Grounding Causality in Bayesian Networks Using Qualitative Reasoning

Mélanie Munch[1], **Kamal Kansou**[2], **Bert Bredeweg**[3,4], **Cédric Baudrit**[1], **Pierre-Henri Wuillemin**[5]

[1] INRAE, Bordeaux University-I2M, Talence, France
[2] INRAE, BIA, Nantes, France
[3] Faculty of Education, Amsterdam University of Applied Sciences, Amsterdam, The Netherlands
[4] Informatics Institute, Faculty of Science, University of Amsterdam, Amsterdam, The Netherlands
[5] Sorbonne University, CNRS, LIP6, Paris, France

Abstract

The complexity of analysing dynamical systems often lies in the difficulty to monitor each of their dynamic properties. In this article, we use qualitative models to present an exhaustive way of representing every possible state of a given system, and combine it with Bayesian networks to integrate quantitative information and reasoning under uncertainty. The result is a combined model able to give explanations relying on expert knowledge to predict the behaviour of a system. We illustrate our approach with a deterministic model to show how the combination is done, then extend this model to integrate uncertainty and demonstrate its benefits.

Introduction

Reasoning about a specific system's behaviour requires a good understanding of the involved **entities**, their **quantities** (*i.e.* their relevant numerical properties), how these are related and the value they can take on. Establishing a **model** able to explain those relations and the general behaviour of the studied system is a complex task, hindered further by the introduction of uncertainty: quantities are not always observed and values tainted with errors can frustrate the interpretations.

Bayesian Networks (Pearl 1985) (BNs), thanks to their graphical aspect, allow to understand the underlying probabilistic dependencies between the quantities (denoted as **variables** in this context). However, they can be impaired by the lack of physical understanding. While the models learned with BNs offer a good quantitative description of the studied system, they might lack explainability (*i.e.* their results do not always match human logical reasoning). This is due to the fact that BNs build correlations, and not causation: in practice, a model could learn **Rain→Grass** ("The rain soaked the lawn") as well as **Grass→Rain** ("The soaked lawn provoked the rain"). To prevent such unwanted results, (Pearl 2009) defined interventions (*i.e.* modifying one quantity without touching the others) to construct causal models. This however is not always doable for practical, ethical or economic reasons: for instance, studying the impact of smoking on health would require to intervene on people to force them to smoke.

Integrating external sources of knowledge can be useful to guide the learning and prune impossible models. The most common way of doing so for BNs is to impose a complete (Baudrit et al. 2022) or partial (Munch et al. 2022) structure, built with experts. This structure is denoted as **theory**, as it reflects the experts' (often causal) knowledge over the considered system. This approach helps to select the relevant variables, and reduces the learning to the parameters (the probabilities). However, this raises the question of the correctness and/or completeness of the fed causal theory: depending on the experts, their number, their area of expertise, ... several can be proposed, each with possible distinct impact over the learning.

On the other hand, qualitative reasoning (QR) builds sound models with solid grounding on causality. By reasoning over quantities and defined relations, they can generate all possible states of a system without relying on data (Forbus 2011). Instead, they allow to define **quantity spaces**, in order to consider only relevant values (*e.g.* $\{\emptyset$, Low, Medium, High$\}$) and to reason on a symbolic level. As such, they give a complete description of the system which can be used to assess the validity of the expert knowledge integrated in the BN's learning.

In this article, we combine BNs with knowledge of the system physics represented as qualitative models (QMs) to learn models able to apprehend uncertain systems with explainable answers. Below, the first section presents the necessary notions and state of the art on QM, BNs and the use of QM for quantitative modeling. The second section presents the principle of our approach illustrated by an example. Finally, the third section compares the results of our approach compared to naive BN learning in order to demonstrate the gain in explainability.

Modeling and simulation have been done using the Dynalearn environment (Bredeweg et al. 2013), which is based on the Garp3 software (https://dynalearn.nl/). BN learning and computing have been done using the PyAgrum library (Ducamp, Gonzales, and Wuillemin 2020).

Background

Qualitative Modeling with Garp3

Garp3 (Bredeweg et al. 2009) defines a qualitative system through (1) the use of entities and their associated quantities and (2) their relations. Quantities are described by their value (magnitude, *e.g.* +) and direction of change (derivative, *e.g.* 0). Values are picked from associated quantity spaces,

which holds every possible values they can take. While magnitudes' quantity spaces can be defined as desired by experts, derivatives' are fixed: negative, null or positive. Following Garp3's notation, they are denoted as $\{\blacktriangledown, \bullet, \blacktriangle\}$, or {-, 0, +}. A combination of magnitude and derivative for each quantity (*e.g.* $<0, +>$) defines a **state**, *i.e.* the behaviour of the system at a certain time. Each state is a unique qualitative behaviour of the system, characterized by a unique set of quantity values and derivatives. Passing from one state to the other represents the evolution of the system: a **graph of state** is defined by a graphical representation of all possible transitions between the different states, where each node is a state and the edges the possible transitions.

