Graph-Theoretical Models for Multivariate Modeling with Imprecise Probabilities

Fabio G. Cozman

University of Sao Paulo, Brazil

Objectives:

- Review probabilistic models that are based on graphs and graph theory
- Review graph-theoretic models that represent probabilistic imprecision and indeterminacy

Why graphs?

- Compact
- Easy to handle, easy to visualize
- Efficient algorithms
- Plausible models of causal relations, of neural activity...

- Computer scientists love graphs

Contents

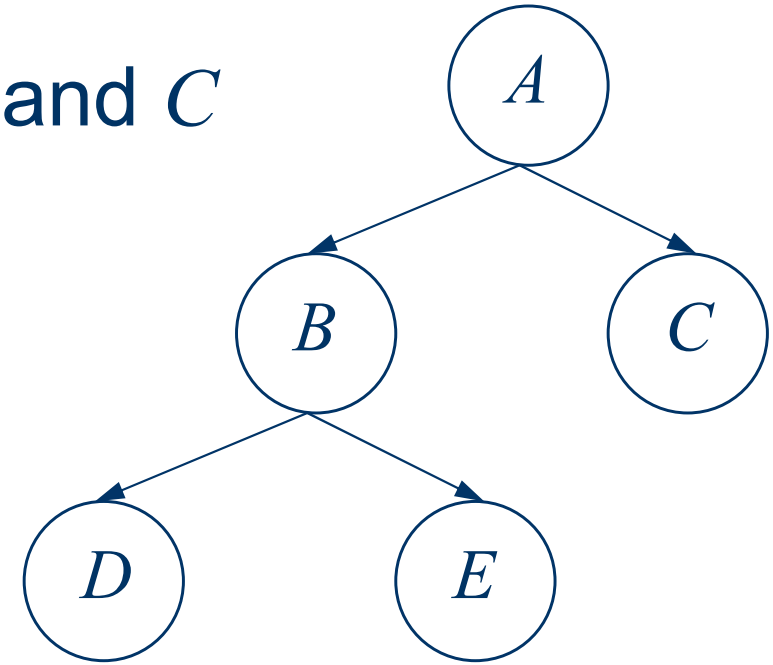
1. Trees and graphs

2. Decision trees, Bayesian networks, Markov random fields, chain graphs, influence diagrams, Markov Decision Processes

3. Imprecision/indeterminacy: graphs for belief functions, sets of probabilities, imprecise Markov Decision Processes

Trees

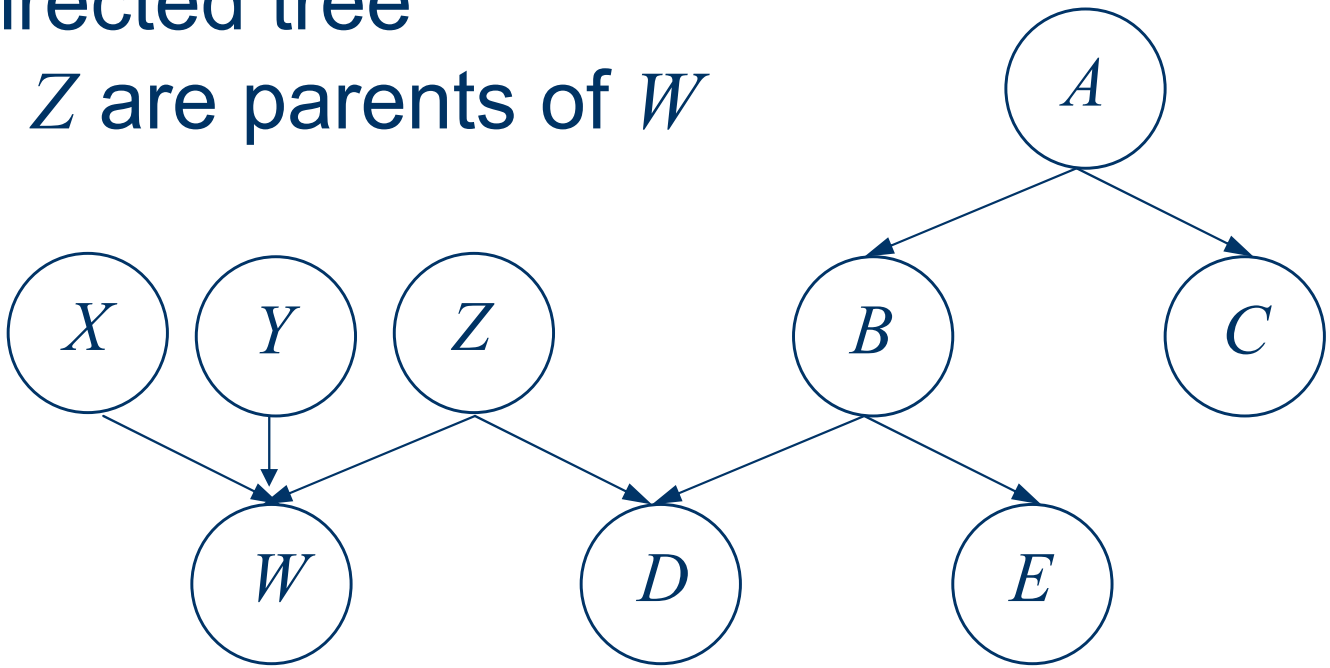
- Nodes and arcs:
  - A is the parent of B and C
  - D is a child of B



