Inference

A. Antonucci, G. Corani, **D. Maua**

{alessandro,giorgio,denis}@idsia.ch

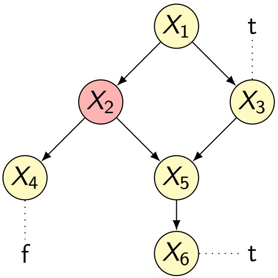
Istituto "Dalle Molle" di Studi sull'Intelligenza Artificiale
Lugano (Switzerland)

IJCAI-13

Inference

Outline

- Updating
    - under strong independence
    - under epistemic irrelevance
- Decision making



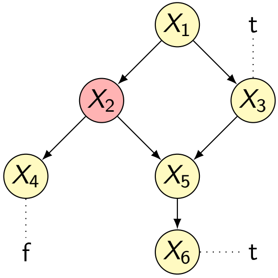
Updating under strong independence

Example

$$\max_P / \min_P P(X_2 = t | X_3 = t, X_4 = f, X_6 = t)\,,$$

where P is a probability measure satisfying:

- Markov condition:
  $P(X_i|\mathrm{Nd}_i) = P(X_i|\mathrm{Pa}_i)$
- Local constraints:
  $P(X_i|\mathrm{Pa}_i = \pi) \in K(X_i|\pi)$



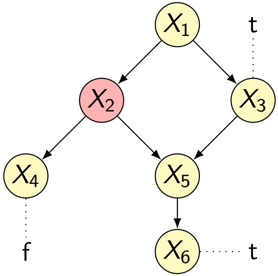
Updating under strong independence

Example

$$\max_P / \min_P P(X_2 = t | X_3 = t, X_4 = f, X_6 = t)\,,$$

where P is a probability measure satisfying:

- Markov condition:
  $P(X_i|\mathrm{Nd}_i) = P(X_i|\mathrm{Pa}_i)$
- Local constraints:
  $P(X_i|\mathrm{Pa}_i = \pi) \in K(X_i|\pi)$



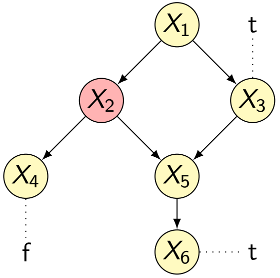
Updating under strong independence

Example

$$\max_P / \min_P P(X_2 = t | X_3 = t, X_4 = f, X_6 = t) \,,$$

where P is a probability measure satisfying:

- Factorization:
  $P(\{X_i\}) = \prod_i P(X_i|\mathrm{Pa}_i)$
- Local constraints:
  $P(X_i|\mathrm{Pa}_i = \pi) \in K(X_i|\pi)$



Updating under strong independence

Definition

$$\begin{aligned}
\text{Given:}\quad & \text{Credal net, query } X_q = x_q,\ \text{evidence } \{X_e = x_e\} \\
\text{Max/Minimize:}\quad & P(x_q|\{x_e\}) \\
\text{Subject to:}\quad & P(x_q|\{x_e\}) = \frac{\sum_{\{X_i\}\sim\{x_q,x_e\}} \prod_i P(X_i|\mathrm{Pa}_i)}{\sum_{\{X_i\}\sim\{x_e\}} \prod_i P(X_i|\mathrm{Pa}_i)} \\
& P(X_i|\mathrm{Pa}_i) \in K(X_i|\mathrm{Pa}_i) \quad [\forall i]
\end{aligned}$$

Updating under strong independence

Very difficult problem:

- Subsumes MAP inference in Bayesian networks with latent variables, which is $\mathrm{NP}^{\mathrm{PP}}$-complete [de Campos & Cozman, 2005]

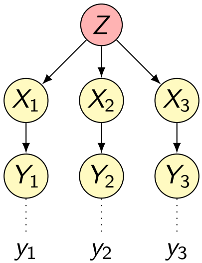
  (insert auxiliary vacuous nodes to represent selection of values for MAP variables)

- Coincides with updating in Bayesian networks if all nodes are precise, in which case is PP-complete
- NP-complete even in tree-shaped networks with a single ternary variable and all others binary [Maua et al., 2013]

Updating under strong independence

Very difficult problem:

- NP-hard even in tree-shaped networks with a single ternary variable and all others binary [Maua et al., 2013]

