads us to have 16 different configurations of possible processes, not considering the case $s = 3$ and $p = 3$ (i.e. a process composed of only three parallel steps without interaction) because it is not interesting.

With these 16 configurations, we can generate several different PRM relational models because of the possible relations between attributes. For example, links between steps are decided randomly, given s and p (see Fig. 5 (a)); moreover, even inside the same class or sequence of two steps, links between attributes are decided randomly given n (see Fig. 5 (b)). We generate 10 PRM relational models for each configuration, that corresponds to a total of 160 processes.



(a) Fixed $s = 5$ and fixed $p = 3$

(b) Fixed $n = 2$ and fixed $d = 3$

Fig. 5. Possible differences with fixed parameters

These PRM relational models will be on the one hand used to generate the test databases and on the other hand considered as the original models, i.e. the ground truths in the experiments' evaluation.

The domain ontologies generation In parallel to the generation of these PRM relational models, we build several domain ontologies, denoted by $PO^2_{dom_i}$, necessary for our ON2PRM learning algorithm. The domain ontology's generation is done using the same process complexity degrees criteria defined above. The number s of steps and the number n of attributes are used to create the

conceptual component of each domain ontology $PO^2_{dom_i}$; the number p of parallel steps are used to create its instance component. For example, the domain ontology PO^2_{stab} of Fig. 2 has $s = 3$, $n = 2$ and $p = 1$.

Databases generation From each of the 160 PRM relational models, we generate 100 times four databases of different sizes as presented in Fig.6: 100 databases of size 50, 100 databases of size 100, 100 databases of size 150 and 100 databases of size 200. The database size refers to the number of examples in it. We therefore generated $16 * 10 * 400 = 64\ 000$ databases for experiments.

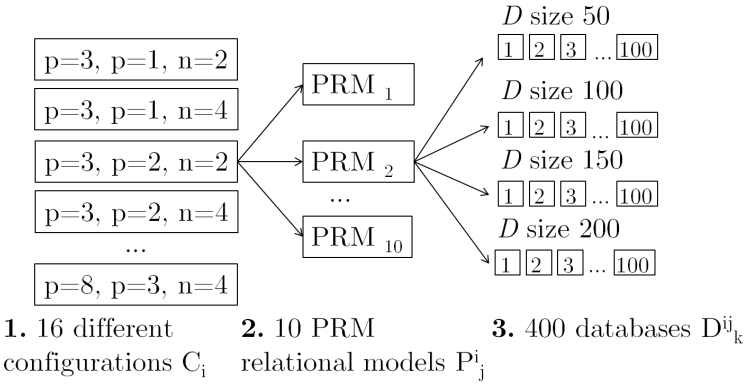


Fig. 6. Databases generation

4.2 Results

We evaluate the performance of ON2PRM in learning the relational model of a PRM by comparing its performance with that of a method that learns probabilistic models without using an ontology.

The learning is performed on the 64 000 databases D_i generated above. We compare the instantiation of the relational models learnt by our algorithm ON2PRM using both learning methods $M1$ and $M2$, denoted by ON2PRM($M1$) and ON2PRM($M2$), with the BNs learnt by $M1$ and $M2$ alone. With the standard approach, the learning is done directly from the database.

The performance of our algorithm is evaluated using structural analysis (i.e. recall, precision and f-score scores) by comparing the structure of the graph learnt to the one of the ground truth. More precisely, let C_i, $i \in [1, 16]$, be one of the 16 possible process configurations, P^i_j, $j \in [1, 10]$, be one of the 10 PRM relational models generated from the configuration C_i and $D^{i,j}_k$, $k \in [1, 4]$, be one of the 400 databases generated from the PRM relational model P^i_j. The model (relational model or BN) learnt from the database $D^{i,j}_k$ with one of the four

different methods ON2PRM($M1$), ON2PRM($M2$), $M1$ and $M2$ is compared with its ground truth P_j^i.

Let us notice that due to the semantic value added, edges orientation is crucial: that is why we consider the presence of arcs as well as their orientation while evaluating the performance.

