Noisy logical connectives in Bayesian networks

Jirka Vomlel

Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
http://www.utia.cz/vomlel

Vienna, 26–28 November 2010

Outline

- Brief introduction to Causal Probabilistic (CP) logic

Outline

- Brief introduction to Causal Probabilistic (CP) logic
- Converting a CP-theory to a Bayesian network

Outline

- Brief introduction to Causal Probabilistic (CP) logic
- Converting a CP-theory to a Bayesian network
- Efficient probabilistic inference with Bayesian networks

A story used throughout the presentation (Meert et al., 2008)

Take a person, named *John*, who may go to the shop to buy dinner. The probability he does is 20%. He chooses to buy either *spaghetti* or *steak*, each with 50% chance.

A story used throughout the presentation (Meert et al., 2008)

Take a person, named *John*, who may go to the shop to buy dinner. The probability he does is 20%. He chooses to buy either *spaghetti* or *steak*, each with 50% chance.

Johns girlfriend *Mary* may also buy dinner. We assume that she cannot contact John during the day. The probability she goes to the shop is 90%. If she goes to the shop she buys either *spaghetti* with probability 30% or *fish* with probability 70%.

A story used throughout the presentation (Meert et al., 2008)

Take a person, named *John*, who may go to the shop to buy dinner. The probability he does is 20%. He chooses to buy either *spaghetti* or *steak*, each with 50% chance.

Johns girlfriend *Mary* may also buy dinner. We assume that she cannot contact John during the day. The probability she goes to the shop is 90%. If she goes to the shop she buys either *spaghetti* with probability 30% or *fish* with probability 70%.

If *John* and *Mary* both buy dinner, it is possible that they both buy *spaghetti*. If they buy something different, they can choose what they will have for dinner, because two meals have been bought.

Necessary notation from predicate logic

Ground terms will be just constant symbols, e.g. *john*, *mary*, *spaghetti*, *steak*, *fish*.

Necessary notation from predicate logic

Ground terms will be just constant symbols, e.g. *john*, *mary*, *spaghetti*, *steak*, *fish*.

Predicate symbols of arity n. E.g. $shops/1$, $bought/1$ will be predicate symbols of arity 1.

Necessary notation from predicate logic

Ground terms will be just constant symbols, e.g. *john*, *mary*, *spaghetti*, *steak*, *fish*.

Predicate symbols of arity n. E.g. $shops/1$, $bought/1$ will be predicate symbols of arity 1.

If p is a predicate symbol of arity n and $a_1, \ldots, a_n$ are ground terms then $p(a_1, \ldots, a_n)$ is a ground atom. E.g., $shops(paul)$, $bought(spaghetti)$.

Necessary notation from predicate logic

Ground terms will be just constant symbols, e.g. *john*, *mary*, *spaghetti*, *steak*, *fish*.

Predicate symbols of arity n. E.g. $shops/1$, $bought/1$ will be predicate symbols of arity 1.

If p is a predicate symbol of arity n and $a_1, \ldots, a_n$ are ground terms then $p(a_1, \ldots, a_n)$ is a ground atom. E.g., $shops(paul)$, $bought(spaghetti)$.

Herbrand base is the set of all ground atoms.

Necessary notation from predicate logic

Ground terms will be just constant symbols, e.g. *john*, *mary*, *spaghetti*, *steak*, *fish*.

Predicate symbols of arity n. E.g. $shops/1$, $bought/1$ will be predicate symbols of arity 1.

If p is a predicate symbol of arity n and $a_1, \ldots, a_n$ are ground terms then $p(a_1, \ldots, a_n)$ is a ground atom. E.g., $shops(paul)$, $bought(spaghetti)$.

Herbrand base is the set of all ground atoms.

Herbrand interpretation is a subset of the Herbrand base that is true.

Causal Probabilistic (CP) Logic (Vennekens, 2007)

Definition

A rule is the statement

$$(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n) \quad \leftarrow \quad \varphi \ ,$$

where φ is a conjuction of ground atoms or their negations, the p_i are ground atoms and the α_i are non-zero probabilities with $\sum \alpha_i \leq 1$.

Causal Probabilistic (CP) Logic (Vennekens, 2007)

Definition

A rule is the statement

$$(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n) \quad \leftarrow \quad \varphi \ ,$$

where φ is a conjuction of ground atoms or their negations, the p_i are ground atoms and the α_i are non-zero probabilities with $\sum \alpha_i \leq 1$.

