Quantum Bayesian Inference

Michał Grabowski   Marcin Przewięźlikowski

Institute of Computer Science, AGH, al. Mickiewicza 30, 30-059 Kraków, Poland
{przewiez,mgrabow}@student.agh.edu.pl

May 29, 2019

Agenda

What are Bayesian Networks?

Classically

- graphical (directed, acyclic) probabilistic models
- **vertices** - random variables (events) with given probabilities of occurences
- **edges** - conditional dependence of variables on each other

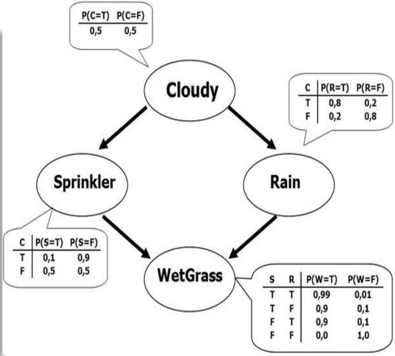


Figure: An example classical Bayesian Network

Use cases of Bayes Nets

Inference

- what is the chance of raining?
- what is the chance that my grass is going to be wet given it rains?
- what is the chance that it was cloudy given my grass is wet but my sprinkler wasn't working?

Structure and parameter learning (irrelevant to this talk)

Given historical data on clouds, rain, sprinkler activity and the wetness of grass I can learn things like:

- clouds cause rain (not the other way around)
- grass tends to be wet more often when it rains

The motivation - non-cooperative games

- multiple players compete with each other
- players employ strategies in order to win - those strategies must account for the rules of the game, as well as the opponents

Nash Equilibrium

No player can benefit by changing his strategy while the other players keep theirs unchanged.

More formally, if σ^* is a strategy profile being Nash equilibrium, then for every player i and every strategy σ_i, the payoff function for player i satisfies: $u_i(\sigma^*) \geq u_i(\sigma_i, \sigma^*_{-i})$

Example - Monty Hall game

- two players - Player and the Host
- three closed doors with a prize behind one of them
- after the Player's first choice of the door, the Host opens one of the remaining two doors, to reveal that it is empty
- the Player can then alter their original choice
- in **classical** case, the Player has $\frac{2}{3}$ chance of winning by altering their choice

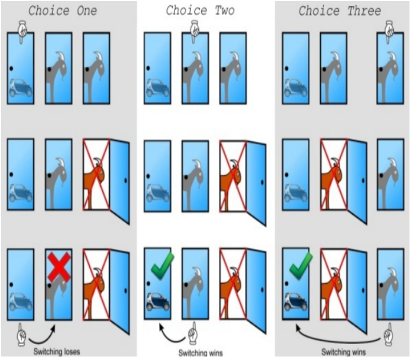


Figure: three choices the Player can make

Beyond classical strategies - quantum games

Research suggests that generalizing to quantum games may yield optimal strategies superior to the classical ones in games such as Monty Hall [Kurzyk and Glos, 2016] and Prisoner's Dilemma [Eisert et al., 1999, Szopa, 2013].

Quantum game

- is a generalization of its classical version and can be reduced to it
- uses concepts from quantum computing, unavailable in classical version
- yields results unavailable in its classical version

Bayesian Networks help us model games

- events in a game can be modeled as a Bayes network
- probability distributions of events reflect the players' strategies
- through Bayesian inference, various scenarios and outcomes of the game may be analyzed

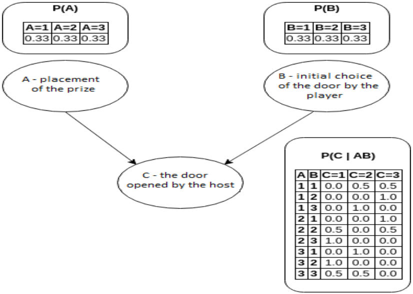


Figure: A classical Bayesian Network modelling the Monty Hall game

Quantum Bayesian networks

- possibility of an **acausal** connection - some of the variables in the network are entangled
- a **system** of entangled variables in the network must be described with joint probability distribution (in this example - ρ_{AB})
- distributions of participants of such systems *can* be calculated, but they don't tell the complete story

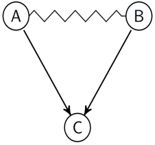


Figure: States of systems A and B are entangled (zigzag line), and there is classic dependence of C on A and B (arrows).

