Generalized Loopy 2U: A New Algorithm for Approximate Inference in Credal Networks

Alessandro Antonucci, Marco Zaffalon, Sun Yi, Cassio de Campos

IDSIA

Lugano, Switzerland

PGM ’08 - Hirtshals

September 19th, 2008

Imprecise probabilities (Walley, 1991)

Models of uncertainty about the state of a categorical variable X

- A probability mass function $P(X)$
- More generally, a *closed convex set* of probability mass functions $K(X)$ This is a *credal set* (Levi, 1980)
- Complete ignorance? A *vacuous* credal set $K_0(X)$
- Lower (and upper) expectation

  $\underline{E}_K[f(X)] = \inf_{P(X) \in K(X)} \sum_X P(x) f(X)$

Imprecise probabilities (Walley, 1991)

Models of uncertainty about the state of a categorical variable X

- A probability mass function $P(X)$
- More generally, a *closed convex set* of probability mass functions $K(X)$
  This is a *credal set* (Levi, 1980)
- Complete ignorance?
  A *vacuous* credal set $K_0(X)$
- Lower (and upper) expectation

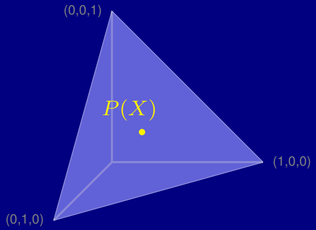
  $\underline{E}_K[f(X)] = \inf_{P(X) \in K(X)} \sum_X P(x) f(X)$

Imprecise probabilities (Walley, 1991)

Models of uncertainty about the state of a categorical variable X

- A probability mass function $P(X)$
- More generally, a *closed convex set* of probability mass functions $K(X)$
  This is a *credal set* (Levi, 1980)
- Complete ignorance?
  A *vacuous* credal set $K_0(X)$
- Lower (and upper) expectation

  $\underline{E}_K[f(X)] = \inf_{P(X) \in K(X)} \sum_X P(x) f(X)$

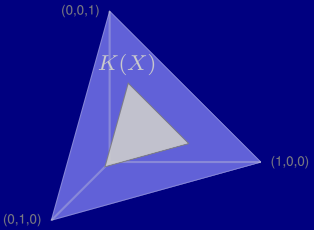


Imprecise probabilities (Walley, 1991)

Models of uncertainty about the state of a categorical variable X

- A probability mass function $P(X)$
- More generally, a *closed convex set* of probability mass functions $K(X)$
  This is a *credal set* (Levi, 1980)
- Complete ignorance?
  A *vacuous* credal set $K_0(X)$
- Lower (and upper) expectation

  $\underline{E}_K[f(X)] = \inf_{P(X) \in K(X)} \sum_X P(x) f(X)$

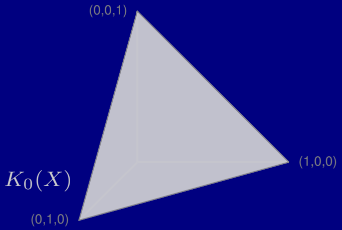


Imprecise probabilities (Walley, 1991)

Models of uncertainty about the state of a categorical variable X

- A probability mass function $P(X)$
- More generally, a *closed convex set* of probability mass functions $K(X)$
  This is a *credal set* (Levi, 1980)
- Complete ignorance?
  A *vacuous* credal set $K_0(X)$
- Lower (and upper) expectation

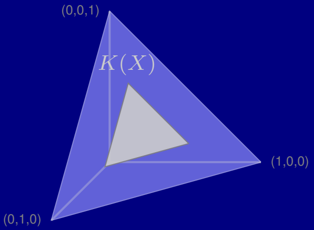
$$\underline{E}_K[f(X)] = \inf_{P(X)\in K(X)} \sum_X P(x) f(X)$$



Imprecise probabilities (Walley, 1991)

Models of uncertainty about the state of a categorical variable X

- A probability mass function $P(X)$
- More generally, a *closed convex set* of probability mass functions $K(X)$
  This is a *credal set* (Levi, 1980)
- Complete ignorance?
  A *vacuous* credal set $K_0(X)$
- Lower (and upper) expectation
  $\underline{E}_K[f(X)] = \inf_{P(X)\in K(X)} \sum_X P(x) f(X)$