- $(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n)$ is the head of the rule and

Causal Probabilistic (CP) Logic (Vennekens, 2007)

Definition

A rule is the statement

$$(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n) \quad \leftarrow \quad \varphi \ ,$$

where φ is a conjuction of ground atoms or their negations, the p_i are ground atoms and the α_i are non-zero probabilities with $\sum \alpha_i \leq 1$.

- $(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n)$ is the head of the rule and
- φ is the body.

Causal Probabilistic (CP) Logic (Vennekens, 2007)

Definition

A rule is the statement

$$(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n) \quad \leftarrow \quad \varphi \ ,$$

where φ is a conjuction of ground atoms or their negations, the p_i are ground atoms and the α_i are non-zero probabilities with $\sum \alpha_i \leq 1$.

- $(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n)$ is the head of the rule and
- φ is the body.

Property φ causes an event, whose effect is that it makes at most one of the properties p_i becomes true, and for each p_i, the probability of it being caused by this event is α_i.

Causal Probabilistic (CP) Logic (Vennekens, 2007)

Definition

A rule is the statement

$$(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n) \quad \leftarrow \quad \varphi \ ,$$

where φ is a conjuction of ground atoms or their negations, the p_i are ground atoms and the α_i are non-zero probabilities with $\sum \alpha_i \leq 1$.

- $(p_1 : \alpha_1) \vee \ldots \vee (p_n : \alpha_n)$ is the head of the rule and
- φ is the body.

Property φ causes an event, whose effect is that it makes at most one of the properties p_i becomes true, and for each p_i, the probability of it being caused by this event is α_i.

Example

$$(bought(spaghetti) : 0.3) \vee (bought(fish) : 0.7) \quad \leftarrow \quad shops(mary)$$

CP-theory

Definition

CP-theory is a set of rules.

CP-theory

Definition

CP-theory is a set of rules.

Example

$$
\begin{aligned}
\mathit{shops}(\mathit{john}) : 0.2 &\leftarrow \cdot \\
\mathit{shops}(\mathit{mary}) : 0.9 &\leftarrow \cdot \\
(\mathit{bought}(\mathit{spaghetti}) : 0.5) \vee (\mathit{bought}(\mathit{steak}) : 0.5) &\leftarrow \mathit{shops}(\mathit{john}) \\
(\mathit{bought}(\mathit{spaghetti}) : 0.3) \vee (\mathit{bought}(\mathit{fish}) : 0.7) &\leftarrow \mathit{shops}(\mathit{mary})
\end{aligned}
$$

Semantics of CP-logic

- The semantics of a CP-theory is defined by its execution model.

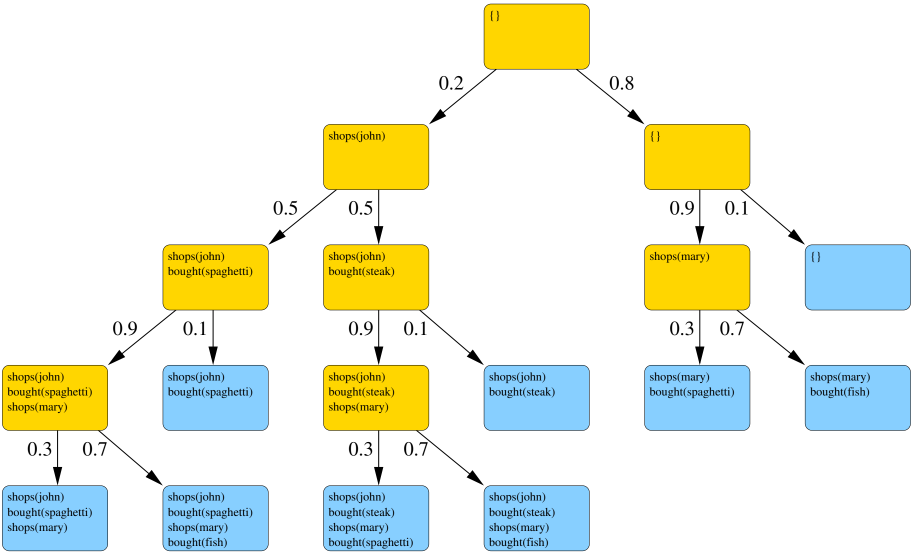
Semantics of CP-logic

- The semantics of a CP-theory is defined by its execution model.
- Execution model of a CP-theory is a probabilistic process.

