Imprecise Probabilistic Graphical Models

Credal Networks and Other Guys

Alessandro Antonucci & Cassio de Campos & Giorgio Corani

{alessandro,cassio,giorgio}@idsia.ch

Dalle Molle Institute for Artificial Intelligence IDSIA

SISPTA School on Imprecise Probability

Durham, September 5, 2010

Imprecise Probability Group @ IDSIA

- IDSIA = *Dalle Molle* Institute for AI

Imprecise Probability Group @ IDSIA

- IDSIA = *Dalle Molle* Institute for AI

Imprecise Probability Group @ IDSIA

- IDSIA = *Dalle Molle* Institute for AI

Imprecise Probability Group @ IDSIA

- IDSIA = *Dalle Molle* Institute for AI

2000

IDSIA

SUPSI

Università della Svizzera italiana

Imprecise Probability Group @ IDSIA

- IDSIA = *Dalle Molle* Institute for AI

Imprecise Probability Group @ IDSIA

- IDSIA = *Dalle Molle* Institute for AI
- Imprecise Probability Group (1 professor, 4 researchers, 1 phd)
  - Theory of imprecise probability
  - Probabilistic graphical models
  - Data mining and classification
  - Observations modelling (missing data)
  - Data fusion and filtering
  - Applications to environmental modelling, military decision making, risk analysis, bioinformatics, biology, tracking, vision, . . .

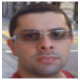
Marco  Zaffalon

Alex  Antonucci

Cassio  de Campos

Giorgio  Corani

Alessio  Benavoli

Denis  Mauà

Probabilistic Graphical Models

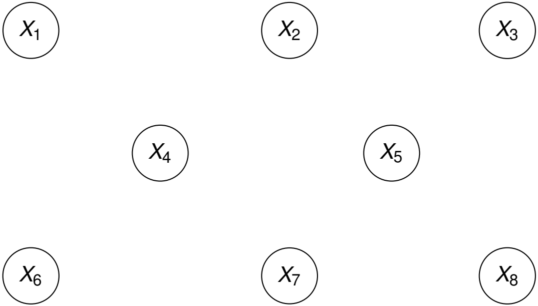
aka **Decomposable** Multivariate Probabilistic Models

(whose decomposability is induced by **independence**)

Probabilistic Graphical Models

aka Decomposable Multivariate Probabilistic Models

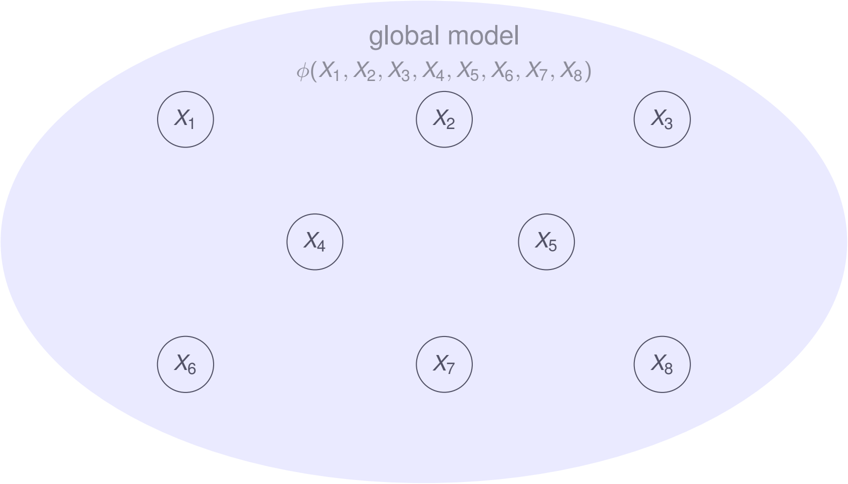
(whose decomposability is induced by independence)



Probabilistic Graphical Models

aka Decomposable Multivariate Probabilistic Models

(whose decomposability is induced by independence)

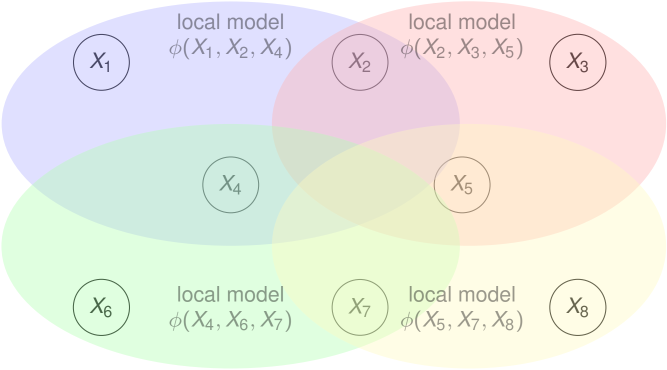


Probabilistic Graphical Models

aka Decomposable Multivariate Probabilistic Models

(whose decomposability is induced by independence)

$$\phi(X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8) = \phi(X_1, X_2, X_4) \otimes \phi(X_2, X_3, X_5) \otimes \phi(X_4, X_6, X_7) \otimes \phi(X_5, X_7, X_8)$$



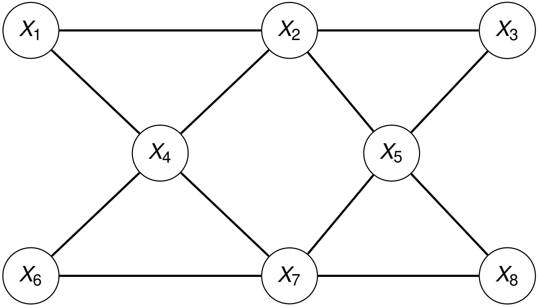
Probabilistic Graphical Models

aka Decomposable Multivariate Probabilistic Models

(whose decomposability is induced by independence)

undirected graphs

precise/imprecise Markov random fields



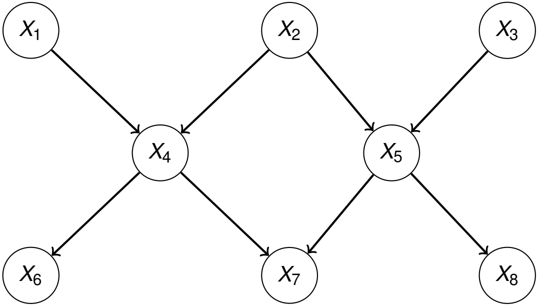
Probabilistic Graphical Models

aka Decomposable Multivariate Probabilistic Models

(whose decomposability is induced by independence)

directed graphs

Bayesian/credal networks



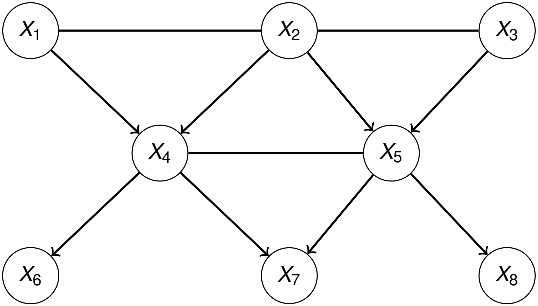
Probabilistic Graphical Models

aka Decomposable Multivariate Probabilistic Models

(whose decomposability is induced by independence)

mixed graphs

chain graphs



[Exe #1] Fault trees (Vesely et al, 1981)

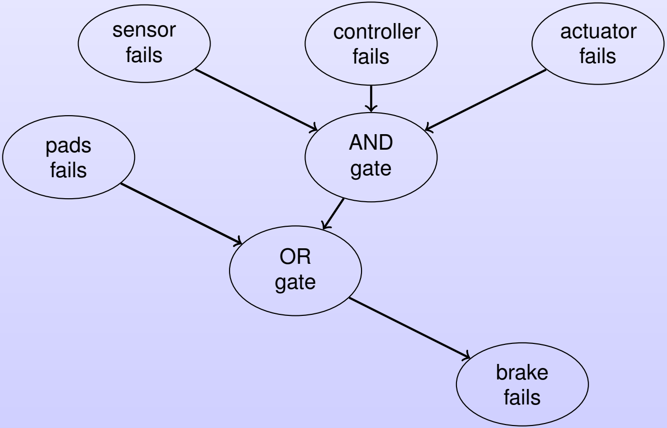
brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

devices failures are independent

[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

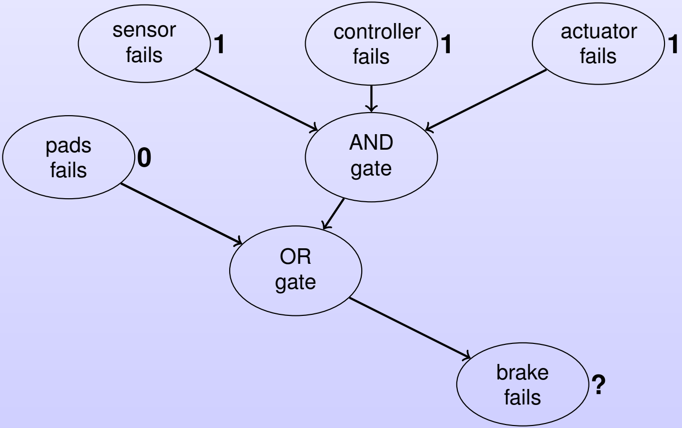
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

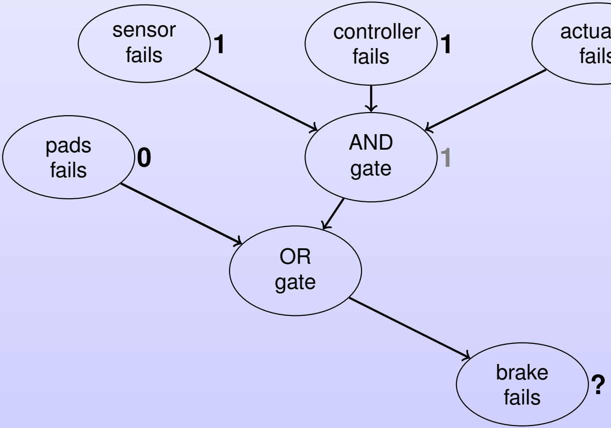
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

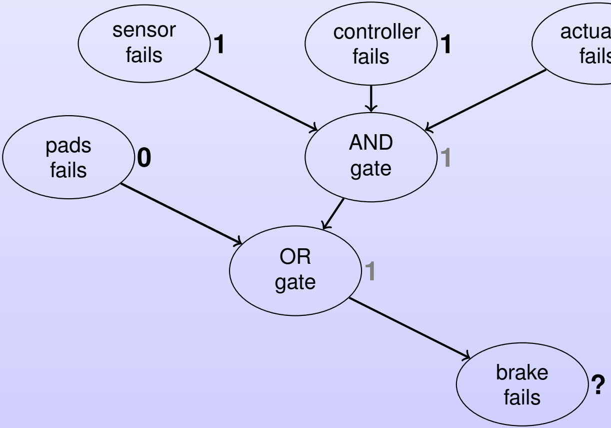
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

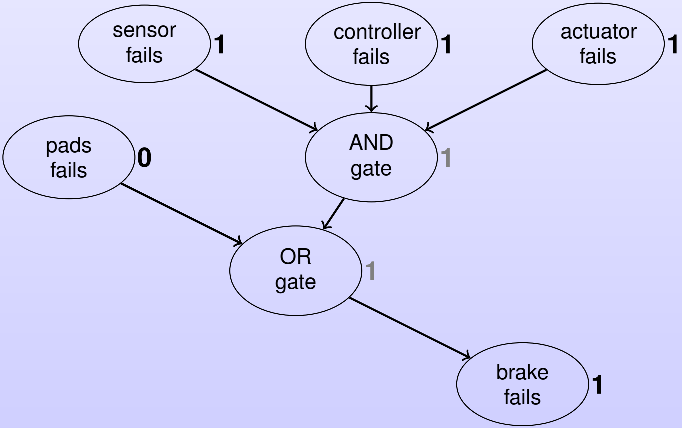
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads ∨ (sensor ∧ controller ∧ actuator)]

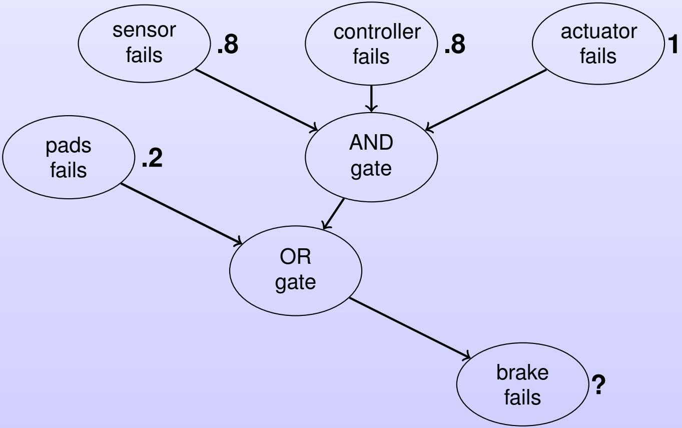
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

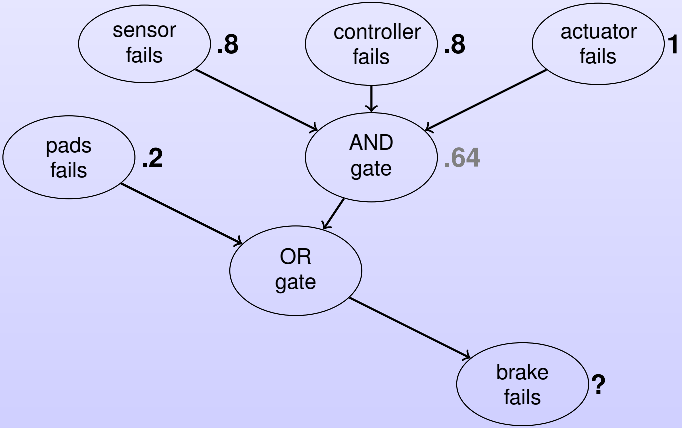
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

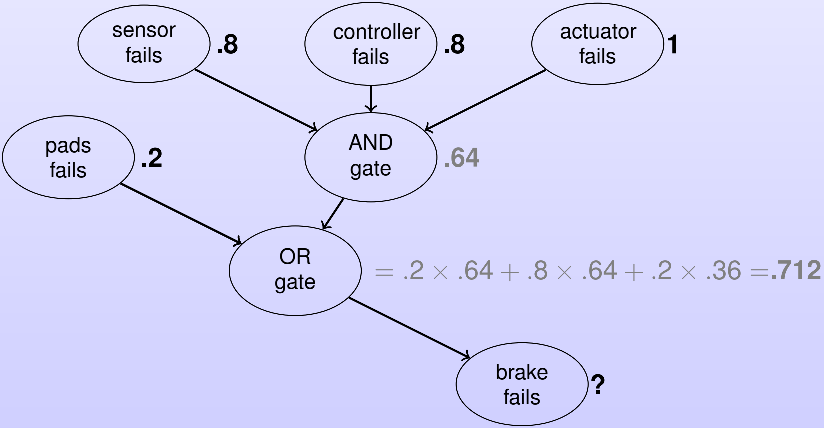
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads ∨ (sensor ∧ controller ∧ actuator)]

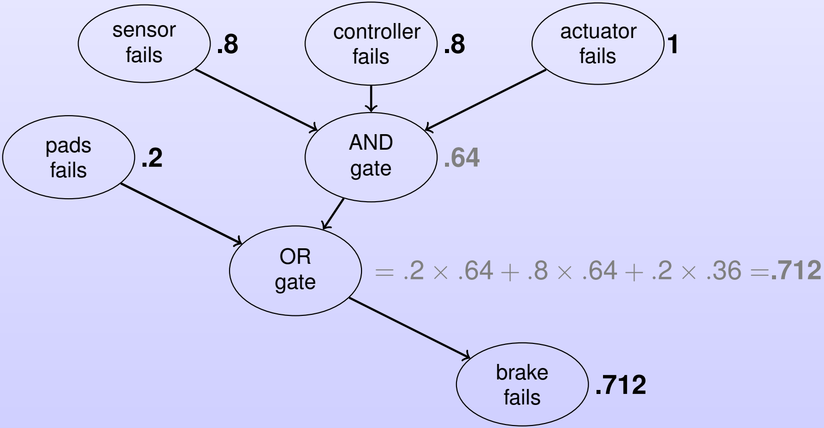
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

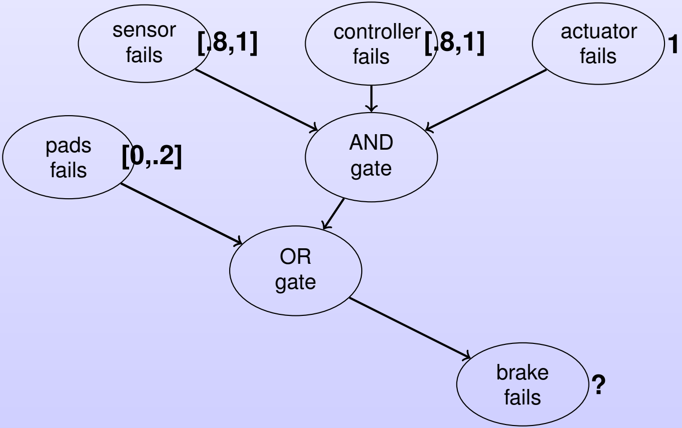
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads ∨ (sensor ∧ controller ∧ actuator)]

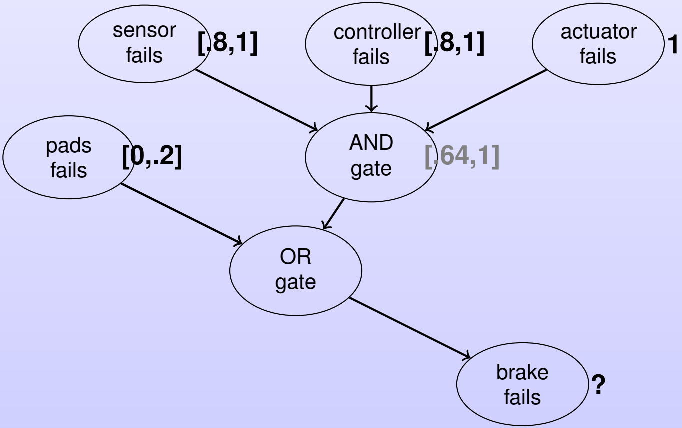
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads ∨ (sensor ∧ controller ∧ actuator)]

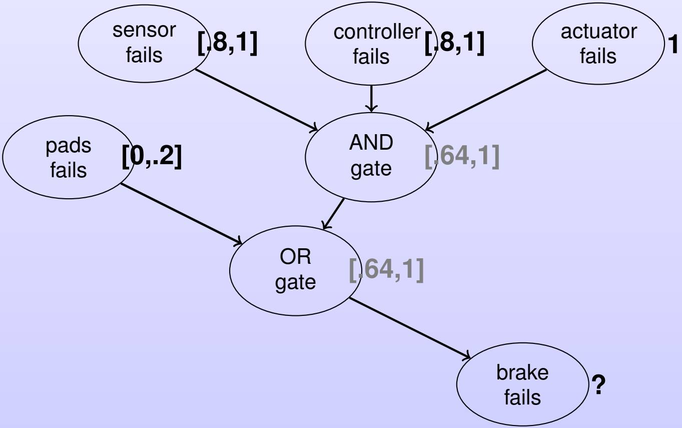
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads ∨ (sensor ∧ controller ∧ actuator)]

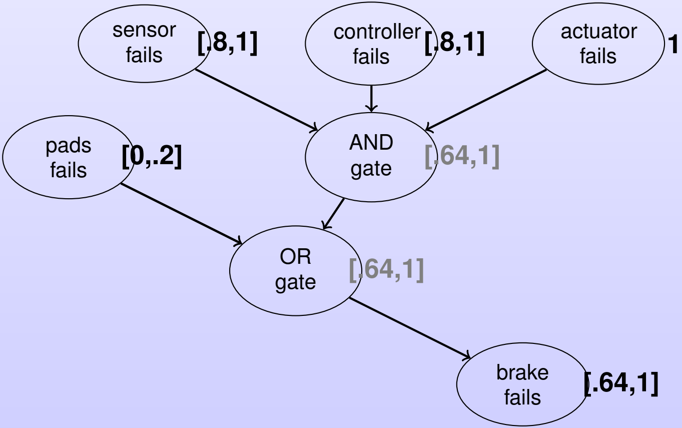
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

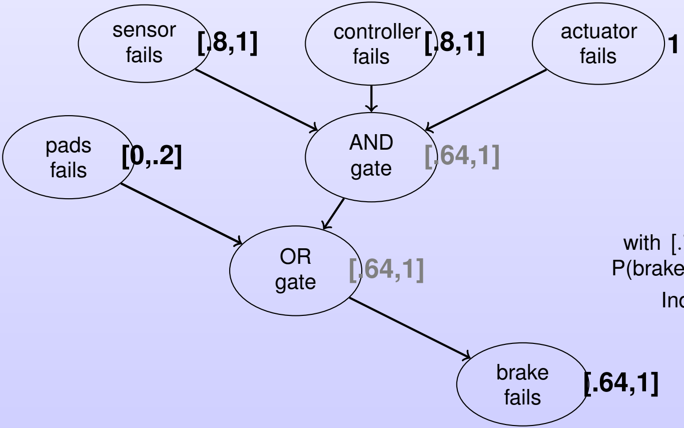
devices failures are independent



[Exe #1] Fault trees (Vesely et al, 1981)

brake fails = [pads $\vee$ (sensor $\wedge$ controller $\wedge$ actuator)]

devices failures are independent



with [.7, 1] instead
P(brake fails)∈[.49,1]

Indecision!

Outline

- Motivations for imprecise probability
- Credal sets (basic concepts and operations)
- Independence relations
- Credal networks
- Modelling observations/missingness
- Decision making
- Inference algorithms
- Other probabilistic graphical models
- Conclusions

Three different levels of knowledge

- FIFA'10 final match between Holland and Spain
- Result of Holland after the regular time? Win, draw or loss?

DETERMINISM

The Dutch goalkeeper is unbeatable and Holland always makes a goal

Holland (certainly) wins

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

UNCERTAINTY

Win is two times more probable than draw, and this being three times more probable than loss

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} .6 \\ .3 \\ .1 \end{bmatrix}$$

IMPRECISION

Win is more probable than draw, and this is more probable than loss

$P(Win) > P(Draw)$
$P(Draw) > P(Loss)$

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} \frac{\alpha}{3} + \beta + \frac{\gamma}{2} \\ \frac{\alpha}{3} + \frac{\gamma}{2} \\ \frac{\alpha}{3} \end{bmatrix}$$

$\forall \alpha, \beta, \gamma$ such that
$\alpha > 0, \beta > 0, \gamma > 0,$
$\alpha + \beta + \gamma = 1$

Three different levels of knowledge

- FIFA'10 final match between Holland and Spain
- Result of Holland after the regular time? Win, draw or loss?

DETERMINISM

The Dutch goalkeeper is unbeatable and Holland always makes a goal

Holland (certainly) wins

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

UNCERTAINTY

Win is two times more probable than draw, and this being three times more probable than loss

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} .6 \\ .3 \\ .1 \end{bmatrix}$$

IMPRECISION

Win is more probable than draw, and this is more probable than loss

$P(Win) > P(Draw)$
$P(Draw) > P(Loss)$

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} \frac{\alpha}{3} + \beta + \frac{\gamma}{2} \\ \frac{\alpha}{3} + \frac{\gamma}{2} \\ \frac{\alpha}{3} \end{bmatrix}$$

$\forall \alpha, \beta, \gamma$ such that
$\alpha > 0, \beta > 0, \gamma > 0,$
$\alpha + \beta + \gamma = 1$

Three different levels of knowledge

- FIFA'10 final match between Holland and Spain
- Result of Holland after the regular time? Win, draw or loss?