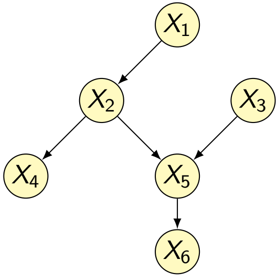


Updating under strong independence

Very difficult problem *unless*

- Network is polytree-shaped and all variables are binary: then 2U finds bounds in polynomial time [Fagiuoli & Zaffalon, 1998]

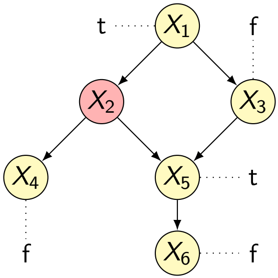
  (2U extends Pearl's belief propagation for updating Bayesian nets to interval propagation)



Updating under strong independence

Very difficult problem *unless*

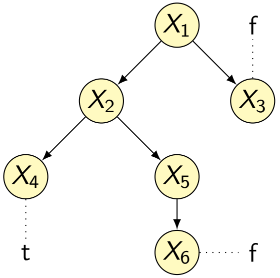
- All variables (but the query) are observed (i.e., there are no missing data)



Updating under strong independence

Very difficult problem *unless*

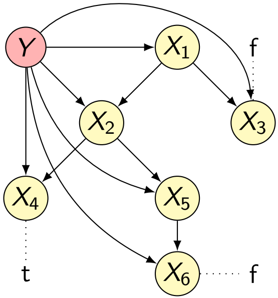
- Network models a tree-augmented classifier: conditional on a root query node, remaining network is a tree over observed variables [Fagiuoli & Zaffalon, 2000]
- Application: robust classification



Updating under strong independence

Very difficult problem *unless*

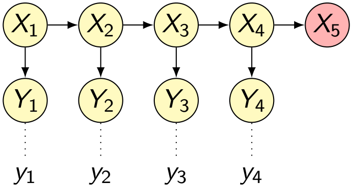
- Network models a tree-augmented classifier: conditional on a root query node, remaining network is a tree over observed variables [Fagiuoli & Zaffalon, 2000]
- Application: robust classification



Updating under strong independence

Very difficult problem unless

- Network models a Hidden Markov Model, query is the “last” node [Maua et. al, 2013]
- Works also for *Markov Chains* (assign uniform $\Pr(Y_i = y_i | X_i)$)
- Applications: robust filtering, prediction



Updating under strong independence

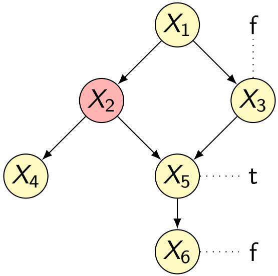
Very difficult even to approximate *unless*

- Network has bounded treewidth, and variable cardinalities are bounded: then there is an FPTAS [Maua et al., 2012]
- In practice: treewidth < 3, ternary variables
- Computes not only bounds, but (inner approx. of) credal sets

Updating under strong independence

Requisite Graph

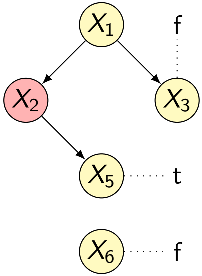
- Remove barren nodes
- Replace observed and query variables with binary variables
- Drop arcs leaving observed variables



Updating under strong independence

Requisite Graph

- Remove barren nodes
- Replace observed and query variables with binary variables
- Drop arcs leaving observed variables



Updating under strong independence

- Mathematical programming approaches
- Combinatorial optimization approaches

Mathematical programming approaches

Multilinear Programming [de Campos & Cozman, 2004]

Max/Minimize: $P(x_q | \{x_e\})$

Subject to: $P(x_q | \{x_e\}) = \dfrac{\sum_{\{X_i\} \sim \{x_q, x_e\}} \prod_i P(X_i | \mathrm{Pa}_i)}{\sum_{\{X_i\} \sim \{x_e\}} \prod_i P(X_i | \mathrm{Pa}_i)}$