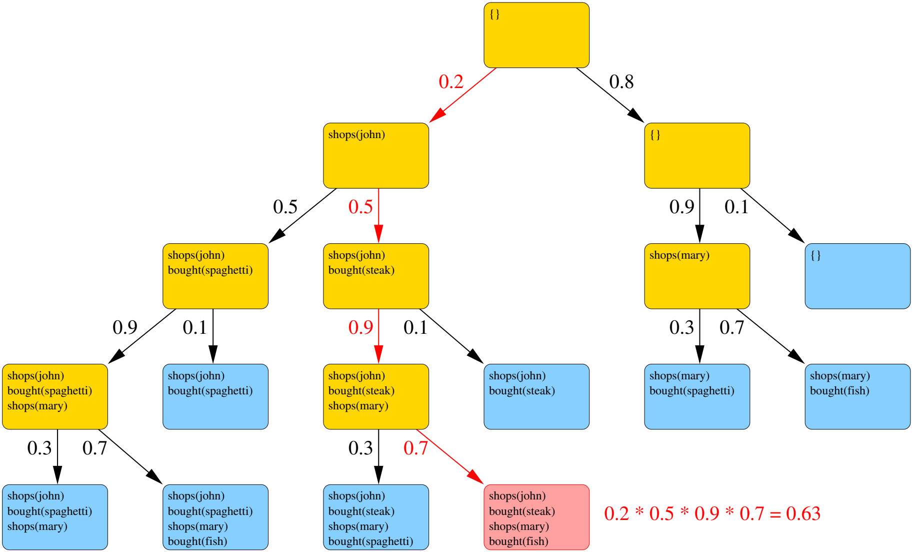
Semantics of CP-logic

- The semantics of a CP-theory is defined by its execution model.
- Execution model of a CP-theory is a probabilistic process.
- Instead of definition, which is quite technical, we provide an example of an execution model.

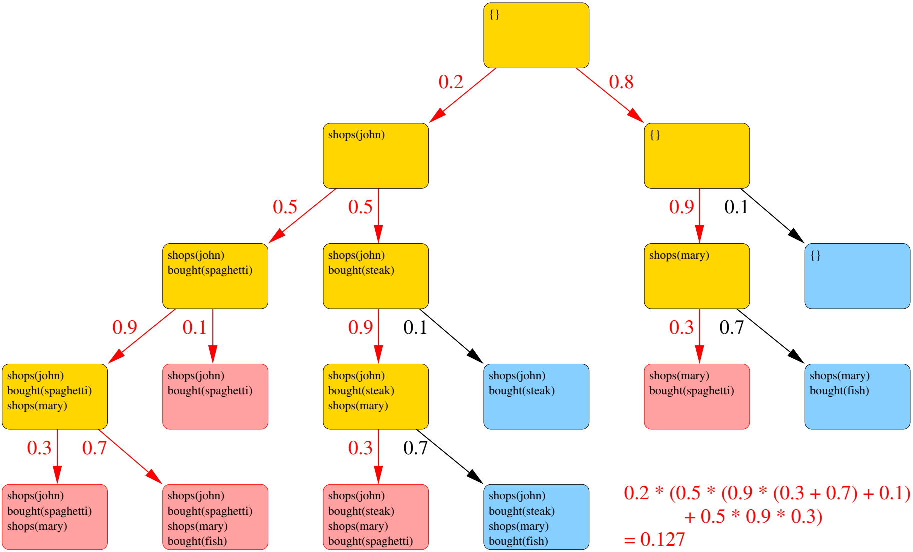
An execution model of a CP-theory



Probability of getting to a leaf



Marginal probability $P(bought(spaghetti))$



$$0.2 * (0.5 * (0.9 * (0.3 + 0.7) + 0.1) + 0.5 * 0.9 * 0.3) = 0.127$$

Construction of the Bayesian network (BN) of a CP-theory (Meert et al., 2008)

1. For every atom in the CP-theory, a Boolean variable is created.

Construction of the Bayesian network (BN) of a CP-theory (Meert et al., 2008)

1. For every atom in the CP-theory, a Boolean variable is created.
2. For every rule in the CP-theory, a choice variable is created. This variable can take $n+1$ values, where n is the number of atoms in the head.

Construction of the Bayesian network (BN) of a CP-theory (Meert et al., 2008)

1. For every atom in the CP-theory, a Boolean variable is created.
2. For every rule in the CP-theory, a choice variable is created. This variable can take $n+1$ values, where n is the number of atoms in the head.
3. Note that the choice variables are unobserved, they are auxiliary nodes that do not correspond to atoms in the domain.