Three different structural parameters were evaluated: **recall**, **precision** and **F-score** [2]. Recall is used to estimate the number of links found out of the total we have to find. Precision allows the estimation of the proportion of true links among the ones found. F-score is the average of recall and precision. In order to compute these parameters, we have to count the number of true positive and true negative (i.e. right learning), and false positive and false negative (i.e. wrong learning). These are defined following the heuristic reported in Table 1.

Table 1. Heuristic used to compare two BNs. TN: True negative. FN: False negative. TP: True positive. FP: False positive

Learned \ Model	$\emptyset$	$\rightarrow$	$\leftarrow$
$\emptyset$	TN	FN	FN
$\rightarrow$	FP	TP	FN
$\leftarrow$	FP	FN	TP

Precision, recall and F-score are computed with the following equations:

$$\text{Recall} = \frac{TP}{TP + FN} \qquad \text{Precision} = \frac{TP}{TP + FP}$$

$$\text{F-score} = \frac{2 * Recall * Precision}{Recall + Precision}$$

In Table 2 we report the F-score of ON2PRM($M1$) compared with $M1$ and ON2PRM($M2$) compared with $M2$ on a database of size 50. In all cases, results with ontology are significantly better than without. This can be explained by the two properties of the ontology that are preserved by the relational schema (as explained in Sect. 3.1): both compartmentalization and the direction constraint drastically reduce the number of possibilities the method M can consider in the ON2PRM(M) algorithm.

Recall and precision are both as significant as F-score; however depending on the methods, performance varies. Precision tends to be, in fact, better with $M1$, while recall is better with $M2$. Since the difference between recall and precision for $M2$ is smaller than for $M1$, it explains why $M2$ has the best F-score. Table 3 shows performances' comparison between different databases' sizes (50, 100, 150 and 200) and between different process complexities (high-complexity processes (b) and low-complexity processes (a)). This score rises with the augmentation of the size of the database.

Even with few data a difference between the two learning approaches appears. Moreover while raising the size of the database, every score increases. In order to quantify and compare the performance of learning with ontology and without,

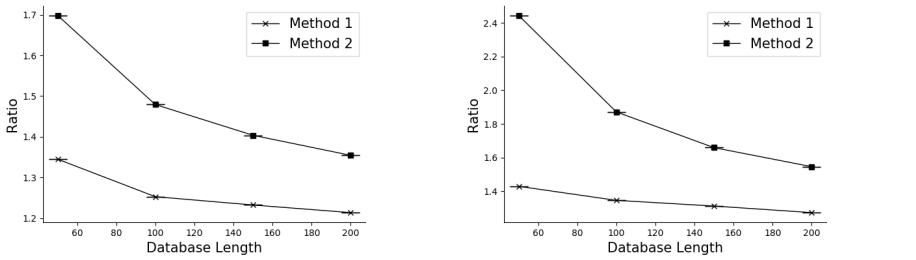
Table 2. Variation of F-score in function of different parameters tested with a database of size 50 with 100 repetitions: (mean [confidence interval 99%]). **bold**: highest value in column, *italic*: lowest value in column. s: number of steps, p: maximal number of parallel steps, n: number of attributes