In order to compute this graph, Garp3's inference engine reasons over two types of qualitative relations which defines causal relations between each other (Forbus 1984): *proportionalities* (changes caused by processes, denoted as P-/+), and *direct influences* (causal propagation of changes, denoted as I-/+). Additional constraints can be added: **correspondences** and **inequalities** allow the user to describe the relations between certain quantity's values and quantities (*e.g.* force the zero value, or force a value to be always higher than another). Finally, reasoning is done over **scenarios**, which define (some) values for the initial state.

Bayesian Networks

Bayesian Networks (BNs) (Pearl 1985) are acyclic graphs G=(V,E), with V and E respectively the sets of all their nodes (representing random variables) and arcs (representing the conditional dependencies). To each variable, a conditional probability table (CPT) is associated, giving the probability distribution for each possible value it can take and how the values of its parents (i.e. variables that have an oriented path toward that variable) influence it (as shown in Fig.1). A joint probability over all nodes V is defined as the product of local probabilities given as:

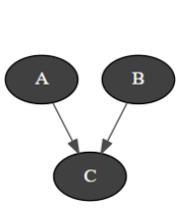
$$P(X_1, ...X_n) = \prod_{i=1}^{n} P(X_i | Pa(X_i))$$

with $P(X_i|Pa(X_i))$ being the conditional probability function associated with random variable X_i, conditioned on its parents $Pa(X_i)$. A probability of 0 describes an impossible event, while a probability of 1 is associated to a certain event.

While models have been proposed to take into account continuity within BN's structure, this article focuses on the discrete part. BN's learning is usually done is two steps: considering a discretized database, the structure G is first learned, then the probabilities. In this study, this last part is tackled, as structure is provided by the addition of expert knowledge from the QM.

Combining Quantitative and Qualitative

Explainable Artificial Intelligence has gained a tremendous attention over the past years (Guidotti et al. 2018), as the need of justifications for supporting a model's predictions is a key-question. More generally, there is an increase in the need of understanding things correctly (*e.g.* science).



(a) BN

A	
0	1
0.1997	0.8003

B	
0	1
0.4488	0.5512

(b) Marginal Distribution

		C	
B	A	0	1
0	0	0.4494	0.5506
	1	0.8381	0.1619
1	0	0.9635	0.0365
	1	0.0424	0.9576

(c) Conditional Distribution

Figure 1: (a) Example of a BN composed of three variables **A**, **B** and **C**. (b) Marginal distributions associated to **A** and **B**. (c) CPT associated to **C**: in this example the probabilities of **C**'s values (columns) depend on **A** and **B**'s values (rows).

Thanks to their graphical component BNs offer explainability for their prediction. However, the lack of causality in their approach leads to inaccurate models, unable to describe real systems. Algorithms such as PC (Spirtes, Glymour, and Scheines 2000) or more recently MIIC (Verny et al. 2017) have been developed to tackle this issue and learn causal structures from data alone. These approaches are however costly in data. On another hand, integrating expert knowledge (*e.g.* as partial node ordering (Parviainen and Koivisto 2013)) during the learning helps reducing the data cost by reducing the search space (Munch et al. 2017). Yet, different causal models lead to different BNs, whose correctness can be difficult to evaluate.

In this article, QM is proposed to define a stable structure able to frame the quantitative reasoning and integrate it into quantitative learning. Such combination has been proposed, for instance to improve simulations based on dynamic equations (Pang, Bruce, and Coghill 2018). In this case, QM allows to define constraints that reduce intervals of simulation for already known equations. It is often proposed to model systems in order to bypass equations and simplify the simulations (Soberl and Bratko 2022; Struss, Reiser, and Kreuzpointner 2018). (Klenk, Nabi, and Arvay 2016) proposes a methodology to compare different explanatory models for co-morbidities, using QM to develop mechanistic explanations. While they do not rely on data, they raise the question of inferences: given a patient and a validated causal model, is it possible to derive conclusions? In the frame of this article, the combination of BNs et QM would allow to answer quantitatively to these questions with probabilities, *i.e.* proposing different possible answers with probabilities of their happenstance. More generally, it aims at answering the three advantages defined by (Forbus and Falkenhainer 1990) for the combination of quantitative simulation with qualitative knowledge: (1) increased automation (*i.e.* no need for manually defining each relevant equation), (2) improved self-monitoring (*i.e.* consistence checking with reality) and (3) better explanations (*i.e.* justifications of predictions based on causal reasoning).

Combining BNs and QMs

This section presents the combination of BNs and QMs as showed in Fig.2, illustrated with a system.