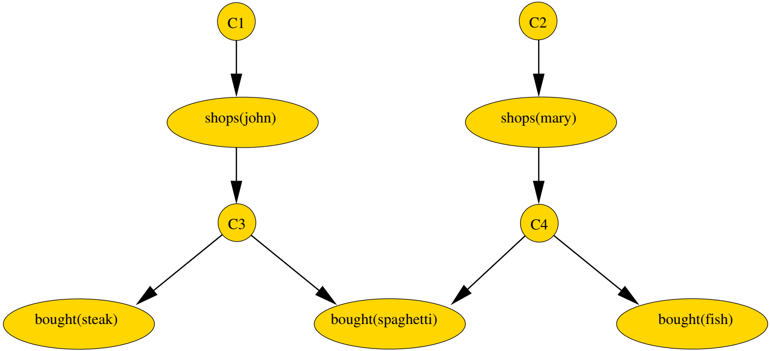
Construction of the Bayesian network (BN) of a CP-theory (Meert et al., 2008)

1. For every atom in the CP-theory, a Boolean variable is created.
2. For every rule in the CP-theory, a choice variable is created. This variable can take $n + 1$ values, where n is the number of atoms in the head.
3. Note that the choice variables are unobserved, they are auxiliary nodes that do not correspond to atoms in the domain.
4. If an atom is in the head of a rule, an edge is created from its corresponding choice node towards the atom's node.

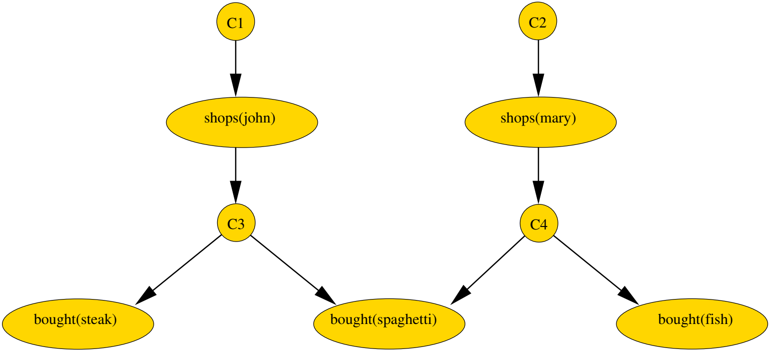
Construction of the Bayesian network (BN) of a CP-theory (Meert et al., 2008)

1. For every atom in the CP-theory, a Boolean variable is created.
2. For every rule in the CP-theory, a choice variable is created. This variable can take $n+1$ values, where n is the number of atoms in the head.
3. Note that the choice variables are unobserved, they are auxiliary nodes that do not correspond to atoms in the domain.
4. If an atom is in the head of a rule, an edge is created from its corresponding choice node towards the atom's node.
5. If a literal is in the body of a rule, an edge is created from the literal's atom node towards the rules choice node.

The BN of a CP-theory - Example

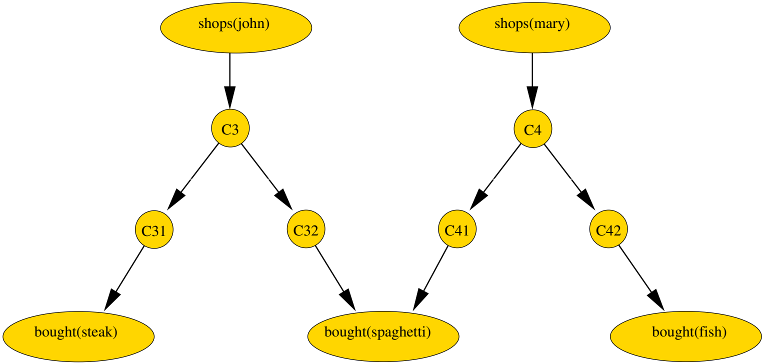


The BN of a CP-theory - Example

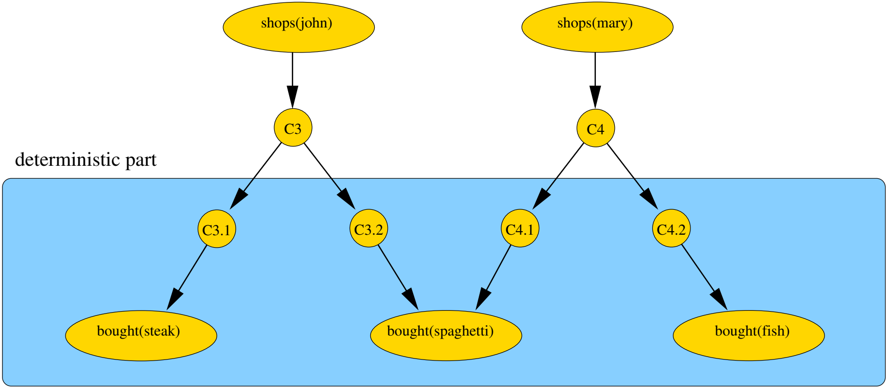


For complete specification of the BN
(including numerical parameters in conditional probability tables)
see `shopping-example-orig.net` in Hugin Light.