$[\forall i]\; P(X_i | \mathrm{Pa}_i) \in K(X_i | \mathrm{Pa}_i)$

Mathematical programming approaches

Multilinear Programming [de Campos & Cozman, 2004]

$$\begin{aligned}
\text{Max/Minimize:}\quad & P(x_q|\{x_e\}) \\
\text{Subject to:}\quad & G(X_q) = \delta_{x_q}(X_q) - P(x_q|\{x_e\}) \\
& \sum_{\{X_i\}\sim\{x_e\}} G(X_q) \prod_i P(X_i|\mathrm{Pa}_i) = 0 \\
& [\forall i]\, P(X_i|\mathrm{Pa}_i) = K(X_i|\mathrm{Pa}_i)
\end{aligned}$$

Mathematical programming approaches

Multilinear Programming [de Campos & Cozman, 2004]

Multilinear Program with variables $\{P(X_i|\mathrm{Pa}_i)\}$, $G(X_q)$ and $P(x_q|\{x_e\})$

$$\begin{aligned}
\text{Max/Minimize: } & P(x_q|\{x_e\}) \\
\text{Subject to: } & G(X_q) = \delta_{x_q}(X_q) - P(x_q|\{x_e\}) \\
& \sum_{\{X_i\}\sim\{x_e\}} G(X_q) \prod_i P(X_i|\mathrm{Pa}_i) = 0 \\
& P(X_i|\mathrm{Pa}_i) = K(X_i|\mathrm{Pa}_i) \quad [\forall i]
\end{aligned}$$

Mathematical programming approaches

Multilinear Programming [de Campos & Cozman, 2004]

- Number of variables (in the program) is exponential in the network treewidth
- Not many solvers available; numerical problems often arise
- Solves networks with at most a few hundred vaiables
- There are other ways to linearization

Mathematical programming approaches

Integer Programming [de Campos & Cozman, 2007]

It is possible to recast the problem as a Mixed-Linear Integer Program [de Campos & Cozman, 2007], which

- has size exponential in the network pathwidth (which is always greater than treewidth)
- can be solved with standard MILP solvers
- usually outperforms multilinear programming (might be due to poor implementation)
- still can only solve networks with at most a few hundred vaiables

Mathematical programming approaches

Linear Relaxation [Antonucci et al., 2013]

Repeat until convergence:
For $k = 1, \ldots, n$ solve

$$\begin{aligned}
\text{Max/Minimize: } & P(x_q | \{x_e\}) \\
\text{Subject to: } & G(X_q) = \delta_{x_q}(X_q) - P(x_q|\{x_e\}) \\
& \sum_{\{X_i\} \sim \{x_e\}} G(X_q) P(X_k|\mathrm{Pa}_k) \prod_{i \neq k} P(X_i|\mathrm{Pa}_i) = 0 \\
& P(X_k|\mathrm{Pa}_k) \in K(X_k|\mathrm{Pa}_k) \\
& [\forall i \neq k]\, P(X_i|\mathrm{Pa}_i) \text{ is fixed}
\end{aligned}$$

Mathematical programming approaches

Linear Relaxation [Antonucci et al., 2013]

- Each iteration of loop requires solving updating in a Bayesian net (with same graph structure) to obtain the constants of the linear program
- No accuracy guarantees
- More accurate than other approximate methods

Combinatorial optimization approaches

Combinatorial Search of Local Vertices [Cano et al. 1994]

$$\begin{aligned}
\text{Max/Minimize:}\quad & P(x_q|\{x_e\}) \\
\text{Subject to:}\quad & P(x_q|\{x_e\}) = \frac{\sum_{\{X_i\}\sim\{x_q,x_e\}} \prod_i P(X_i|\mathrm{Pa}_i)}{\sum_{\{X_i\}\sim\{x_e\}} \prod_i P(X_i|\mathrm{Pa}_i)} \\
& [\forall i]\; P(X_i|\mathrm{Pa}_i) \in \mathrm{ext} K(X_i|\mathrm{Pa}_i)
\end{aligned}$$

Combinatorial optimization approaches

Search Combination of Vertices of Local Credal [Cano et al., 1994]

- Requires pre-computing the vertices of each credal set (there might be exponentially many)
- Requires Bayesian network updating (in a network of same grapg structure) to evaluate each candidate solution
- Common combinatorial optimization approaches can be used: genetic programming, tabu-search, etc.