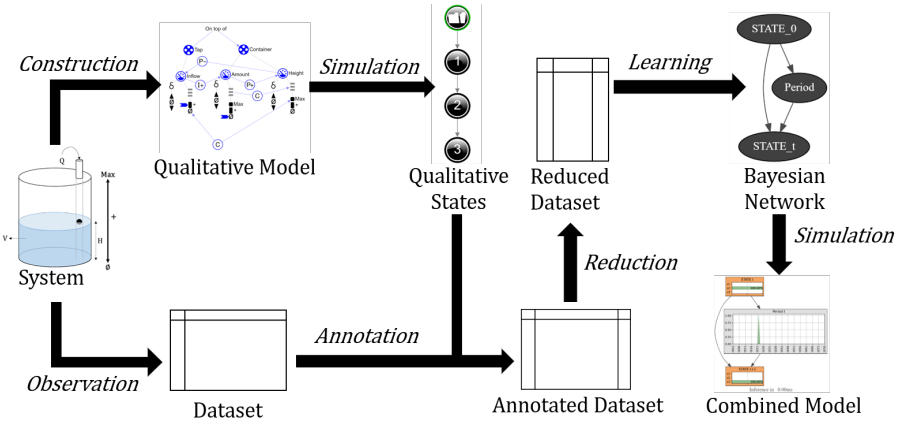


Figure 2: Summary of the approach. From a given system, a QM is constructed then used in a simulation to extract the different possible qualitative states. From a dataset observing the different values of the systems quantities, an annotated dataset is constructed, using the qualitative states to describe the dataset. This dataset is then transformed to allow the learning of a BN.

Example: the Container System

Consider a container with a floating cap being filled with water described by three quantities (illustrated in Fig.3 (a)):

Q The inflow of water going in the container through a tap. Initial flow is denoted Q_{in}.

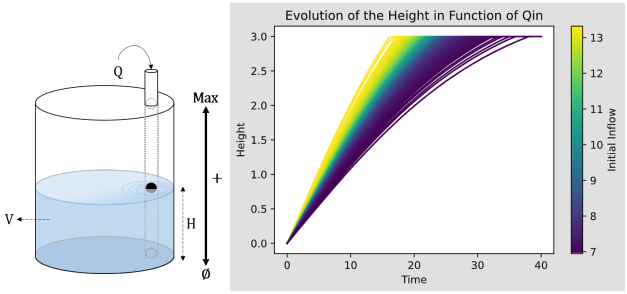
V The current volume of water in the container. Maximum volume is denoted V_{max}.

H The current height of the floating cap. Maximum height is denoted H_{max}.

Starting with an empty container, water is introduced at a given flow, which arbitrarily decreases while height increases. A floating cap is present such that, once the container is filled, it interrupts the flow. For the following, a dataset describing the values of the different quantities through different simulation is considered. Each simulation is initialized using:

- $H_{max} = 3$
- $V_{max} = 3\pi$
- $Q_{in} \hookrightarrow \mathcal{N}(10, 1)$

While this approach is able to address deterministic systems, randomness is introduced to demonstrate its robustness when facing uncertainty. Fig.3 (b) shows the influence of Q_{in} on the filling rate speed.



(a) Container system (b) Impact of Qin

Figure 3: Qualitative modeling of the container system. (a) System. (b) Impact of the initial inflow $\mathbf{Q}_{in}$ value over the time required to reach the maximal height.

Qualitative Model and States

The first step propose for the dynamic of this system a QM. Following the system's description, we consider two objects and three quantities:

- The ***tap***, associated to the **inflow** quantity **Q**. The quantity space is $\{\emptyset, +\}$, referring respectively to the absence and presence of flow.
- The ***container***, associated to the **amount V** of water and **height H** of the cap. Their quantity spaces is $\{\emptyset, +, Max\}$, with $<Max>$ respectively the maximum volume and height. For both, $<\emptyset>$ refers to the null value (*i.e.* no water), while $<+>$ refers to the amount and height in between.

Fig.4 (a) presents the relations existing between the defined objects and variables as well as the initial values used for the simulation: **Inflow** is $<+>$, while the **Amount** is $<\emptyset>$. This creates three possibles states for the system, denoted in the rest of this article as $s1$, $s2$ and $s3$:

$s1$ The tank is empty: water starts flowing through the tap. The volume and height are null, but increasing.

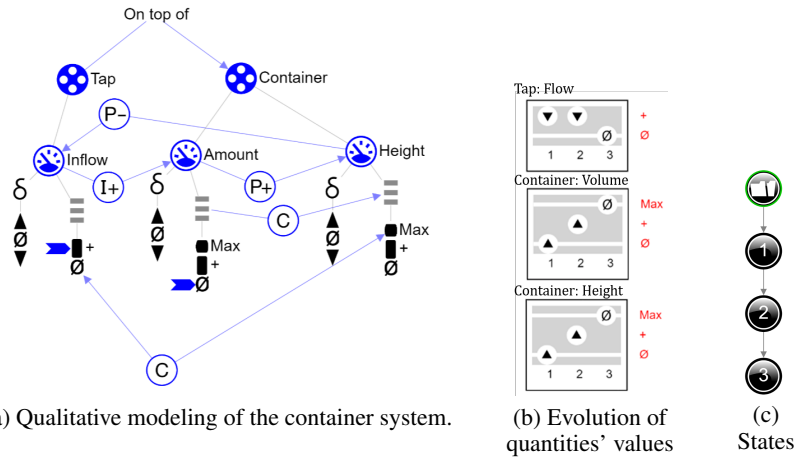
$s2$ The tank is being filled: the volume and height are not null and increasing, while the flow decreases.