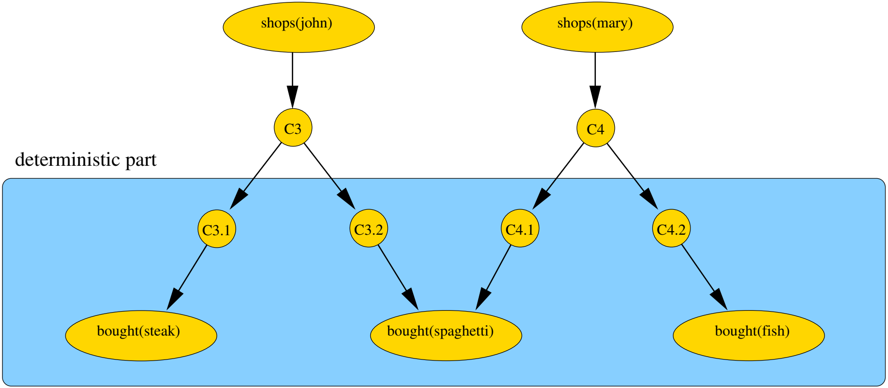
Modified BN of a CP-theory - Example



Modified BN of a CP-theory - Example

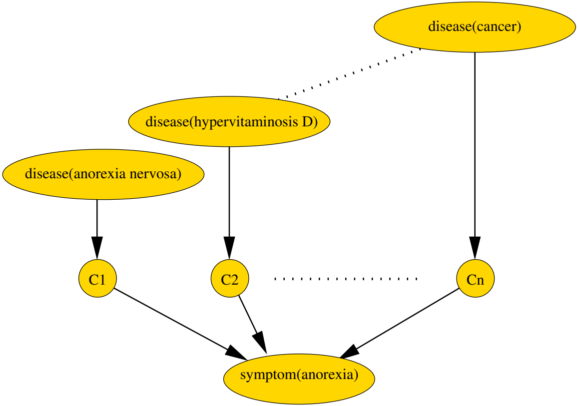


Modified BN of a CP-theory - Example

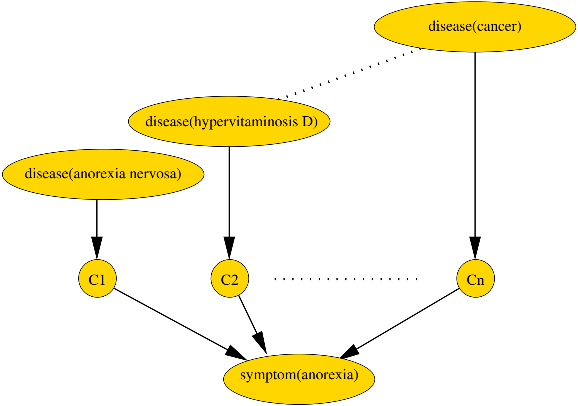


For modified BN and computations with this BN see `shopping-example.net` in Hugin Light. Compare behavior with evidence $bought(spaghetti)$ and $\neg bought(spaghetti)$.

A part of a BN with the OR function



A part of a BN with the OR function



A well-known model often called noisy-OR.

Efficient probabilistic inference in BNs containing noisy-OR

Probabilistic inference is the computation of $P(X_i|\mathbf{e})$ for all variables X_i of the BN where $\mathbf{e}$ stands for evidence collected so far.

Efficient probabilistic inference in BNs containing noisy-OR

Probabilistic inference is the computation of $P(X_i|\mathbf{e})$ for all variables X_i of the BN where $\mathbf{e}$ stands for evidence collected so far.

A standard method for probabilistic inference is the junction tree method which exploits the global structure of the model, specifically, conditional independence among variables.

Efficient probabilistic inference in BNs containing noisy-OR

Probabilistic inference is the computation of $P(X_i|\mathbf{e})$ for all variables X_i of the BN where $\mathbf{e}$ stands for evidence collected so far.