From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1,\ldots,x_n)=\prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- ...?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1,\ldots,x_n)=\prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- . . . ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1, \ldots, X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- . . . ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions
  $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function
  $P(x_1,\ldots,x_n) = \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$
  NP-hard (Cooper, 1989)
- BP efficiently updates polytrees
  (Pearl, 1988)
- Loopy BP for multi-connected
  (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets
  $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension)
  $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$
  $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees
  (Zaffalon, 1998)
- Loopy 2U for multi-connected binary
  (Ide & Cozman, 2002)
- . . . ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1, \ldots, X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- $\ldots$?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1,\ldots,x_n)=\prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- … ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1, \ldots, X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- $\ldots$?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1, \ldots, X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- $\ldots$?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions
  $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function
  $P(x_1,\ldots,x_n)=\prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$
  NP-hard (Cooper, 1989)
- BP efficiently updates polytrees
  (Pearl, 1988)
- Loopy BP for multi-connected
  (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets
  $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension)
  $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$
  $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees
  (Zaffalon, 1998)
- Loopy 2U for multi-connected binary
  (Ide & Cozman, 2002)
- … ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1,\ldots,x_n)=\prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- …?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1,\ldots,x_n)=\prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- $\ldots$?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions
  $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function
  $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$
  NP-hard (Cooper, 1989)
- BP efficiently updates polytrees
  (Pearl, 1988)
- Loopy BP for multi-connected
  (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets
  $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension)
  $K(X_1, \ldots, X_n)$
- updating = compute $\underline{P}(X_q|x_E)$
  $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees
  (Zaffalon, 1998)
- Loopy 2U for multi-connected binary
  (Ide & Cozman, 2002)
- …?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1,\ldots,x_n)=\prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- . . . ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1,\ldots,x_n)=\prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- . . . ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1,\ldots,x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1,\ldots,X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- . . . ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1, \ldots, X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- . . . ?

Updating non-binary credal nets?

From Bayesian to credal nets

Bayesian nets (Pearl, 1988)

- (stochastic) independence by a DAG
- conditional mass functions $P(X_i|\mathrm{pa}(X_i))$
- joint probability mass function $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- updating = compute $P(X_q|x_E)$ NP-hard (Cooper, 1989)
- BP efficiently updates polytrees (Pearl, 1988)
- Loopy BP for multi-connected (Murphy, 1999)

Credal nets (Cozman, 2000)

- strong independence by a DAG
- conditional credal sets $K(X_i|\mathrm{pa}(X_i))$
- joint credal set (strong extension) $K(X_1, \ldots, X_n)$
- updating = compute $\underline{P}(X_q|x_E)$ $\mathrm{NP}^{\mathrm{PP}}$-hard (Campos & Cozman, 2005)
- 2U: fast alg for binary polytrees (Zaffalon, 1998)
- Loopy 2U for multi-connected binary (Ide & Cozman, 2002)
- . . . ?

Updating non-binary credal nets?

Two main results

Theorem (about representation)

“Every credal net can be **equivalently represented** as a credal net over **binary** variables”
(and the transformation takes only polynomial time)

Corollary (about inference)

“Algorithms for binary credal nets can be **applied to credal nets of any kind**”
(loopy 2U can update credal nets of any kind)

Two main results

Theorem (about representation)

"Every credal net can be **equivalently represented**
as a credal net over **binary** variables"
(and the transformation takes only polynomial time)

Corollary (about inference)

"Algorithms for binary credal nets can be
applied to credal nets of any kind"
(loopy 2U can update credal nets of any kind)

Two main results

Theorem (about representation)

"Every credal net can be **equivalently represented**
as a credal net over **binary** variables"
(and the transformation takes only polynomial time)

Corollary (about inference)

"Algorithms for binary credal nets can be
applied to credal nets of any kind"
(loopy 2U can update credal nets of any kind)

Binarization (graph)

- Nodes binarization
  - State of a variable as a joint state of a number of "bits"
    $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \dots$
- Arcs binarization
  - For each arc between two variables, all the relative bits are linked
  - The bits of the same variable are completely connected

Binarization (graph)

- Nodes binarization
  - State of a variable as a joint state of a number of "bits"

    $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \dots$

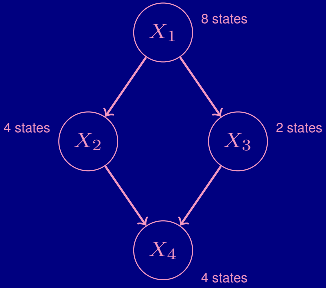
- Arcs binarization
  - For each arc between two variables, all the relative bits are linked
  - The bits of the same variable are completely connected

Binarization (graph)

- Nodes binarization
  - State of a variable as a joint state of a number of “bits”

    $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \dots$

- Arcs binarization
  - For each arc between two variables, all the relative bits are linked
  - The bits of the same variable are completely connected