DETERMINISM

The Dutch goalkeeper is unbeatable and Holland always makes a goal

Holland (certainly) wins

$$\begin{array}{l} P(Win) \\ P(Draw) \\ P(Loss) \end{array} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

UNCERTAINTY

Win is two times more probable than draw, and this being three times more probable than loss

$$\begin{array}{l} P(Win) \\ P(Draw) \\ P(Loss) \end{array} = \begin{bmatrix} .6 \\ .3 \\ .1 \end{bmatrix}$$

IMPRECISION

Win is more probable than draw, and this is more probable than loss

$P(Win) > P(Draw)$
$P(Draw) > P(Loss)$

$$\begin{array}{l} P(Win) \\ P(Draw) \\ P(Loss) \end{array} = \begin{bmatrix} \frac{\alpha}{3} + \beta + \frac{\gamma}{2} \\ \frac{\alpha}{3} + \frac{\gamma}{2} \\ \frac{\alpha}{3} \end{bmatrix}$$

$\forall \alpha, \beta, \gamma$ such that
$\alpha > 0, \beta > 0, \gamma > 0,$
$\alpha + \beta + \gamma = 1$

Three different levels of knowledge

- FIFA'10 final match between Holland and Spain
- Result of Holland after the regular time? Win, draw or loss?

DETERMINISM

The Dutch goalkeeper is unbeatable and Holland always makes a goal

Holland (certainly) wins

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

UNCERTAINTY

Win is two times more probable than draw, and this being three times more probable than loss

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} .6 \\ .3 \\ .1 \end{bmatrix}$$

IMPRECISION

Win is more probable than draw, and this is more probable than loss

$P(Win) > P(Draw)$
$P(Draw) > P(Loss)$

$$\begin{matrix} P(Win) \\ P(Draw) \\ P(Loss) \end{matrix} = \begin{bmatrix} \frac{\alpha}{3} + \beta + \frac{\gamma}{2} \\ \frac{\alpha}{3} + \frac{\gamma}{2} \\ \frac{\alpha}{3} \end{bmatrix}$$

$\forall \alpha, \beta, \gamma$ such that
$\alpha > 0, \beta > 0, \gamma > 0,$
$\alpha + \beta + \gamma = 1$

Three different levels of knowledge (ii)

DETERMINISM

UNCERTAINTY

IMPRECISION

Three different levels of knowledge (ii)

DETERMINISM UNCERTAINTY IMPRECISION

Three different levels of knowledge (ii)

DETERMINISM

UNCERTAINTY

IMPRECISION

Three different levels of knowledge (ii)

DETERMINISM

UNCERTAINTY

IMPRECISION

Three different levels of knowledge (ii)

DETERMINISM

UNCERTAINTY

IMPRECISION

Limit when learning from large (complete) data sets

Three different levels of knowledge (ii)

DETERMINISM

UNCERTAINTY

IMPRECISION

Limit when learning from large (complete) data sets

Propositional (Boolean) Logic

Bayesian Theory of Probability

Walley's Theory of Coherent Lower Previsions

Three different levels of knowledge (ii)

DETERMINISM

UNCERTAINTY

IMPRECISION

Limit when learning from large (complete) data sets

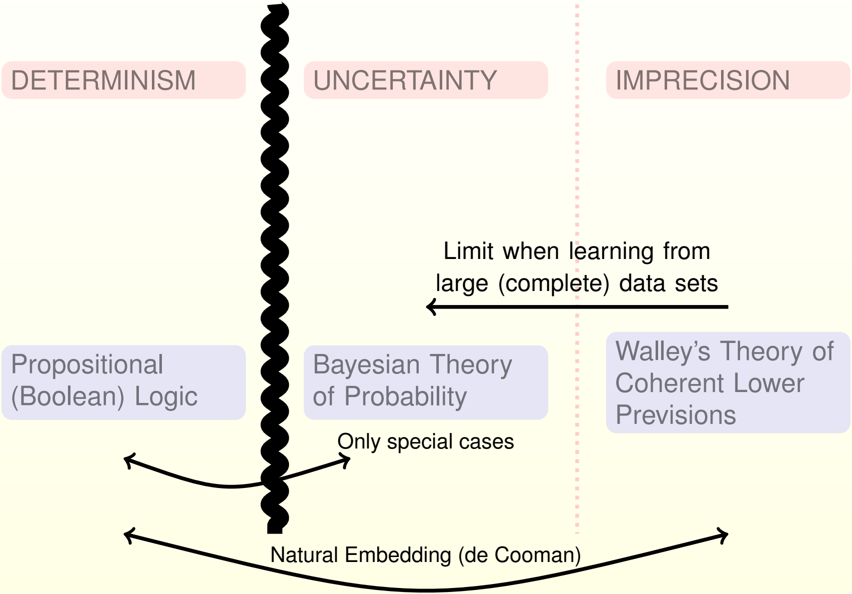
Propositional (Boolean) Logic

Bayesian Theory of Probability

Walley's Theory of Coherent Lower Previsions

Natural Embedding (de Cooman)

Three different levels of knowledge (ii)



Three different levels of knowledge (ii)

DETERMINISM

Propositional (Boolean) Logic

UNCERTAINTY

Bayesian Theory of Probability

Only special cases

IMPRECISION

Small/incomplete data

Expert's (qualitative) knowledge

Vague/incomplete observations

Limit when learning from large (complete) data sets

Walley's Theory of Coherent Lower Previsions

Natural Embedding (de Cooman)

From CLPs to Credal Sets

Modelling knowledge about X, taking values in $\mathcal{X}$

From CLPs to Credal Sets

Modelling knowledge about X, taking values in $\mathcal{X}$

Bayesian / precise

probability distribution
$p : \mathcal{X} \to \mathbb{R}$

$$\begin{cases} p(x) \geq 0 \forall x \in \mathcal{X} \\ \sum_{x \in \mathcal{X}} p(x) = 1 \end{cases}$$

coherent **linear** prevision
$P : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} P(f+g) = P(f) + P(g) \\ P(f) \geq \inf f, \forall f, g \in \mathcal{L}(\mathcal{X}) \end{cases}$$

From CLPs to Credal Sets

Modelling knowledge about X, taking values in $\mathcal{X}$

Bayesian / precise

probability distribution
$p : \mathcal{X} \to \mathbb{R}$

$$\begin{cases} p(x) \geq 0 \forall x \in \mathcal{X} \\ \sum_{x \in \mathcal{X}} p(x) = 1 \end{cases}$$

$\underline{P}(f) = \sum_{x \in \mathcal{X}} p(x) f(x)$

⟷

$P(x) = \underline{P}(\mathcal{I}_{\{x\}})$

coherent **linear** prevision
$P : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} P(f+g) = P(f) + P(g) \\ P(f) \geq \inf f, \forall f, g \in \mathcal{L}(\mathcal{X}) \end{cases}$$

From CLPs to Credal Sets

Modelling knowledge about X, taking values in $\mathcal{X}$

Bayesian / precise

$\underline{P}(f) = \sum_{x\in\mathcal{X}} p(x)f(x)$

probability distribution
$p : \mathcal{X} \to \mathbb{R}$

$\Longleftrightarrow$

coherent **linear** prevision
$P : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$P(x) = \underline{P}(\mathcal{I}_{\{x\}})$

$$\begin{cases} p(x) \geq 0 \forall x \in \mathcal{X} \\ \sum_{x\in\mathcal{X}} p(x) = 1 \end{cases}$$

$$\begin{cases} P(f+g) = P(f) + P(g) \\ P(f) \geq \inf f, \forall f, g \in \mathcal{L}(\mathcal{X}) \end{cases}$$

Credal / imprecise

set of distributions
$K(X) = \{\ p(X) \mid \mathit{constraints}\ \}$

coherent lower prevision
$\underline{P} : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} \underline{P}(f) \geq \min f\ \forall f \in \mathcal{L}(\mathcal{X}) \\ \underline{P}(\lambda f) = \lambda \underline{P}(f) \forall \lambda > 0 \\ \underline{P}(f+g) \geq \underline{P}(f) + \underline{P}(g) \end{cases}$$

From CLPs to Credal Sets

Modelling knowledge about X, taking values in $\mathcal{X}$

Bayesian / precise

probability distribution
$p : \mathcal{X} \to \mathbb{R}$

$$\begin{cases} p(x) \geq 0 \forall x \in \mathcal{X} \\ \sum_{x \in \mathcal{X}} p(x) = 1 \end{cases}$$

$\underline{P}(f) = \sum_{x \in \mathcal{X}} p(x) f(x)$

$P(x) = \underline{P}(\mathcal{I}_{\{x\}})$

coherent **linear** prevision
$P : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} P(f+g) = P(f) + P(g) \\ P(f) \geq \inf f, \forall f, g \in \mathcal{L}(\mathcal{X}) \end{cases}$$

Credal / imprecise

set of distributions
$K(X) = \{ \; p(X) \mid \textit{constraints} \; \}$

coherent lower prevision
$\underline{P} : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} \underline{P}(f) \geq \min f \; \forall f \in \mathcal{L}(\mathcal{X}) \\ \underline{P}(\lambda f) = \lambda \underline{P}(f) \forall \lambda > 0 \\ \underline{P}(f+g) \geq \underline{P}(f) + \underline{P}(g) \end{cases}$$

$\exists$ set $\mathcal{M}$ of linear previsions s.t.
$\underline{P}(f) = \inf_{P \in \mathcal{M}} P(f) \; \forall f \in \mathcal{L}(\mathcal{X})$

From CLPs to Credal Sets

Modelling knowledge about X, taking values in $\mathcal{X}$

Bayesian / precise

probability distribution
$p : \mathcal{X} \to \mathbb{R}$

$\underline{P}(f) = \sum_{x \in \mathcal{X}} p(x) f(x)$

$P(x) = \underline{P}(\mathcal{I}_{\{x\}})$

coherent **linear** prevision
$P : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} p(x) \geq 0 \forall x \in \mathcal{X} \\ \sum_{x \in \mathcal{X}} p(x) = 1 \end{cases}$$

$$\begin{cases} P(f+g) = P(f) + P(g) \\ P(f) \geq \inf f, \forall f, g \in \mathcal{L}(\mathcal{X}) \end{cases}$$

Credal / imprecise

set of distributions
$K(X) = \{\ p(X) \mid constraints\ \}$

coherent lower prevision
$\underline{P} : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} \underline{P}(f) \geq \min f\ \forall f \in \mathcal{L}(\mathcal{X}) \\ \underline{P}(\lambda f) = \lambda \underline{P}(f) \forall \lambda > 0 \\ \underline{P}(f+g) \geq \underline{P}(f) + \underline{P}(g) \end{cases}$$

$\exists$ set $\mathcal{M}$ of linear previsions s.t.
$\underline{P}(f) = \inf_{P \in \mathcal{M}} P(f)\ \forall f \in \mathcal{L}(\mathcal{X})$

lower envelope Theorem

(a set and its convex closure have the same lower envelope!)

From CLPs to Credal Sets

Modelling knowledge about X, taking values in $\mathcal{X}$

Bayesian / precise

probability distribution
$p : \mathcal{X} \to \mathbb{R}$

$$\begin{cases} p(x) \geq 0 \forall x \in \mathcal{X} \\ \sum_{x \in \mathcal{X}} p(x) = 1 \end{cases}$$

$\underline{P}(f) = \sum_{x \in \mathcal{X}} p(x) f(x)$

$P(x) = \underline{P}(\mathcal{I}_{\{x\}})$

coherent **linear** prevision
$P : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} P(f+g) = P(f) + P(g) \\ P(f) \geq \inf f, \forall f, g \in \mathcal{L}(\mathcal{X}) \end{cases}$$

Credal / imprecise

set of distributions **convex closed**
$K(X) = \{ \ p(X) \mid \textit{constraints} \ \}$

coherent lower prevision
$\underline{P} : \mathcal{L}(\mathcal{X}) \to \mathbb{R}$

$$\begin{cases} \underline{P}(f) \geq \min f \ \forall f \in \mathcal{L}(\mathcal{X}) \\ \underline{P}(\lambda f) = \lambda \underline{P}(f) \forall \lambda > 0 \\ \underline{P}(f+g) \geq \underline{P}(f) + \underline{P}(g) \end{cases}$$

$\exists$ set $\mathcal{M}$ of linear previsions s.t.
$\underline{P}(f) = \inf_{P \in \mathcal{M}} P(f) \ \forall f \in \mathcal{L}(\mathcal{X})$

lower envelope Theorem

(a set and its convex closure have the same lower envelope!)

Credal sets (Levi, 1980)

- A closed convex set $K(X)$ of probability mass functions
- Equivalently described by its *extreme points* $\mathrm{ext}[K(X)]$
- Focus on categorical variables ($|\mathcal{X}| < +\infty$) and finitely generated CSs (polytopes, $|\mathrm{ext}[K(X)]| < +\infty$)

V-representation	⟷	H-representation
enumerate the	LRS software	linear constraints
extreme points	(Avis & Fukuda)	on the probabilities

- Given a function (gamble) $f(X)$, lower expectation: $\underline{E}[f(X)] := \min_{P(X)\in K(X)} \sum_{x\in\mathcal{X}} P(x)f(x)$
- LP task: the optimum is on an extreme! $\underline{E}[f(X)] = \min_{P(X)\in \mathrm{ext}[K(X)]} \sum_{x\in\mathcal{X}} P(x)f(x)$

Credal sets (Levi, 1980)

- A closed convex set $K(X)$ of probability mass functions
- Equivalently described by its *extreme points* $\mathrm{ext}[K(X)]$
- Focus on categorical variables ($|\mathcal{X}| < +\infty$) and finitely generated CSs (polytopes, $|\mathrm{ext}[K(X)]| < +\infty$)

V-representation	⟷	H-representation
enumerate the	LRS software	linear constraints
extreme points	(Avis & Fukuda)	on the probabilities

- Given a function (gamble) $f(X)$, lower expectation: $\underline{E}[f(X)] := \min_{P(X) \in K(X)} \sum_{x \in \mathcal{X}} P(x) f(x)$
- LP task: the optimum is on an extreme! $\underline{E}[f(X)] = \min_{P(X) \in \mathrm{ext}[K(X)]} \sum_{x \in \mathcal{X}} P(x) f(x)$

Credal sets (Levi, 1980)

- A closed convex set $K(X)$ of probability mass functions
- Equivalently described by its *extreme points* $\mathrm{ext}[K(X)]$
- Focus on categorical variables ($|\mathcal{X}| < +\infty$)

  and finitely generated CSs (polytopes, $|\mathrm{ext}[K(X)]| < +\infty$)

V-representation	⟷	H-representation
enumerate the	LRS software	linear constraints
extreme points	(Avis & Fukuda)	on the probabilities

- Given a function (gamble) $f(X)$, lower expectation:

  $\underline{E}[f(X)] := \min_{P(X) \in K(X)} \sum_{x \in \mathcal{X}} P(x) f(x)$
- LP task: the optimum is on an extreme!

  $\underline{E}[f(X)] = \min_{P(X) \in \mathrm{ext}[K(X)]} \sum_{x \in \mathcal{X}} P(x) f(x)$

Credal sets (Levi, 1980)

- A closed convex set $K(X)$ of probability mass functions
- Equivalently described by its *extreme points* $\mathrm{ext}[K(X)]$
- Focus on categorical variables ($|\mathcal{X}| < +\infty$) and finitely generated CSs (polytopes, $|\mathrm{ext}[K(X)]| < +\infty$)

V-representation	⟷	H-representation
enumerate the	LRS software	linear constraints
extreme points	(Avis & Fukuda)	on the probabilities

- Given a function (gamble) $f(X)$, lower expectation: $\underline{E}[f(X)] := \min_{P(X) \in K(X)} \sum_{x \in \mathcal{X}} P(x) f(x)$
- LP task: the optimum is on an extreme! $\underline{E}[f(X)] = \min_{P(X) \in \mathrm{ext}[K(X)]} \sum_{x \in \mathcal{X}} P(x) f(x)$

Credal sets (Levi, 1980)

- A closed convex set $K(X)$ of probability mass functions
- Equivalently described by its *extreme points* $\mathrm{ext}[K(X)]$
- Focus on categorical variables ($|\mathcal{X}| < +\infty$)
  and finitely generated CSs (polytopes, $|\mathrm{ext}[K(X)]| < +\infty$)

V-representation	⟷	H-representation
enumerate the	LRS software	linear constraints
extreme points	(Avis & Fukuda)	on the probabilities

- Given a function (gamble) $f(X)$, lower expectation:
  $\underline{E}[f(X)] := \min_{P(X)\in K(X)} \sum_{x\in\mathcal{X}} P(x)f(x)$
- LP task: the optimum is on an extreme!
  $\underline{E}[f(X)] = \min_{P(X)\in \mathrm{ext}[K(X)]} \sum_{x\in\mathcal{X}} P(x)f(x)$

Credal sets (Levi, 1980)

- A closed convex set $K(X)$ of probability mass functions
- Equivalently described by its *extreme points* $\mathrm{ext}[K(X)]$
- Focus on categorical variables ($|\mathcal{X}| < +\infty$)
  and finitely generated CSs (polytopes, $|\mathrm{ext}[K(X)]| < +\infty$)

V-representation		H-representation
enumerate the	⟷ LRS software	linear constraints
extreme points	(Avis & Fukuda)	on the probabilities

- Given a function (gamble) $f(X)$, lower expectation:
  $\underline{E}[f(X)] := \min_{P(X)\in K(X)} \sum_{x\in\mathcal{X}} P(x)f(x)$
- LP task: the optimum is on an extreme!
  $\underline{E}[f(X)] = \min_{P(X)\in \mathrm{ext}[K(X)]} \sum_{x\in\mathcal{X}} P(x)f(x)$

Credal sets (Levi, 1980)

- A closed convex set $K(X)$ of probability mass functions
- Equivalently described by its *extreme points* $\mathrm{ext}[K(X)]$
- Focus on categorical variables ($|\mathcal{X}| < +\infty$)
  and finitely generated CSs (polytopes, $|\mathrm{ext}[K(X)]| < +\infty$)

V-representation	⟷	H-representation
enumerate the	LRS software	linear constraints
extreme points	(Avis & Fukuda)	on the probabilities

- Given a function (gamble) $f(X)$, lower expectation:
  $\underline{E}[f(X)] := \min_{P(X)\in K(X)} \sum_{x\in\mathcal{X}} P(x)f(x)$
- LP task: the optimum is on an extreme!
  $\underline{E}[f(X)] = \min_{P(X)\in \mathrm{ext}[K(X)]} \sum_{x\in\mathcal{X}} P(x)f(x)$

Credal Sets over Boolean Variables

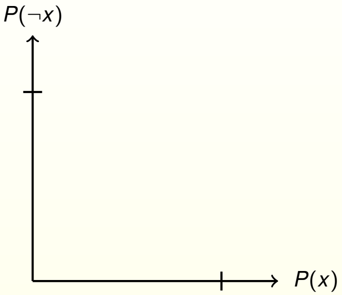
Credal Sets over Boolean Variables

- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$

Credal Sets over Boolean Variables

- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$
- Determinism $\equiv$ degenerate mass f

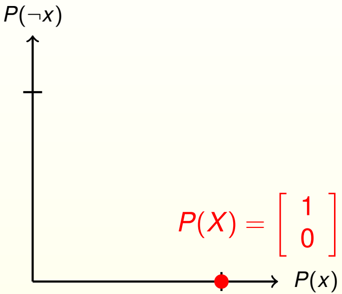
  E.g., $X = x \iff P(X) = \left[\begin{array}{c} 1 \\ 0 \end{array} \right]$



Credal Sets over Boolean Variables

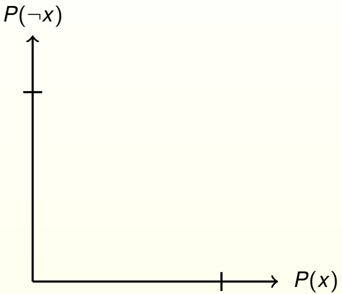
- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$
- Determinism $\equiv$ degenerate mass f

E.g., $X = x \iff P(X) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$



Credal Sets over Boolean Variables

- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$
- Determinism $\equiv$ degenerate mass f
  E.g., $X = x \iff P(X) = \left[\begin{array}{c} 1 \\ 0 \end{array}\right]$
- Uncertainty $\equiv$ prob mass function
  $P(X) = \left[\begin{array}{c} p \\ 1-p \end{array}\right]$ with $p \in [0, 1]$

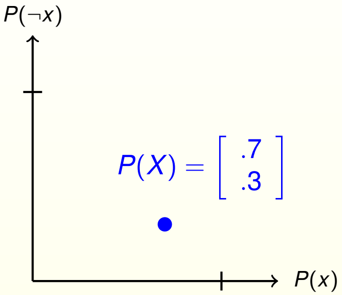


Credal Sets over Boolean Variables

- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$
- Determinism $\equiv$ degenerate mass f

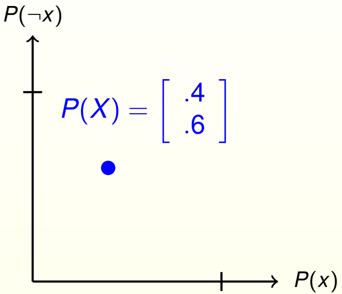
  E.g., $X = x \iff P(X) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- Uncertainty $\equiv$ prob mass function

  $P(X) = \begin{bmatrix} p \\ 1-p \end{bmatrix}$ with $p \in [0, 1]$



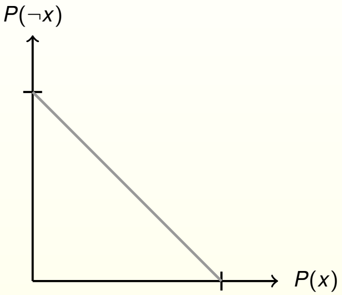
Credal Sets over Boolean Variables

- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$
- Determinism $\equiv$ degenerate mass f
  E.g., $X = x \iff P(X) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- Uncertainty $\equiv$ prob mass function
  $P(X) = \begin{bmatrix} p \\ 1-p \end{bmatrix}$ with $p \in [0, 1]$



Credal Sets over Boolean Variables

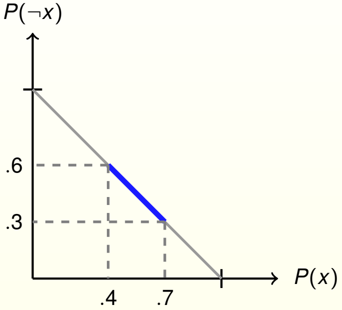
- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$
- Determinism $\equiv$ degenerate mass f
  E.g., $X = x \iff P(X) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- Uncertainty $\equiv$ prob mass function
  $P(X) = \begin{bmatrix} p \\ 1-p \end{bmatrix}$ with $p \in [0, 1]$
- Imprecision credal set
  on the *probability simplex*



Credal Sets over Boolean Variables

- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$
- Determinism $\equiv$ degenerate mass f
  E.g., $X = x \iff P(X) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- Uncertainty $\equiv$ prob mass function
  $P(X) = \begin{bmatrix} p \\ 1-p \end{bmatrix}$ with $p \in [0, 1]$
- Imprecision credal set
  on the *probability simplex*

$$K(X) \equiv \left\{ P(X) = \begin{bmatrix} p \\ 1-p \end{bmatrix} \middle| .4 \leq p \leq .7 \right\}$$



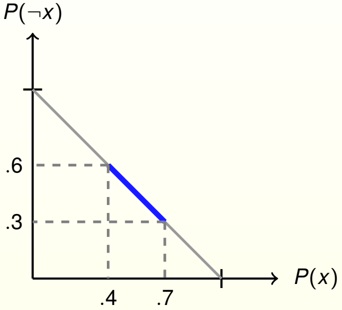
Credal Sets over Boolean Variables

- Boolean X, values in $\mathcal{X} = \{x, \neg x\}$
- Determinism $\equiv$ degenerate mass f
  E.g., $X = x \iff P(X) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- Uncertainty $\equiv$ prob mass function
  $P(X) = \begin{bmatrix} p \\ 1-p \end{bmatrix}$ with $p \in [0,1]$
- Imprecision credal set
  on the *probability simplex*

$$K(X) \equiv \left\{ P(X) = \begin{bmatrix} p \\ 1-p \end{bmatrix} \middle| .4 \leq p \leq .7 \right\}$$

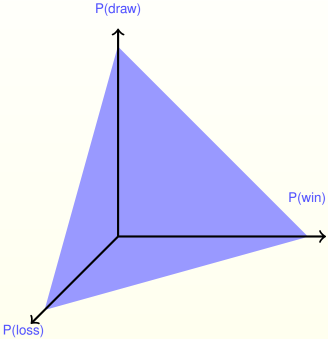
- A CS over a Boolean variable cannot have more than two vertices!