A standard method for probabilistic inference is the junction tree method which exploits the global structure of the model, specifically, conditional independence among variables.

Additionally, the local structure of noisy-OR can be exploited as well to perform computations more efficiently.

Efficient probabilistic inference in BNs containing noisy-OR

Probabilistic inference is the computation of $P(X_i|\mathbf{e})$ for all variables X_i of the BN where $\mathbf{e}$ stands for evidence collected so far.

A standard method for probabilistic inference is the junction tree method which exploits the global structure of the model, specifically, conditional independence among variables.

Additionally, the local structure of noisy-OR can be exploited as well to perform computations more efficiently.

- Parent divorcing method (Olesen et al., 1989).

Efficient probabilistic inference in BNs containing noisy-OR

Probabilistic inference is the computation of $P(X_i|\mathbf{e})$ for all variables X_i of the BN where $\mathbf{e}$ stands for evidence collected so far.

A standard method for probabilistic inference is the junction tree method which exploits the global structure of the model, specifically, conditional independence among variables.

Additionally, the local structure of noisy-OR can be exploited as well to perform computations more efficiently.

- Parent divorcing method (Olesen et al., 1989).
- Quickscore (Heckerman, 1990).

Efficient probabilistic inference in BNs containing noisy-OR

Probabilistic inference is the computation of $P(X_i|\mathbf{e})$ for all variables X_i of the BN where $\mathbf{e}$ stands for evidence collected so far.

A standard method for probabilistic inference is the junction tree method which exploits the global structure of the model, specifically, conditional independence among variables.

Additionally, the local structure of noisy-OR can be exploited as well to perform computations more efficiently.

- Parent divorcing method (Olesen et al., 1989).
- Quickscore (Heckerman, 1990).
- Factorization of noisy-MAX (Díez & Galán, 2003).

Efficient probabilistic inference in BNs containing noisy-OR

Probabilistic inference is the computation of $P(X_i|\mathbf{e})$ for all variables X_i of the BN where $\mathbf{e}$ stands for evidence collected so far.

A standard method for probabilistic inference is the junction tree method which exploits the global structure of the model, specifically, conditional independence among variables.

Additionally, the local structure of noisy-OR can be exploited as well to perform computations more efficiently.

- Parent divorcing method (Olesen et al., 1989).
- Quickscore (Heckerman, 1990).
- Factorization of noisy-MAX (Díez & Galán, 2003).
- Rank-one decomposition (Savický & Vomlel, 2007).

Comparison of ROD and PD

Comparison of the size of arithmetic circuits used for probabilistic inference in BN2O networks after:

- the parent divorcing method (horizontal axis)

Comparison of ROD and PD

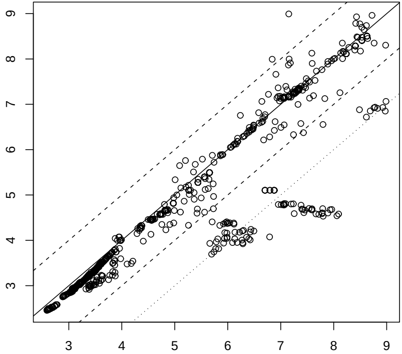
Comparison of the size of arithmetic circuits used for probabilistic inference in BN2O networks after:

- the parent divorcing method (horizontal axis)
- the rank-one decomposition (vertical axis).

Comparison of ROD and PD

Comparison of the size of arithmetic circuits used for probabilistic inference in BN2O networks after:

- the parent divorcing method (horizontal axis)
- the rank-one decomposition (vertical axis).



Decadic logarithm scale is used.

Current and future work

Current work

Current and future work

Current work

- Graph triangulation methods for BN2O networks.

Current and future work

Current work

- Graph triangulation methods for BN2O networks.
- The rank-one decomposition of tables of ℓ-out-of-k functions.

Current and future work

Current work

- Graph triangulation methods for BN2O networks.
- The rank-one decomposition of tables of ℓ-out-of-k functions.

Future work

Current and future work

Current work

- Graph triangulation methods for BN2O networks.
- The rank-one decomposition of tables of ℓ-out-of-k functions.

Future work

- Aproximate inference based on rank-one decomposition.

Current and future work

Current work

- Graph triangulation methods for BN2O networks.
- The rank-one decomposition of tables of ℓ-out-of-k functions.

Future work

- Aproximate inference based on rank-one decomposition.
- An application in medicine.