s	p	n	ON2PRM($M1$)	$M1$	ON2PROM($M2$)	$M2$
3	1	2	**0.40** [0.03]	0.27 [0.03]	0.56 [0.03]	**0.33** [0.03]
		4	0.33 [0.02]	0.25 [0.02]	0.45 [0.02]	0.26 [0.02]
	2	2	*0.24* [0.03]	*0.17* [0.03]	0.43 [0.03]	0.26 [0.03]
		4	0.25 [0.01]	0.20 [0.01]	*0.38* [0.02]	0.24 [0.02]
5	1	2	**0.40** [0.02]	**0.29** [0.02]	0.54 [0.02]	0.27 [0.02]
		4	0.30 [0.01]	0.22 [0.01]	0.43 [0.01]	0.22 [0.01]
	2	2	0.37 [0.02]	0.27 [0.02]	0.54 [0.02]	0.27 [0.02]
		4	0.29 [0.01]	0.21 [0.01]	0.42 [0.01]	0.21 [0.01]
	3	2	0.37 [0.02]	0.28 [0.02]	0.52 [0.02]	0.27 [0.02]
		4	0.28 [0.01]	0.22 [0.01]	0.41 [0.01]	0.21 [0.01]
8	1	2	0.45 [0.01]	**0.29** [0.02]	**0.58** [0.01]	0.25 [0.01]
		4	0.31 [0.01]	0.21 [0.01]	0.43 [0.01]	*0.17* [0.01]
	2	2	0.37 [0.02]	0.25 [0.02]	0.52 [0.02]	0.22 [0.01]
		4	0.31 [0.01]	0.22 [0.01]	0.44 [0.01]	0.18 [0.01]
	3	2	0.34 [0.02]	0.24 [0.02]	0.52 [0.02]	0.22 [0.01]
		4	0.31 [0.01]	0.22 [0.01]	0.43 [0.01]	0.18 [0.01]

we introduce the following ratio of the performances:

$$\text{ratio} = \frac{\text{performance with ON2PRM}}{\text{performance without ON2PRM}}$$

The more the ratio is above 1, the more the learning with the ON2PRM algorithm is efficient. We have used this value to compare the evolution of scores with processes complexity and the different complexity degrees defined (number of step s, number of parent p and number of attribute n).



(a) (b)

Simple process (s = 3, p = 1, n = 5) Complex process (s = 8, p = 3, n = 5)

Fig. 7. Evolution of F-score ratio for two different processes with the database length

Table 3. Comparison of performances for recall, precision and F-score for $M1$ and $M2$ with different sizes of the database (mean [confidence interval 99%]).

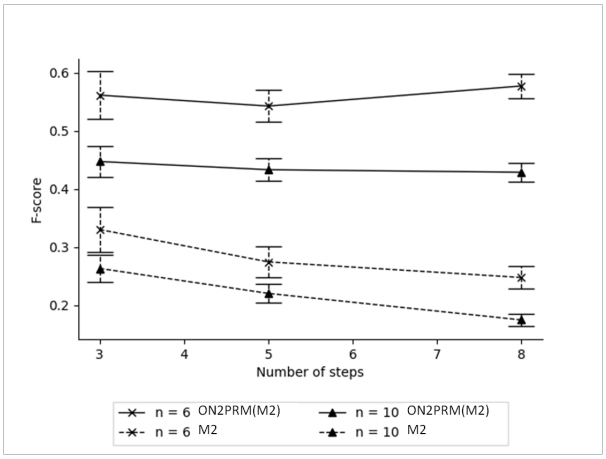
Method	Length	Recall		Precision		Fscore	
		ON2PRM(M)	M	ON2PRM(M)	M	ON2PRM(M)	M
M1	50	**0.26** [0.04]	0.16 [0.03]	**0.95** [0.05]	0.81 [0.09]	**0.4** [0.05]	0.27 [0.04]
	100	**0.39** [0.04]	0.24 [0.04]	**0.97** [0.02]	0.87 [0.07]	**0.54** [0.05]	0.37 [0.05]
	150	**0.47** [0.04]	0.28 [0.04]	**0.97** [0.02]	0.86 [0.06]	**0.62** [0.04]	0.41 [0.05]
	200	**0.51** [0.04]	0.31 [0.04]	**0.97** [0.02]	0.88 [0.06]	**0.66** [0.04]	0.44 [0.05]
M2	50	**0.44** [0.04]	0.27 [0.04]	**0.82** [0.04]	0.46 [0.05]	**0.56** [0.04]	0.33 [0.04]
	100	**0.53** [0.04]	0.33 [0.04]	**0.90** [0.03]	0.61 [0.06]	**0.66** [0.04]	0.42 [0.05]
	150	**0.57** [0.04]	0.38 [0.05]	**0.92** [0.03]	0.69 [0.05]	**0.70** [0.03]	0.48 [0.05]
	200	**0.61** [0.04]	0.4 [0.04]	**0.94** [0.02]	0.72 [0.05]	**0.73** [0.03]	0.50 [0.05]