Polytrees

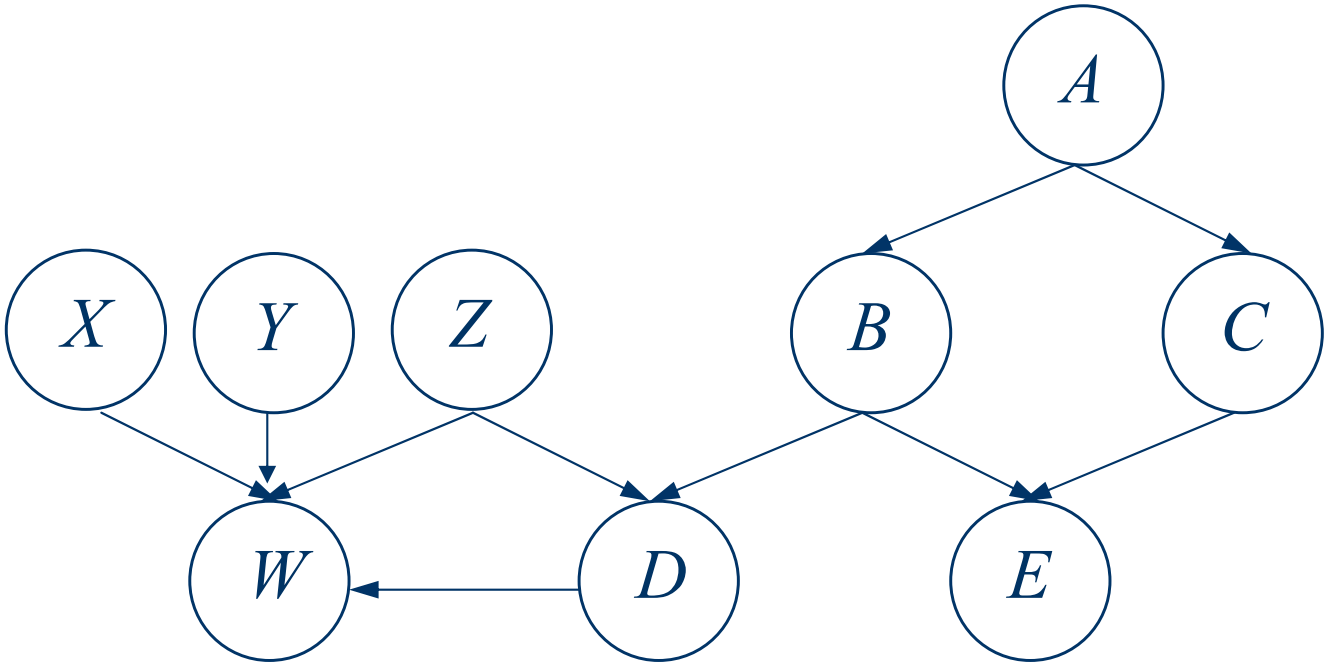
- An “undirected tree”