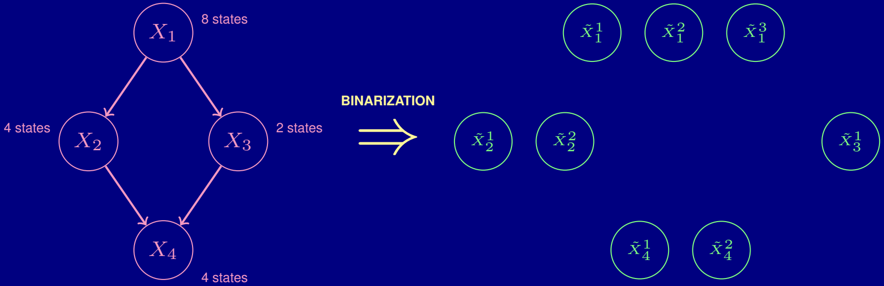


Binarization (graph)

- Nodes binarization
  - State of a variable as a joint state of a number of "bits"

    $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$

- Arcs binarization
  - For each arc between two variables, all the relative bits are linked
  - The bits of the same variable are completely connected

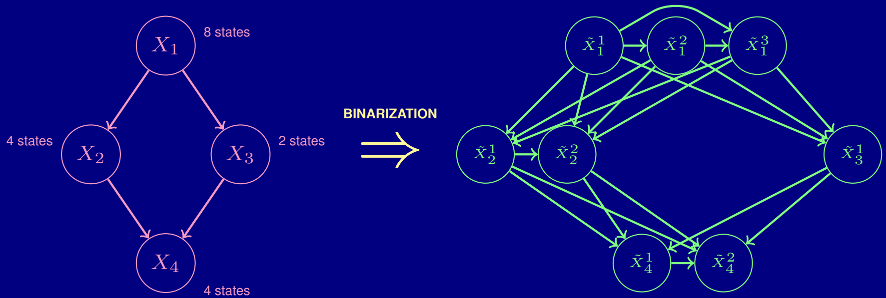


Binarization (graph)

- Nodes binarization
  - State of a variable as a joint state of a number of "bits"

    $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$

- Arcs binarization
  - For each arc between two variables, all the relative bits are linked
  - The bits of the same variable are completely connected



Binarization (probabilities)

- Bayesian nets
  - All the conditional mass function $P(\tilde{X}_i^j|\mathrm{pa}(\tilde{X}_i^j))$ of the bits of X_i can be computed from the conditional mass function $P(X_i|\mathrm{pa}(X_i))$

$$P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) \propto \sum'_{x_i} P(x_i|\mathrm{pa}(X_i))$$

- Credal nets
  - Same calculations iterated over the conditional credal set $K(X_i|\pi_i)$

$$\underline{P}(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) = \inf_{P(X_i|\mathrm{pa}(X_i))\in K(X_i|\mathrm{pa}(X_i))} P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j))$$

- A “binarized” Bayesian/credal net is obtained
- The transformation takes only linear time!

Binarization (probabilities)

- Bayesian nets
  - All the conditional mass function $P(\tilde{X}_i^j|\mathrm{pa}(\tilde{X}_i^j))$ of the bits of X_i can be computed from the conditional mass function $P(X_i|\mathrm{pa}(X_i))$

$$P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) \propto {\sum}'_{x_i} P(x_i|\mathrm{pa}(X_i))$$

- Credal nets
  - Same calculations iterated over the conditional credal set $K(X_i|\pi_i)$

$$\underline{P}(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) = \inf_{P(X_i|\mathrm{pa}(X_i)) \in K(X_i|\mathrm{pa}(X_i))} P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j))$$

- A “binarized” Bayesian/credal net is obtained
- The transformation takes only linear time!

Binarization (probabilities)

- Bayesian nets
  - All the conditional mass function $P(\tilde{X}_i^j|\mathrm{pa}(\tilde{X}_i^j))$ of the bits of X_i can be computed from the conditional mass function $P(X_i|\mathrm{pa}(X_i))$

$$P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) \propto \sum'_{x_i} P(x_i|\mathrm{pa}(X_i))$$

- Credal nets
  - Same calculations iterated over the conditional credal set $K(X_i|\pi_i)$

$$\underline{P}(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) = \inf_{P(X_i|\mathrm{pa}(X_i))\in K(X_i|\mathrm{pa}(X_i))} P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j))$$

- A “binarized” Bayesian/credal net is obtained
- The transformation takes only linear time!

Binarization (probabilities)

- Bayesian nets
  - All the conditional mass function $P(\tilde{X}_i^j|\mathrm{pa}(\tilde{X}_i^j))$ of the bits of X_i can be computed from the conditional mass function $P(X_i|\mathrm{pa}(X_i))$

$$P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) \propto \sum'_{x_i} P(x_i|\mathrm{pa}(X_i))$$

- Credal nets
  - Same calculations iterated over the conditional credal set $K(X_i|\pi_i)$

$$\underline{P}(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) = \inf_{P(X_i|\mathrm{pa}(X_i))\in K(X_i|\mathrm{pa}(X_i))} P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j))$$

- A “binarized” Bayesian/credal net is obtained
- The transformation takes only linear time!