$$\mathrm{ext}[K(X)] = \left\{ \begin{bmatrix} .7 \\ .3 \end{bmatrix}, \begin{bmatrix} .4 \\ .6 \end{bmatrix} \right\}$$



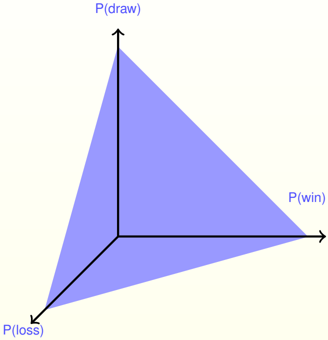
Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{\text{win,draw,loss}\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\text{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



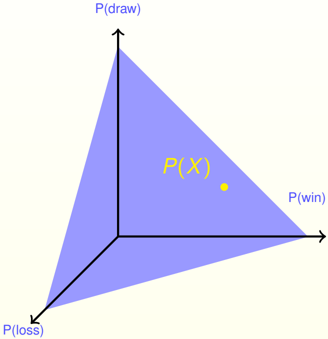
Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\mathrm{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



Geometric Representation of CSs (ternary variables)

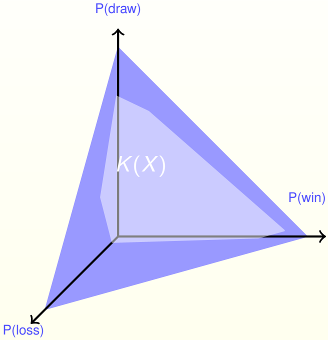
- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\text{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



$$P(X) = \begin{bmatrix} .6 \\ .3 \\ .1 \end{bmatrix}$$

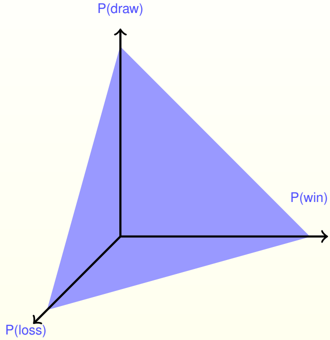
Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\mathrm{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



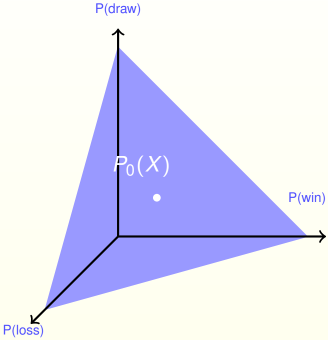
Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\mathrm{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



Geometric Representation of CSs (ternary variables)

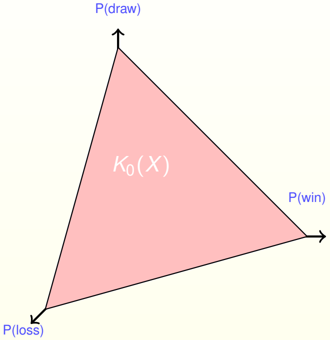
- Ternary X (e.g., $\mathcal{X} = \{\text{win,draw,loss}\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\text{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



$P_0(x) = \frac{1}{|\Omega_X|}$

Geometric Representation of CSs (ternary variables)

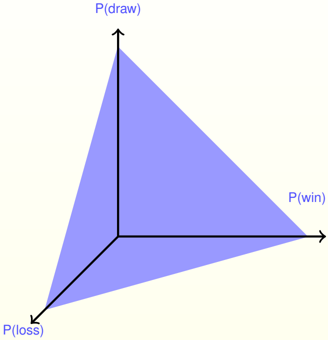
- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\mathrm{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



$$K_0(X) = \left\{ P(X) \;\middle|\; \begin{array}{l} \sum_x P(x) = 1, \\ P(x) \geq 0 \end{array} \right\}$$

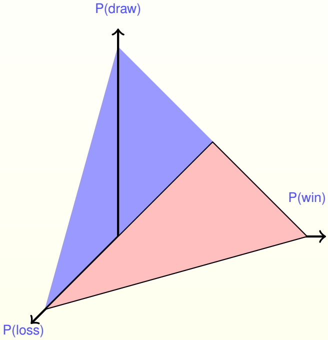
Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\mathrm{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



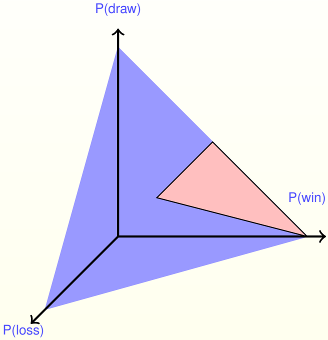
Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\mathrm{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\mathrm{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\text{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data

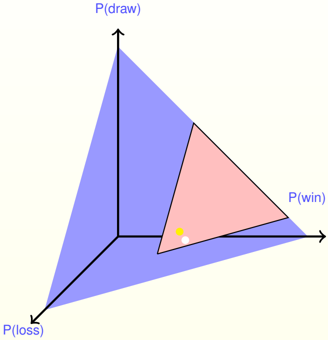
From natural language to linear contraints on probabilities

(Walley, 1991)

extremely probable $P(x) \geq 0.98$
very high probability $P(x) \geq 0.9$
highly probable $P(x) \geq 0.85$
very probable $P(x) \geq 0.75$
has a very good chance $P(x) \geq 0.65$
quite probable $P(x) \geq 0.6$
$P(x) \geq 0.5$
has a good chance $0.4 \leq P(x) \leq 0.85$
is improbable (unlikely) $P(x) \leq 0.5$
is somewhat unlikely $P(x) \leq 0.4$
is very unlikely $P(x) \leq 0.25$
has little chance $P(x) \leq 0.2$
is highly improbable $P(x) \leq 0.15$
is has very low probability $P(x) \leq 0.1$
is extremely unlikely $P(x) \leq 0.02$

Geometric Representation of CSs (ternary variables)

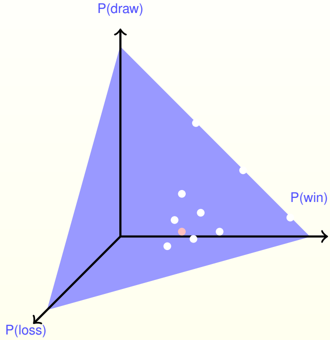
- Ternary X (e.g., $\mathcal{X} = \{\text{win,draw,loss}\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\text{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



Previous matches:
Holland 4 wins,
Draws 1,
Spain 3 wins

Geometric Representation of CSs (ternary variables)

- Ternary X (e.g., $\mathcal{X} = \{$win,draw,loss$\}$)
- $P(X) \equiv$ point in the space (simplex)
- No bounds to $|\text{ext}[K(X)]|$
- Modelling ignorance
  - Uniform models indifference
  - Vacuous credal set
- Expert qualitative knowledge
  - Win is more probable than draw, which more probable than loss
- Learning from small datasets
  - Learning with multiple priors (e.g., IDM with $s = 2$)
- Learning from incomplete data
  - Considering all the possible explanation of the missing data



1957:	Spain	vs.	Holland	5 — 1	
1973:	Holland	vs.	Spain	3 — 2	
1980:	Spain	vs.	Holland	1 — 0	
1983:	Spain	vs.	Holland	1 — 0	
1983:	Holland	vs.	Spain	2 — 1	
1987:	Spain	vs.	Holland	1 — 1	
2000:	Spain	vs.	Holland	1 — 2	
2001:	Holland	vs.	Spain	1 — 0	
2005:	Spain	vs.	Holland	$*$ — $*$	(missing)
2008:	Holland	vs.	Spain	$*$ — $*$	(missing)

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X,Y)$	$K(X,Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x,y)$	$K(X) = \left\{ P(X) \,\middle\vert\, \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \,\middle\vert\, \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Combination	$P(x,y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y)=P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X,Y)$	$K(X,Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x,y)$	$K(X) = \left\{ P(X) \,\middle\vert\, \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \,\middle\vert\, \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Combination	$P(x,y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y)=P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X, Y)$	$K(X, Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x, y)$	$K(X) = \left\{ P(X) \,\middle\vert\, \begin{array}{l} P(x) = \sum_y P(x, y) \\ P(X, Y) \in K(X, Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \,\middle\vert\, \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X, Y) \in K(X, Y) \end{array} \right\}$
Combination	$P(x, y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X, Y) \,\middle\vert\, \begin{array}{l} P(x, y) = P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X,Y)$	$K(X,Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x,y)$	$K(X) = \left\{ P(X) \,\middle\vert\, \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \,\middle\vert\, \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Combination	$P(x,y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y)=P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X,Y)$	$K(X,Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x,y)$	$K(X) = \left\{ P(X) \,\middle|\, \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \,\middle|\, \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Combination	$P(x,y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \,\middle|\, \begin{array}{l} P(x,y)=P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X,Y)$	$K(X,Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x,y)$	$K(X) = \left\{ P(X) \middle\vert \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \middle\vert \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Combination	$P(x,y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \middle\vert \begin{array}{l} P(x,y) = P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X,Y)$	$K(X,Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x,y)$	$K(X) = \left\{ P(X) \,\middle\vert\, \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \,\middle\vert\, \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Combination	$P(x,y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y) = P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X,Y)$	$K(X,Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x,y)$	$K(X) = \left\{ P(X) \,\middle\vert\, \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \,\middle\vert\, \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Combination	$P(x,y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y)=P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets

	PRECISE Mass functions	IMPRECISE Credal sets
Joint	$P(X,Y)$	$K(X,Y)$
Marginalization	$P(X)$ s.t. $p(x) = \sum_y p(x,y)$	$K(X) = \left\{ P(X) \,\middle\vert\, \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Conditioning	$P(X\|y)$ s.t. $p(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)}$	$K(X\|y) = \left\{ P(X\|y) \,\middle\vert\, \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$
Combination	$P(x,y) = P(x\|y)P(y)$	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y) = P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$

Basic operations with credal sets (vertices)

	IMPRECISE Credal sets	IMPRECISE Extremes
Joint	$K(X,Y)$	
Marginalization	$K(X) = \left\{ P(X) \;\middle|\; \begin{array}{l} P(x) = \sum_y P(x,y) \\ P(X,Y) \in K(X,Y) \end{array} \right\}$	
Conditioning	$K(X\|y) = \left\{ P(X\|y) \;\middle|\; \begin{array}{l} P(x\|y) = \frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y) \in K(X,Y) \end{array} \right\}$	
Combination	$K(X\|Y) \otimes K(Y) = \left\{ P(X,Y) \;\middle|\; \begin{array}{l} P(x,y)=P(x\|y)P(y) \\ P(X\|y) \in K(X\|y) \\ P(Y) \in K(Y) \end{array} \right\}$	

Basic operations with credal sets (vertices)

	IMPRECISE Credal sets	IMPRECISE Extremes
Joint	$K(X,Y)$	$= \mathrm{CH}\left\{P_j(X,Y)\right\}_{j=1}^{n_v}$
Marginalization	$K(X) = \left\{P(X) \,\middle\vert\, \begin{array}{l} P(x)=\sum_y P(x,y) \\ P(X,Y)\in K(X,Y)\end{array}\right\}$	$= \mathrm{CH} \left\{P(X) \,\middle\vert\, \begin{array}{l} P(x)=\sum_y P(x,y) \\ P(X,Y)\in \mathrm{ext}[K(X,Y)]\end{array}\right\}$
Conditioning	$K(X\vert y) = \left\{P(X\vert y) \,\middle\vert\, \begin{array}{l} P(x\vert y)=\frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y)\in K(X,Y)\end{array}\right\}$	$= \mathrm{CH} \left\{P(X\vert y) \,\middle\vert\, \begin{array}{l} P(x\vert y)=\frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y)\in \mathrm{ext}[K(X,Y)]\end{array}\right\}$
Combination	$K(X\vert Y)\otimes K(Y) = \left\{P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y)=P(x\vert y)P(y) \\ P(X\vert y)\in K(X\vert y) \\ P(Y)\in K(Y)\end{array}\right\}$	$= \mathrm{CH} \left\{P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y)=P(x\vert y)P(y) \\ P(X\vert y)\in \mathrm{ext}[K(X\vert y)] \\ P(Y)\in \mathrm{ext}[K(Y)]\end{array}\right\}$

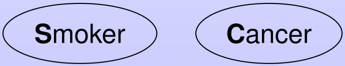
Basic operations with credal sets (vertices)

	IMPRECISE Credal sets	IMPRECISE Extremes
Joint	$K(X,Y)$	$= \mathrm{CH}\left\{P_j(X,Y)\right\}_{j=1}^{n_v}$
Marginalization	$K(X) = \left\{P(X) \,\middle\vert\, \begin{array}{l} P(x)=\sum_y P(x,y) \\ P(X,Y)\in K(X,Y)\end{array}\right\}$	$= \mathrm{CH} \left\{P(X) \,\middle\vert\, \begin{array}{l} P(x)=\sum_y P(x,y) \\ P(X,Y)\in \mathrm{ext}[K(X,Y)]\end{array}\right\}$
Conditioning	$K(X\vert y) = \left\{P(X\vert y) \,\middle\vert\, \begin{array}{l} P(x\vert y)=\frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y)\in K(X,Y)\end{array}\right\}$	$= \mathrm{CH} \left\{P(X\vert y) \,\middle\vert\, \begin{array}{l} P(x\vert y)=\frac{P(x,y)}{\sum_y P(x,y)} \\ P(X,Y)\in \mathrm{ext}[K(X,Y)]\end{array}\right\}$
Combination	$K(X\vert Y)\otimes K(Y) = \left\{P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y)=P(x\vert y)P(y) \\ P(X\vert y)\in K(X\vert y) \\ P(Y)\in K(Y)\end{array}\right\}$	$= \mathrm{CH} \left\{P(X,Y) \,\middle\vert\, \begin{array}{l} P(x,y)=P(x\vert y)P(y) \\ P(X\vert y)\in \mathrm{ext}[K(X\vert y)] \\ P(Y)\in \mathrm{ext}[K(Y)]\end{array}\right\}$

[EXE]
Prove it!

[Exe #2] An imprecise bivariate (graphical?) model

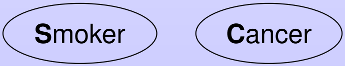
- Two Boolean variables: **S**moker, Lung **C**ancer



[Exe #2] An imprecise bivariate (graphical?) model

- Two Boolean variables: **S**moker, Lung **C**ancer
- Eight "Bayesian" phisicians, each one assessing $P_j(S, C)$

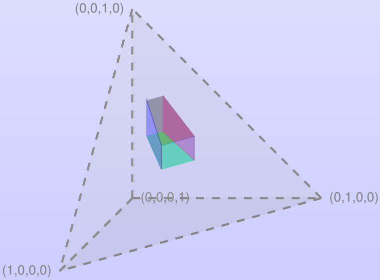
j	$P_j(s, c)$	$P_j(s, \neg c)$	$P_j(\neg s, c)$	$P_j(\neg s, \neg c)$
1	1/8	1/8	3/8	3/8
2	1/8	1/8	9/16	3/16
3	3/16	1/16	3/8	3/8
4	3/16	1/16	9/16	3/16
5	1/4	1/4	1/4	1/4
6	1/4	1/4	3/8	1/8
7	3/8	1/8	1/4	1/4
8	3/8	1/8	3/8	1/8

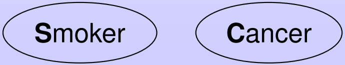


[Exe #2] An imprecise bivariate (graphical?) model

- Two Boolean variables: **S**moker, Lung **C**ancer
- Eight "Bayesian" phisicians, each one assessing $P_j(S,C)$
- $K(S,C) = \mathrm{CH}\left\{P_j(S,C)\right\}_{j=1}^{8}$

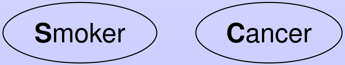
j	$P_j(s,c)$	$P_j(s,\neg c)$	$P_j(\neg s,c)$	$P_j(\neg s,\neg c)$
1	1/8	1/8	3/8	3/8
2	1/8	1/8	9/16	3/16
3	3/16	1/16	3/8	3/8
4	3/16	1/16	9/16	3/16
5	1/4	1/4	1/4	1/4
6	1/4	1/4	3/8	1/8
7	3/8	1/8	1/4	1/4
8	3/8	1/8	3/8	1/8



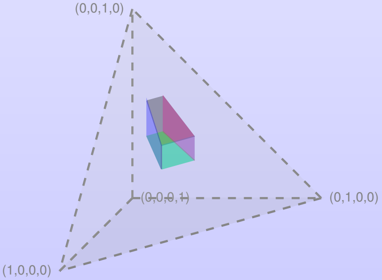


[Exe #2] An imprecise bivariate (graphical?) model

- Two Boolean variables: **S**moker, Lung **C**ancer
- Eight “Bayesian” phisicians, each one assessing $P_j(S, C)$
- $K(S, C) = \mathrm{CH}\left\{P_j(S, C)\right\}_{j=1}^{8}$
- Compute:
  - Marginal $K(S)$
  - Conditioning $K(C|S) := \{K(C|s), K(C|s)\}$
  - Combination (marg ext) $K'(C, S) := K(C|S) \otimes K(S)$

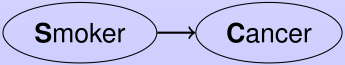


j	$P_j(s, c)$	$P_j(s, \neg c)$	$P_j(\neg s, c)$	$P_j(\neg s, \neg c)$
1	1/8	1/8	3/8	3/8
2	1/8	1/8	9/16	3/16
3	3/16	1/16	3/8	3/8
4	3/16	1/16	9/16	3/16
5	1/4	1/4	1/4	1/4
6	1/4	1/4	3/8	1/8
7	3/8	1/8	1/4	1/4
8	3/8	1/8	3/8	1/8

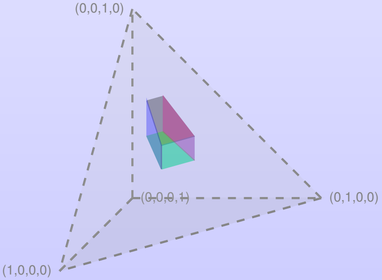


[Exe #2] An imprecise bivariate (graphical?) model

- Two Boolean variables: **S**moker, Lung **C**ancer
- Eight “Bayesian” phisicians, each one assessing $P_j(S,C)$
- $K(S,C) = \mathrm{CH}\left\{P_j(S,C)\right\}_{j=1}^{8}$
- Compute:
  - Marginal $K(S)$
  - Conditioning $K(C|S) := \{K(C|s), K(C|s)\}$
  - Combination (marg ext) $K'(C,S) := K(C|S) \otimes K(S)$
- Is this a (I)PGM?

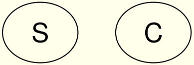


j	$P_j(s,c)$	$P_j(s,\neg c)$	$P_j(\neg s,c)$	$P_j(\neg s,\neg c)$
1	1/8	1/8	3/8	3/8
2	1/8	1/8	9/16	3/16
3	3/16	1/16	3/8	3/8
4	3/16	1/16	9/16	3/16
5	1/4	1/4	1/4	1/4
6	1/4	1/4	3/8	1/8
7	3/8	1/8	1/4	1/4
8	3/8	1/8	3/8	1/8



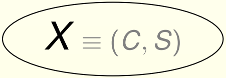
Cano-Cano-Moral Transformation

Cano-Cano-Moral Transformation



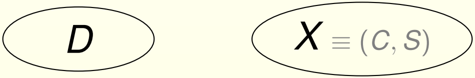
Cano-Cano-Moral Transformation

- Joint variable $X := (C, S)$, $K(X) = \{P_j(X)\}_{j=1}^{n_v}$ $(|\mathcal{X} = 4|$ and $n_v = 8)$



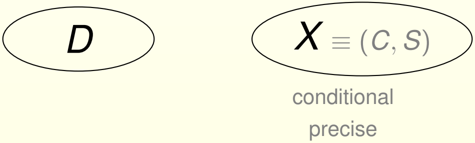
Cano-Cano-Moral Transformation

- Joint variable $X := (C, S)$, $K(X) = \{P_j(X)\}_{j=1}^{n_v}$ ($|\mathcal{X} = 4|$ and $n_v = 8$)
- Auxiliary variable D to enumerate elements of $\mathrm{ext}[K(X)]$



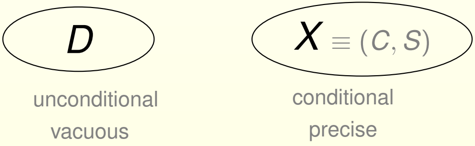
Cano-Cano-Moral Transformation

- Joint variable $X := (C, S)$, $K(X) = \{P_j(X)\}_{j=1}^{n_v}$ ($|\mathcal{X} = 4|$ and $n_v = 8$)
- Auxiliary variable D to enumerate elements of $\mathrm{ext}[K(X)]$
  - Precise $K(X|D)$, with $P(X|d_j) := P_j(X)$ with $|\mathcal{D}| = 8$



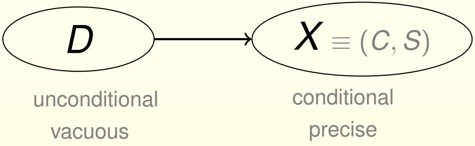
Cano-Cano-Moral Transformation

- Joint variable $X := (C, S)$, $K(X) = \{P_j(X)\}_{j=1}^{n_v}$ ($|\mathcal{X} = 4|$ and $n_v = 8$)
- Auxiliary variable D to enumerate elements of $\mathrm{ext}[K(X)]$
  - Precise $K(X|D)$, with $P(X|d_j) := P_j(X)$ with $|\mathcal{D}| = 8$
  - Vacuous $K(D)$ (8 vertices)



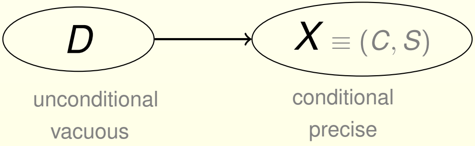
Cano-Cano-Moral Transformation

- Joint variable $X := (C, S)$, $K(X) = \{P_j(X)\}_{j=1}^{n_v}$ ($|\mathcal{X} = 4|$ and $n_v = 8$)
- Auxiliary variable D to enumerate elements of $\mathrm{ext}[K(X)]$
  - Precise $K(X|D)$, with $P(X|d_j) := P_j(X)$ with $|\mathcal{D}| = 8$
  - Vacuous $K(D)$ (8 vertices)
- Marginal extension $K(X, D) = K(X|D) \otimes K(D)$



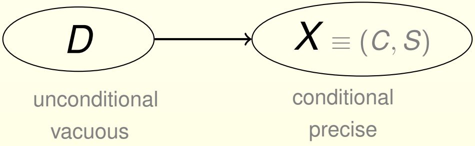
Cano-Cano-Moral Transformation

- Joint variable $X := (C, S)$, $K(X) = \{P_j(X)\}_{j=1}^{n_v}$ ($|\mathcal{X} = 4|$ and $n_v = 8$)
- Auxiliary variable D to enumerate elements of $\mathrm{ext}[K(X)]$
  - Precise $K(X|D)$, with $P(X|d_j) := P_j(X)$ with $|\mathcal{D}| = 8$
  - Vacuous $K(D)$ (8 vertices)
- Marginal extension $K(X, D) = K(X|D) \otimes K(D)$
- After marginalization of D, $K(X, D)$ becomes $K(X)$



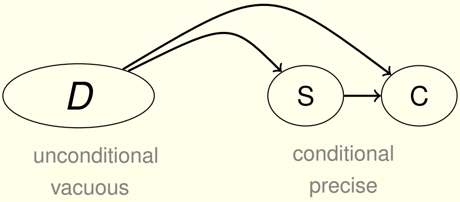
Cano-Cano-Moral Transformation

- Joint variable $X := (C, S)$, $K(X) = \{P_j(X)\}_{j=1}^{n_v}$ ($|\mathcal{X} = 4|$ and $n_v = 8$)
- Auxiliary variable D to enumerate elements of $\mathrm{ext}[K(X)]$
  - Precise $K(X|D)$, with $P(X|d_j) := P_j(X)$ with $|\mathcal{D}| = 8$
  - Vacuous $K(D)$ (8 vertices)
- Marginal extension $K(X, D) = K(X|D) \otimes K(D)$
- After marginalization of D, $K(X, D)$ becomes $K(X)$
- IP models as combination of precise models and vacuous priors
- Hyerarchical model with a vacuous second order knowledge