$s3$ The tank is filled: water stops flowing. All quantities' derivative are null, the system is at equilibrium.

Table 1 recaps the states different values, while Fig.4 (b) and (c) presents the simulation's results.

Annotated Dataset

While QM reasons over states and transitions between those, quantitative models such as BNs are dedicated to the study



(a) Qualitative modeling of the container system. (b) Evolution of quantities' values (c) States

Figure 4: Qualitative modeling of the container system. (a) Model and initial conditions. (b) Evolution of the quantities' values during simulation. (c) State graph of the simulation.

State	**Q**	**V**	**H**
$s1$	<+, ▼>	<Ø, ▲>	<Ø, ▲>
$s2$	<+, ▼>	<+, ▲>	<+, ▲>
$s3$	<Ø, ●>	<+, ●>	<Max, ●>

Table 1: Description of the quantities <Magnitude, Derivative> for each state.

variable's evolution across fixed intervals. This difference of focus requires the definition of a novel quantity in order to bridge between the two representations: the **Time Step**. In practice, learning a quantitative model requires values to reflect on; when learning a dynamic model, it helps to have data describing the system at regular intervals (the time steps). This is particularly important as the aim of the final model is to be able to describe precisely the evolution (*i.e.*, the passage or not from one state to the other) of the system: irregular time steps would scramble the predictions. For the following, the time step quantity refers to the time at which the system is described.

Using Table 1 states, each time step is associated to one:

1. By discretizing the quantity's value with its quantity space. For instance, if **Height** = 0, then its discretized value is <Ø>; H_{max} becomes <Max>; otherwise, it is discretized to <+>.
2. By looking at the derivative for each quantity: if the difference between the quantity's value at time t and $t+1$ is negative, then the derivative is also *negative* (▼); if the values are equal, then the derivative is *null* (●); if it is positive, then the derivative is *positive* (▲).

In the end, using the combination of the discretized value and the derivative for each quantity, each time step can be associated to a QM state. A new quantity is also introduced for interval of Time Steps: the **Period**. While **Time Step** marks the passing of time, the **Period** indicates how long the system has been in the current state. For the rest of this article, given a quantity **X**, the variable $\mathbf{X}_t$ denotes its value at time step t. Table 2 shows an example of the whole discretization process.

Time	H	V	Q	dH	dV	dQ	State	Period
1	0	0	10	▲	▲	▼	$s1$	1
2	2.9	9.1	7.3	▲	▲	▼	$s2$	1
3	4.6	14.4	3.7	▲	▲	▼	$s2$	2
4	5	15.7	0	●	●	●	$s3$	1
5	5	15.7	0	?	?	?	$s3$	2

Table 2: Example of a discretization using the QM, considering Q_{in}=10, H_{max}=5 and 5 time steps. Since Step 3 is an equilibrium state, we assume that the observation at time 5 still matches state 3, although the derivatives are unknown.

State$_t$	**State**$_{t+1}$	**Period**$_t$	**State**$_t$	**State**$_{t+1}$	**Period**$_t$
$s1$	$s2$	1	$s3$	$s3$	1
$s2$	$s2$	1	$s3$	$s3$	2
$s2$	$s3$	2			

Table 3: Transformation of the database of Table 2 into a database suitable for the BN learning.

Reduced Dataset

Since the model aims to learn the evolution of the system, *i.e.* the transitions between steps, a new dataset is composed from the **State**$_t$, **State**$_{t+1}$ and **Period**$_t$. This way, each lines brings information of the system's state, how long it has been this way, and whether it will remain the same (or transition) in the next time step. Table 3 shows the transformation applied to Table 2 in order to be able to learn a BN.

Bayesian Network

Structure Definition Once the database is prepared, a structure is manually defined to guide the BN learning, based on two assumptions:

		STATE t+1		
Period t	STATE t	s1	s2	s3
t022	s1	0.5000	0.5000	0.0000
	s2	0.0000	0.6897	0.3103
	s3	0.0000	0.0000	1.0000

Table 4: Excerpt of the CPT showing the probabilities of passing from $\mathbf{State}_t$ to $\mathbf{State}_{t+1}$ if $\mathbf{Period}_t = 22$.

- The $\mathbf{Period}_t$ value depends only on the value of $\mathbf{State}_t$;
- The probability of passage from $\mathbf{State}_t$ to $\mathbf{State}_{t+1}$ depends on $\mathbf{State}_t$ and the $\mathbf{Period}_t$;

This defines the following structure: $\mathbf{State}_t \rightarrow \mathbf{State}_{t+1} \leftarrow \mathbf{Period}_t \rightarrow \mathbf{State}_t$.

Parameters Learning Once the structure defined, parameters are learned through a statistical learning whose goal is to maximize the likelihood by estimating the probability of an event according to its frequency in the considered database. In case an event is never observed (*e.g.* if the system never stays more than one time step in $s1$, then the combination $\{\mathbf{State}_t{=}s1,\ \mathbf{Period}{=}10\}$ is never observed), the probabilities are by default equiprobable: all possible outcomes are considered as likely. The learned BN thus encompasses the QM model in its structure, and heavily depends on the data only for its parameters.