Binarization (probabilities)

- Bayesian nets
  - All the conditional mass function $P(\tilde{X}_i^j|\mathrm{pa}(\tilde{X}_i^j))$ of the bits of X_i can be computed from the conditional mass function $P(X_i|\mathrm{pa}(X_i))$

$$P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) \propto \sum'_{x_i} P(x_i|\mathrm{pa}(X_i))$$

- Credal nets
  - Same calculations iterated over the conditional credal set $K(X_i|\pi_i)$

$$\underline{P}(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j)) = \inf_{P(X_i|\mathrm{pa}(X_i))\in K(X_i|\mathrm{pa}(X_i))} P(\tilde{x}_i^j|\mathrm{pa}(\tilde{X}_i^j))$$

- A “binarized” Bayesian/credal net is obtained
- The transformation takes only linear time!

Binarization (results)

BNs binarization is exact

- Let $P(\mathbf{X})$ be the joint probability mass function of a BN,
- and $\tilde{P}(\tilde{\mathbf{X}})$ the corresponding p.m.f. on the binarized BN.
- Then, $P(\mathbf{X}) = \tilde{P}(\tilde{\mathbf{X}})$.

CNs binarization is an outer approximation

- Let $K(\mathbf{X})$ be the strong extension of a CN,
- and $\tilde{K}(\tilde{\mathbf{X}})$ the strong extension of its binarization.
- Then, $K(\mathbf{X}) \subseteq \tilde{K}(\tilde{\mathbf{X}})$.

We can do better!
But another transformation should be applied
before the binarization.

Binarization (results)

BNs binarization is exact

- Let $P(\mathbf{X})$ be the joint probability mass function of a BN,
- and $\tilde{P}(\tilde{\mathbf{X}})$ the corresponding p.m.f. on the binarized BN.
- Then, $P(\mathbf{X}) = \tilde{P}(\tilde{\mathbf{X}})$.

CNs binarization is an outer approximation

- Let $K(\mathbf{X})$ be the strong extension of a CN,
- and $\tilde{K}(\tilde{\mathbf{X}})$ the strong extension of its binarization.
- Then, $K(\mathbf{X}) \subseteq \tilde{K}(\tilde{\mathbf{X}})$.

We can do better!
But another transformation should be applied
before the binarization.

Binarization (results)

BNs binarization is exact

- Let $P(\mathbf{X})$ be the joint probability mass function of a BN,
- and $\tilde{P}(\tilde{\mathbf{X}})$ the corresponding p.m.f. on the binarized BN.
- Then, $P(\mathbf{X}) = \tilde{P}(\tilde{\mathbf{X}})$.

CNs binarization is an outer approximation

- Let $K(\mathbf{X})$ be the strong extension of a CN,
- and $\tilde{K}(\tilde{\mathbf{X}})$ the strong extension of its binarization.
- Then, $K(\mathbf{X}) \subseteq \tilde{K}(\tilde{\mathbf{X}})$.

We can do better!
But another transformation should be applied
before the binarization.

Binarization (results)

BNs binarization is exact

- Let $P(\mathbf{X})$ be the joint probability mass function of a BN,
- and $\tilde{P}(\tilde{\mathbf{X}})$ the corresponding p.m.f. on the binarized BN.
- Then, $P(\mathbf{X}) = \tilde{P}(\tilde{\mathbf{X}})$.

CNs binarization is an outer approximation

- Let $K(\mathbf{X})$ be the strong extension of a CN,
- and $\tilde{K}(\tilde{\mathbf{X}})$ the strong extension of its binarization.
- Then, $K(\mathbf{X}) \subseteq \tilde{K}(\tilde{\mathbf{X}})$.

We can do better!

But another transformation should be applied before the binarization.

Binarization (results)

BNs binarization is exact

- Let $P(\mathbf{X})$ be the joint probability mass function of a BN,
- and $\tilde{P}(\tilde{\mathbf{X}})$ the corresponding p.m.f. on the binarized BN.
- Then, $P(\mathbf{X}) = \tilde{P}(\tilde{\mathbf{X}})$.

CNs binarization is an outer approximation

- Let $K(\mathbf{X})$ be the strong extension of a CN,
- and $\tilde{K}(\tilde{\mathbf{X}})$ the strong extension of its binarization.
- Then, $K(\mathbf{X}) \subseteq \tilde{K}(\tilde{\mathbf{X}})$.