  X, Y, Z are parents of W

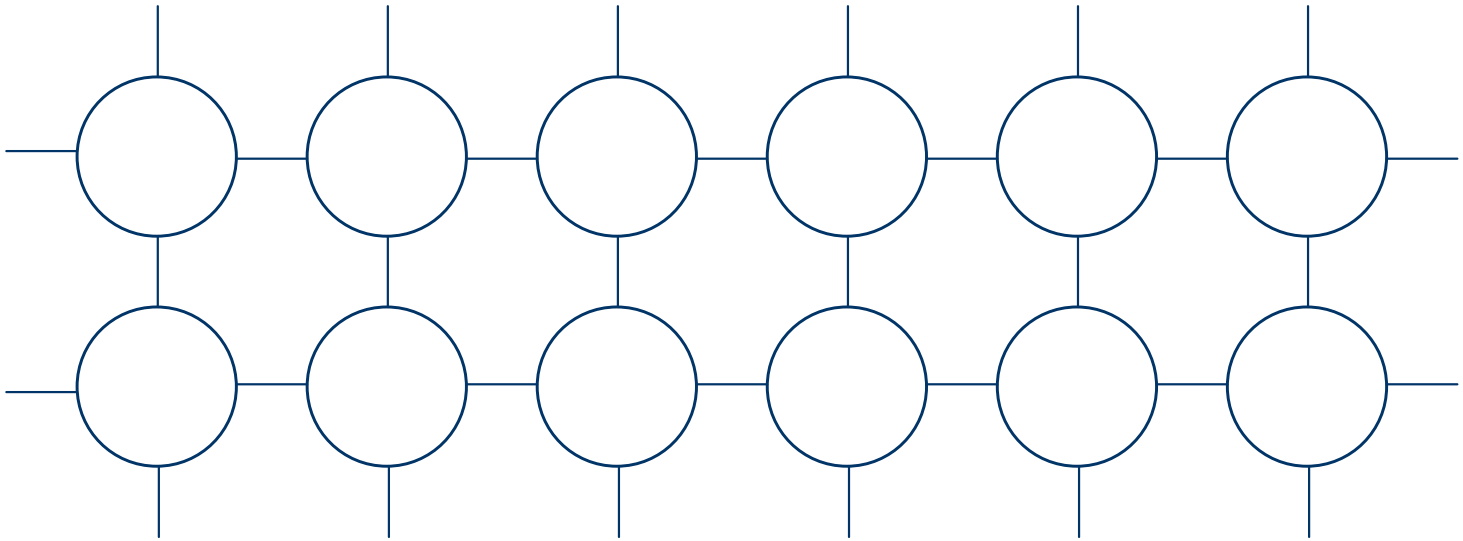


Directed graphs

- A directed *acyclic* graph:

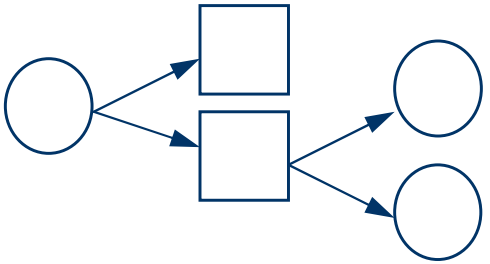


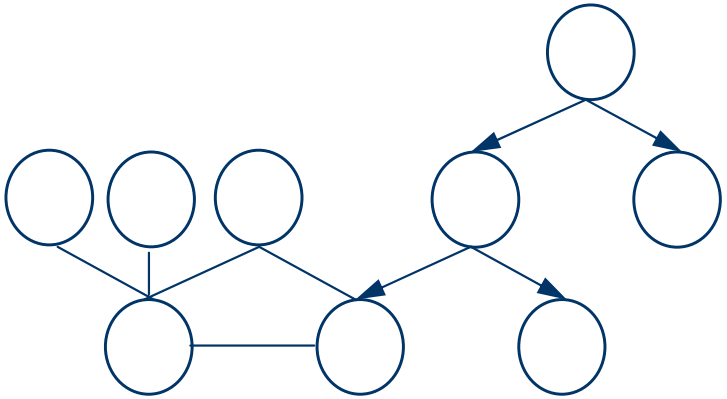
Undirected graphs



Other graphs...