Combined Model

In this article, two applications are presented in order to demonstrate the reasoning offered by the learned model:

1. **State Prediction.** Reading the CPT, the probability of passing from one state to the other knowing the period can be deduced. Table 4 presents an excerpt, focusing on the passage from one step to the other after a period of 22 ($\mathbf{Period}_t = 22$). It shows that depending of $\mathbf{State}_t$, the most probable value of $\mathbf{State}_{t+1}$ depends: if $\mathbf{State}_t = s2$, then it has a probability of 0.69 of staying $s2$; on another hand, if $\mathbf{State}_t = s3$, then it will stay $s3$ (which is logical, since it is a equilibrium state). To be noted, if $\mathbf{State}_t = s1$, then the probability of transitioning is equiprobable between $s1$ and $s2$ ($s3$ is not considered as $s1 \rightarrow s3$ is not possible according the state graph): this is due to the fact that the system has no information about cases where a system has stayed 22 time steps in $s1$.
2. **Period Prediction.** Another way of exploiting the probabilistic relations is to make inferences: knowing the value of some variables, it is possible to compute the most probable values of the others. Fig.5 presents such an example: knowing that $\mathbf{State}_t = s2$ and $\mathbf{State}_{t+1} = s3$ (in orange to indicate it is *observed*), the most probable period (in grey to indicate it is *computed*) is 22.

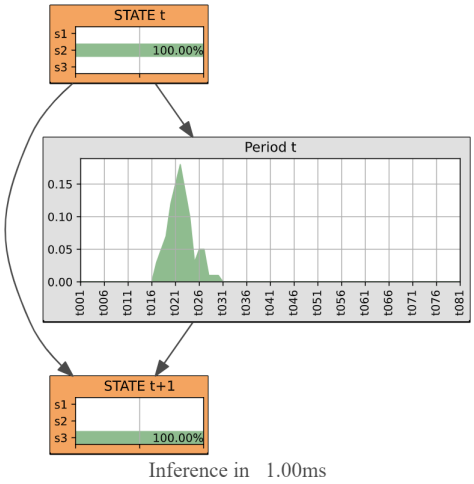


Figure 5: Example of an inference using the BN.

Comparison to Naive BN[1]

In order to evaluate its performances, the combined model is compared to naive BNs. To do so, two naive BNs integrating different expert constraints are presented. Evaluation is done for both (1) the graph and (2) predictions, which are compared against a ground truth.

Naive BNs Learning

"Naive" refers to the fact that the QM's model is not known during the learning, the main difference being that state knowledge is not taken into account. This section presents two versions, with different degree of the model's understanding:

1. Learning is approached with no information from the system at all. Discretization is made using quantiles (instead of the system space's values), and the structure is constrained only so that variables from the past ($\mathbf{Q}_t$, $\mathbf{V}_t$, $\mathbf{H}_t$) can be used to predict the future ($\mathbf{Q}_{t+1}$, $\mathbf{V}_{t+1}$, $\mathbf{H}_{t+1}$), but not the contrary. This approach represents the most naive learning, and gives an idea on how BNs handle this kind of data without prior knowledge of the system. It is denoted as the unguided approach.
2. A second learning is made to include more system's knowledge. The QM's space's values is used for the discretization, and the structure is forced in order to transcribe the expert knowledge used in the QM. This approach is denoted as the guided approach.

Unguided Approach Seven variables are considered: six to capture the values of the quantities **Q**, **V** and **H** at times t and $t+1$, and one to capture the **Time Steps**. To be noted, the **Time Steps** variable in this context is different from the $\mathbf{Period}_t$ one presented until now: since states are not known, time refers here to the beginning of the simulation, and not to the time passed in a certain state. The structure

[1]All code used in this article are available at https://gitlab.com/melanie.munch/qr23-submission

is learned through a classical greedy algorithm (Chickering 2003), with the only constraint that variables at time t can explain variables at time $t+1$, but not the contrary (temporal constraints). Discretization is done such that (1) **Q**, **V** and **H** are discretized in 5 quantiles; (2) V_{max} and H_{max} are a 6^{th} category in order to capture when the tank is filled; **Time Steps** is not categorized to keep track of the time as precisely as possible.

Guided Approach This approach still considers seven variables, but handles them differently. First, discretization is done following the QM's space values; secondly, structure is oriented so that additionally to the temporal constraints forced in the naive version, it also takes into account (1) the expert knowledge integrated in the QM (**Q**→**V**→**H**), (2) the influence of the **Time Steps** variable over the values measured at time t and (3) for each variable its value at time t to predict its value at $t+1$.

Simulations

Given a database of 1000 simulations, three models are learned using the same sample of 100 experiments:

1. **Combined Model:** A model learned using the method presented in the previous section.
2. **Unguided Model:** A model learned using the unguided approach presented in this section.
3. **Guided Model:** A model learned using the guided approach presented in this section.