We can do better!

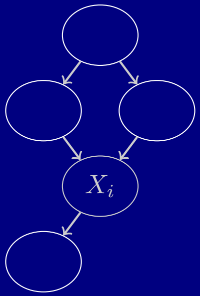
But another transformation should be applied

before the binarization.

Decision-theoretic specification of CNs

(Antonucci & Zaffalon, PGM '06/IJAR 2008)

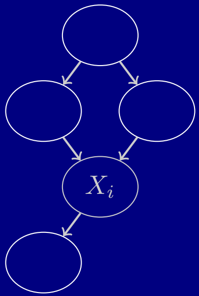
- For each $i = 1, \ldots, n$, add a node T_i, parent of X_i, between X_i and $\mathrm{pa}(X_i)$
- Decision nodes $\{T_i\}_{i=1}^n$ indexing the possible specifications of each mass function given the values of the parents
- Decision nodes can be regarded as chance nodes with a "vacuous" specification of the relative credal set
- An equivalent credal net is obtained!



Decision-theoretic specification of CNs

(Antonucci & Zaffalon, PGM '06/IJAR 2008)

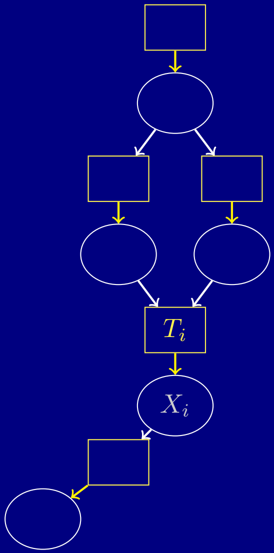
- For each $i = 1, \ldots, n$, add a node T_i, parent of X_i, between X_i and $\mathrm{pa}(X_i)$
- Decision nodes $\{T_i\}_{i=1}^n$ indexing the possible specifications of each mass function given the values of the parents
- Decision nodes can be regarded as chance nodes with a "vacuous" specification of the relative credal set
- An equivalent credal net is obtained!



Decision-theoretic specification of CNs

(Antonucci & Zaffalon, PGM ’06/IJAR 2008)

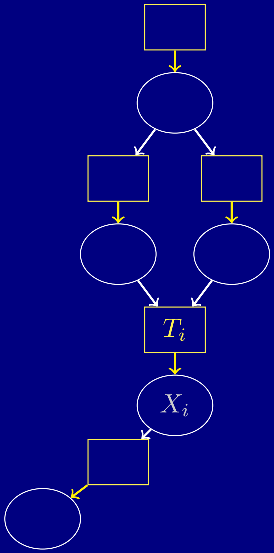
- For each $i = 1, \ldots, n$, add a node T_i, parent of X_i, between X_i and $\mathrm{pa}(X_i)$
- Decision nodes $\{T_i\}_{i=1}^n$ indexing the possible specifications of each mass function given the values of the parents
- Decision nodes can be regarded as chance nodes with a “vacuous” specification of the relative credal set
- An equivalent credal net is obtained!



Decision-theoretic specification of CNs

(Antonucci & Zaffalon, PGM '06/IJAR 2008)

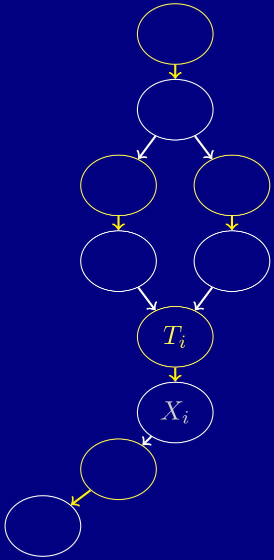
- For each $i = 1, \ldots, n$, add a node T_i, parent of X_i, between X_i and $\mathrm{pa}(X_i)$
- Decision nodes $\{T_i\}_{i=1}^n$ indexing the possible specifications of each mass function given the values of the parents
- Decision nodes can be regarded as chance nodes with a "vacuous" specification of the relative credal set
- An equivalent credal net is obtained!



Decision-theoretic specification of CNs

(Antonucci & Zaffalon, PGM ’06/IJAR 2008)

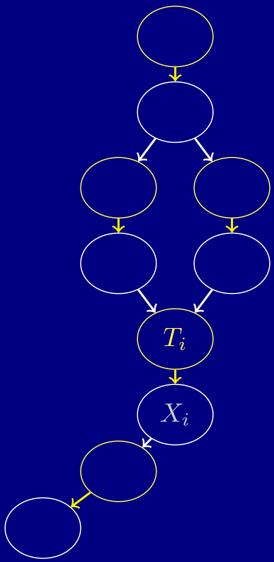
- For each $i = 1, \ldots, n$, add a node T_i, parent of X_i, between X_i and $\mathrm{pa}(X_i)$
- Decision nodes $\{T_i\}_{i=1}^{n}$ indexing the possible specifications of each mass function given the values of the parents
- Decision nodes can be regarded as chance nodes with a “vacuous” specification of the relative credal set
- An equivalent credal net is obtained!