Combinatorial optimization approaches

Variable Elimination

Provably accurate:

- Propagating extreme points [Cano & Moral, 1999]: vertex-enumeration implementation (although polynomial) are not very fast
- Propagating Pareto-dominated points [Maua et al., 2012]: much faster than vertex-enumeration; can be combined with vertex-enumeration.
- Grid-like coarsening: partition space of locally propagated potentials and keep only one potential per partition. This leads to an FPTAS if network treewidth and variable cardinalities are bounded [Maua et al, 2012].

Combinatorial optimization approaches

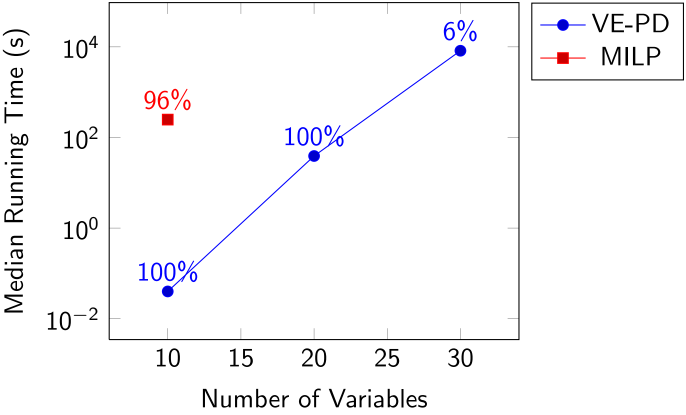
Message Passing

Provably efficient:

- AR: Extend Pearl's Belief Propagation Algorithm to propagate upper and lower probabilities in polytree-shaped networks [Tessem, 1992]
- 2U: interval propagation is exact in polytrees with binary variables [Fagiuoli & Zaffalon, 1998]
- L2U: Apply 2U in loopy graphs: solution usually degrades with increase in number of cycles or evidence variables [Ide & Cozman, 2004]
- GL2U: Binarize network then apply L2U: binarization introduces many cicles [Antonucci et al., 2010]

Updating under strong independence

What inferences can we compute? [Maua et. al, 2012]:



MILP using CPLEX solver with default parameters

Updating under strong independence

What about large networks? [Antonucci et. al, 2013]:

NETWORK	INST.	LP	GL2U
Alarm	973	0.0474	0.1218
Insurance	650	0.0767	0.0795
Polytrees	6162	0.0816	0.1724
Loopy	2963	0.0855	0.1594

Results report average mean squared error

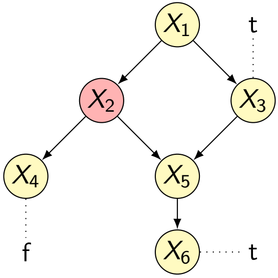
Updating under epistemic irrelevance

Example

$$\max_P / \min_P P(X_2 = t | X_3 = t, X_4 = f, X_6 = t)\,,$$

where P is a probability measure satisfying:

- Markov condition:
  $P(X_i | \mathrm{Nd}_i) = Q_{\mathrm{Nd}_i}(X_i | \mathrm{Pa}_i)$
- Local constraints:
  $Q_{\mathrm{Nd}_i}(X_i | \mathrm{Pa}_i = \pi) \in K(X_i | \pi)$



Updating under epistemic irrelevance

Definition

$$\begin{aligned} \text{Given: } & \text{Credal net, query } X_q = x_q, \text{ evidence } \{X_e = x_e\} \\ \text{Max/Minimize: } & P(x_q|\{x_e\}) \\ \text{Subject to: } & P(x_q|\{x_e\}) = \frac{\sum_{\{X_i\}\sim\{x_q,x_e\}} P(\{X_i\})}{\sum_{\{X_i\}\sim\{x_e\}} P(\{X_i\})} \\ & P(X_i|\mathrm{Nd}_i) \in K(X_i|\mathrm{Pa}_i) \quad [\forall i] \end{aligned}$$

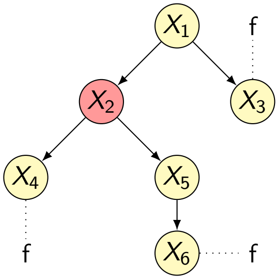
Unlike inference under strong independence, $P(\{X_i\})$ need not to factorize as a product of local conditional probability distributions $P(X_i|\mathrm{Pa}_i)$.