Quantum probability operators I

A classical distribution σ_A of a discrete random variable A can be generalized by a density operator $\rho_A \in L(H_A)$, where:

- $\rho_A = \rho_A^\dagger$
- $\rho_A \geq 0$, $Tr(\rho_a) = 1$
- $L(H)$ is a set of linear operators on complex Hilbert space H

Generalizing classical to quantum probability

- $\sigma_V = [p_0, p_1, ..., p_{n-1}]$
- p_i - probability of the i-th state
- $\sum_{i=0}^{N-1} p_i = 1$
- $|i\rangle$ - vector representing the i-th state
- $\rho_V = \sum_{i=1}^{N} p_i |i\rangle\langle i|$ - density matrix

Quantum probability operators II

Density matrices of quantum superpositions

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

$$|\alpha|^2 + |\beta|^2 = 1$$

$$\rho_\psi = |\psi\rangle\langle\psi| = \begin{pmatrix} |\alpha|^2 & \alpha\overline{\beta} \\ \overline{\alpha}\beta & |\beta|^2 \end{pmatrix}$$

Classical mixture of N quantum states

$|\psi_i\rangle$ - a quantum state achieved with probability p_i

$$\rho_\psi = \sum_{i=1}^{N} p_i|\psi_i\rangle\langle\psi_i|$$

Generalizing Bayesian theory to the quantum realm I

Joint probability distribution of a system of variables

- $\rho_{AB} \in L(H_{AB}) = L(H_A \otimes H_B)$ - joint distribution of (A, B)
- $\rho_A = Tr_B(\rho_{AB})$ - partial tracing

$\star$ operator

$$X \star Y = Y^{\frac{1}{2}} X Y^{\frac{1}{2}}$$

$$X, Y \in L(H)$$

non-commutative
non-associative

Conditionality

- B conditionally dependent on A
- $\rho_{B|A} \in L(H_{AB})$
- $Tr_B(\rho_{B|A}) = I_A$
- I_A - an identity operator $\in L(H_A)$
- $\rho_{AB} = \rho_{B|A} \star (\rho_A \otimes I_B)$

Generalizing Bayesian theory to the quantum realm II

Applying evidence

Knowledge that the actual state of A is $|a\rangle$ transforms ρ_{AB} into

$$\rho_{A=a,B} = \frac{(|a\rangle\langle a| \otimes I_B)\rho_{AB}(|a\rangle\langle a| \otimes I_B)}{(|a\rangle\langle a| \otimes I_B)\rho_{AB}} \quad (1)$$

Inference in Bayes Nets

- Inference in Bayesian Networks is an NP-hard problem [Kwisthout, 2015].

Naive algorithm

Given a Bayes Net with variables $[_1, \dots_n]$:

1. compute the joint probability of the whole net:

$$\rho_{V_1,\dots,V_n} = \star_{i=1}^{n}(I_1 \otimes I_2 \otimes \dots I_{i-1} \otimes \rho_i \otimes I_{i+1} \otimes \dots \otimes I_n) \quad (2)$$

2. apply evidence (if any) using (1)
3. trace the result to obtain the desired joint probability

Exponential memory complexity!

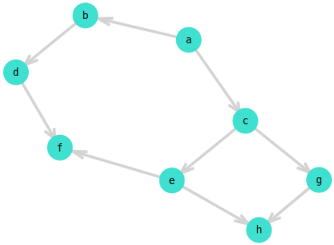
Belief Propagation algorithm I

1. network is converted to a tree structure ("Junction tree")
2. each vertex in a tree contains a subset of variables from the network with *partially* initialized probabilities
3. evidence is applied
4. vertices pass "messages" between each other in order to gain full knowledge about the state of the tree

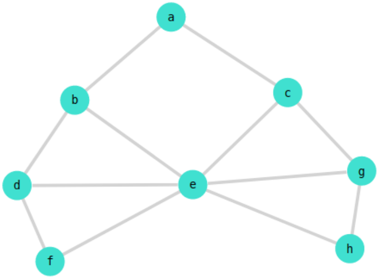
Steps (1) and (2) are identical regardless of whether the network is classical or quantum. In step (3) equations for messages with quantum states slightly differ in quantum version.

We are still researching a correct implementation of the quantum version of the algorithm.

Belief Propagation algorithm II

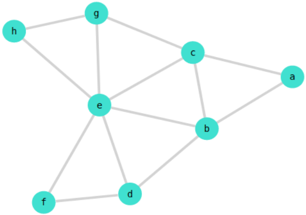


(a) A Bayes net

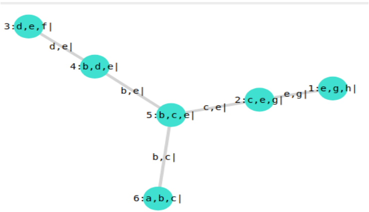


(b) Moralization of the graph and building a clique of the variables we're infering