Decision-theoretic specification of CNs

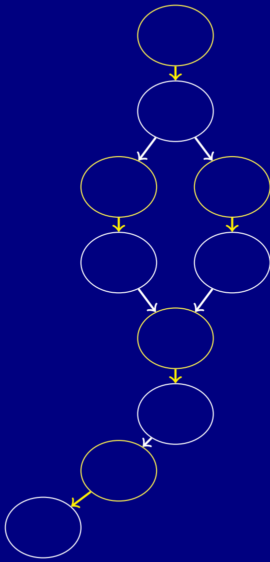
(Antonucci & Zaffalon, PGM ’06/IJAR 2008)

- For each $i = 1, \ldots, n$, add a node T_i, parent of X_i, between X_i and $\mathrm{pa}(X_i)$
- Decision nodes $\{T_i\}_{i=1}^n$ indexing the possible specifications of each mass function given the values of the parents
- Decision nodes can be regarded as chance nodes with a “vacuous” specification of the relative credal set
- An equivalent credal net is obtained!



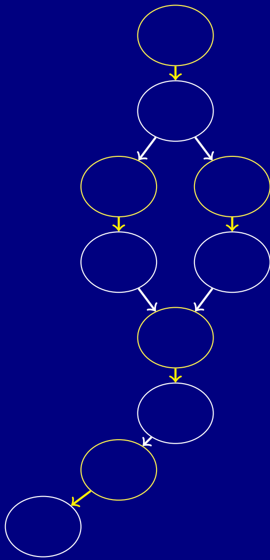
Binarization of DT-specified CNs

- Binarization can be implemented locally
- After DT specification, the nodes are either precise or vacuous
  - For precise specifications binarization is exact (result for BNs)
  - Also for vacuous specifications binarization can be proved to be exact
- Binarization of DT-specified CNs is exact!
- But any CN can be DT-specified
- Any CN admits an exact binarization



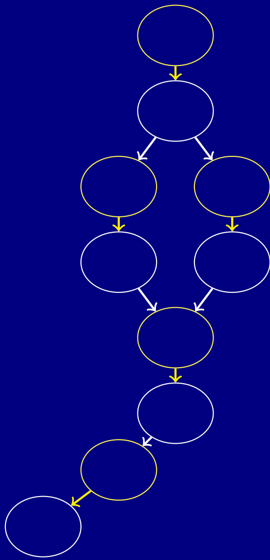
Binarization of DT-specified CNs

- Binarization can be implemented locally
- After DT specification, the nodes are either precise or vacuous
  - For precise specifications binarization is exact (result for BNs)
  - Also for vacuous specifications binarization can be proved to be exact
- Binarization of DT-specified CNs is exact!
- But any CN can be DT-specified
- Any CN admits an exact binarization



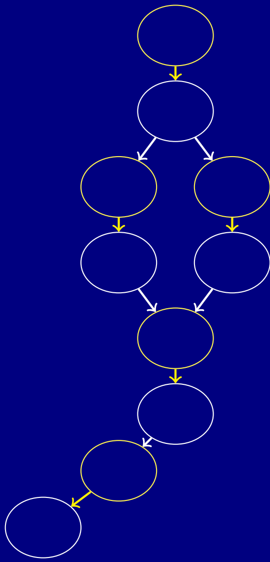
Binarization of DT-specified CNs

- Binarization can be implemented locally
- After DT specification, the nodes are either precise or vacuous
  - For precise specifications binarization is exact (result for BNs)
  - Also for vacuous specifications binarization can be proved to be exact
- Binarization of DT-specified CNs is exact!
- But any CN can be DT-specified
- Any CN admits an exact binarization



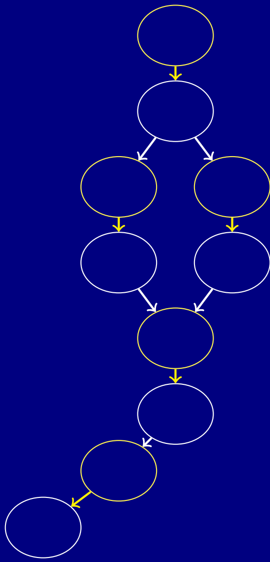
Binarization of DT-specified CNs

- Binarization can be implemented locally
- After DT specification, the nodes are either precise or vacuous
  - For precise specifications binarization is exact (result for BNs)
  - Also for vacuous specifications binarization can be proved to be exact
- Binarization of DT-specified CNs is exact!
- But any CN can be DT-specified
- Any CN admits an exact binarization