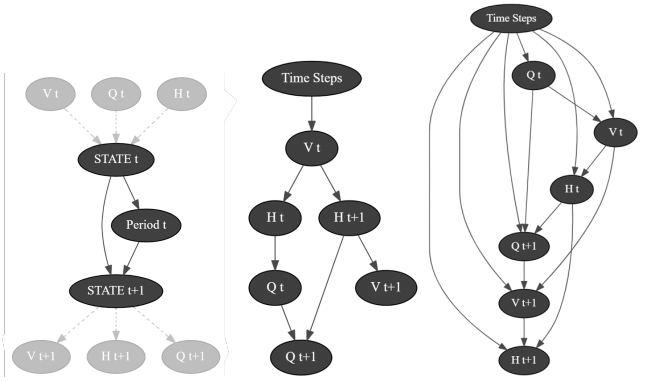
The database of 1000 experiments represents the ground truth that the learned models aim to reach.

Graph Evaluation

Result of the learning are presented in Fig.6. For the sake of explainability, variables **Q**, **V** and **H** have been represented in the combined model (a), so that it can be compared to the other structures. Since it was learned without knowledge, unguided structure (b) differs the most from the QM structure of Fig.4 (a), leading to non causal relations (*e.g.*, $\mathbf{V}_t \rightarrow \mathbf{H}_t \rightarrow \mathbf{Q}_t$). As such, the learned relation are not able to explain the system in a causal way: it only displays correlations, and cannot generate sound explanations to justify the model's prediction. On another hand, the guided approach (c) presents a structure coherent with the QM. Moreover, on the contrary of the combined structure, it directly displays the relations between the variables, instead of having them hidden between the states transitions. While this is an advantage in term of readability for systems with only a few variables, this can become a hassle when considering bigger systems.

Predictions Evaluation

Generation For each model, 1000 simulations are done using the principle illustrated in Fig.7: starting from the same initialization ($\mathbf{Q}_t$>0, $\mathbf{V}_t$=0, $\mathbf{H}_t$=0, **Time Period/Period**$_t$=1), marginal laws for the next step are computed (*i.e.*, probabilistic distribution for the possible values). Using these laws, new values for the variables $\mathbf{Q}_{t+1}$, $\mathbf{V}_{t+1}$ and $\mathbf{H}_{t+1}$ are drawn. If either $\mathbf{H}_{t+1}$ takes the maximal value



(a) Combined Structure (b) Unguided Structure (c) Guided Approach Structure

Figure 6: Models' structures comparison.

or the number of time steps exceed 100, then the simulation is finished. In the first case, the number of time steps is kept; in the second, it means that the model could not reach the end of the simulation and thus the run is incorrect. Frequency of the number of time steps required to conclude the simulation (*i.e.* to fill the container) are compared to the frequency measured in the initial dataset.

Results Results are presented in Fig.8. The first notable thing is that despite the fact that only 10% of the original dataset was used to learn the models, all models have an average time of filling close to the ground truth's. A Kolmogorov-Smirnov goodness of fit test is performed in order to compare each distribution to the baseline: H_0 means that both distributions are identical, while H_1 means they are distinct. H_1 is rejected for both combined and guided models (with p-values respectively of 0.6 and 0.3), while it is validated by the unguided approach (p-value$<10^{-5}$). This means that the unguided approach did not manage to capture the underlying distribution of the dataset. On another hand, both combined and guided are statically indistinguishable, both having an average expectancy of time steps (*i.e.* the average time taken to fill the container) close to the truth's (respectively 24.3 and 23.2 against 24.7).

The main difference between combined and guided models lies in the evolution of the different values. Fig.9 shows three independent simulation results for each model. Guided and unguided are characterized by (1) a decorrelation between the three variables (*e.g.* **V** reaches *Max* value before **H**); and (2) impossible evolution of the values (*e.g.* **Q** increasing). This shows that even if the guided model is close in structure and (for this particular problem) of the ground truth's predictions, it fails at providing an explanation grounded into the causal model. Combined model, on the contrary, is able to provide a description of the system which is consistent with the QM.

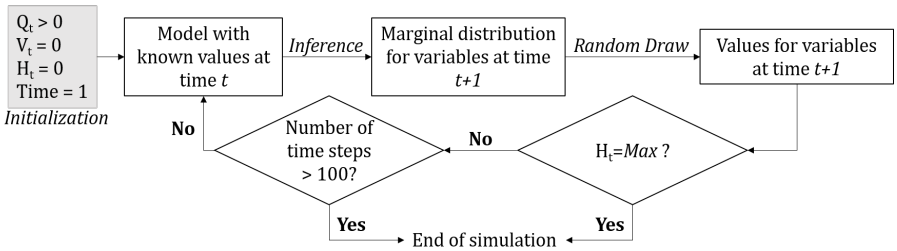
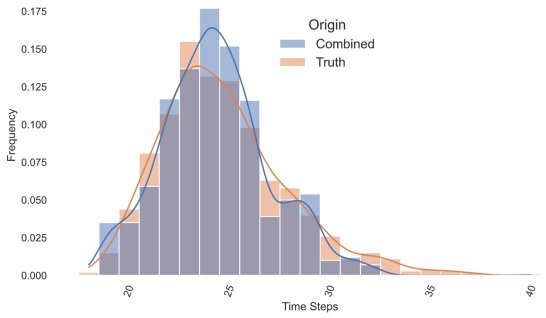
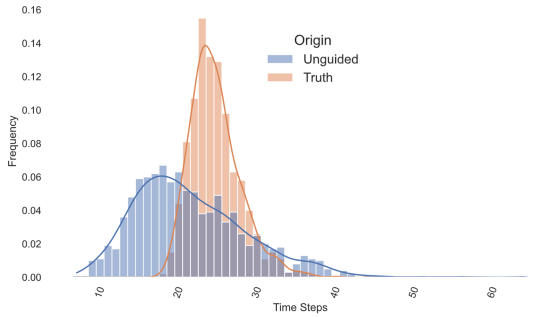