Cano-Cano-Moral Transformation

- Joint variable $X := (C, S)$, $K(X) = \{P_j(X)\}_{j=1}^{n_v}$ ($|\mathcal{X} = 4|$ and $n_v = 8$)
- Auxiliary variable D to enumerate elements of $\text{ext}[K(X)]$
  - Precise $K(X|D)$, with $P(X|d_j) := P_j(X)$ with $|\mathcal{D}| = 8$
  - Vacuous $K(D)$ (8 vertices)
- Marginal extension $K(X, D) = K(X|D) \otimes K(D)$
- After marginalization of D, $K(X, D)$ becomes $K(X)$
- IP models as combination of precise models and vacuous priors
- Hyerarchical model with a vacuous second order knowledge



Independence

Independence

Stochastic independence/irrelevance (precise case)

- X and Y stochastically independent: $P(x, y) = P(x)P(y)$
- Y stochastically irrelevant to X: $P(X|y) = P(X)$
- independence $\Rightarrow$ irrelevance

Independence

Stochastic independence/irrelevance (precise case)

- X and Y stochastically independent: $P(x,y) = P(x)P(y)$
- Y stochastically irrelevant to X: $P(X|y) = P(X)$
- independence $\Rightarrow$ irrelevance

Strong independence/irrelevance (imprecise case)

- X and Y strongly independent: stochastic independence $\forall P(X,Y) \in \mathrm{ext}[K(X,Y)]$
- Y strongly irrelevant to X: $K(X|y) = K(X)$
- independence $\Rightarrow$ irrelevance

Independence

Stochastic independence/irrelevance (precise case)

- X and Y stochastically independent: $P(x, y) = P(x)P(y)$
- Y stochastically irrelevant to X: $P(X|y) = P(X)$
- independence $\Rightarrow$ irrelevance

Strong independence/irrelevance (imprecise case)

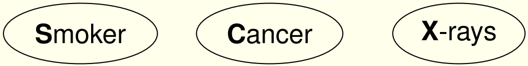
- X and Y strongly independent: stochastic independence $\forall P(X, Y) \in \mathrm{ext}[K(X, Y)]$
- Y strongly irrelevant to X: $K(X|y) = K(X)$
- independence $\Rightarrow$ irrelevance

Every notion admits a conditional formulation

Other IP independence concepts (epistemic, Kuznetzov, strict)

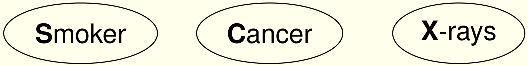
A tri-variate example

- 3 Boolean variables: **S**moker, Lung **C**ancer, **X**-rays



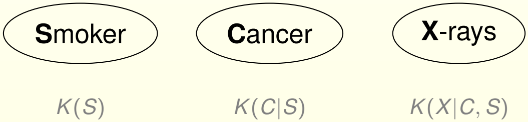
A tri-variate example

- 3 Boolean variables: **S**moker, Lung **C**ancer, **X**-rays
- Given cancer, no relation between smoker and X-rays
- IP language: given C, S and X strongly independent



A tri-variate example

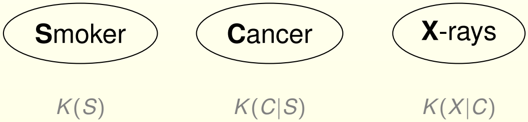
- 3 Boolean variables: **S**moker, Lung **C**ancer, **X**-rays
- Given cancer, no relation between smoker and X-rays
- IP language: given C, S and X strongly independent
- Marginal extension (iterated two times)
  $K(S,C,X) = K(X|C,S) \otimes K(C,S) = K(X|C,S) \otimes K(C|S) \otimes K(S)$



A tri-variate example

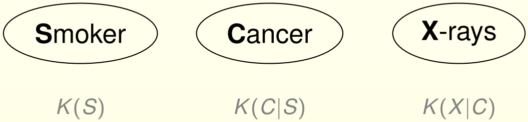
- 3 Boolean variables: **S**moker, Lung **C**ancer, **X**-rays
- Given cancer, no relation between smoker and X-rays
- IP language: given *C*, *S* and *X* strongly independent
- Marginal extension (iterated two times)
  $K(S,C,X) = K(X|C,S) \otimes K(C,S) = K(X|C,S) \otimes K(C|S) \otimes K(S)$
- Independence implies irrelevance: given *C*, *S* irrelevant to *X*

$$K(S,C,X) = K(X|C) \otimes K(C|S) \otimes K(S)$$



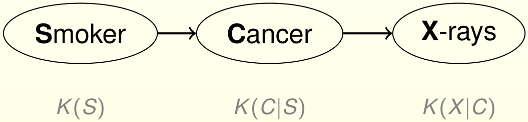
A tri-variate example

- 3 Boolean variables: **S**moker, Lung **C**ancer, **X**-rays
- Given cancer, no relation between smoker and X-rays
- IP language: given C, S and X strongly independent
- Marginal extension (iterated two times)
  $K(S,C,X) = K(X|C,S) \otimes K(C,S) = K(X|C,S) \otimes K(C|S) \otimes K(S)$
- Independence implies irrelevance: given C, S irrelevant to X
  $$K(S,C,X) = K(X|C) \otimes K(C|S) \otimes K(S)$$
- Global model decomposed in 3 "local" models



A tri-variate example

- 3 Boolean variables: **S**moker, Lung **C**ancer, **X**-rays
- Given cancer, no relation between smoker and X-rays
- IP language: given C, S and X strongly independent
- Marginal extension (iterated two times)
  $K(S,C,X) = K(X|C,S) \otimes K(C,S) = K(X|C,S) \otimes K(C|S) \otimes K(S)$
- Independence implies irrelevance: given C, S irrelevant to X
  $$K(S,C,X) = K(X|C) \otimes K(C|S) \otimes K(S)$$
- Global model decomposed in 3 “local” models
- A true PGM! Needed: language to express independencies



Markov Condition

- Probabilistic model over set of variables $(X_1, \ldots, X_n)$
  in one-to-one correspondence with the nodes of a graph

Markov Condition

- Probabilistic model over set of variables $(X_1, \ldots, X_n)$ in one-to-one correspondence with the nodes of a graph

Undirected Graphs

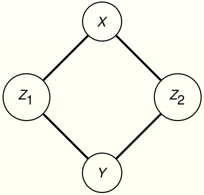
X and Y are independent given Z
if any path between X and Y
containts an element of Z

Markov Condition

- Probabilistic model over set of variables $(X_1, \ldots, X_n)$ in one-to-one correspondence with the nodes of a graph

Undirected Graphs

X and Y are independent given Z
if any path between X and Y
containts an element of Z

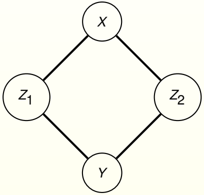


Markov Condition

- Probabilistic model over set of variables $(X_1, \ldots, X_n)$ in one-to-one correspondence with the nodes of a graph

Undirected Graphs

X and Y are independent given Z
if any path between X and Y
containts an element of Z



Directed Graphs

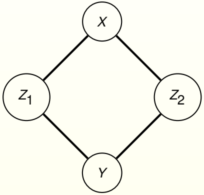
Given its parents, every node is independent of
its non-descendants non-parents

Markov Condition

- Probabilistic model over set of variables $(X_1, \ldots, X_n)$ in one-to-one correspondence with the nodes of a graph

Undirected Graphs

X and Y are independent given Z if any path between X and Y containts an element of Z



Directed Graphs

Given its parents, every node is independent of its non-descendants non-parents

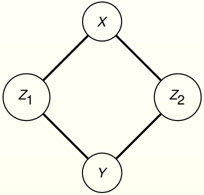
X and Y are d-separated by Z if, along every path between X and Y there is a W such that either W has converging arrows and is not in Z and none of its descendants are in Z, or W has no converging arrows and is in Z

Markov Condition

- Probabilistic model over set of variables $(X_1, \ldots, X_n)$ in one-to-one correspondence with the nodes of a graph

Undirected Graphs

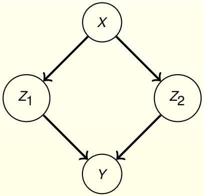
X and Y are independent given Z if any path between X and Y containts an element of Z



Directed Graphs

Given its parents, every node is independent of its non-descendants non-parents

X and Y are d-separated by Z if, along every path between X and Y there is a W such that either W has converging arrows and is not in Z and none of its descendants are in Z, or W has no converging arrows and is in Z



Credal networks

Credal networks

- A directed acyclic graph $\mathcal{G}$ whose nodes are in 1-to-1 correspondence with a set of (categorical) variables $(X_1, \ldots, X_n)$

Credal networks

- A directed acyclic graph $\mathcal{G}$ whose nodes are in 1-to-1 correspondence with a set of (categorical) variables $(X_1, \ldots, X_n)$
- Graph semantics by Markov condition (for directed graphs, strong independence)

Credal networks

- A directed acyclic graph $\mathcal{G}$ whose nodes are in 1-to-1 correspondence with a set of (categorical) variables $(X_1, \ldots, X_n)$
- Graph semantics by Markov condition (for directed graphs, strong independence)
- Variables in topological order (X_j descendant of X_i implies $j > i$)

$$K(X_1, \ldots, X_n) = \otimes_{j=n}^{1} K(X_j | X_{j-1}, \ldots, X_1) = \otimes_{j=n}^{1} K(X_j | \mathrm{Pa}(X_j))$$

Credal networks

- A directed acyclic graph $\mathcal{G}$ whose nodes are in 1-to-1 correspondence with a set of (categorical) variables $(X_1, \ldots, X_n)$
- Graph semantics by Markov condition (for directed graphs, strong independence)
- Variables in topological order (X_j descendant of X_i implies $j > i$)

$$K(X_1, \ldots, X_n) = \otimes_{j=n}^{1} K(X_j | X_{j-1}, \ldots, X_1) = \otimes_{j=n}^{1} K(X_j | \mathrm{Pa}(X_j))$$

directed graph
(with Markov condition)
$\mathcal{G}$

conditional credal sets
$\{K(X_i | \mathrm{Pa}(X_i))\}_{i=1}^{n}\}$

credal network
specification

Credal networks

- A directed acyclic graph $\mathcal{G}$ whose nodes are in 1-to-1 correspondence with a set of (categorical) variables $(X_1, \ldots, X_n)$
- Graph semantics by Markov condition (for directed graphs, strong independence)
- Variables in topological order (X_j descendant of X_i implies $j > i$)

$$K(X_1, \ldots, X_n) = \otimes_{j=n}^{1} K(X_j | X_{j-1}, \ldots, X_1) = \otimes_{j=n}^{1} K(X_j | \mathrm{Pa}(X_j))$$

directed graph
(with Markov condition)
$\mathcal{G}$

conditional credal sets
$\{K(X_i|\mathrm{Pa}(X_i))\}_{i=1}^{n}\}$

→

joint credal set
$K(X_1, \ldots, X_n) \;=\; \otimes_{i=1}^{n} K(X_i|\mathrm{Pa}(X_i))$

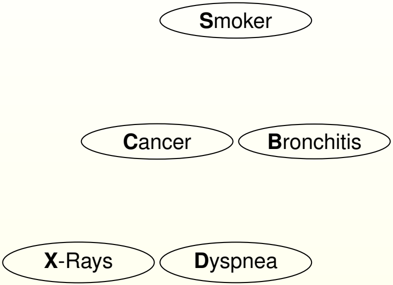
credal network specification

credal network strong extension

Medical diagnosis by CNs (a simple example of)

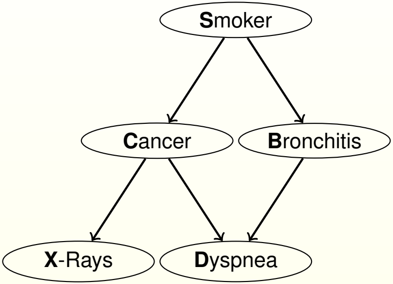
Medical diagnosis by CNs (a simple example of)

- Five Boolean vars



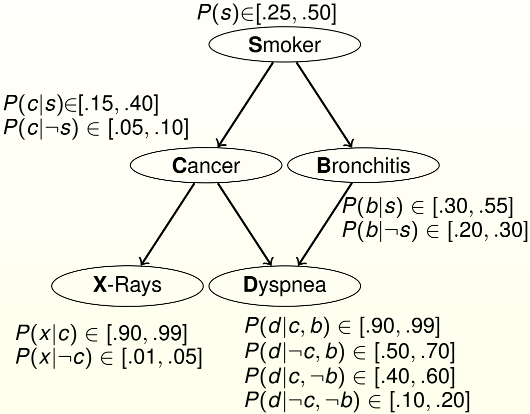
Medical diagnosis by CNs (a simple example of)

- Five Boolean vars
- Conditional independence relations by a DAG



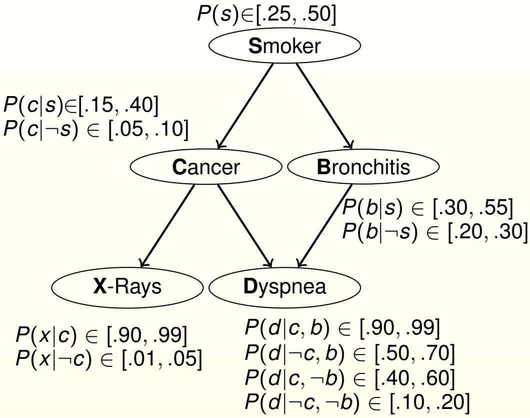
Medical diagnosis by CNs (a simple example of)

- Five Boolean vars
- Conditional independence relations by a DAG
- Elicitation of the local (conditional) CSs



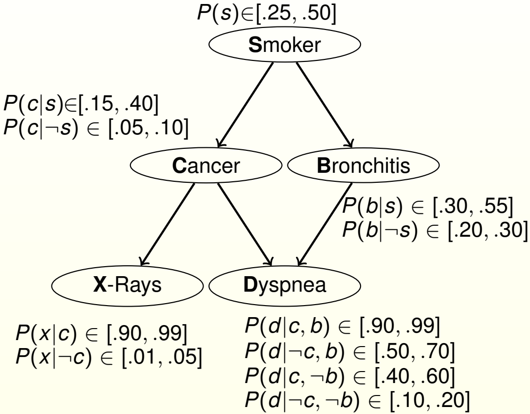
Medical diagnosis by CNs (a simple example of)

- Five Boolean vars
- Conditional independence relations by a DAG
- Elicitation of the local (conditional) CSs
- This is a CN specification



Medical diagnosis by CNs (a simple example of)

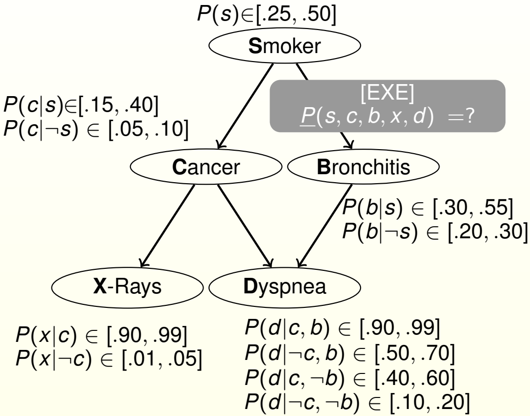
- Five Boolean vars
- Conditional independence relations by a DAG
- Elicitation of the local (conditional) CSs
- This is a CN specification
- The strong extension $K(S, C, B, X, D) =$



$$\mathrm{CH}\left\{ P(S, C, B, X, D) \;\middle|\; \begin{array}{l} P(s, c, b, x, d) = P(s)P(c|s)P(b|s)P(x|c)P(d|c, b) \\ P(S) \in K(S) \\ P(C|s) \in K(C|s), P(C|\neg s) \in K(C|\neg s) \\ \dots \end{array} \right\}$$

Medical diagnosis by CNs (a simple example of)

- Five Boolean vars
- Conditional independence relations by a DAG
- Elicitation of the local (conditional) CSs
- This is a CN specification
- The strong extension $K(S, C, B, X, D) =$



$$\mathrm{CH}\left\{ P(S, C, B, X, D) \;\middle|\; \begin{array}{l} P(s, c, b, x, d) = P(s)P(c|s)P(b|s)P(x|c)P(d|c, b) \\ P(S) \in K(S) \\ P(C|s) \in K(C|s), P(C|\neg s) \in K(C|\neg s) \\ \ldots \end{array} \right\}$$

Inside the strong extension

- A CN whose conditional credal sets are all precise?
  $K(X|\mathrm{pa}(X_i)) = \{P(X|\mathrm{pa}(X_i))\}$

Inside the strong extension

- A CN whose conditional credal sets are all precise?
  $K(X|\mathrm{pa}(X_i)) = \{P(X|\mathrm{pa}(X_i)\}$
- This is a Bayesian network (Pearl, 1988).
  Its SE still precise: $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$

Inside the strong extension

- A CN whose conditional credal sets are all precise? $K(X|\mathrm{pa}(X_i)) = \{P(X|\mathrm{pa}(X_i)\}$
- This is a Bayesian network (Pearl, 1988). Its SE still precise: $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- "Operational" definition of SE of a CN:$K(X_1, \ldots, X_n) =$

$$\mathrm{CH}\left\{ P(X_1, \ldots, X_n) \middle| \begin{array}{l} P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i)) \\ P(X_i|\mathrm{pa}(X_i)) \in \mathrm{ext}[K(X_i|\mathrm{pa}(X_i))] \end{array} \right\}$$

Inside the strong extension

- A CN whose conditional credal sets are all precise?
  $K(X|\mathrm{pa}(X_i)) = \{P(X|\mathrm{pa}(X_i)\}$
- This is a Bayesian network (Pearl, 1988).
  Its SE still precise: $P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))$
- "Operational" definition of SE of a CN: $K(X_1, \ldots, X_n) =$

$$\mathrm{CH}\left\{ P(X_1, \ldots, X_n) \middle| \begin{array}{l} P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i)) \\ P(X_i|\mathrm{pa}(X_i)) \in \mathrm{ext}[K(X_i|\mathrm{pa}(X_i))] \end{array} \right\}$$

- The vertices of the SE correspond to Bayesian networks!
- A CN = collection of BN (all with the same graph) *n* exponential
- Sensitivity analysis interpretation
  $\mathrm{ext}[K(X_1, \ldots, X_n)] = \{P_j(X_1, \ldots, X_n)\}_{j=1}^{n}$

Non-separately specified CNs

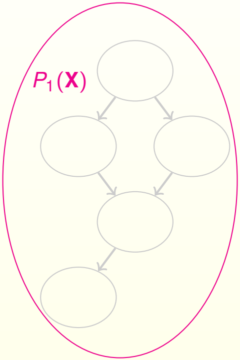
- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*

Non-separately specified CNs

- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*

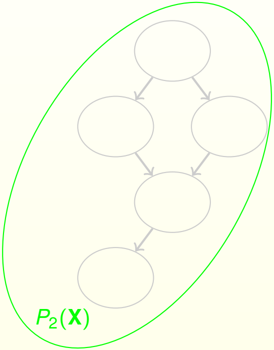
Non-separately specified CNs

- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*



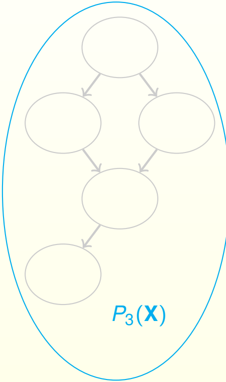
Non-separately specified CNs

- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*



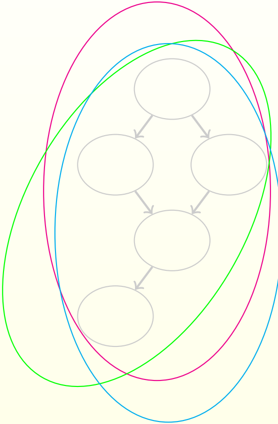
Non-separately specified CNs

- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*



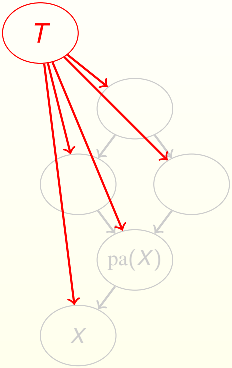
Non-separately specified CNs

- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*



Non-separately specified CNs

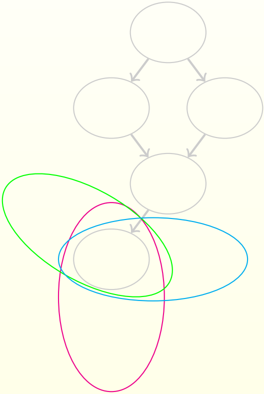
- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*



$$P(X|\mathrm{pa}(X), T = t_j) = P_j(X|\mathrm{pa}(X))$$

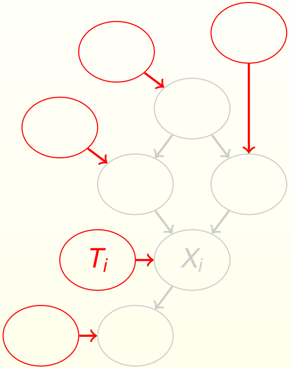
Non-separately specified CNs

- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*



Non-separately specified CNs

- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*



Non-separately specified CNs

- Constraints among different conditional mass functions of a CN
- Explicit enumeration of the relative BNs
  - Auxiliary parent selecting the conditional probabilities *(Cano, Cano, Moral, 1994)* with a vacuous prior
- "Extensive" specification
  - Constraints among conditional mass functions of the same variable
  - Each CPT takes values from a set of tables an auxiliary parent selecting the tables
- An unconstrained (i.e., separated) specification is always possible *(Antonucci & Zaffalon, IJAR, 2008)*

Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case:

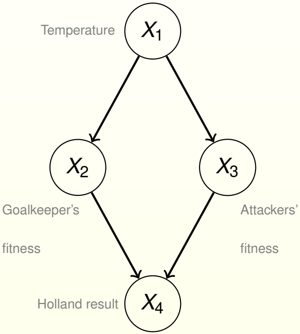
$$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$

with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$

- Fast algorithm for singly-connected BNs
- Credal case:

$$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$

- Fast algorithm for binary singly-connected CNs



Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case:

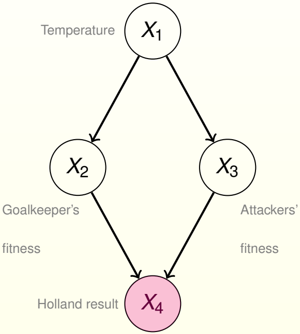
$$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$

with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$

- Fast algorithm for singly-connected BNs
- Credal case:

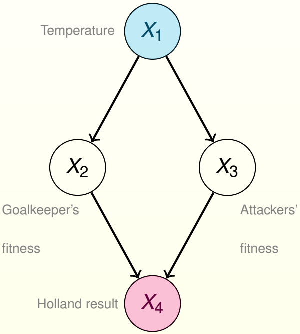
$$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$

- Fast algorithm for binary singly-connected CNs



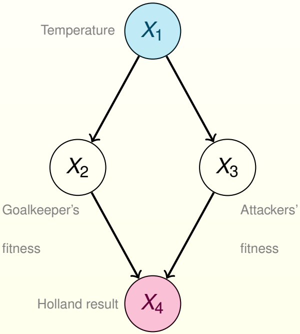
Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case:
  $$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$
  with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$
- Fast algorithm for singly-connected BNs
- Credal case:
  $$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$
- Fast algorithm for binary singly-connected CNs



Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case:
  $$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$
  with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$
- Fast algorithm for singly-connected BNs
- Credal case:
  $$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$
- Fast algorithm for binary singly-connected CNs



Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case:

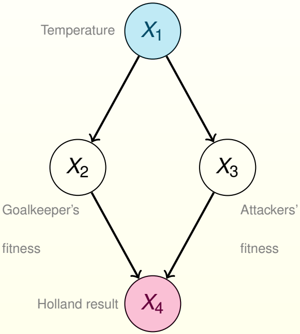
$$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$

with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$

- Fast algorithm for singly-connected BNs
- Credal case:

$$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$

- Fast algorithm for binary singly-connected CNs



$$P(x_4|x_1) = \frac{\sum_{x_2, x_3, x_4} P(\ldots)}{\sum_{x_2, x_3} P(\ldots)}$$

Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case: NP-hard (Cooper, 1989)

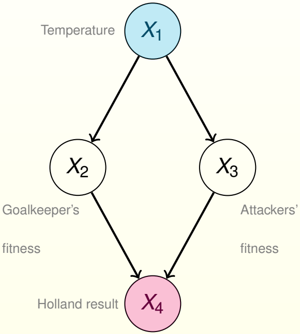
$$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$

with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$

- Fast algorithm for singly-connected BNs
- Credal case:

$$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$

- Fast algorithm for binary singly-connected CNs



$$P(x_4|x_1) = \frac{\sum_{x_2,x_3,x_4} P(\ldots)}{\sum_{x_2,x_3} P(\ldots)}$$

Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case:

NP-hard (Cooper, 1989)

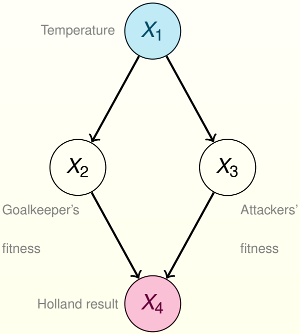
$$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$

with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$

- Fast algorithm for singly-connected BNs
- Credal case:

$$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$

- Fast algorithm for binary singly-connected CNs



$$P(x_4|x_1) = \frac{\sum_{x_2,x_3,x_4} P(\ldots)}{\sum_{x_2,x_3} P(\ldots)}$$

Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case: NP-hard (Cooper, 1989)

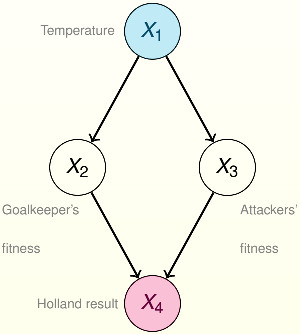
$$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$

with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$

- Fast algorithm for singly-connected BNs
- Credal case:

$$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$

- Fast algorithm for binary singly-connected CNs



$$\underline{P}(x_4|x_1) = \min_j \frac{\sum_{x_2,x_3,x_4} P_j(\ldots)}{\sum_{x_2,x_3} P_j(\ldots)}$$

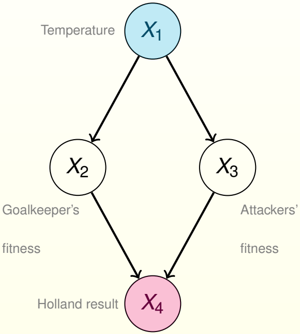
Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case: NP-hard (Cooper, 1989)

  $$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$

  with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$
- Fast algorithm for singly-connected BNs
- Credal case: NP-hard even on singly-connected (de Campos, 2005)

  $$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^{n} P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^{n} P(x_i|\mathrm{pa}(X_i))}$$
- Fast algorithm for binary singly-connected CNs



$$\underline{P}(x_4|x_1) = \min_j \frac{\sum_{x_2,x_3,x_4} P_j(\ldots)}{\sum_{x_2,x_3} P_j(\ldots)}$$

Updating credal nets

- A variable of interest X_q (query)
- Information x_E about the state of some other variables X_E (evidence)
- Updating posterior beliefs about the queried variable given the available evidence
- Bayesian case: NP-hard (Cooper, 1989)

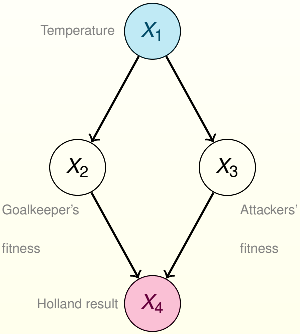
$$P(x_q|x_E) = \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$

with $X_M = (X_1, \ldots, X_n) \setminus (X_q, X_E)$

- Fast algorithm for singly-connected BNs
- Credal case: NP-hard even on singly-connected (de Campos, 2005)

$$\underline{P}(x_q|x_E) = \min_{j=1,\ldots,v} \frac{\sum_{x_M \in \Omega_{X_M}} \prod_{i=1}^n P_j(x_i|\mathrm{pa}(X_i))}{\sum_{x_M \in \Omega_{X_M}, x_q \in \Omega_{X_q}} \prod_{i=1}^n P(x_i|\mathrm{pa}(X_i))}$$

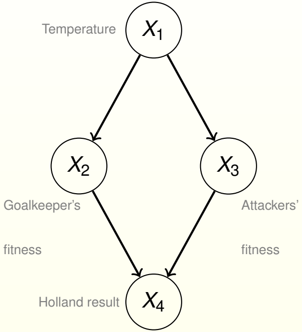
- Fast algorithm for binary singly-connected CNs



$$\underline{P}(x_4|x_1) = \min_j \frac{\sum_{x_2,x_3,x_4} P_j(\ldots)}{\sum_{x_2,x_3} P_j(\ldots)}$$

Updating with incomplete observations

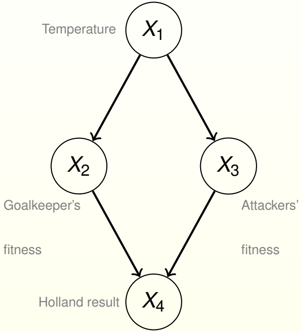
- $\underline{P}(X_q = x_q | X_E = x_E, X_M = *)$
  $= \underline{P}(X_q = x_q | X_E = x_E)$
  right only if *missing at random* assumption holds
- *Conservative inference rule* (CIR)
  $\underline{P}(x_q | x_E, *) = \min_{x_M \in \Omega_{X_M}} \underline{P}(x_q | x_E, x_M)$
  near-ignorance about the process preventing some variable from being observed *(de Cooman & Zaffalon, 2004)*
- CIR on CNs?
  - Add a (dummy) binary child for each missing, with vacuous quantification
  - Use standard updating algorithms



$\underline{P}(x_4 | X_1 = warm, X_2 = *)$

Updating with incomplete observations

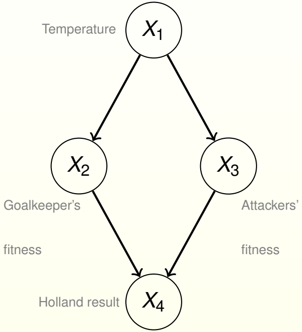
- $\underline{P}(X_q = x_q | X_E = x_E, X_M = *)$ $= \underline{P}(X_q = x_q | X_E = x_E)$ right only if *missing at random* assumption holds
- *Conservative inference rule* (CIR) $\underline{P}(x_q | x_E, *) = \min_{x_M \in \Omega_{X_M}} P(x_q | x_E, x_M)$ near-ignorance about the process preventing some variable from being observed *(de Cooman & Zaffalon, 2004)*
- CIR on CNs?
  - Add a (dummy) binary child for each missing, with vacuous quantification
  - Use standard updating algorithms



$\underline{P}(x_4 | X_1 = warm, X_2 = *)$

Updating with incomplete observations

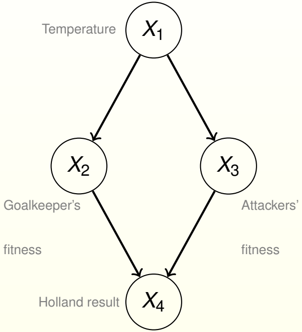
- $\underline{P}(X_q = x_q | X_E = x_E, X_M = *)$
  $= \underline{P}(X_q = x_q | X_E = x_E)$
  right only if *missing at random* assumption holds
- *Conservative inference rule* (CIR)
  $\underline{P}(x_q | x_E, *) = \min_{x_M \in \Omega_{X_M}} P(x_q | x_E, x_M)$
  near-ignorance about the process preventing some variable from being observed *(de Cooman & Zaffalon, 2004)*
- CIR on CNs?
  - Add a (dummy) binary child for each missing, with vacuous quantification
  - Use standard updating algorithms



$$\underline{P}(x_4 | X_1 = warm, X_2 = *)$$
$$= \min\{\underline{P}(x_4 | warm, good), \underline{P}(x_4 | warm, bad)\}$$

Updating with incomplete observations

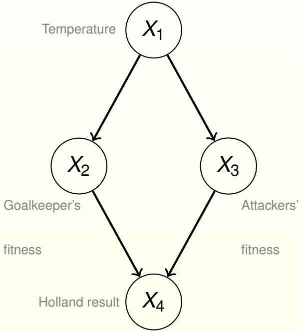
- $\underline{P}(X_q = x_q | X_E = x_E, X_M = *)$
  $= \underline{P}(X_q = x_q | X_E = x_E)$
  right only if *missing at random* assumption holds
- *Conservative inference rule* (CIR)
  $\underline{P}(x_q | x_E, *) = \min_{x_M \in \Omega_{X_M}} P(x_q | x_E, x_M)$
  near-ignorance about the process preventing some variable from being observed *(de Cooman & Zaffalon, 2004)*
- CIR on CNs?
  - Add a (dummy) binary child for each missing, with vacuous quantification
  - Use standard updating algorithms



$$\underline{P}(x_4 | X_1 = warm, X_2 = *)$$
$$= \min\{\underline{P}(x_4 | warm, good), \underline{P}(x_4 | warm, bad)\}$$

Updating with incomplete observations

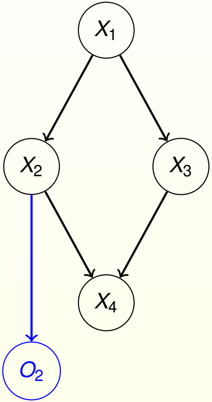
- $\underline{P}(X_q = x_q | X_E = x_E, X_M = *)$
  $= \underline{P}(X_q = x_q | X_E = x_E)$
  right only if *missing at random* assumption holds
- *Conservative inference rule* (CIR)
  $\underline{P}(x_q | x_E, *) = \min_{x_M \in \Omega_{X_M}} P(x_q | x_E, x_M)$
  near-ignorance about the process preventing some variable from being observed *(de Cooman & Zaffalon, 2004)*
- CIR on CNs?
  - Add a (dummy) binary child for each missing, with vacuous quantification
  - Use standard updating algorithms



$\underline{P}(x_4 | X_1 = warm, X_2 = *)$
$= \min\{\underline{P}(x_4 | warm, good), \underline{P}(x_4 | warm, bad)\}$

Updating with incomplete observations

- $\underline{P}(X_q = x_q | X_E = x_E, X_M = *)$
  $= \underline{P}(X_q = x_q | X_E = x_E)$
  right only if *missing at random* assumption holds
- *Conservative inference rule* (CIR)
  $\underline{P}(x_q | x_E, *) = \min_{x_M \in \Omega_{X_M}} P(x_q | x_E, x_M)$
  near-ignorance about the process preventing some variable from being observed (de Cooman & Zaffalon, 2004)
- CIR on CNs?
  - Add a (dummy) binary child for each missing, with vacuous quantification
  - Use standard updating algorithms

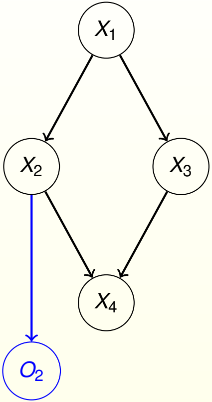


$\underline{K}(O_2|x_2)$ vacuous CS

$\underline{P}(x_4 | X_1 = warm, X_2 = *)$
$= \min\{\underline{P}(x_4 | warm, good), \underline{P}(x_4 | warm, bad)\}$

Updating with incomplete observations

- $\underline{P}(X_q = x_q | X_E = x_E, X_M = *)$
  $= \underline{P}(X_q = x_q | X_E = x_E)$
  right only if *missing at random* assumption holds
- *Conservative inference rule* (CIR)
  $\underline{P}(x_q | x_E, *) = \min_{x_M \in \Omega_{X_M}} P(x_q | x_E, x_M)$
  near-ignorance about the process preventing some variable from being observed *(de Cooman & Zaffalon, 2004)*
- CIR on CNs?
  - Add a (dummy) binary child for each missing, with vacuous quantification
  - Use standard updating algorithms

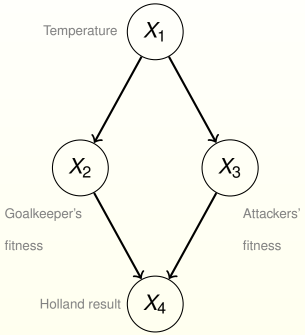


$\underline{K}(O_2 | x_2)$ vacuous CS

$$\underline{P}(x_4 | X_1 = warm, X_2 = *)$$
$$= \min\{\underline{P}(x_4 | warm, good), \underline{P}(x_4 | warm, bad)\}$$
$$= \underline{P}(x_4 | warm, O_2 = 1)$$

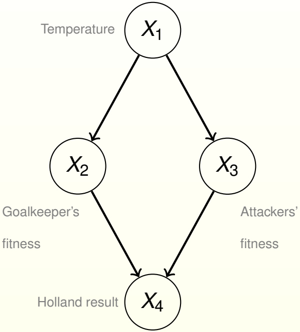
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem $\underline{P}(X_q|O_E = x_E)$



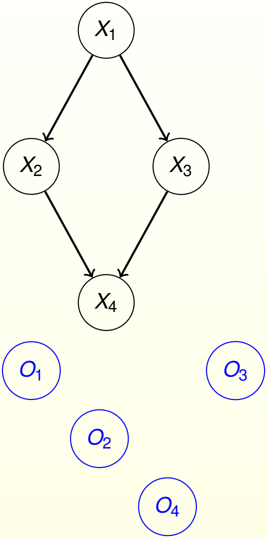
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem $\underline{P}(X_q|O_E = x_E)$



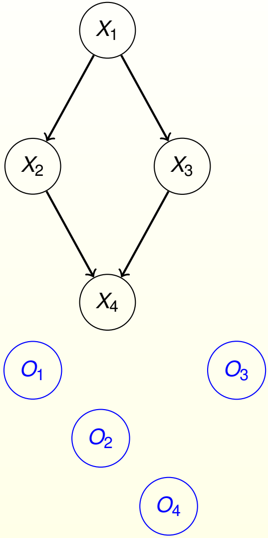
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation
  $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem
  $\underline{P}(X_q|O_E = x_E)$



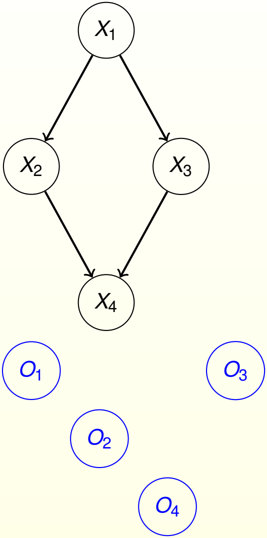
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem $\underline{P}(X_q|O_E = x_E)$



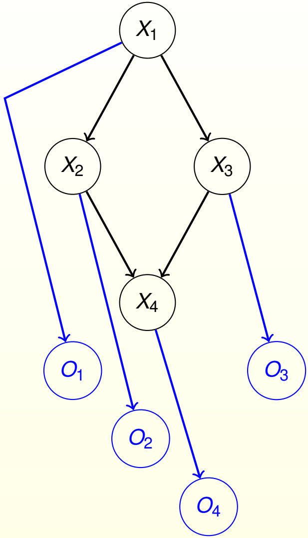
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem $\underline{P}(X_q|O_E = x_E)$



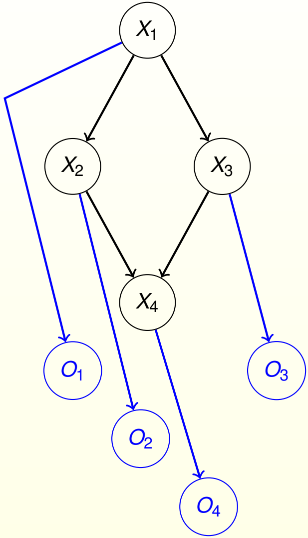
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem $\underline{P}(X_q|O_E = x_E)$



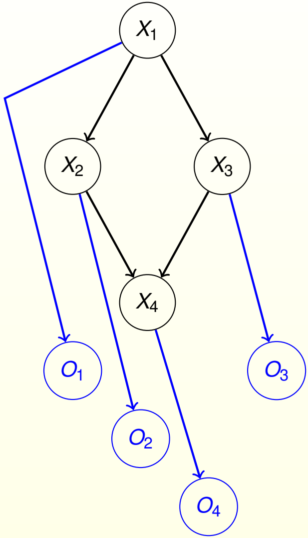
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem $\underline{P}(X_q|O_E = x_E)$



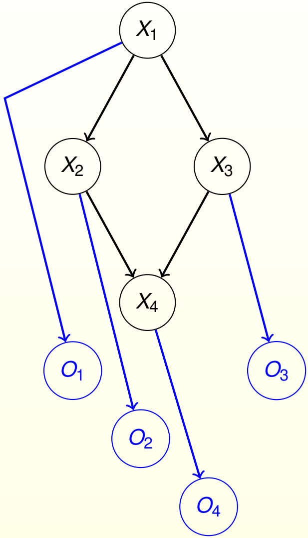
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem $\underline{P}(X_q|O_E = x_E)$



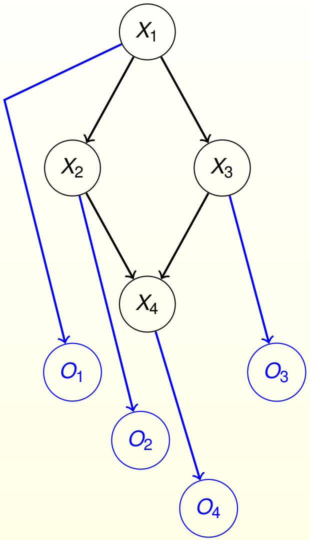
Modelling the observational process

- Each X as a *latent* variable
- For each X a *manifest* variable O_X modelling the observation $\Omega_O = \Omega_X \cup \{*\}$
- Conditional independence, given X between O and the other variables (or weaker conditions)
- Quantifying link between O and X (observational process)
- A CS $K(O|x)$ might a realistic model! (better than $P(O|X)$)
- Standard updating problem $\underline{P}(X_q|O_E = x_E)$



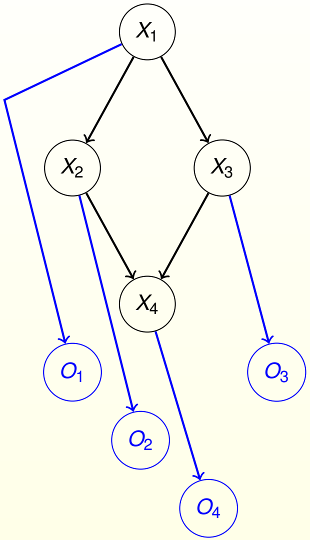
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



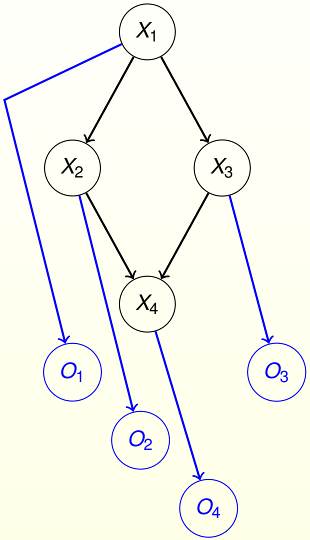
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent
  $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



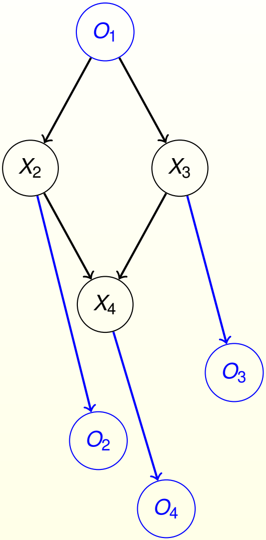
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



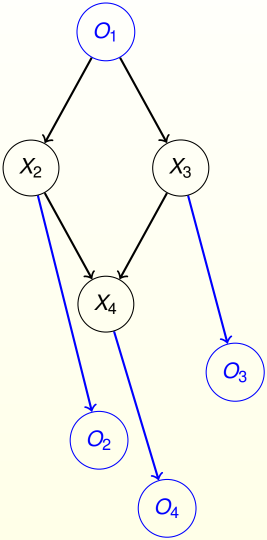
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent
  $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation:
    $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



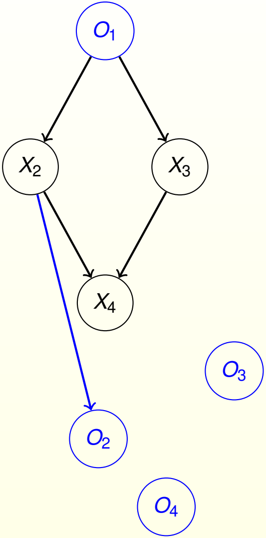
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



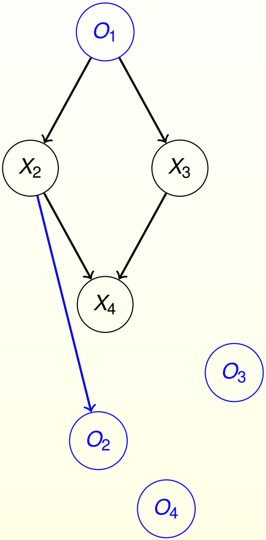
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



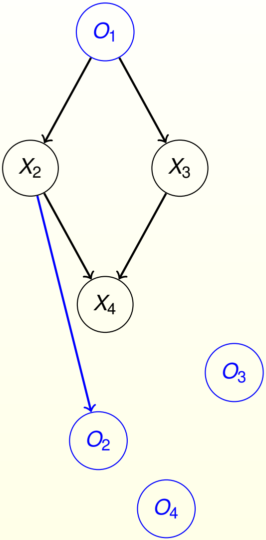
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



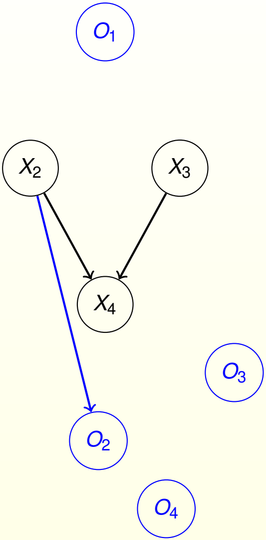
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



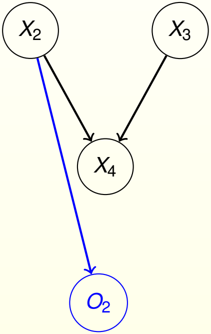
Modelling the observational process (ii)

- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



Modelling the observational process (ii)

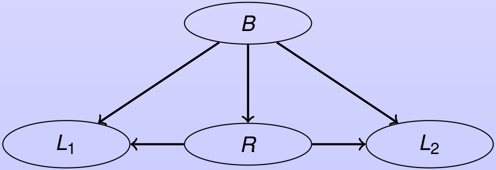
- Manifest variables reduced to binary variables (coarsen to $\{o, \neg o\}$)
- Elicit only lower/upper likelihoods of observation given the latent $\underline{P}(o|x) \leq P(o|x) \leq \overline{P}(o|x)$
  - Perfect observation: $\underline{P}(o|x) = \overline{P}(o|x) = \delta_{o,x}$
  - MAR: $\underline{P}(o|x) = \overline{P}(o|x) = k$
  - CIR: $\underline{P}(o|x) = 0, \overline{P}(o|x) = 1$
  - Imprecise likelihood ratio (and Jeffrey's rule)
- Hard evidence? Drop leaving arcs!
- Only the subnet connected to the query node



[Exe # 3] Is the ball in or out?

