Support for high-level quantum Bayesian inference

Marcin Przewięźlikowski[1], Michał Grabowski[1], Dariusz Kurzyk[2,3], and Katarzyna Rycerz[1]

[1] Institute of Computer Science, AGH, al. Mickiewicza 30, 30-059 Kraków, Poland
[2] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland
{przewiez,mgrabow}@student.agh.edu.pl, dkurzyk@iitis.pl, kzajac@agh.edu.pl

Abstract. In this paper, we present `AcausalNets.jl` - a library supporting inference in a quantum generalization of Bayesian networks and their application to quantum games. The proposed solution is based on modern approach to numerical computing provided by Julia language. The library provides a high-level functions for Bayesian inference that can be applied to both classical and quantum Bayesian networks.

Keywords: Quantum Bayesian networks · Quantum games · Julia language

1 Introduction

Bayesian networks [10] are probabilistic models which, among their numerous use cases, allow to model complex systems of interconnected random events in games of chance and their influence on each other. There are several approaches to generalizing Bayesian probability theory into the quantum realm [7,8,10].

Introducing quantum probability into game theory opens up new opportunities for finding optimal strategies in various games of chance. Apart from well known work on representing quantum strategies as unitary gates [9] or applying quantum entanglement to find optimal strategies [2], there is also relatively new approach to apply quantum Bayesian networks for that purpose [5]. This new approach has not yet been fully explored, therefore we focus on methods, algorithms and numerical support for researchers working this topic.

Proper numerical tools are required due to high complexity of computations essential to perform such experiments. In particular, performing Bayesian inference in Acausal networks, a quantum generalization of Bayesian networks, is not yet fully supported among numerical libraries.

In this paper we present `AcausalNets.jl` - a library providing a high-level functions for Bayesian inference that can be applied to both classical and quantum Bayesian networks. The library takes advantage of the new approach to numerical computing offered by Julia language [1].

Organization of the paper The paper is organized as follows: in Section 2 we summarize relevant related work which provided basis and inspiration for

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_76

this paper. In Section 3 we describe Bayesian networks and their usage in higher detail. We also briefly delve into the inference algorithms implemented in `AcausalNets.jl`. In Section 4 we provide in brief detail the principles we followed when implementing the library. Section 5 sums up our results when recreating and expanding experiments first conducted and described in [5]. In Section 6 we provide a brief overview of our ideas for further improvements and enhancements of `AcausalNets.jl`.

2 Related work

Belief Propagation algorithms that we adapt for quantum Bayes networks in `AcausalNets.jl` library are covered to a fuller extent in [4] and [10]. The adaptation was done using theoretical foundations for generalizing Bayesian probability theory into the quantum realm described in [8,7]. To our best knowledge, there is no numerical support for quantum version of that algorithm.

Additionally, our work was inspired by `BayesNets.jl` [3] - a `Julia` library designed for high-level operations on classical Bayesian Networks.

In general, there is a huge variety of software related to quantum information implemented in numerous programming languages, most notably `QuantumInformation.jl` [3] implemented in `Julia`. The set of functionalities these libraries provide, while wide, is focused mainly on low-level optimized matrix operations. `AcausalNets.jl` makes use of matrix operations implemented in `QuantumInformation.jl` in its implementation of Belief Propagation algorithms.

3 Classical and Quantum Bayesian Networks

Bayesian Networks are probabilistic graphical models used for describing systems of random variables and correlations between their probability distributions. Those networks take a form of a directed acyclic graph, where vertices denote the variables and edges - correlations between them. Bayesian network is usually represented using set of multivariate distributions, which accounts for both variables distributions and their correlations. Typical applications of Bayesian networks include: calculating probability of a given variable values configuration or inferring probability distributions of given variables based on known states of other variables in the network. The variable connection types differ depending on their relation is classical or quantum.

Classical version - conditional dependence. In classical version, as in Fig 1, the variables in a Bayesian Network may be causally dependent on one other, which means their distributions are conditional. If distribution of variable V_2 is conditionally dependent on the distribution of variable V_1, a *happens-before* relationship between V_1 and V_2 is implied.