Updating under epistemic irrelevance

- Less studied than inference under strong independence (means fewer algorithms and complexity results)
- More imprecise: bounds computed under epistemic irrelevance are looser

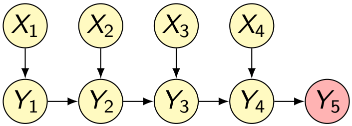
Updating under epistemic irrelevance

Tractable in tree-shaped networks
[de Cooman et. al, 2010]



Updating under epistemic irrelevance

- NP^{PP}-hard
- NP-hard in polytrees even if variables are at most ternary



Updating under epistemic irrelevance

Multlinear programming [de Campos & Cozman, 2007]

- Similar to multilinear programming approach under strong independence, except that factorization of the joint is much "coarser" under epistemic irrelevance
- Solves only very small networks (say, a dozen binary variables)

Updating

Summary

MODEL	STRONG	EPISTEMIC
Imprecise HMM (filtering)	P	P
Imprecise HMM (smoothing)	Unknown	P
Credal trees	NP-hard	P
Credal polytrees with binary variables	P	Unknown
Credal polytrees with ternary variables	NP-hard	NP-hard
Bounded treewidth networks	NP-hard	NP-hard
Credal networks	NP^{PP}-hard	NP^{PP}-hard

Decision making

Maximality

- Defines strict partial order $\succ$ on the values of query variable X_q
- $x_q \succ x'_q$ if
$$\min_P P(x_q|\{x_e\}) - P(x'_q|\{x_e\}) > 0$$
- Verifying above inequality can be reduced to checking the sign of an updating problem

Decision making

Maximality

- Maximal states: $\mathcal{X}_q = \{x_q : \nexists x'_q \text{ s.t. } x_q \succ x'_q\}$
- n-ary Single-variable case: can be reduced to $n^2 - n$ updatings
- Complexity: (asymptotically) equal to updating in the same network
- Applications: classification, decision support systems

Decision making

Maximality

- Multivariate case: There might be exponentially many maximal vectors
- $x_{q_1}, \ldots, x_{q_m} \succ x'_{q_1}, \ldots, x'_{q_m}$ if

$$\min_P P(x_{q_1}, \ldots, x_{q_m} | \{x_e\}) - P(x'_{q_1}, \ldots, x'_{q_m} | \{x_e\}) > 0$$

- HMMs: Can be computed in time linear in number of maximal vectors [de Bock & de Cooman, 2011]
- Applications: robust smoothing, multilabel classification

That's it!

Questions?

References

- Tessem, B. Interval probability propagation. Int. J. Approx. Reasoning 7(3-4): 95-120 (1992)
- Fagiuoli, E., Zaffalon, M. A note about redundancy in influence diagrams. Int. J. Approx. Reasoning 19(34), 1998.
- Fagiuoli, E., Zaffalon, M. Tree-augmented naive credal classifiers. In IPMU 2000.
- de Campos, C.P., Cozman, F.G. Inference in credal networks using multilinear programming. In STAIRS 2004.
- de Campos, C.P., Cozman, F.G. The inferential complexity of Bayesian and credal networks. In IJCAI 2005.
- de Campos, C.P., Cozman, F.G. Inference in credal networks through integer programming. In ISIPTA 2007.
- de Campos, C.P., Cozman, F.G.. Computing lower and upper expectations under epistemic independence. Int. J. Approx. Reasoning, 44(3), 2007.
- Antonucci, A., Yi, S., de Campos, C.P., Zaffalon, M. Generalized loopy 2U: a new algorithm for approximate inference in credal networks. Int. J. Approx. Reasoning 55(5), 2010.
- de Cooman, G., Hermans, F., Antonucci, A., Zaffalon, M. Epistemic irrelevance in credal nets: the case of imprecise markov trees. Int. J. Approx. Reasoning 51(9), 2010.
- Mauá, D.D., de Campos, C.P., Zaffalon, M. Updating credal networks is approximable in polynomial time. Int. J. Approx. Reasoning 53(8), 2012.
- Antonucci, A., de Campos, C.P., Huber, D., Zaffalon, M. (2013). Approximating credal network inferences by linear programming. In ECSQARU 2013.
- Mauá, D.D., de Campos, C.P., Benavoli, A., Antonucci, A. On the Complexity of Strong and Epistemic Credal Networks. In UAI 2013.