- B, with $\mathcal{B} = \{1, 0\}$, means the ball was in
- R, L_1, L_2 are the opinions/observation of the referee/linesmen
- A CN over these variables
- Given B, the three opinions are independent? Not really, the referee has an influence on the linesmen
- Compute bounds of
  $P(B = 1|R = 1, L_1 = 1, L_2 = 1)$
  $\in [.896, .962]$
  $P(B = 1|R = 0, L_1 = 1, L_2 = 1)$
  $P(B = 1|R = 1, L_1 = 0, L_2 = 0)$
  $P(B = 1|R = 0, L_1 = 0, L_2 = 1)$

$P(B = 1) = .50$
$P(R = 1|B = 1) \in [.80, .90]$
$P(R = 1|B = 0) \in [.20, .30]$
$P(L_j = 1|B = 1, R = 1) \in [.90, .95]$
$P(L_j = 1|B = 1, R = 0) \in [.50, .60]$
$P(L_j = 1|B = 0, R = 1) \in [.40, .50]$
$P(L_j = 1|B = 0, R = 0) \in [.10, .20]$



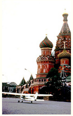
No-fly zones surveyed by the Air Force

- Around important potential targets (eg. WEF, dams, nuke plants)
- Twofold circle wraps the target
  - External no-fly zone (sensors)
  - Internal no-fly zone (anti-air units)
- An aircraft entering the zone (to be called **intruder**)
- Its presence, speed, height, and other features revealed by the sensors
- A team of military experts decides:
  - what the intruder intends to do (external zone / credal level)
  - what to do with the intruder (internal zone / pignistic level)

Identifying intruder's goal

- Four possible (exclusive, exhaustive) options for intruder's goal

renegade



provocateur

damaged

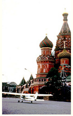
erroneous

- This identification is difficult
  - Sensors reliabilities are affected by geo/meteo conditions
  - **Information fusion** from several sensors

Identifying intruder’s goal

- Four possible (exclusive, exhaustive) options for intruder’s goal


renegade


provocateur


damaged


erroneous

- This identification is difficult
  - Sensors reliabilities are affected by geo/meteo conditions
  - **Information fusion** from several sensors

Why credal networks?

- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

Why credal networks?

- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

Why credal networks?

- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

Why credal networks?

- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

Why credal networks?

- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

Why credal networks?

- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

Why credal networks?

- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

Why credal networks?

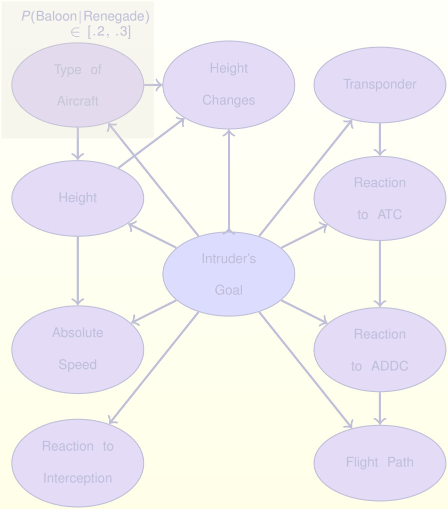
- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

Why credal networks?

- Why a **probabilistic** model?
  - No deterministic relations between the different variables
  - Pervasive uncertainty in the observations
- Why a **graphical** model?
  - Many independence relations among the different variables
- Why an **imprecise** (probabilistic) model?
  - Expert evaluations are mostly based on qualitative judgements
  - The model should be (over)cautious

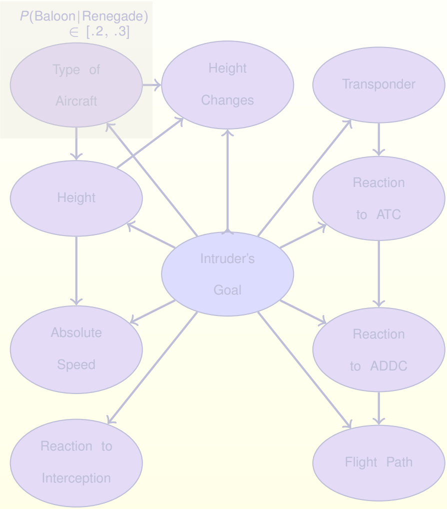
Network core

- Intruder's goal and features as **categorical variables**
- Independencies depicted by a **directed graph** (acyclic)
- Experts provide interval-valued probabilistic assessments, we compute **credal sets**
- A (small) **credal network**
- Complex observation process!



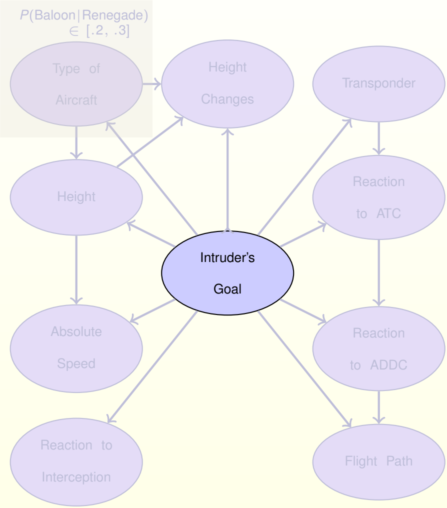
Network core

- Intruder's goal and features as **categorical variables**
- Independencies depicted by a **directed graph** (acyclic)
- Experts provide interval-valued probabilistic assessments, we compute **credal sets**
- A (small) **credal network**
- Complex observation process!



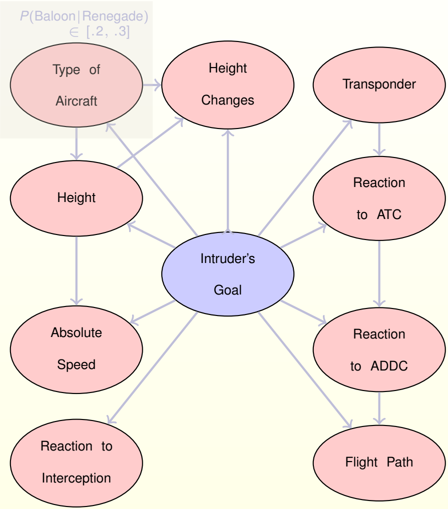
Network core

- Intruder's goal and features as **categorical variables**
- Independencies depicted by a **directed graph** (acyclic)
- Experts provide interval-valued probabilistic assessments, we compute **credal sets**
- A (small) **credal network**
- Complex observation process!



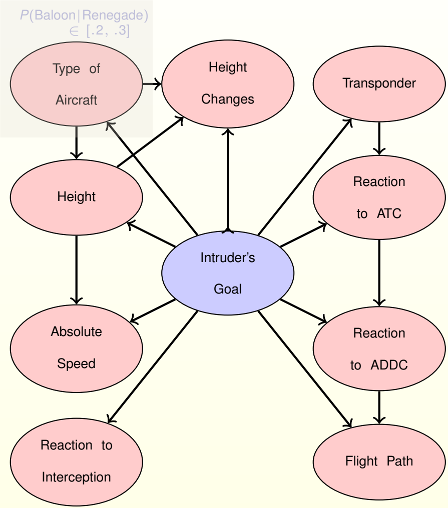
Network core

- Intruder's goal and features as **categorical variables**
- Independencies depicted by a **directed graph** (acyclic)
- Experts provide interval-valued probabilistic assessments, we compute **credal sets**
- A (small) **credal network**
- Complex observation process!



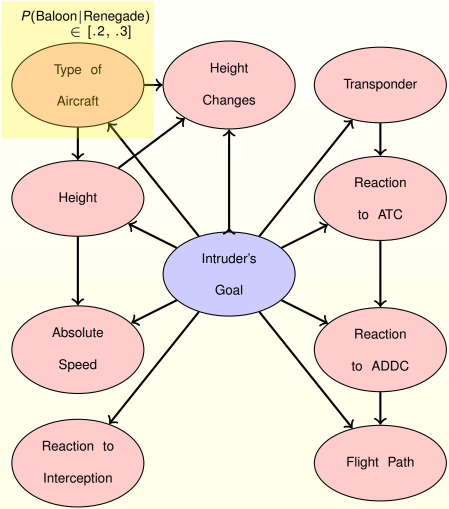
Network core

- Intruder's goal and features as **categorical variables**
- Independencies depicted by a **directed graph** (acyclic)
- Experts provide interval-valued probabilistic assessments, we compute **credal sets**
- A (small) **credal network**
- Complex observation process!



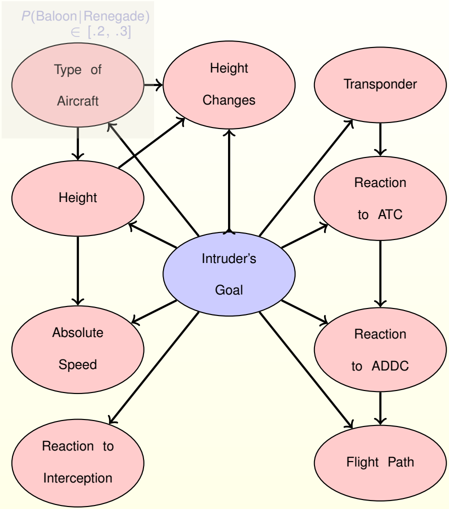
Network core

- Intruder's goal and features as **categorical variables**
- Independencies depicted by a **directed graph** (acyclic)
- Experts provide interval-valued probabilistic assessments, we compute **credal sets**
- A (small) **credal network**
- Complex observation process!



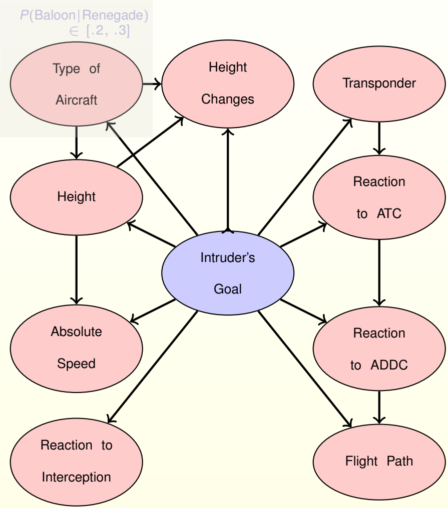
Network core

- Intruder's goal and features as **categorical variables**
- Independencies depicted by a **directed graph** (acyclic)
- Experts provide interval-valued probabilistic assessments, we compute **credal sets**
- A (small) **credal network**
- Complex observation process!



Network core

- Intruder's goal and features as **categorical variables**
- Independencies depicted by a **directed graph** (acyclic)
- Experts provide interval-valued probabilistic assessments, we compute **credal sets**
- A (small) **credal network**
- Complex observation process!



Observations Modeling and Fusion by Credal Nets

- Each sensor modeled by an auxiliary child of the (ideal) variable to be observed
- $P(\text{sensor}|\text{ideal})$ models sensor reliability

  (eg. identity matrix = perfectly reliable sensor)
- Many sensors? Many children!

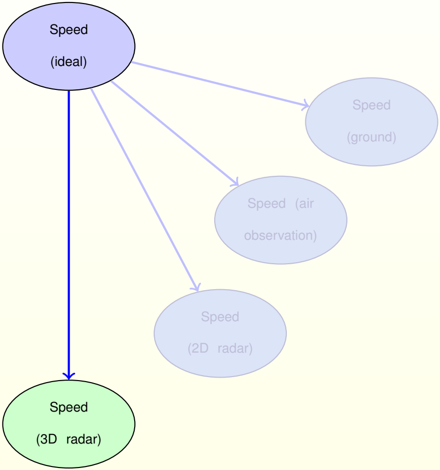
  (conditional independence between sensors given the ideal)

Observations Modeling and Fusion by Credal Nets

- Each sensor modeled by an auxiliary child of the (ideal) variable to be observed
- $P(\text{sensor}|\text{ideal})$ models sensor reliability

  (eg. identity matrix = perfectly reliable sensor)
- Many sensors? Many children!

  (conditional independence between sensors given the ideal)

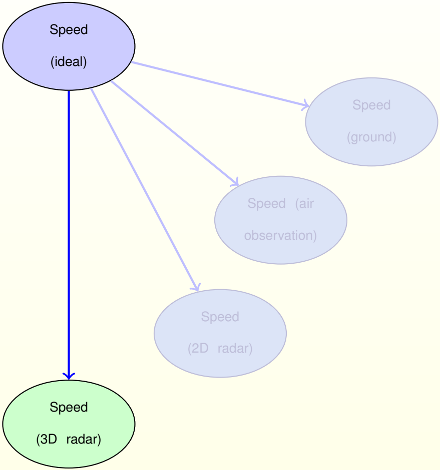


Observations Modeling and Fusion by Credal Nets

- Each sensor modeled by an auxiliary child of the (ideal) variable to be observed
- $P(\text{sensor}|\text{ideal})$ models sensor reliability

  (eg. identity matrix = perfectly reliable sensor)
- Many sensors? Many children!

  (conditional independence between sensors given the ideal)

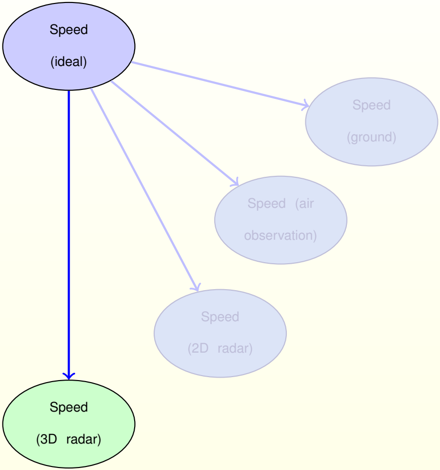


Observations Modeling and Fusion by Credal Nets

- Each sensor modeled by an auxiliary child of the (ideal) variable to be observed
- $P(\text{sensor}|\text{ideal})$ models sensor reliability

  (eg. identity matrix = perfectly reliable sensor)
- Many sensors? Many children!

  (conditional independence between sensors given the ideal)

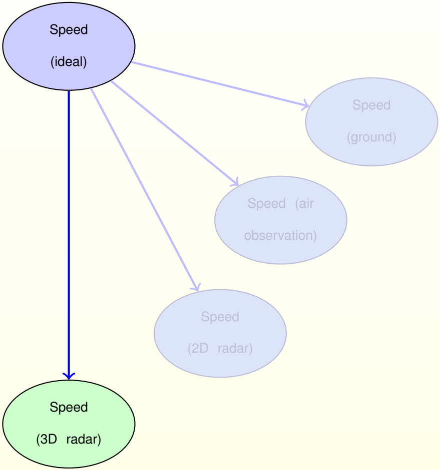


Observations Modeling and Fusion by Credal Nets

- Each sensor modeled by an auxiliary child of the (ideal) variable to be observed
- $P(\text{sensor}|\text{ideal})$ models sensor reliability

  (eg. identity matrix = perfectly reliable sensor)
- Many sensors? Many children!

  (conditional independence between sensors given the ideal)

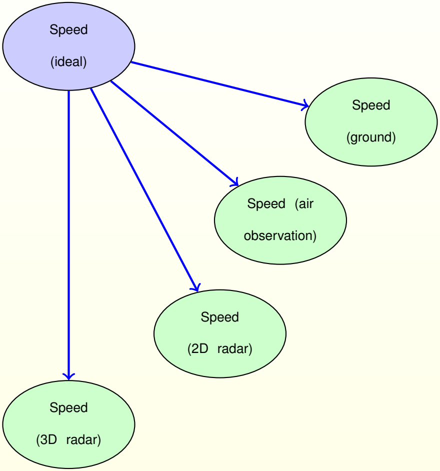


Observations Modeling and Fusion by Credal Nets

- Each sensor modeled by an auxiliary child of the (ideal) variable to be observed
- $P(\text{sensor}|\text{ideal})$ models sensor reliability

  (eg. identity matrix = perfectly reliable sensor)
- Many sensors? Many children!

  (conditional independence between sensors given the ideal)

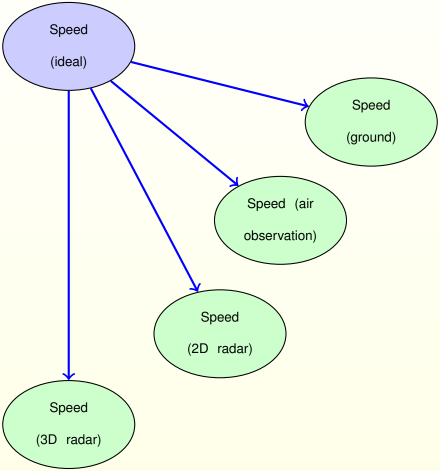


Observations Modeling and Fusion by Credal Nets

- Each sensor modeled by an auxiliary child of the (ideal) variable to be observed
- $P(\text{sensor}|\text{ideal})$ models sensor reliability

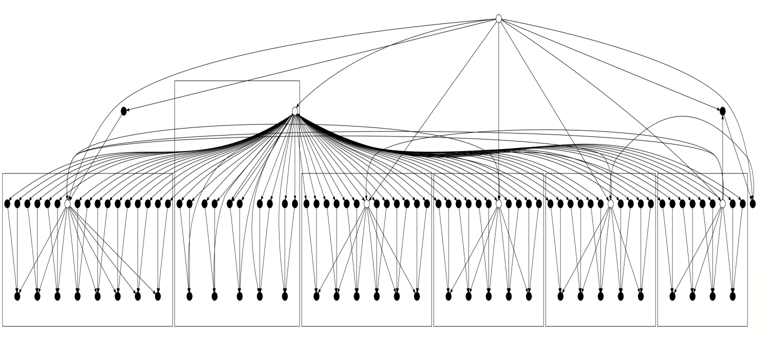
  (eg. identity matrix = perfectly reliable sensor)
- Many sensors? Many children!

  (conditional independence between sensors given the ideal)



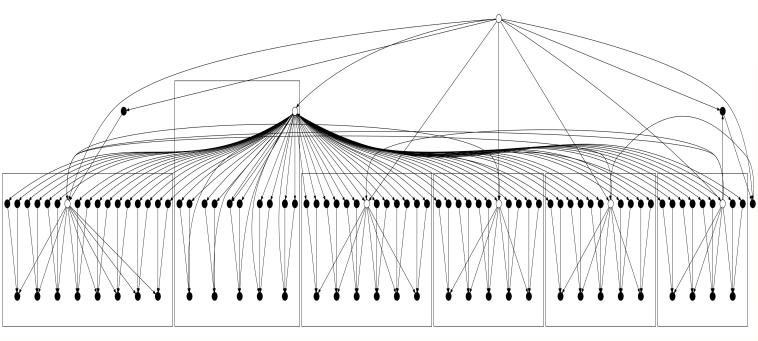
The whole network

- We conclude a huge multiply-connected credal network
- Approximate algorithm:
  1. Local specification [Antonucci and Zaffalon, PGM'06]
  2. Binarization [Antonucci, Zaffalon, Ide and Cozman, STAIRS 2006]
  3. L2U [Ide and Cozman, STAIRS 2004]



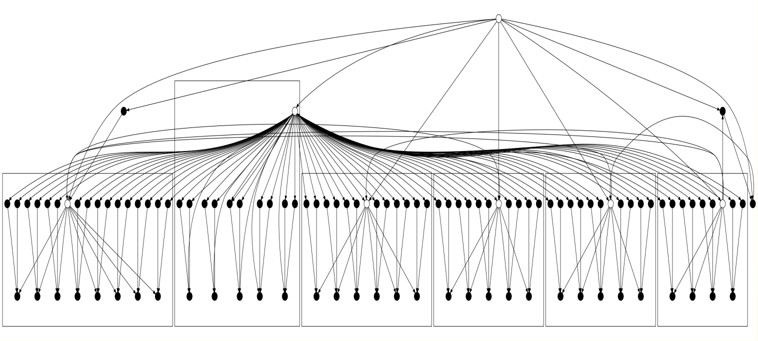
The whole network

- We conclude a huge multiply-connected credal network
- Approximate algorithm:
  1. Local specification [Antonucci and Zaffalon, PGM'06]
  2. Binarization [Antonucci, Zaffalon, Ide and Cozman, STAIRS 2006]
  3. L2U [Ide and Cozman, STAIRS 2004]



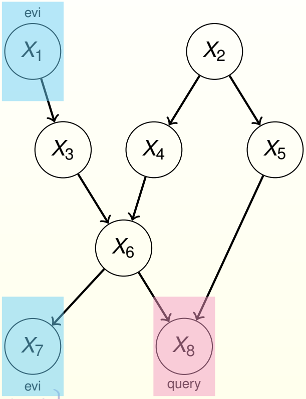
The whole network

- We conclude a huge multiply-connected credal network
- Approximate algorithm:
  1. Local specification [Antonucci and Zaffalon, PGM'06]
  2. Binarization [Antonucci, Zaffalon, Ide and Cozman, STAIRS 2006]
  3. L2U [Ide and Cozman, STAIRS 2004]



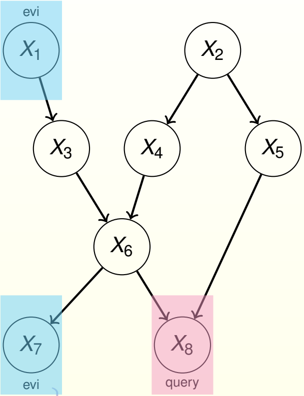
Decision Making with CNs

- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q = ?$
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute
  $\underline{P}(X_q|x_E)$
- What about the state of $X_q = ?$
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$



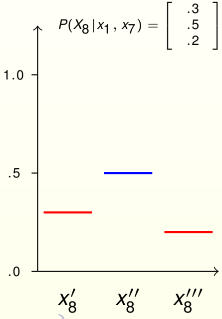
Decision Making with CNs

- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q =$?
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute
  $\underline{P}(X_q|x_E)$
- What about the state of $X_q =$?
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$



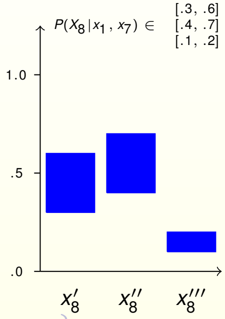
Decision Making with CNs

- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q = ?$
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute
  $\underline{P}(X_q|x_E)$
- What about the state of $X_q = ?$
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$



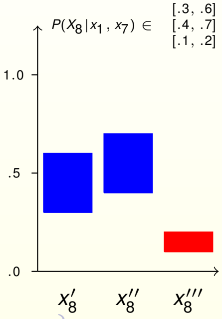
Decision Making with CNs

- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q =?$
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute $\underline{P}(X_q|x_E)$
- What about the state of $X_q =?$
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$
  - Maximality in CNs by auxiliary Boolean child
    $P_{x'_q,x''_q}(Y|x_q) = \frac{\delta_{x_q,x'_q} - \delta_{x_q,x''_q} + 1}{2}$
    $\underline{P}_{x'_q,x''_q}(Y=1|x_E) > \frac{1}{2} \Leftrightarrow P(x'_q|x_E) > P(x''_q|x_E) \; \forall P(X_q|x_E) \in K(X_q|x_E)$



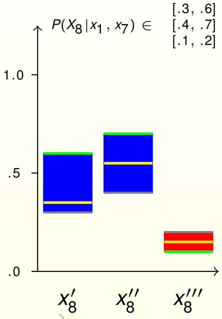
Decision Making with CNs

- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q =?$
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute $\underline{P}(X_q|x_E)$
- What about the state of $X_q =?$
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$
  - Maximality in CNs by auxiliary Boolean child
    $P_{x'_q,x''_q}(Y|x_q) = \frac{\delta_{x_q,x'_q} - \delta_{x_q,x''_q} + 1}{2}$
    $\underline{P}_{x'_q,x''_q}(Y=1|x_E) > \frac{1}{2} \Leftrightarrow P(x'_q|x_E) > P(x''_q|x_E) \; \forall P(X_q|x_E) \in K(X_q|x_E)$



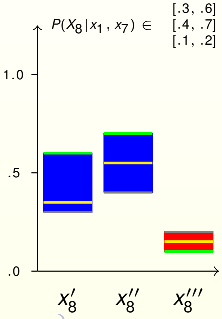
Decision Making with CNs

- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q = ?$
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute
  $\underline{P}(X_q|x_E)$
- What about the state of $X_q = ?$
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$
  - Maximality in CNs by auxiliary Boolean child
    $P_{x'_q,x''_q}(Y|x_q) = \frac{\delta_{x_q,x'_q} - \delta_{x_q,x''_q} + 1}{2}$
    $\underline{P}_{x'_q,x''_q}(Y=1|x_E) > \frac{1}{2} \Leftrightarrow P(x'_q|x_E) > P(x''_q|x_E)\ \forall P(X_q|x_E) \in K(X_q|x_E)$