[3] https://github.com/sisl/BayesNets.jl

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_76

Quantum version - acausal relationships. When generalized to the quantum domain, Bayesian networks can also describe acausal relationships between variables, which may be interpreted as a quantum entanglement between them. As opposed to a causal relationship, such a system of variables is not described as one system at two times, but rather as two systems in a single time frame. An example of such a network is shown in Fig. 2.

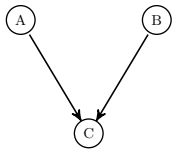

Fig. 1: A Bayesian network where there is a dependency of C on A and B (arrows). Source: [5]

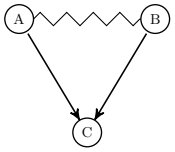

Fig. 2: States of systems A and B are entangled (zigzag line), and there is classic dependence of C on A and B (arrows). Source: [5]

Inference algorithms. Inference in Bayesian networks - calculating the probability distribution of a subsystem of variables based on already known variables - is an NP-hard problem [6]. However, there are algorithms which successfully approximate such computations with considerably lower computational complexity. `AcausalNets.jl` provides implementations of two algorithms for performing inference in Bayesian networks: a non-optimal naive algorithm, as well as the Belief Propagation algorithm [4,10]. Moreover, the second algorithm has been generalized to the quantum domain based on [7].

4 Design of the library

`AcausalNets.jl` API has been designed with simplicity of computations regarding discrete quantum probability systems in mind. It allows the user to perform inference in Bayesian Networks described in Section 3.

Defining Bayesian networks. To define a Bayesian network, one must first define systems of random variables which make up the network. Then a graph is built of those variables by inserting them in topological order. This requirement is essential, since Bayesian network is a directed acyclic graph. An example of the code which defines a network is shown in Fig. 3.

Performing inference. Inference can be performed with or without evidence - the known state of some of the random variables in the network. Known states are represented by a system with appropriate variables and their known distribution. Next, function *infer(network, variables, evidence, strategy)* is used for conducting inference for variables and evidence passed as arguments. By default, naive inference algorithm is used, but it can be changed in library's *Inference*

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_76

```
# 1.Defining variables
var_a = Variable(:a, 3) # location of the prize
var_b = Variable(:b, 3) # player's first choice
var_c = Variable(:c, 3) # host's choice

# 2.Defining system distribution for variable C
roCwAB = QuantumDistribution(diagm(0 =>[
0,1/2,1/2,  0,0,1,  0,1,0,  0,0,1,
1/2,0,1/2, 1,0,0, 1/2,1/2,0]))

# 3.Defining other systems using previously declared
# distribution and variables
sys_c_ab = DiscreteQuantumSystem( [var_a, var_b],[var_c],
    roCwAB)
sys_b= ...
sys_a=...
.....
# 4.Defining a Bayesian network
network = AcausalNet()
push!(network, sys_a)
push!(network, sys_b)
push!(network, sys_c_ab)
```

Fig. 3: Using `AcasualNets.jl` API to build a bayesian network analogous to Fig. 1

module. The example code for Monty Hall game use case [5] is shown in the Fig. 4.

```
# 5.Declaring evidence
ev = Evidence{QuantumDistribution}[
    Evidence{QuantumDistribution}(
        [var_b],  QuantumDistribution(ketbra(1,1,3))),
    Evidence{QuantumDistribution}(
       [var_c], QuantumDistribution(ketbra(3,3,3)))
    ]
# 6.Performing inference on created network
inferred = infer(network, [var_a], evidence)
```

Fig. 4: Using `AcasualNets.jl` API to perform Bayesian inference.

Usage of the Julia language. The software benefits strongly from Julia type-system [1], which allows to implement general operation on Bayesian Networks based on the type of the distributions the network is dealing with. It is possible because of Julia's type parametrization properties. More specifically, as shown in the Fig. 5, `BayesNet` type is parametrized with the type of `DiscreteSystem`, which specifies the math operations to be perform on probability distributions of the variables. For example, in case of `DiscreteQuantumSystem` such opera-

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_76

tions are performed in accordance with definitions of quantum conditional operators [8].