Belief Propagation algorithm III



(c) Triangulation of the graph



(d) Clique of the triangulated graph become vertices of the junction tree

Figure: Building an optimal junction tree

Belief Propagation algorithm IV

Knowledge gathering by vertex W

- W updates the knowledge about its state by gathering "messages" from its neighbors in the tree

$$\rho_W = \frac{1}{Y} \cdot \mu_W \star \prod_{v \in neighbors(W)} m_{v \to W}$$

- $m_{v \to u}$ carries the knowledge about the state of variables shared by v and u from v's perspective

$$m_{v \to u} = \frac{1}{Y} Tr_u(\mu_v \star [(\prod_{v' \in neighbors(v)/u} m_{v' \to v}) \star \nu_{v:u}])$$

- knowledge is gathered recursively by the vertex containing the variables we want

AcausalNets.jl

- a Julia (1.0) library supporting inference in a quantum generalization of Bayesian networks
- available on GitHub:
  - https://github.com/mikegpl/AcausalNets.jl
- the repository also contains usage examples in a form of Jupyter Notebooks
  - https://github.com/mikegpl/AcausalNets.jl/tree/master/notebooks
- M. Przewiezlikowski, M. Grabowski, D. Kurzyk and K. Rycerz "Support for high-level quantum Bayesian inference" accepted for ICCS 2019 12-14 June, Faro, Portugal.

Implementation details I

Our goals

- simplicity of computations
- abstracting out as much math as possible
- extendability of the library

Functionalities

- defining random variables, systems of random variables
- building Bayesian networks as graphs with systems as vertices
- performing inference
  - naive algorithm works for all networks
  - we are still working on belief propagation for quantum networks

Implementation details II

Listing 1: Defining variables, systems and Bayesian network in `AcausalNets.jl`

```
# discrete variables
var_a = Variable(:a, 3)
var_b = Variable(:b, 3)
var_c = Variable(:c, 3)
# distributions
roA = diagm(0 => [1/3,1/3,1/3])
roB = diagm(0 => [1/3,1/3,1/3])
roCwAB = diagm(0 =>[0,1/2,1/2, 0,0,1, 0,1,0,
    0,0,1, 1/2,0,1/2, 1,0,0,
    0,1,0, 1,0,0, 1/2,1/2,0 ])
# variable systems
sys_a = DiscreteQuantumSystem([var_a], roA)
sys_b = DiscreteQuantumSystem([var_b], roB)
sys_c_ab = DiscreteQuantumSystem([var_a, var_b], [var_c], roCwAB)
# defining the network
an = AcausalNet()
push!(an, sys_ab)
push!(an, sys_c_ab)
```

Implementation details III

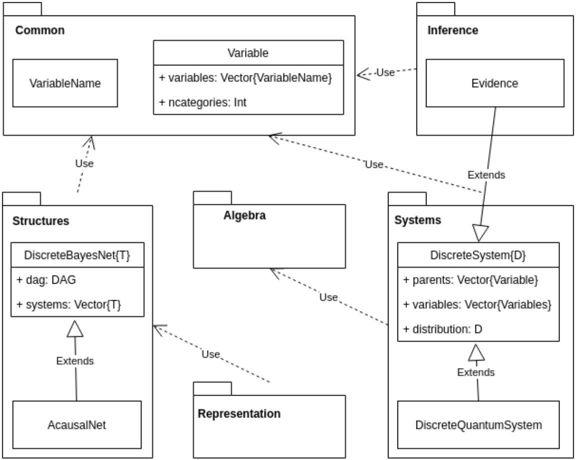


Figure: Submodules of `AcausalNets.jl`

More on the Julia language

Benefits

- a unique, well-designed approach to typing - dynamic, nominative, parametric
- methods overriding mechanism
- speed of the language, especially when it comes to numeric computations
- good documentation
- Jupyter Notebook support

Drawbacks

- the language has only recently achieved maturity (July 2018)
- third-party packages' developers don't always keep up with the development of the language
- relative unpopularity → small amount of development tools
- compiling slows development down

Experiments with Monty Hall I

- we use `AcausalNets.jl` to reproduce results of the research on quantum strategies in Monty Hall [Kurzyk and Glos, 2016]
- we aim to find a quantum state AB in which the Player and the Host both have 50% chance of winning - a Nash equilibrium
- for that purpose, we build appropriate Bayesian networks and analyze the probabilities of $|a\rangle = |c\rangle$ under various conditions
- `https://github.com/mikegpl/AcausalNets.jl/blob/master/notebooks/inferrer_monty_hall.ipynb`