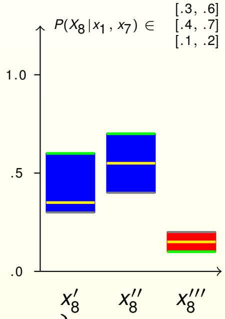
Decision Making with CNs

- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q = ?$
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute $\underline{P}(X_q|x_E)$
- What about the state of $X_q = ?$
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$
  - Maximality in CNs by auxiliary Boolean child
    $P_{x'_q,x''_q}(Y|x_q) = \frac{\delta_{x_q,x'_q} - \delta_{x_q,x''_q} + 1}{2}$
    $\underline{P}_{x'_q,x''_q}(Y=1|x_E) > \frac{1}{2} \Leftrightarrow P(x'_q|x_E) > P(x''_q|x_E) \; \forall P(X_q|x_E) \in K(X_q|x_E)$



Decision Making with CNs

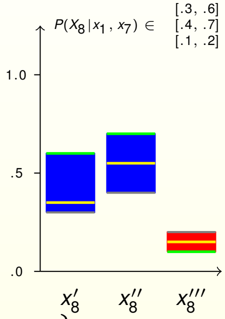
- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q = ?$
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute $\underline{P}(X_q|x_E)$
- What about the state of $X_q = ?$
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$



- Maximality in CNs by auxiliary Boolean child
  $P_{x'_q,x''_q}(Y|x_q) = \frac{\delta_{x_q,x'_q} - \delta_{x_q,x''_q} + 1}{2}$
  $\underline{P}_{x'_q,x''_q}(Y = 1|x_E) > \frac{1}{2} \Leftrightarrow P(x'_q|x_E) > P(x''_q|x_E) \; \forall P(X_q|x_E) \in K(X_q|x_E)$

Decision Making with CNs

- BN updating compute $P(X_q|x_E)$
- What about the state of $X_q = ?$
  $x_q^* = \arg\max_{x_q \in \Omega_{X_q}} P(x_q|x_E)$
- CN updating should compute $K(X_q|x_E)$
  In practice algorithms only compute $\underline{P}(X_q|x_E)$
- What about the state of $X_q = ?$
- State(s) of X_q by interval dominance
  $\Omega^*_{X_q} = \left\{ x_q \middle| \nexists x'_q \text{ s.t. } \underline{P}(x'_q|x_E) > \overline{P}(x_q|x_E) \right\}$
- More informative criterion: maximality
  $\left\{ x_q \middle| \nexists x'_q \text{ s.t. } P(x'_q|x_E) > P(x_q|x_E) \forall P(X_q|x_E) \in K(X_q|x_E) \right\}$
  - Maximality in CNs by auxiliary Boolean child
    $P_{x'_q, x''_q}(Y|x_q) = \frac{\delta_{x_q, x'_q} - \delta_{x_q, x''_q} + 1}{2}$
    $\underline{P}_{x'_q, x''_q}(Y = 1|x_E) > \frac{1}{2} \Leftrightarrow P(x'_q|x_E) > P(x''_q|x_E) \; \forall P(X_q|x_E) \in K(X_q|x_E)$



Simulations (military application)

- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)

Simulations (military application)

- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)

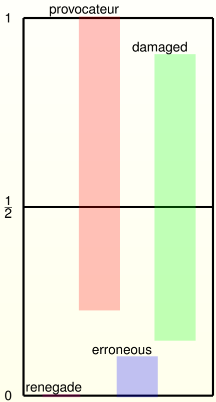
Simulations (military application)

- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)

Simulations (military application)

- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)

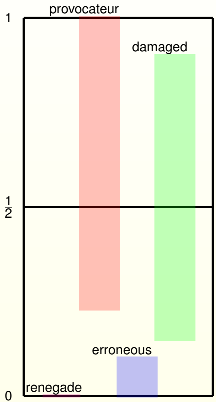
SIMULATION #1



Simulations (military application)

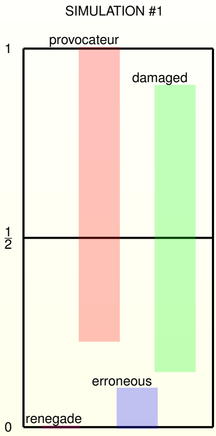
- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)

SIMULATION #1



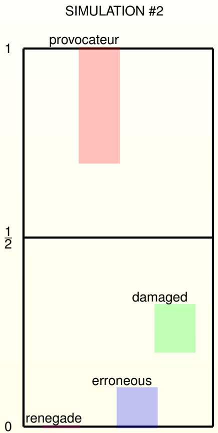
Simulations (military application)

- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)



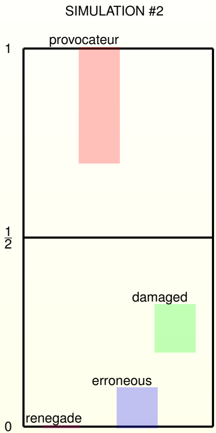
Simulations (military application)

- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)



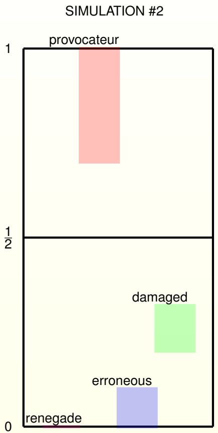
Simulations (military application)

- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)



Simulations (military application)

- Simulating a dam in the Swiss Alps, with no interceptors, relatively good coverage for other sensors, discontinuous low clouds and daylight
- Sensors return:
  - Height = very low / very low / very low / low
  - Type = helicopter / helicopter
  - Flight Path = U-path / U-path / U-path / U-path / U-path / missing
  - Height Changes = descent / descent / descent / descent / missing
  - Speed = slow / slow / slow / slow / slow
  - ADDC reaction = positive / positive / positive / positive / positive / positive
- We reject renegade and damaged, but indecision between provocateur and erroneous
- Assuming higher levels of reliability
- We conclude the aircraft is a provocateur!
- Same conclusion with maximality (no need of higher reliability)



An application: debris flows risk assessment

- Debris flows are very destructive natural hazards
- Still partially understood
- Human expertize is still fundamental!
- An artificial expert system supporting human experts?

An application: debris flows risk assessment

- Debris flows are very destructive natural hazards
- Still partially understood
- Human expertize is still fundamental!
- An artificial expert system supporting human experts?

An application: debris flows risk assessment

- Debris flows are very destructive natural hazards
- Still partially understood
- Human expertize is still fundamental!
- An artificial expert system supporting human experts?

An application: debris flows risk assessment

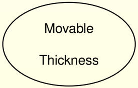
- Debris flows are very destructive natural hazards
- Still partially understood
- Human expertize is still fundamental!
- An artificial expert system supporting human experts?

An application: debris flows risk assessment

- Debris flows are very destructive natural hazards
- Still partially understood
- Human expertize is still fundamental!
- An artificial expert system supporting human experts?

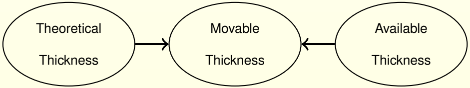
Building the causal network

Proxy indicator of the level of risk

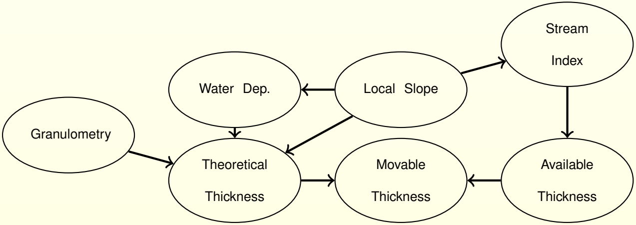


Building the causal network

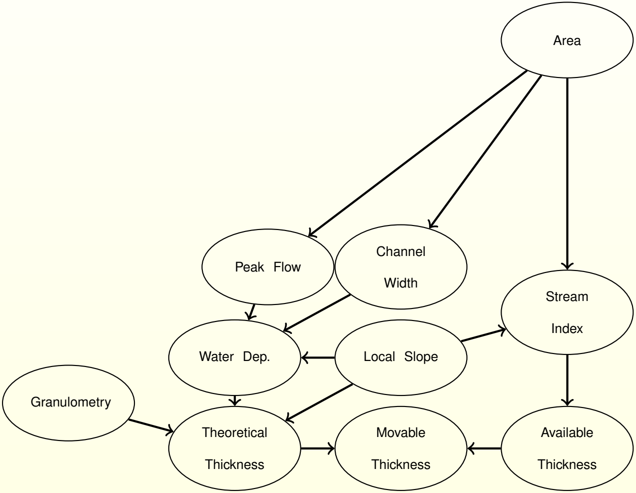
Triggering Factors



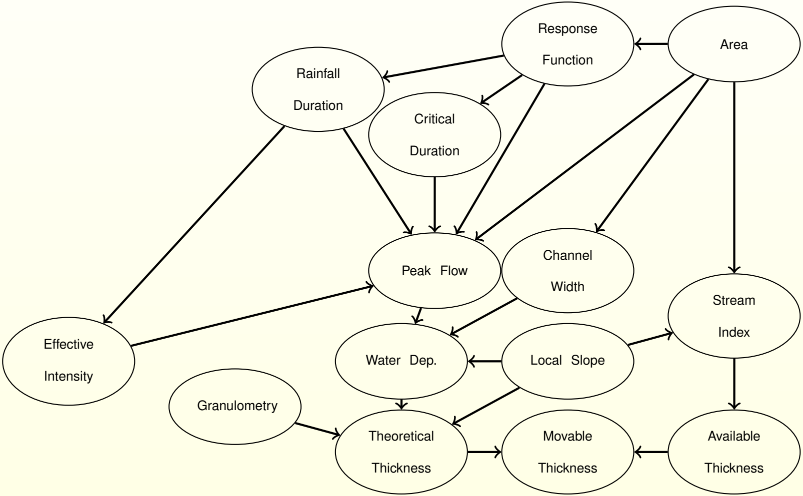
Building the causal network



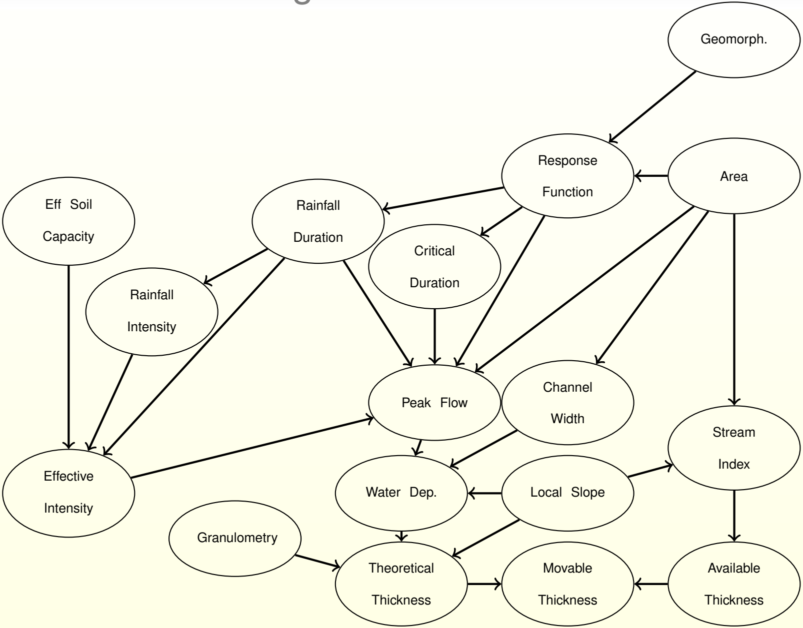
Building the causal network



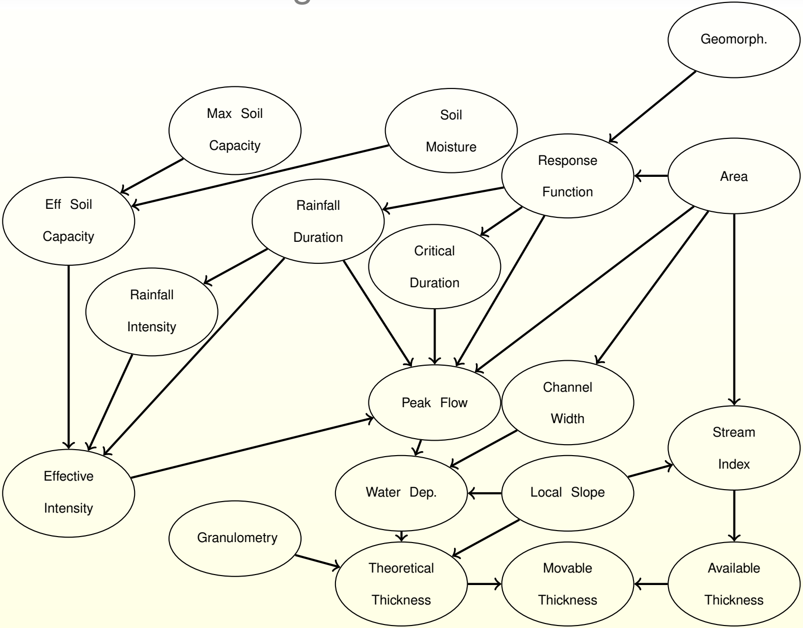
Building the causal network



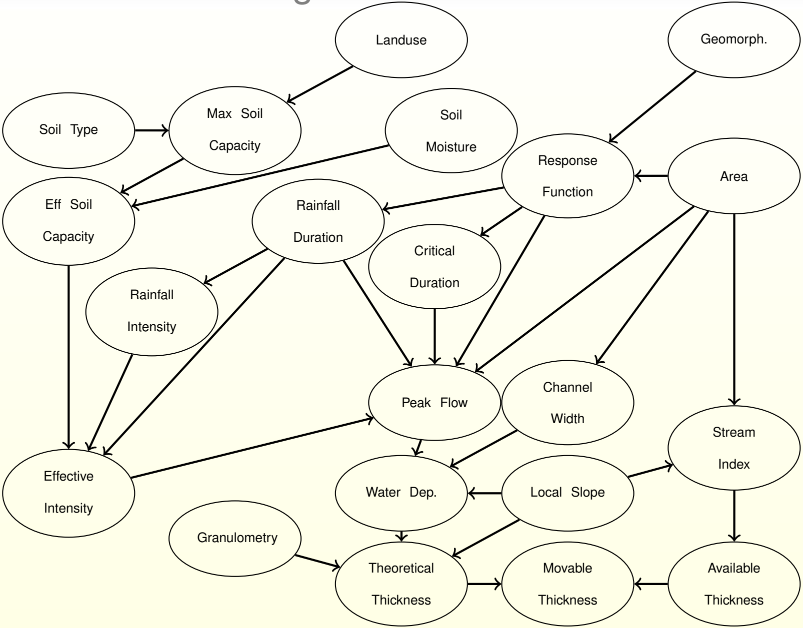
Building the causal network



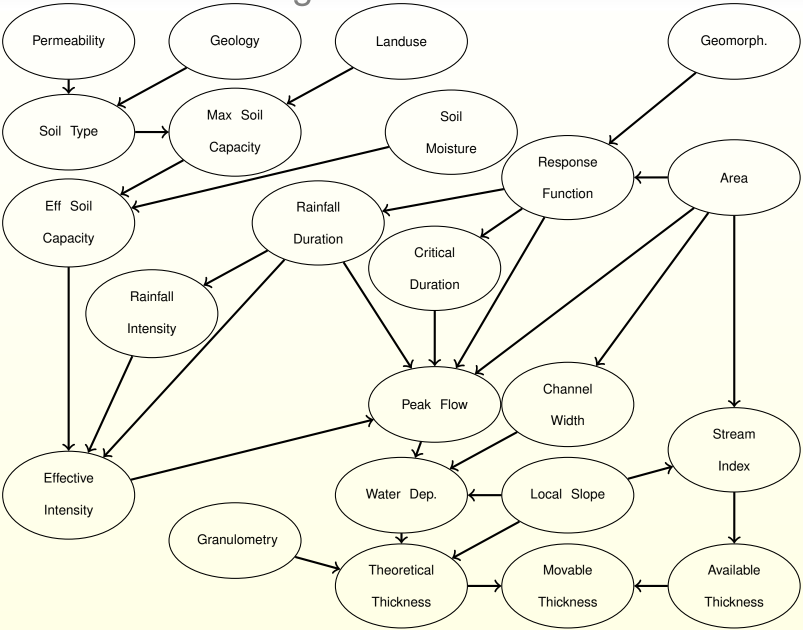
Building the causal network



Building the causal network

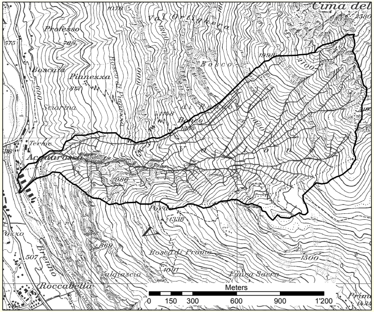


Building the causal network



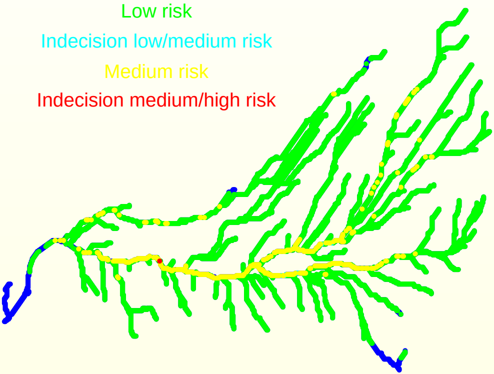
Debris flow hazard assessment by CNs

- Extensive simulations in a debris flow prone watershed Acquarossa Creek Basin (area 1.6 Km^2, length 3.1 Km)



Debris flow hazard assessment by CNs

- Extensive simulations in a debris flow prone watershed Acquarossa Creek Basin (area 1.6 Km^2, length 3.1 Km)



Inference based on message propagation

Bayesian nets

- Pearl's message propagation
  Efficient for polytrees
- Multiply connected BNs?
  Loopy belief propagation

Credal nets

- Only outer approximation for general polytrees
  (Tessem, 1992)
  (da Rocha & Cozman, A/R+, 2005)
- Exact for binary polytrees
  (2U, Zaffalon, 1998)
- Loopy version of 2U for binary multiply connected
  (Ide & Cozman, 2004)

$$
\begin{aligned}
&p(x|e) = \alpha \Lambda(x)\pi(x),\\
&\Lambda(x) = \Lambda_X(x) \textstyle\prod_j \Lambda_{Y_j}(x),\\
&\pi(x) = \textstyle\sum_u p(x|u) \prod_k \pi_X(u_k),\\
&\Lambda_X(u_i) =\\
&\alpha \textstyle\sum_x \Lambda(x) \sum_{u_k:k\neq i} p(x|u) \prod_{k\neq i} \pi_X(u_k),\\
&\pi_{Y_j}(x) = \alpha \pi(x)\Lambda_X(x) \textstyle\prod_{k\neq j} \Lambda_{Y_k}(x)
\end{aligned}
$$

Updating non-binary CNs?

Inference based on message propagation

BAYESIAN NETS

- Pearl's message propagation
  Efficient for polytrees
- Multiply connected BNs?
  Loopy belief propagation

CREDAL NETS

- Only outer approximation for general polytrees
  (Tessem, 1992)
  (da Rocha & Cozman, A/R+, 2005)
- Exact for binary polytrees
  (2U, Zaffalon, 1998)
- Loopy version of 2U for binary multiply connected
  (Ide & Cozman, 2004)

$$
\begin{aligned}
&p(x|e) = \alpha \Lambda(x)\pi(x),\\
&\Lambda(x) = \Lambda_X(x)\textstyle\prod_j \Lambda_{Y_j}(x),\\
&\pi(x) = \textstyle\sum_u p(x|u)\prod_k \pi_X(u_k),\\
&\Lambda_X(u_i) =\\
&\alpha \textstyle\sum_x \Lambda(x)\sum_{u_k:k\neq i} p(x|u)\prod_{k\neq i}\pi_X(u_k),\\
&\pi_{Y_j}(x) = \alpha\pi(x)\Lambda_X(x)\textstyle\prod_{k\neq j}\Lambda_{Y_k}(x)
\end{aligned}
$$

Updating non-binary CNs?

Inference based on message propagation

Bayesian nets

- Pearl's message propagation
  Efficient for polytrees
- Multiply connected BNs?
  Loopy belief propagation

Credal nets

- Only outer approximation for general polytrees
  (Tessem, 1992)
  (da Rocha & Cozman, A/R+, 2005)
- Exact for binary polytrees
  (2U, Zaffalon, 1998)
- Loopy version of 2U for binary multiply connected
  (Ide & Cozman, 2004)

$$
\begin{aligned}
&p(x|e) = \alpha \Lambda(x)\pi(x),\\
&\Lambda(x) = \Lambda_X(x) \textstyle\prod_j \Lambda_{Y_j}(x),\\
&\pi(x) = \textstyle\sum_u p(x|u) \prod_k \pi_X(u_k),\\
&\Lambda_X(u_i) =\\
&\alpha \textstyle\sum_x \Lambda(x) \sum_{u_k:k\neq i} p(x|u) \prod_{k\neq i} \pi_X(u_k),\\
&\pi_{Y_j}(x) = \alpha\pi(x)\Lambda_X(x) \textstyle\prod_{k\neq j} \Lambda_{Y_k}(x)
\end{aligned}
$$

Updating non-binary CNs?

Inference based on message propagation

BAYESIAN NETS

- Pearl's message propagation
  Efficient for polytrees
- Multiply connected BNs?
  Loopy belief propagation

CREDAL NETS

- Only outer approximation for general polytrees
  (Tessem, 1992)
  (da Rocha & Cozman, A/R+, 2005)
- Exact for binary polytrees
  (2U, Zaffalon, 1998)
- Loopy version of 2U for binary multiply connected
  (Ide & Cozman, 2004)

$$p(x|e) = \alpha \Lambda(x) \pi(x),$$
$$\Lambda(x) = \Lambda_X(x) \prod_j \Lambda_{Y_j}(x),$$
$$\pi(x) = \sum_u p(x|u) \prod_k \pi_X(u_k),$$
$$\Lambda_X(u_i) =$$
$$\alpha \sum_x \Lambda(x) \sum_{u_k : k \neq i} p(x|u) \prod_{k \neq i} \pi_X(u_k),$$
$$\pi_{Y_j}(x) = \alpha \pi(x) \Lambda_X(x) \prod_{k \neq j} \Lambda_{Y_k}(x)$$

Updating non-binary CNs?

Inference based on message propagation

Bayesian nets

- Pearl's message propagation
  Efficient for polytrees
- Multiply connected BNs?
  Loopy belief propagation

Credal nets

- Only outer approximation for general polytrees
  (Tessem, 1992)
  (da Rocha & Cozman, A/R+, 2005)
- Exact for binary polytrees
  (2U, Zaffalon, 1998)
- Loopy version of 2U for binary multiply connected
  (Ide & Cozman, 2004)

$$p(x|e) = \alpha \Lambda(x)\pi(x),$$
$$\Lambda(x) = \Lambda_X(x) \prod_j \Lambda_{Y_j}(x),$$
$$\pi(x) = \sum_u p(x|u) \prod_k \pi_X(u_k),$$
$$\Lambda_X(u_i) =$$
$$\alpha \sum_x \Lambda(x) \sum_{u_k : k \neq i} p(x|u) \prod_{k \neq i} \pi_X(u_k),$$
$$\pi_{Y_j}(x) = \alpha \pi(x) \Lambda_X(x) \prod_{k \neq j} \Lambda_{Y_k}(x)$$

Updating non-binary CNs?