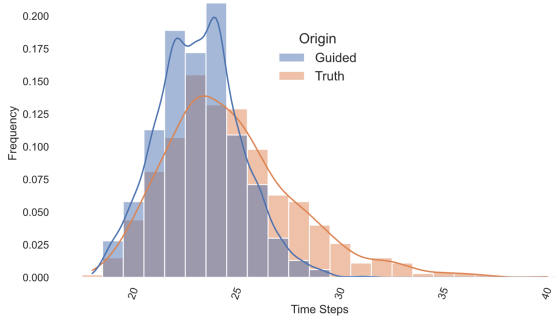
Figure 7: General flowchart for the simulations. To be noted, in the case of the combined model, variables are encompassed in the **State** variable and **Time** is replaced by the **Period** variable.



(a) Combined Model (24.3)



(b) Unguided Model (21.6)



(c) Guided Model (23.2)

Figure 8: Frequency of the times taken to finish the simulation (average number of time step). Truth has an average number of time step of 24.7.

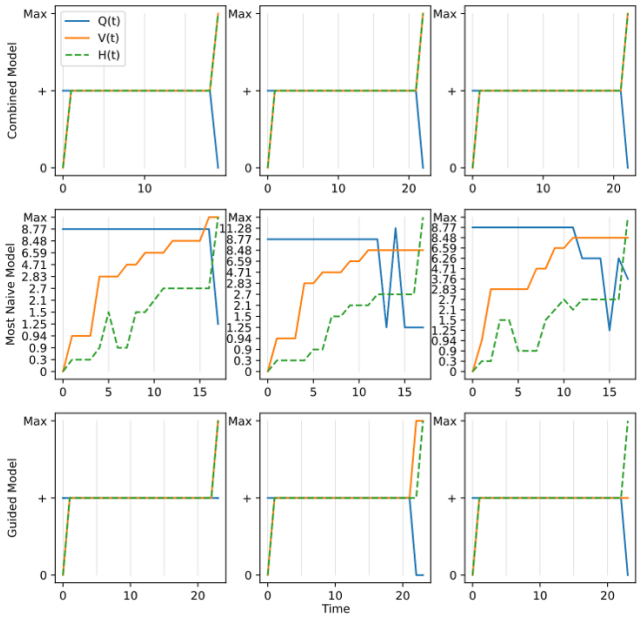


Figure 9: Example of three independent simulations for each model.

Discussion

Comparison to naive BNs has shown that grounding causal knowledge from QM to BN's learning results in a model able to provide simulation close to the reality of the studied system. This is particularly due to the distinction between **Period** and **Time Steps** quantities: while the combined model is able to reason only on state transitions, naive models can only consider total times. As such, they cannot reason about the state they are in, but only how long the simulation has been running. In this simulation, the simplest case was considered, as only $s2$ had a non-constant time: $s1$, for instance, always lasts one time step. Further experiments should be done on systems with more complex state graphs (*e.g.* with cycles and branching paths), in order to assess whether the combined model can adapt. While the approach presented in this article only requires a dataset and a QM as inputs, more work should be done on its automation. More broadly, it should be interesting to see how the combined model can scale on systems with more quantities. Especially, it is im-

portant to also take into account the challenges brought by introducing more quantities, as some could be uncertain or missing from the dataset: states could then be uncertain as well, if not all quantities are known.

Another lead to explore would be to use the combine model to assess the adequacy between a theory and a dataset. (Kansou et al. 2017) proposes two tests to define whether a model can be well described by a QM or not: the encompassment (the adequacy between the QM and the dataset) and the sufficiency (the adequacy between the QM and the model's behaviour). To pass this verification, it is important to consider technical aspects:

- The choice of time steps has an influence: if too great, it is possible to skip some combinations of value (and thus states) when annotating the dataset. This would result in a model not respecting the state graph. For instance, if passing from states takes 2 time steps ($s1 \rightarrow s1 \rightarrow s2 \rightarrow s2 \rightarrow s3$), then having a time step of 3 would lead to a model predicting a passage from 1 directly to 3 ($s1 \rightarrow s3$).
- Another critical point is the computation of derivatives. The same way the choice of time steps influences the model's learning, data's sensibility can influence the derivatives' precision. Indeed, depending of the precision of measurement, zero derivative can be hard to catch, as it usually concerns one data point.