Experiments with Monty Hall II

We consider the following quantum states of (A, B)

$$\tilde{\rho}_{AB} = \frac{1}{3}(|00\rangle + |11\rangle + |22\rangle)(\langle 00| + \langle 11| + \langle 22|) \tag{3}$$

$$\begin{aligned}\hat{\rho}_{AB} = &\frac{1}{6}(|01\rangle + |10\rangle)(\langle 01| + \langle 10|) \\ &+ \frac{1}{6}(|02\rangle + |20\rangle)(\langle 02| + \langle 20|) \\ &+ \frac{1}{6}(|12\rangle + |21\rangle)(\langle 12| + \langle 21|)\end{aligned} \tag{4}$$

$\tilde{\rho}_{AB}$

always $|a\rangle = |b\rangle$

$\hat{\rho}_{AB}$

always $|a\rangle \neq |b\rangle$

Experiments with Monty Hall III

What is the chance of Player winning if their choices are entangled with prize placement?

- $\lambda \in [0, 1]$
- we examine combinations $\lambda \tilde{\rho}_{AB} + (1 - \lambda)\hat{\rho}_{AB}$
- we look for a λ for which there is a 50% chance that $|a\rangle = |b\rangle$
- we assume prior knowledge of $|b\rangle, |c\rangle$ and infer $|a\rangle$

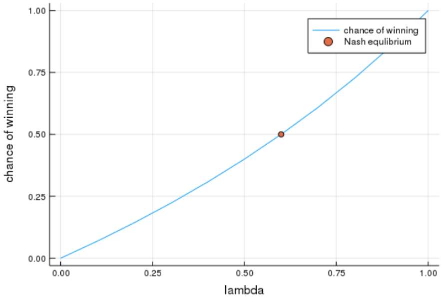


Figure: Probability of $|a\rangle = |b\rangle$ for various combinations $\lambda \tilde{\rho}_{AB} + (1 - \lambda)\hat{\rho}_{AB}$

Related work

- original paper with Monty Hall experiments we reproduced: [Kurzyk and Glos, 2016]
- examples of quantum games and strategies beyond Monty Hall: [Szopa, 2013, Eisert et al., 1999]
- classical Bayesian Networks and inference: [Huang and Darwiche, 1996, Yedidia et al., 2003]
- quantum Bayesian theory, generalization of inference: [Leifer and Poulin, 2008]
- Julia programming language: [Bezanson et al., 2017], `https://julialang.org/`

Summary

- employing quantum strategies in games can yield results impossible with the classical ones
- Bayesian networks are graphical probabilistic models which help model related random variables
  - among other use cases, they are useful for modeling events in games
  - in order to analyze quantum games, we need a generalization of Bayesian nets where acausal connections between variables are possible
- `AcausalNets.jl` is a Julia library which helps model Bayesian networks with acausal connections
- using `AcausalNets.jl`, we perform experiments with a quantum version of Monty Hall game

Bibliography I

J. Bezanson, A. Edelman, S. Karpinski, and V. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi: 10.1137/141000671. URL https://doi.org/10.1137/141000671.

Jens Eisert, Martin Wilkens, and Maciej Lewenstein. Quantum games and quantum strategies. *Phys. Rev. Lett.*, 83:3077–3080, Oct 1999. doi: 10.1103/PhysRevLett.83.3077. URL https://link.aps.org/doi/10.1103/PhysRevLett.83.3077.

Cecil Huang and Adnan Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15 (3):225 – 263, 1996. ISSN 0888-613X. doi: https://doi.org/10.1016/S0888-613X(96)00069-2. URL http://www.sciencedirect.com/science/article/pii/S0888613X96000692.

Bibliography II

Dariusz Kurzyk and Adam Glos. Quantum inferring acausal structures and the monty hall problem. *Quantum Information Processing*, 15 (12):4927–4937, Dec 2016. ISSN 1573-1332. doi: 10.1007/s11128-016-1431-8. URL https://doi.org/10.1007/s11128-016-1431-8.

Johan Kwisthout. Lecture notes : Computational complexity of bayesian networks. 2015.

M.S. Leifer and D. Poulin. Quantum graphical models and belief propagation. *Annals of Physics*, 323(8):1899 – 1946, 2008. ISSN 0003-4916. doi: https://doi.org/10.1016/j.aop.2007.10.001. URL http://www.sciencedirect.com/science/article/pii/S0003491607001509.

Marek Szopa. Dlaczego w dylemat więźnia warto grać kwantowo? 01 2013.

Bibliography III

Jonathan S. Yedidia, William T. Freeman, and Yair Weiss. Exploring artificial intelligence in the new millennium. chapter Understanding Belief Propagation and Its Generalizations, pages 239–269. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2003. ISBN 1-55860-811-7. URL http://dl.acm.org/citation.cfm?id=779343.779352.