Inference based on message propagation

BAYESIAN NETS

- Pearl's message propagation
  Efficient for polytrees
- Multiply connected BNs?
  Loopy belief propagation

CREDAL NETS

- Only outer approximation for general polytrees
  (Tessem, 1992)
  (da Rocha & Cozman, A/R+, 2005)
- Exact for binary polytrees
  (2U, Zaffalon, 1998)
- Loopy version of 2U for binary multiply connected
  (Ide & Cozman, 2004)

$$p(x|e) = \alpha\Lambda(x)\pi(x),$$
$$\Lambda(x) = \Lambda_X(x)\prod_j \Lambda_{Y_j}(x),$$
$$\pi(x) = \sum_u p(x|u)\prod_k \pi_X(u_k),$$
$$\Lambda_X(u_i) =$$
$$\alpha\sum_x \Lambda(x)\sum_{u_k:k\neq i} p(x|u)\prod_{k\neq i}\pi_X(u_k),$$
$$\pi_{Y_j}(x) = \alpha\pi(x)\Lambda_X(x)\prod_{k\neq j}\Lambda_{Y_k}(x)$$

Updating non-binary CNs?

Inference based on message propagation

Bayesian nets

- Pearl's message propagation
  Efficient for polytrees
- Multiply connected BNs?
  Loopy belief propagation

Credal nets

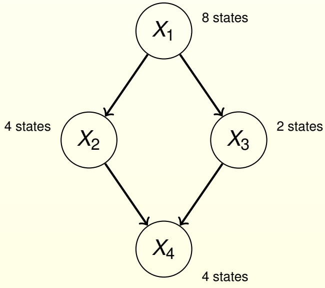
- Only outer approximation for general polytrees
  (Tessem, 1992)
  (da Rocha & Cozman, A/R+, 2005)
- Exact for binary polytrees
  (2U, Zaffalon, 1998)
- Loopy version of 2U for binary multiply connected
  (Ide & Cozman, 2004)

$p(x|e) = \alpha \Lambda(x)\pi(x)$,
$\Lambda(x) = \Lambda_X(x) \prod_j \Lambda_{Y_j}(x)$,
$\pi(x) = \sum_u p(x|u) \prod_k \pi_X(u_k)$,
$\Lambda_X(u_i) =$
$\alpha \sum_x \Lambda(x) \sum_{u_k : k \neq i} p(x|u) \prod_{k \neq i} \pi_X(u_k)$,
$\pi_{Y_j}(x) = \alpha \pi(x) \Lambda_X(x) \prod_{k \neq j} \Lambda_{Y_k}(x)$

Updating non-binary CNs?

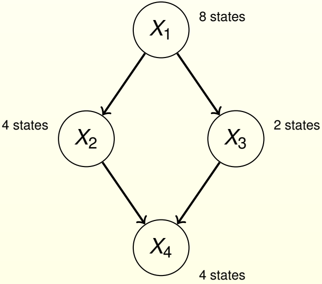
Binarizing non-binary credal nets

- State of a variable as a joint state of a number of "bits"
  $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$
- For each arc between two variables, all the relative bits are linked, bits of the same variable are completely connected
- Local computations for the probabilities
- A "binarized" equivalent CN is obtained
- L2U can update it *(GL2U, Antonucci et al. 2010)*



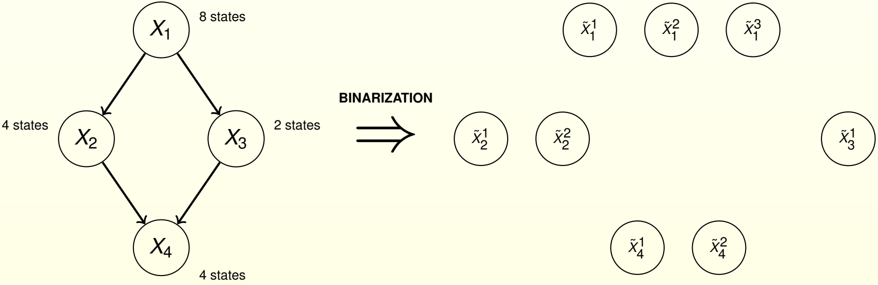
Binarizing non-binary credal nets

- State of a variable as a joint state of a number of “bits”
  $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$
- For each arc between two variables, all the relative bits are linked, bits of the same variable are completely connected
- Local computations for the probabilities
- A “binarized” equivalent CN is obtained
- L2U can update it *(GL2U, Antonucci et al. 2010)*



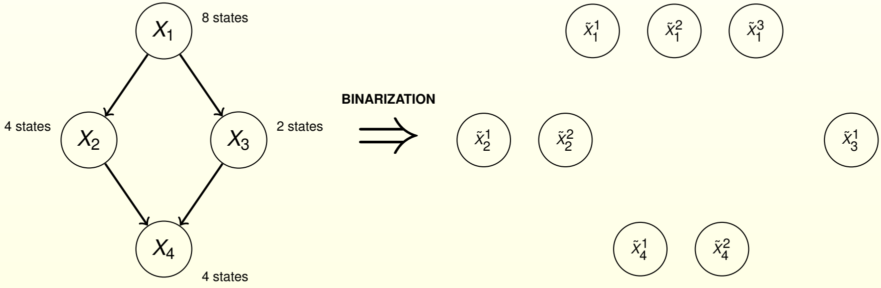
Binarizing non-binary credal nets

- State of a variable as a joint state of a number of “bits”
  $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$
- For each arc between two variables, all the relative bits are linked, bits of the same variable are completely connected
- Local computations for the probabilities
- A “binarized” equivalent CN is obtained
- L2U can update it *(GL2U, Antonucci et al. 2010)*



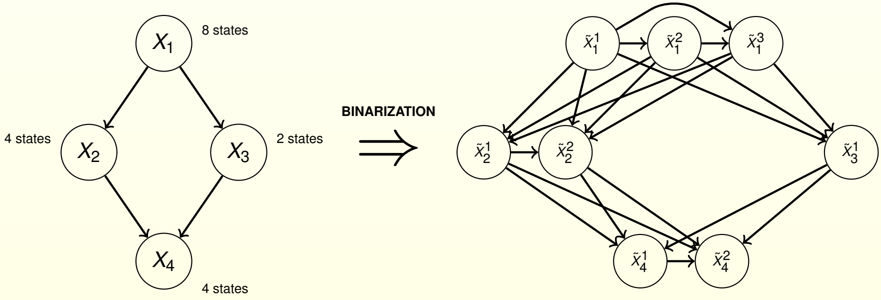
Binarizing non-binary credal nets

- State of a variable as a joint state of a number of "bits"
  $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \dots$
- For each arc between two variables, all the relative bits are linked, bits of the same variable are completely connected
- Local computations for the probabilities
- A "binarized" equivalent CN is obtained
- L2U can update it *(GL2U, Antonucci et al. 2010)*



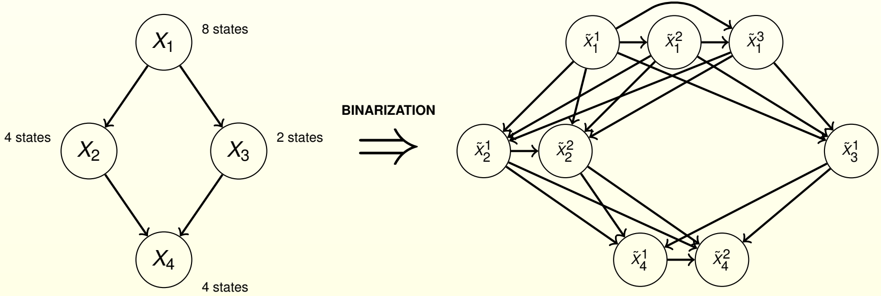
Binarizing non-binary credal nets

- State of a variable as a joint state of a number of "bits"
  $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$
- For each arc between two variables, all the relative bits are linked, bits of the same variable are completely connected
- Local computations for the probabilities
- A "binarized" equivalent CN is obtained
- L2U can update it *(GL2U, Antonucci et al. 2010)*



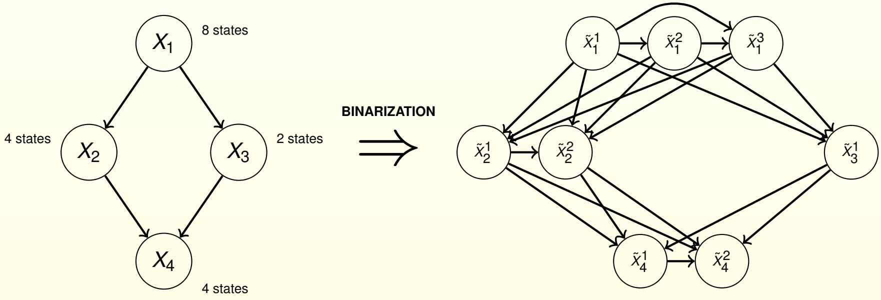
Binarizing non-binary credal nets

- State of a variable as a joint state of a number of "bits"
  $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$
- For each arc between two variables, all the relative bits are linked, bits of the same variable are completely connected
- Local computations for the probabilities
- A "binarized" equivalent CN is obtained
- L2U can update it *(GL2U, Antonucci et al. 2010)*



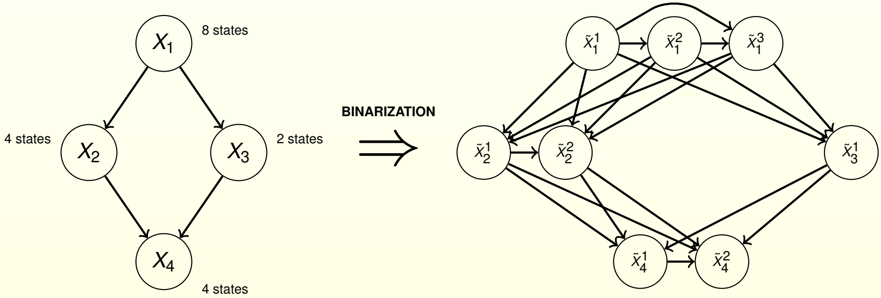
Binarizing non-binary credal nets

- State of a variable as a joint state of a number of "bits" $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$
- For each arc between two variables, all the relative bits are linked, bits of the same variable are completely connected
- Local computations for the probabilities
- A "binarized" equivalent CN is obtained
- L2U can update it *(GL2U, Antonucci et al. 2010)*



Binarizing non-binary credal nets

- State of a variable as a joint state of a number of "bits"
  $X = x \iff (\tilde{X}^1 = \tilde{x}^1) \wedge (\tilde{X}^2 = \tilde{x}^2) \wedge \ldots$
- For each arc between two variables, all the relative bits are linked, bits of the same variable are completely connected
- Local computations for the probabilities
- A "binarized" equivalent CN is obtained
- L2U can update it *(GL2U, Antonucci et al. 2010)*



Exact inference: Variable elimination

BAYESIAN NETS

- Choose an ordering of the variables (query last)
- Create a *pool* of functions with all local distributions
- For each X:

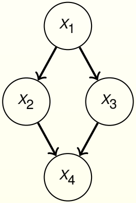
  Insert all functions that contain X in a structure called *bucket* of X and remove them from the pool

  Multiply these functions and marginalize out X

  Insert the results in the pool

bucket elimination (Dechter, 1996)
fusion algorithm for valuation algebras (Shenoy & Kohlas, 1994)

Compute $P(X_4)$ with ordering X_1, X_2, X_3, X_4
Pool $\equiv \{P(X_1), p(X_2|X_1), p(X_3|X_1), p(X_4|X_2, X_3)\}$
Bucket(X_1): $\sum_{X_1} P(X_2|X_1)p(X_3|X_1)p(X_1) = p(X_2, X_3) \to$ pool
Bucket(X_2): $\sum_{X_2} P(X_4|X_2, X_3)p(X_2, X_3) = p(X_3, X_4) \to$ pool
Bucket(X_3): $\sum_{X_3} P(X_4, X_3) = p(X_4) \to$ pool
Bucket(X_4): just get $P(X_4)$ from the pool



CREDAL NETS

- Symbolic variable elimination
  multilinear constraints
- Updating $\equiv$ multilinear optimization
  (de Campos & Cozman, 2004)

Exact inference: Variable elimination

BAYESIAN NETS

- Choose an ordering of the variables (query last)
- Create a *pool* of functions with all local distributions
- For each X:

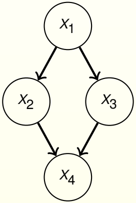
  Insert all functions that contain X in a structure called *bucket* of X and remove them from the pool

  Multiply these functions and marginalize out X

  Insert the results in the pool

bucket elimination (Dechter, 1996)
fusion algorithm for valuation algebras (Shenoy & Kohlas, 1994)

Compute $P(X_4)$ with ordering X_1, X_2, X_3, X_4
Pool $\equiv \{P(X_1), p(X_2|X_1), p(X_3|X_1), p(X_4|X_2, X_3)$
Bucket(X_1): $\sum_{X_1} P(X_2|X_1)p(X_3|X_1)p(X_1) = p(X_2, X_3) \to$ pool
Bucket(X_2): $\sum_{X_2} P(X_4|X_2, X_3)p(X_2, X_3) = p(X_3, X_4) \to$ pool
Bucket(X_3): $\sum_{X_3} P(X_4, X_3) = p(X_4) \to$ pool
Bucket(X_4): just get $P(X_4)$ from the pool



CREDAL NETS

- Symbolic variable elimination
  multilinear constraints
- Updating $\equiv$ multilinear optimization
  (de Campos & Cozman, 2004)

Exact inference: Variable elimination

BAYESIAN NETS

- Choose an ordering of the variables (query last)
- Create a *pool* of functions with all local distributions
- For each X:

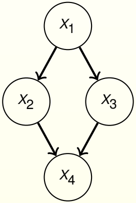
  Insert all functions that contain X in a structure called *bucket* of X and remove them from the pool

  Multiply these functions and marginalize out X

  Insert the results in the pool

bucket elimination (Dechter, 1996)
fusion algorithm for valuation algebras (Shenoy & Kohlas, 1994)

Compute $P(X_4)$ with ordering X_1, X_2, X_3, X_4
Pool $\equiv \{P(X_1), p(X_2|X_1), p(X_3|X_1), p(X_4|X_2, X_3)\}$
Bucket(X_1): $\sum_{X_1} P(X_2|X_1)p(X_3|X_1)p(X_1) = p(X_2, X_3) \rightarrow$ pool
Bucket(X_2): $\sum_{X_2} P(X_4|X_2, X_3)p(X_2, X_3) = p(X_3, X_4) \rightarrow$ pool
Bucket(X_3): $\sum_{X_3} P(X_4, X_3) = p(X_4) \rightarrow$ pool
Bucket(X_4): just get $P(X_4)$ from the pool



CREDAL NETS

- Symbolic variable elimination
  multilinear constraints
- Updating $\equiv$ multilinear optimization
  (de Campos & Cozman, 2004)

Exact inference: Variable elimination

BAYESIAN NETS

- Choose an ordering of the variables (query last)
- Create a *pool* of functions with all local distributions
- For each X:

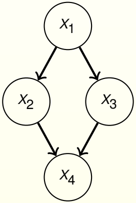
  Insert all functions that contain X in a structure called *bucket* of X and remove them from the pool

  Multiply these functions and marginalize out X

  Insert the results in the pool

bucket elimination *(Dechter, 1996)*
fusion algorithm for valuation algebras *(Shenoy & Kohlas, 1994)*

Compute $P(X_4)$ with ordering X_1, X_2, X_3, X_4
Pool $\equiv \{P(X_1), p(X_2|X_1), p(X_3|X_1), p(X_4|X_2, X_3)$
Bucket(X_1): $\sum_{X_1} P(X_2|X_1)p(X_3|X_1)p(X_1) = p(X_2, X_3) \rightarrow$ pool
Bucket(X_2): $\sum_{X_2} P(X_4|X_2, X_3)p(X_2, X_3) = p(X_3, X_4) \rightarrow$ pool
Bucket(X_3): $\sum_{X_3} P(X_4, X_3) = p(X_4) \rightarrow$ pool
Bucket(X_4): just get $P(X_4)$ from the pool



CREDAL NETS

- Symbolic variable elimination
  multilinear constraints
- Updating $\equiv$ multilinear optimization
  (de Campos & Cozman, 2004)

Exact inference: Variable elimination

BAYESIAN NETS

- Choose an ordering of the variables (query last)
- Create a *pool* of functions with all local distributions
- For each X:

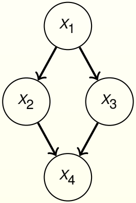
  Insert all functions that contain X in a structure called *bucket* of X and remove them from the pool

  Multiply these functions and marginalize out X

  Insert the results in the pool

bucket elimination *(Dechter, 1996)*
fusion algorithm for valuation algebras
(Shenoy & Kohlas, 1994)

Compute $P(X_4)$ with ordering X_1, X_2, X_3, X_4
Pool $\equiv \{P(X_1), p(X_2|X_1), p(X_3|X_1), p(X_4|X_2, X_3)$
Bucket(X_1): $\sum_{X_1} P(X_2|X_1)p(X_3|X_1)p(X_1) = p(X_2, X_3) \to$ pool
Bucket(X_2): $\sum_{X_2} P(X_4|X_2, X_3)p(X_2, X_3) = p(X_3, X_4) \to$ pool
Bucket(X_3): $\sum_{X_3} P(X_4, X_3) = p(X_4) \to$ pool
Bucket(X_4): just get $P(X_4)$ from the pool



CREDAL NETS

- Symbolic variable elimination
  multilinear constraints
- Updating $\equiv$ multilinear optimization
  (de Campos & Cozman, 2004)

Exact inference: Variable elimination

BAYESIAN NETS

- Choose an ordering of the variables (query last)
- Create a *pool* of functions with all local distributions
- For each X:

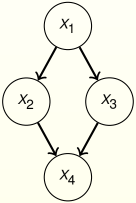
  Insert all functions that contain X in a structure called *bucket* of X and remove them from the pool

  Multiply these functions and marginalize out X

  Insert the results in the pool

bucket elimination *(Dechter, 1996)*
fusion algorithm for valuation algebras
(Shenoy & Kohlas, 1994)

Compute $P(X_4)$ with ordering X_1, X_2, X_3, X_4
Pool $\equiv \{P(X_1), p(X_2|X_1), p(X_3|X_1), p(X_4|X_2, X_3)\}$
Bucket(X_1): $\sum_{X_1} P(X_2|X_1)p(X_3|X_1)p(X_1) = p(X_2, X_3) \to$ pool
Bucket(X_2): $\sum_{X_2} P(X_4|X_2, X_3)p(X_2, X_3) = p(X_3, X_4) \to$ pool
Bucket(X_3): $\sum_{X_3} P(X_4, X_3) = p(X_4) \to$ pool
Bucket(X_4): just get $P(X_4)$ from the pool



CREDAL NETS

- Symbolic variable elimination
  multilinear constraints
- Updating $\equiv$ multilinear optimization
  (de Campos & Cozman, 2004)

Other algorithms for inference on CNs

- Inner approximation by iterative local search
  - Choose a BN consistent with the CN,
    vary parameters of a single node to improve the solution
    (da Rocha, Campos & Cozman, 2003)
- Outer approximation with *probability trees* *(Cano & Moral, 2002)*
- Integer linear programming *(de Campos & Cozman, 2007)*
- Branch and bound techniques on vertices
- Instead of propagating all the elements in the convex hull
  only the elements in the Pareto set (reduce complexity!)
  (de Campos, 2010)

Other algorithms for inference on CNs

- Inner approximation by iterative local search
  - Choose a BN consistent with the CN,
    vary parameters of a single node to improve the solution
    (da Rocha, Campos & Cozman, 2003)
- Outer approximation with *probability trees* *(Cano & Moral, 2002)*
- Integer linear programming *(de Campos & Cozman, 2007)*
- Branch and bound techniques on vertices
- Instead of propagating all the elements in the convex hull only the elements in the Pareto set (reduce complexity!)
  (de Campos, 2010)

Other algorithms for inference on CNs

- Inner approximation by iterative local search
  - Choose a BN consistent with the CN,
    vary parameters of a single node to improve the solution
    (da Rocha, Campos & Cozman, 2003)
- Outer approximation with *probability trees* *(Cano & Moral, 2002)*
- Integer linear programming *(de Campos & Cozman, 2007)*
- Branch and bound techniques on vertices
- Instead of propagating all the elements in the convex hull
  only the elements in the Pareto set (reduce complexity!)
  (de Campos, 2010)

Other algorithms for inference on CNs

- Inner approximation by iterative local search
  - Choose a BN consistent with the CN,
    vary parameters of a single node to improve the solution
    (da Rocha, Campos & Cozman, 2003)
- Outer approximation with *probability trees* *(Cano & Moral, 2002)*
- Integer linear programming *(de Campos & Cozman, 2007)*
- Branch and bound techniques on vertices
- Instead of propagating all the elements in the convex hull only the elements in the Pareto set (reduce complexity!)
  (de Campos, 2010)

Other algorithms for inference on CNs

- Inner approximation by iterative local search
  - Choose a BN consistent with the CN,
    vary parameters of a single node to improve the solution
    (da Rocha, Campos & Cozman, 2003)
- Outer approximation with *probability trees* *(Cano & Moral, 2002)*
- Integer linear programming *(de Campos & Cozman, 2007)*
- Branch and bound techniques on vertices
- Instead of propagating all the elements in the convex hull
  only the elements in the Pareto set (reduce complexity!)
  (de Campos, 2010)

Other algorithms for inference on CNs

- Inner approximation by iterative local search
  - Choose a BN consistent with the CN,
    vary parameters of a single node to improve the solution
    (da Rocha, Campos & Cozman, 2003)
- Outer approximation with *probability trees* *(Cano & Moral, 2002)*
- Integer linear programming *(de Campos & Cozman, 2007)*
- Branch and bound techniques on vertices
- Instead of propagating all the elements in the convex hull
  only the elements in the Pareto set (reduce complexity!)
  (de Campos, 2010)

Other guys

Other IPGMs

- CNs with epistemic irrelevance (de Cooman) and epistemic independence (Cozman)
- Imprecise Markov Chains (Skulj)
- Hierarchical models (Cattaneo)
- Imprecise Markov decision processes (MDP) (Cozman)
- Qualitative probabilistic nets (Van der Gaag)
- Possibilistic networks (PGM with BFs)
- Imprecise decision Trees (Ekenberg, Jaffray)

Still to be formalized

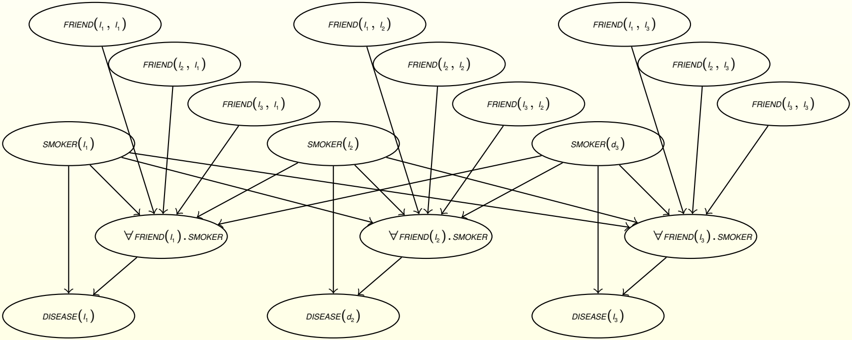
- Imprecise Markov random fields and iHMM
- Imprecise influence diagrams

Links with CNs

- Precise influence diagrams, MAP problems on BNs, . . .

$CR\mathcal{ALC}$ probabilistic logic with IPs *(Cozman, 2008)*

- Description logic with interval of probabilities
- N individuals $(I_1, \ldots, I_n)$,
  $P(smoker(I_i)) \in [.3, .5]$, $P(friend(I_j, I_i)) \in [.0, .5]$,
  $P(disease(I_i)|smoker(I_i), \forall friend(I_j, I_i).smoker(I_i)) = \ldots$
- $\underline{P}(disease)$? Inference $\equiv$ updating of a (large) binary CN
- In a sens symbolic (or OO) CNs



Future directions for CNs

- Inference algorithms
  - Inference based on Pareto set (de Campos)
  - Gibb's sampling
  - Joint tree
- Learning CNs from data
  - Structural learning (next talk)
  - Imprecise EM
- More "bridges" with BNs world
- Continuous variables (Benavoli)
- Undirected Models (random Markov fields with imprecision)
- Applications, applications, applications, applications, applications