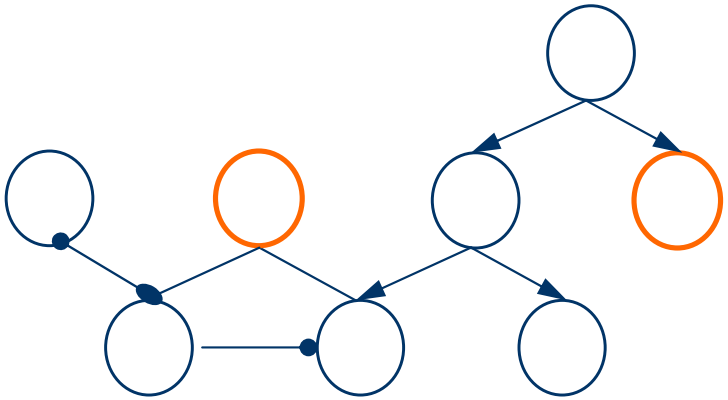
Different kinds of nodes



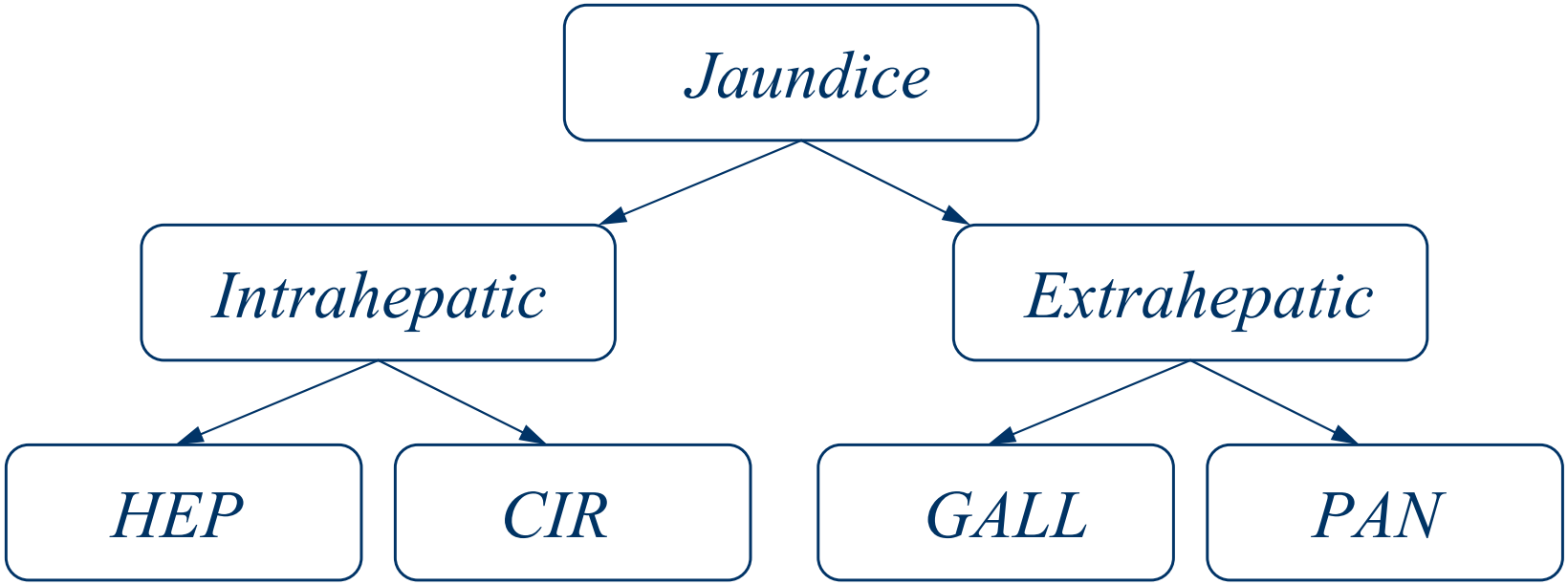


Directed and undirected arcs

Colored, marked, ...