(a) Parameters of the process: s = 3, p = 1, n = 2

Method	Length	Recall		Precision		Fscore	
		ON2PRM(M)	M	ON2PRM(M)	M	ON2PRM(M)	M
M1	50	**0.19** [0.01]	0.13 [0.01]	**0.91** [0.02]	0.61 [0.03]	**0.31** [0.02]	0.22 [0.01]
	100	**0.29** [0.01]	0.21 [0.01]	**0.93** [0.01]	0.73 [0.03]	**0.44** [0.01]	0.33 [0.02]
	150	**0.36** [0.01]	0.27 [0.01]	**0.94** [0.01]	0.77 [0.02]	**0.52** [0.02]	0.40 [0.02]
	200	**0.42** [0.01]	0.32 [0.02]	**0.94** [0.01]	0.8 [0.02]	**0.58** [0.02]	0.46 [0.02]
M2	50	**0.33** [0.02]	0.19 [0.01]	**0.61** [0.02]	0.16 [0.01]	**0.43** [0.02]	0.18 [0.01]
	100	**0.42** [0.02]	0.26 [0.02]	**0.78** [0.02]	0.32 [0.02]	**0.54** [0.02]	0.29 [0.02]
	150	**0.48** [0.02]	0.32 [0.02]	**0.84** [0.02]	0.44 [0.02]	**0.61** [0.02]	0.37 [0.02]
	200	**0.52** [0.02]	0.36 [0.02]	**0.87** [0.01]	0.52 [0.02]	**0.65** [0.01]	0.42 [0.02]

(b) Parameters of the process: s = 8, p = 3, n = 4

Fig. 7 illustrates the evolution of the ratio for two processes. The ratio is always above 1, but it is also decreasing with the augmentation of the database size. Depending on the methods this decrease can be narrower or wider: while $M1$ stays practically stable $M2$ drops faster. Moreover the ratio varies equally with the complexity for $M2$: ON2PRM efficiency is higher with a complex process.



s	3		5		8	
n	3	5	3	5	3	5
Ratio	1.70		1.95		2.35	2.45

Fig. 8. Evolution of F-score in function of n (p = 1, D size = 50) and ratio evolution in function of n and s for M2.

Fig. 8 shows correlation between the number of attributes and the number of steps: the more a process is complex in terms of attributes, the lower is the F-score. Moreover, for the same number of attribute, the F-score decreases in case of the learning without ontology while it stays stable with ON2PRM. This explains that the ratio measure increases with the number s of steps (b).

5 Conclusion

In this paper, we presented an approach to learn a PRM by using expert knowledge extracted from an ontology. The advantage of our approach is that (i) the high level knowledge structure organization needed to construct the relational schema can be found directly in the ontology and (ii) this relational schema combined with the semantic knowledge represented by the ontology eases the learning afterward. Thanks to the addition of this semantic knowledge the learning's complexity is reduced and the learnt models are more meaningful than those learnt with a simple direct learning. In our experiments we demonstrated the efficiency of our approach compared to the one without prior knowledge, even in low-complexity processes or with few data.

In future works, we will generalize this approach to consider different temporal relations such as, for example, the time duration of the observations. In this paper, we identified the mapping between classes in the ontology and the PRM, and the temporal dependency properties from the concepts' structuration and the relations of an ontology, following the approach presented in [11]. A global framework of completely automated ON2PRM extraction from an ontology conceptual component and its instance component will face two main challenges: the discovery of oriented relations in ontologies (from data, from expert, etc.) and more complex relations between concepts in ontology and classes in PRM (more complex than a one-to-one mapping). We intend to address these issues by testing our approach and its limits on different ontologies.