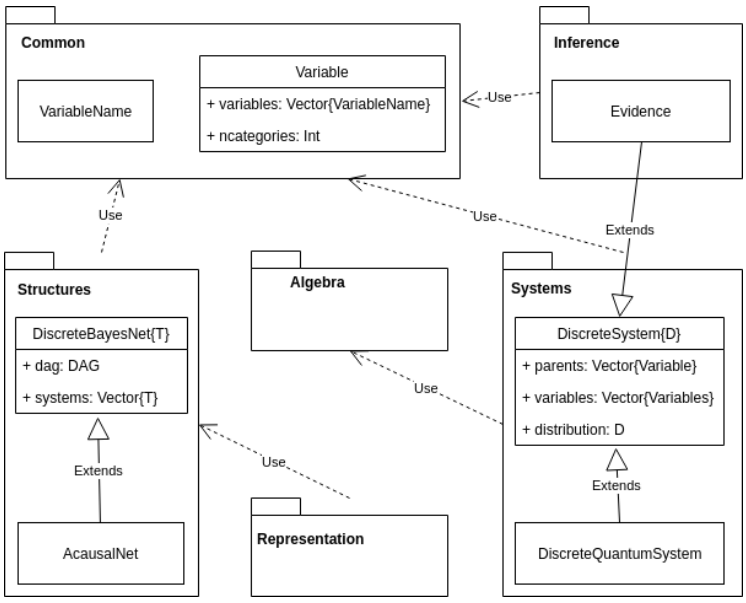

Fig. 5: The structure of `AcausalNets.jl` modules and types.

`AcausalNets.jl` has been implemented in version 1.0 of Julia language [1]. Source code is publicly available on [4]. The repository also contains more example use cases [5] in a convenient form of interactive Jupyter Notebooks [6].

5 Monty Hall game example use case

As a use case we use results of quantum Monty Hall game from [5] modeled as Bayesian network shown in Fig. 2. We consider a case where there occur quantum effects between the event A - the placement of the prize and B - the initial choice of the player. We choose two example quantum probability distributions. Eq.(1) models the situation where A and B are entangled in the way that due to quantum effects the placement of the prize always turns out to be the same as the initial choice of the player.

$$\rho_{AB\_same} = \frac{1}{3}(|00\rangle + |11\rangle + |22\rangle)(\langle 00| + \langle 11| + \langle 22|) \tag{1}$$

[4] `https://github.com/mikegpl/AcausalNets.jl`
[5] `https://github.com/mikegpl/AcausalNets.jl/tree/master/notebooks`
[6] `https://jupyter.org`

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_76

Eq.(2) models the situation where A and B are entangled in the way that due to quantum effects placement of the prize always turns out to be different that the initial choice of the player.

$$\rho_{AB\_diff} = \frac{1}{6}(|01\rangle + |10\rangle)(\langle 01| + \langle 10|) + \frac{1}{6}(|02\rangle + |20\rangle)(\langle 02| + \langle 20|) + \frac{1}{6}(|12\rangle + |21\rangle)(\langle 12| + \langle 21|) \tag{2}$$

Next, we construct linear combination of these two situations for $\lambda \in (0, 1)$:

$$\rho_{AB} = \lambda\rho_{AB\_same} + (1 - \lambda)\rho_{AB\_diff} \tag{3}$$

We aim to find how the probability of the player wining the game by staying with his initial choice depends on λ. We construct the Bayesian network as on Fig. 2 with an appropriate ρ_{AB} and perform inference to obtain the probabilities. We show our results in Fig. 6.We find that for $\lambda = 0.6$, the game is fair.