Finally, it is important to consider that the data depends on multiple parameters not represented as quantities in the QM. For instance, the required time to fill the container depends on V_{max} (maximal height and radius): if a model is learned only on high and/or large containers, its predictions will not be relevant for smaller containers.

Conclusion

In this article, a new approach of combining BNs with QM has been presented, with the goal of improving BN's modeling by integrating expert knowledge. Comparison with naive BNs has displayed better results for the combined model in term of prediction and explainability. In conclusion, the resulting model is able to provide explainable answers and simulations over an uncertain system. The learning is based only on a dataset and the expert knowledge encompassed in the QM, which dispenses the modeller with the prior definition of system equations.

References

Baudrit, C.; Buche, P.; Leconte, N.; Fernandez, C.; Belna, M.; and Gésan-Guiziou, G. 2022. Decision support tool for the agri-food sector using data annotated by ontology and Bayesian network: a proof of concept applied to milk microfiltration. *International Journal of Agricultural and Environmental Information Systems*, 13(1).

Bredeweg, B.; Liem, J.; Beek, W.; Linnebank, F.; Gracia, J.; Lozano, E.; Wissner, M.; Bühling, R.; Salles, P.; Noble, R.; Zitek, A.; Borisova, P.; and Mioduser, D. 2013. DynaLearn - An Intelligent Learning Environment for Learning Conceptual Knowledge. *Ai Magazine*, 34: 46–65.

Bredeweg, B.; Linnebank, F.; Bouwer, A.; and Liem, J. 2009. Garp3 — Workbench for qualitative modelling and simulation. *Ecological Informatics*, 4(5): 263–281.

Chickering, D. M. 2003. Optimal Structure Identification with Greedy Search. *J. Mach. Learn. Res.*, 3: 507–554.

Ducamp, G.; Gonzales, C.; and Wuillemin, P.-H. 2020. aGrUM/pyAgrum : a Toolbox to Build Models and Algorithms for Probabilistic Graphical Models in Python. In *10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, 609–612. Skørping, Denmark.

Forbus, K. D. 1984. Qualitative process theory. *Artificial Intelligence*, 24(1): 85–168.

Forbus, K. D. 2011. Qualitative modeling. *WIREs Cognitive Science*, 2(4): 374–391.

Forbus, K. D.; and Falkenhainer, B. 1990. Self-Explanatory Simulations: An Integration of Qualitative and Quantitative Knowledge. 380–387.

Guidotti, R.; Monreale, A.; Ruggieri, S.; Turini, F.; Giannotti, F.; and Pedreschi, D. 2018. A Survey of Methods for Explaining Black Box Models. *ACM Comput. Surv.*, 51(5).

Kansou, K.; Rémond, C.; Païs, G.; Bonnin, E.; Tayeb, J.; and Bredeweg, B. 2017. Testing scientific models using qualitative reasoning: Application to cellulose hydrolysis. *Scientific Reports*, 7.

Klenk, M.; Nabi, M.; and Arvay, A. 2016. Challenges in Formulating Explanatory Models for Co-morbidities.

Munch, M.; Buche, P.; Dervaux, S.; Dibie, J.; Ibanescu, L. L.; Manfredotti, C.; Wuillemin, P.-H.; and Angellier-Coussy, H. 2022. Combining ontology and probabilistic models for the design of bio-based product transformation processes. *Expert Systems with Applications*, 203: 117406.

Munch, M.; Wuillemin, P.-H.; Manfredotti, C.; Dibie, J.; and Dervaux, S. 2017. Learning Probabilistic Relational Models Using an Ontology of Transformation Processes. In Panetto, H.; Debruyne, C.; Gaaloul, W.; Papazoglou, M.; Paschke, A.; Ardagna, C. A.; and Meersman, R., eds., *On the Move to Meaningful Internet Systems. OTM 2017 Conferences*, 198–215. Cham: Springer International Publishing. ISBN 978-3-319-69459-7.

Pang, W.; Bruce, A.; and Coghill, G. 2018. Non-constructive interval simulation of dynamic systems.

Parviainen, P.; and Koivisto, M. 2013. Finding Optimal Bayesian Networks Using Precedence Constraints. *Journal of Machine Learning Research*, 14: 1387–1415.

Pearl, J. 1985. Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning. In *Proc. of Cognitive Science Society (CSS-7)*.

Pearl, J. 2009. *Causality: Models, Reasoning and Inference*. New York, USA: Cambridge University Press, 2nd edition. ISBN 052189560X, 9780521895606.

Soberl, D.; and Bratko, I. 2022. Qualitative control of cart and pole on uneven terrain.

Spirtes, P.; Glymour, C.; and Scheines, R. 2000. *Causation, Prediction, and Search*. MIT press, 2nd edition.

Struss, P.; Reiser, R.; and Kreuzpointner, F. 2018. A Qualitative Gait Model for Diagnosis of Juvenile Idiopathic Arthritis.

Verny, L.; Sella, N.; Affeldt, S.; Singh, P. P.; and Isambert, H. 2017. Learning causal networks with latent variables from multivariate information in genomic data. *PLOS Computational Biology*, 13(10): e1005662.