References

1. Ben Ishak, M., Leray, P., Ben Amor, N.: Ontology-based generation of object oriented bayesian networks. In: BMAW 2011. pp. 9–17. Spain (2011), https://hal.archives-ouvertes.fr/hal-00644992
2. Christopher D., M., Prabhakar, R., Hinrich, S.: Introduction to Information Retrieval. Cambridge University Press (2008)
3. Cutic, D., Gini, G.: Creating causal representations from ontologies and bayesian networks. In: Arai, Gini, P.E. (ed.) Proc. Workshop NRF- IAS. pp. 1–12. Venice, Italy (18-19th July 2014), http://home.deib.polimi.it/gini/papers/2014NFR.pdf
4. Ettouzi, N., Leray, P., Messaoud, M.B.: An exact approach to learning probabilistic relational model. In: Antonucci, A., Corani, G., Campos, C.P. (eds.) Proceedings of the Eighth International Conference on Probabilistic Graphical Models. pp. 171–182 (2016)

5. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Dean, T. (ed.) 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. Morgan Kaufmann (1999), http://ijcai.org/Proceedings/99-2/Papers/090.pdf
6. Getoor, L., Taskar, B.: Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning). The MIT Press (2007)
7. Gonzales, C., Torti, L., Wuillemin, P.H.: aGrUM: a Graphical Universal Model framework. In: International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems. Proceedings of the 30th International Conference on Industrial Engineering, Other Applications of Applied Intelligent Systems, Arras, France (Jun 2017), https://hal.archives-ouvertes.fr/hal-01509651
8. Helsper, E.M., van der Gaag, L.C.: Building bayesian networks through ontologies. In: Proceedings of the 15th Eureopean Conference on Artificial Intelligence, ECAI'2002, Lyon, France, July 2002. pp. 680–684 (2002)
9. Ibanescu, L., Dibie, J., Dervaux, S., Guichard, E., Raad, J.: Po^2 - A process and observation ontology in food science. application to dairy gels. In: Garoufallou, E., Coll, I.S., Stellato, A., Greenberg, J. (eds.) Metadata and Semantics Research - 10th International Conference, MTSR 2016, Göttingen, Germany, November 22-25, 2016, Proceedings. Communications in Computer and Information Science, vol. 672, pp. 155–165 (2016), https://doi.org/10.1007/978-3-319-49157-8_13
10. Li, X., Zhou, Z.: Structure learning of probabilistic relational models from incomplete relational data. In: Kok, J.N., Koronacki, J., de Mántaras, R.L., Matwin, S., Mladenic, D., Skowron, A. (eds.) Machine Learning: ECML 2007, 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007, Proceedings. Lecture Notes in Computer Science, vol. 4701, pp. 214–225. Springer (2007), https://doi.org/10.1007/978-3-540-74958-5_22
11. Manfredotti, C.E., Baudrit, C., Dibie-Barthélemy, J., Wuillemin, P.: Mapping ontology with probabilistic relational models. In: Fred, A.L.N., Dietz, J.L.G., Aveiro, D., Liu, K., Filipe, J. (eds.) KEOD 2015 - Proceedings of the International Conference on Knowledge Engineering and Ontology Development, part of the 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2015), Volume 2, Lisbon, Portugal, November 12-14, 2015. pp. 171–178. SciTePress (2015), https://doi.org/10.5220/0005590001710178
12. Neapolitan, R.E.: Learning Bayesian Networks. Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2003)
13. Staab, S., Studer, R. (eds.): Handbook on Ontologies. International Handbooks on Information Systems, Springer (2009), https://doi.org/10.1007/978-3-540-92673-3
14. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press (2005)
15. Torti, L., Wuillemin, P.H., Gonzales, C.: Reinforcing the Object-Oriented Aspect of Probabilistic Relational Models. In: PGM 2010 - The Fifth European Workshop on Probabilistic Graphical Models. pp. 273–280. Helsinki, Finland (Sep 2010), https://hal.archives-ouvertes.fr/hal-00627823
16. Wuillemin, P., Torti, L.: Structured probabilistic inference. Int. J. Approx. Reasoning 53(7), 946–968 (2012), https://doi.org/10.1016/j.ijar.2012.04.004