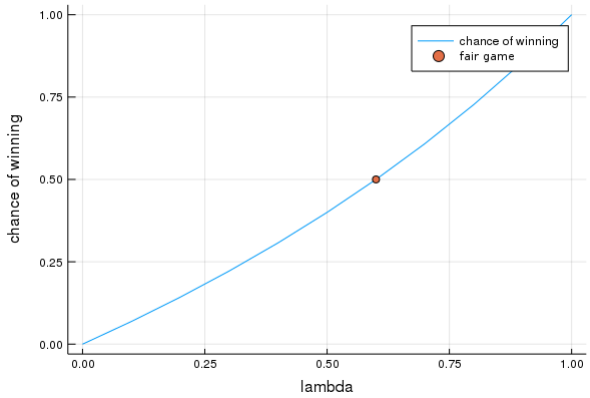

Fig. 6: Probability of winning the prize, when player does not change the door for the initial state given by $\rho_{AB} = \lambda\rho_{AB\_same} + (1 - \lambda)\rho_{AB\_diff}$

The above results show that using high level API of `AcausalNets.jl` library gives the same results as direct calculations presented in [5].

6 Conclusions and future work

High-level tools enable researchers to better organize, perform and document their experiments, although sometimes for the price of their flexibility.
`AcausalNets.jl` has been created with aim to abstract out tedious calculations, provide a high-level API for quantum Bayesian inference and leverage

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_76

the numerical potential of Julia language. We have successfuly used our library to reproduce the results of experiments on the quantum version of Monty Hall originally presented in [5]. Moreover, through automating a lot of necessary computations, `AcausalNets.jl` helps experiment with more complex Bayesian networks. In the future we plan to fully implement the quantum version of Belief Propagation algorithm and test its efficiency on bigger networks. This may lead to interesting new research results in quantum information as well as machine learning in the future.

Acknowledgements This research was partly supported by the National Science Centre, Poland – project number 2014/15/B/ST6/05204.

References

1. Bezanson, J., Edelman, A., Karpinski, S., Shah, V.: Julia: A fresh approach to numerical computing. SIAM Review **59**(1), 65–98 (2017). https://doi.org/10.1137/141000671, https://doi.org/10.1137/141000671
2. Eisert, J., Wilkens, M., Lewenstein, M.: Quantum games and quantum strategies. Phys. Rev. Lett. **83**, 3077–3080 (Oct 1999). https://doi.org/10.1103/PhysRevLett.83.3077, https://link.aps.org/doi/10.1103/PhysRevLett.83.3077
3. Gawron, P., Kurzyk, D., Pawela, Ł.: QuantumInformation.jl—a julia package for numerical computation in quantum information theory. PLOS ONE **13**(12), e0209358 (dec 2018). https://doi.org/10.1371/journal.pone.0209358, https://doi.org/10.1371/journal.pone.0209358
4. Huang, C., Darwiche, A.: Inference in belief networks: A procedural guide. International Journal of Approximate Reasoning **15**(3), 225 – 263 (1996). https://doi.org/https://doi.org/10.1016/S0888-613X(96)00069-2, http://www.sciencedirect.com/science/article/pii/S0888613X96000692
5. Kurzyk, D., Glos, A.: Quantum inferring acausal structures and the monty hall problem. Quantum Information Processing **15**(12), 4927–4937 (Dec 2016). https://doi.org/10.1007/s11128-016-1431-8, https://doi.org/10.1007/s11128-016-1431-8
6. Kwisthout, J.: Lecture notes : Computational complexity of bayesian networks (2015)
7. Leifer, M.S., Spekkens, R.W.: Towards a formulation of quantum theory as a causally neutral theory of bayesian inference. Phys. Rev. A **88**, 052130 (Nov 2013). https://doi.org/10.1103/PhysRevA.88.052130, https://link.aps.org/doi/10.1103/PhysRevA.88.052130
8. Leifer, M., Poulin, D.: Quantum graphical models and belief propagation. Annals of Physics **323**(8), 1899 – 1946 (2008). https://doi.org/https://doi.org/10.1016/j.aop.2007.10.001, http://www.sciencedirect.com/science/article/pii/S0003491607001509
9. Meyer, D.A.: Quantum strategies. Phys. Rev. Lett. **82**, 1052–1055 (Feb 1999). https://doi.org/10.1103/PhysRevLett.82.1052, https://link.aps.org/doi/10.1103/PhysRevLett.82.1052
10. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Exploring artificial intelligence in the new millennium. chap. Understanding Belief Propagation and Its Generalizations, pp. 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2003), http://dl.acm.org/citation.cfm?id=779343.779352

ICCS Camera Ready Version 2019
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-22750-0_76