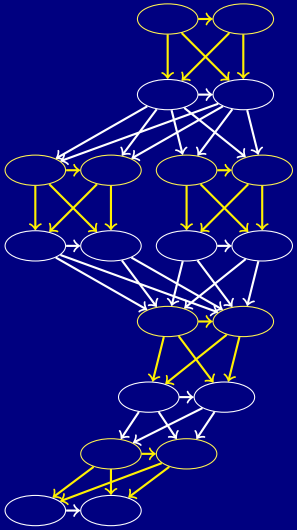
Binarization of DT-specified CNs

- Binarization can be implemented locally
- After DT specification, the nodes are either precise or vacuous
  - For precise specifications binarization is exact (result for BNs)
  - Also for vacuous specifications binarization can be proved to be exact
- Binarization of DT-specified CNs is exact!
- But any CN can be DT-specified
- Any CN admits an exact binarization



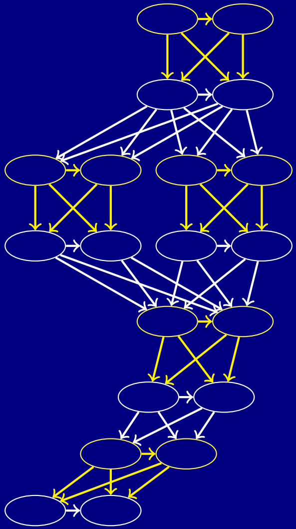
Binarization of DT-specified CNs

- Binarization can be implemented locally
- After DT specification, the nodes are either precise or vacuous
  - For precise specifications binarization is exact (result for BNs)
  - Also for vacuous specifications binarization can be proved to be exact
- Binarization of DT-specified CNs is exact!
- But any CN can be DT-specified
- Any CN admits an exact binarization



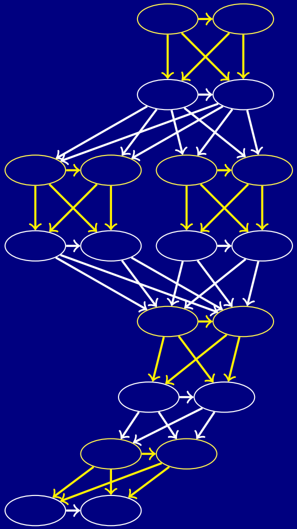
Binarization of DT-specified CNs

- Binarization can be implemented locally
- After DT specification, the nodes are either precise or vacuous
  - For precise specifications binarization is exact (result for BNs)
  - Also for vacuous specifications binarization can be proved to be exact
- Binarization of DT-specified CNs is exact!
- But any CN can be DT-specified
- Any CN admits an exact binarization



Binarization of DT-specified CNs

- Binarization can be implemented locally
- After DT specification, the nodes are either precise or vacuous
  - For precise specifications binarization is exact (result for BNs)
  - Also for vacuous specifications binarization can be proved to be exact
- Binarization of DT-specified CNs is exact!
- But any CN can be DT-specified
- Any CN admits an exact binarization



Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\max}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\max}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\text{max}}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\max}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\text{max}}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\text{max}}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\max}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
    - Approx only because of loopy!
    - State-of-the-art updating algorithm for CNs updating
    - Good accuracy and scalability $O(e^{\text{indegree}_{\max}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	**1.40**	1.81
Multi-10	1.95	**1.07**	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	**1.89**	6.93
Multi-25	2.31	**1.60**	1.84
Multi-25	2.48	**2.04**	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	**1.90**	3.02
Alarm	3.31	**2.39**	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\text{max}}})$ is better than $O(e^{\text{treewidth}})$

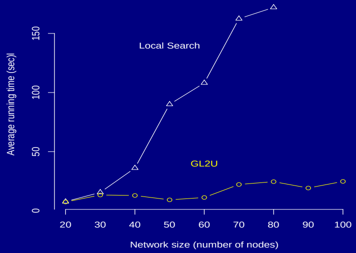
	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	**1.40**	1.81
Multi-10	1.95	**1.07**	3.38
Multi-10	**1.20**	1.75	3.08
Multi-10	**0.27**	1.25	2.22
Multi-10	2.34	**1.89**	6.93
Multi-25	2.31	**1.60**	1.84
Multi-25	2.48	**2.04**	3.03
Polyt-50	**1.12**	1.93	2.89
Polyt-50	**1.45**	2.21	3.92
Insurance	**0.55**	1.17	1.75
Insurance	**1.13**	1.32	1.93
Alarm	2.90	**1.90**	3.02
Alarm	3.31	**2.39**	4.23

Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability

$O(e^{\text{indegree}_{\max}})$ is better than $O(e^{\text{treewidth}})$

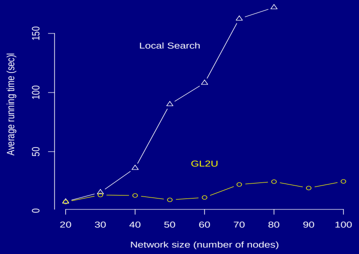
	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23



Generalized loopy 2U (GL2U)

- Multi-connected (non-binary) CNs?
- Binarization + L2U (twofold approx)
- DT + Bin + L2U = GL2U is better!
  - Approx only because of loopy!
  - State-of-the-art updating algorithm for CNs updating
  - Good accuracy and scalability $O(e^{\text{indegree}_{\max}})$ is better than $O(e^{\text{treewidth}})$

	Loc Search	GL2U	Bin+L2U
Multi-10	1.89	1.40	1.81
Multi-10	1.95	1.07	3.38
Multi-10	1.20	1.75	3.08
Multi-10	0.27	1.25	2.22
Multi-10	2.34	1.89	6.93
Multi-25	2.31	1.60	1.84
Multi-25	2.48	2.04	3.03
Polyt-50	1.12	1.93	2.89
Polyt-50	1.45	2.21	3.92
Insurance	0.55	1.17	1.75
Insurance	1.13	1.32	1.93
Alarm	2.90	1.90	3.02
Alarm	3.31	2.39	4.23



Conclusions and outlooks

- Exact binarization of BNs and CNs
- A state-of-the-art algorithm for CNs updating
- The algorithm of choice for very large nets?
- A Python/C++ implementation available (ask Sun Yi)
- Challenges
  - In numerical tests (G)L2U always converges.
    A formal proof of that?
  - For non-binary targets, accuracy can be improved
    with an alternative binarization of the target.

Conclusions and outlooks

- Exact binarization of BNs and CNs
- A state-of-the-art algorithm for CNs updating
- The algorithm of choice for very large nets?
- A Python/C++ implementation available (ask Sun Yi)
- Challenges
  - In numerical tests (G)L2U always converges.
    A formal proof of that?
  - For non-binary targets, accuracy can be improved
    with an alternative binarization of the target.

Conclusions and outlooks

- Exact binarization of BNs and CNs
- A state-of-the-art algorithm for CNs updating
- The algorithm of choice for very large nets?
- A Python/C++ implementation available (ask Sun Yi)
- Challenges
  - In numerical tests (G)L2U always converges.
    A formal proof of that?
  - For non-binary targets, accuracy can be improved
    with an alternative binarization of the target.

Conclusions and outlooks

- Exact binarization of BNs and CNs
- A state-of-the-art algorithm for CNs updating
- The algorithm of choice for very large nets?
- A Python/C++ implementation available (ask Sun Yi)
- Challenges
  - In numerical tests (G)L2U always converges.
    A formal proof of that?
  - For non-binary targets, accuracy can be improved
    with an alternative binarization of the target.

Conclusions and outlooks

- Exact binarization of BNs and CNs
- A state-of-the-art algorithm for CNs updating
- The algorithm of choice for very large nets?
- A Python/C++ implementation available (ask Sun Yi)
- Challenges
  - In numerical tests (G)L2U always converges.
    A formal proof of that?
  - For non-binary targets, accuracy can be improved
    with an alternative binarization of the target.

Conclusions and outlooks

- Exact binarization of BNs and CNs
- A state-of-the-art algorithm for CNs updating
- The algorithm of choice for very large nets?
- A Python/C++ implementation available (ask Sun Yi)
- Challenges
  - In numerical tests (G)L2U always converges.
    A formal proof of that?
  - For non-binary targets, accuracy can be improved
    with an alternative binarization of the target.

Conclusions and outlooks

- Exact binarization of BNs and CNs
- A state-of-the-art algorithm for CNs updating
- The algorithm of choice for very large nets?
- A Python/C++ implementation available (ask Sun Yi)
- Challenges
  - In numerical tests (G)L2U always converges.
    A formal proof of that?
  - For non-binary targets, accuracy can be improved
    with an alternative binarization of the target.

Conclusions and outlooks

- Exact binarization of BNs and CNs
- A state-of-the-art algorithm for CNs updating
- The algorithm of choice for very large nets?
- A Python/C++ implementation available (ask Sun Yi)
- Challenges
  - In numerical tests (G)L2U always converges.
    A formal proof of that?
  - For non-binary targets, accuracy can be improved
    with an alternative binarization of the target.