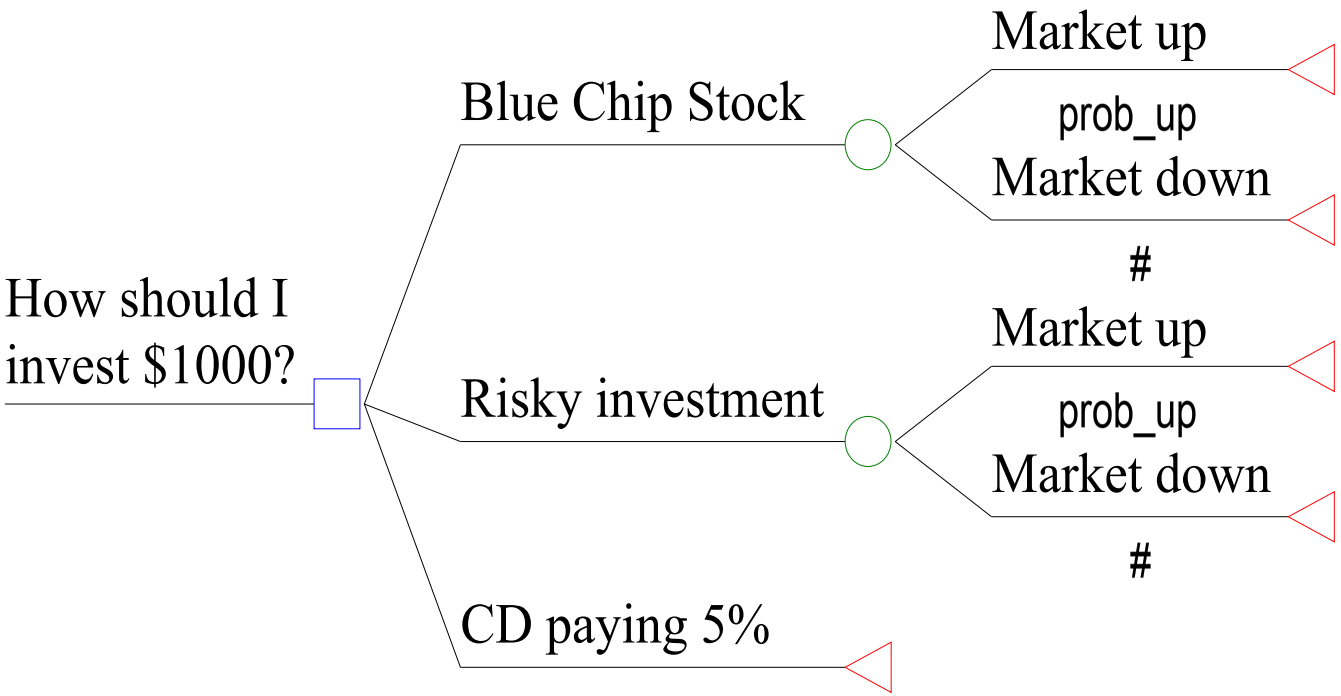


A tree for medical knowledge



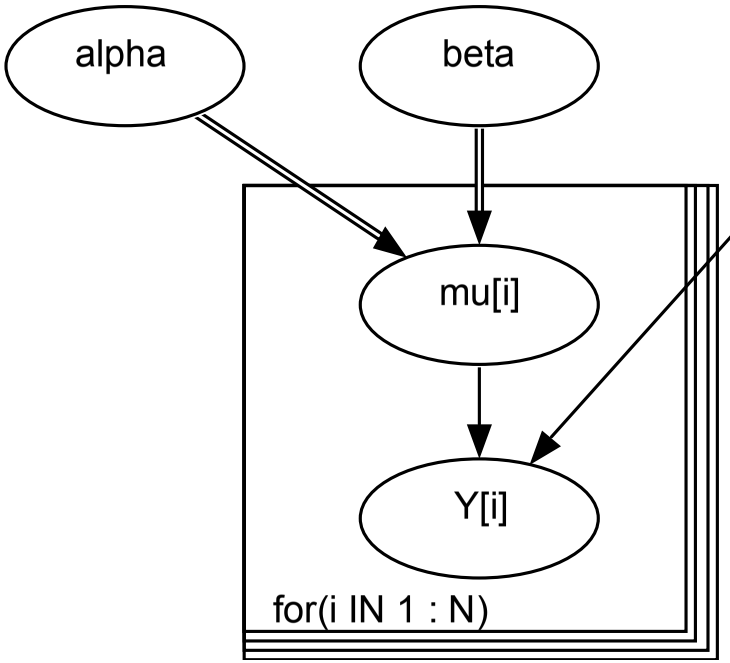
- Chandrasekharan et al, 1989
  [Shortliffe & Buchanan, 1985]

Decision trees



- Data, by TreeAge (www.treeage.com)

Hierarchical models



$$Y_i \sim N(\mu_i, \tau), \qquad i \in \{1, \ldots, N\}$$

$$\mu_i = \alpha + \beta(X_i - \bar{X})$$

- BUGS system (http://www.mrc-bsu.cam.ac.uk/bugs)

Such hierarchical models are...

- compact, easy to describe and visualize
- amenable to Monte Carlo schemes
- known under different names (*latent variable models*, *pedigrees*, *Bayesian networks*)

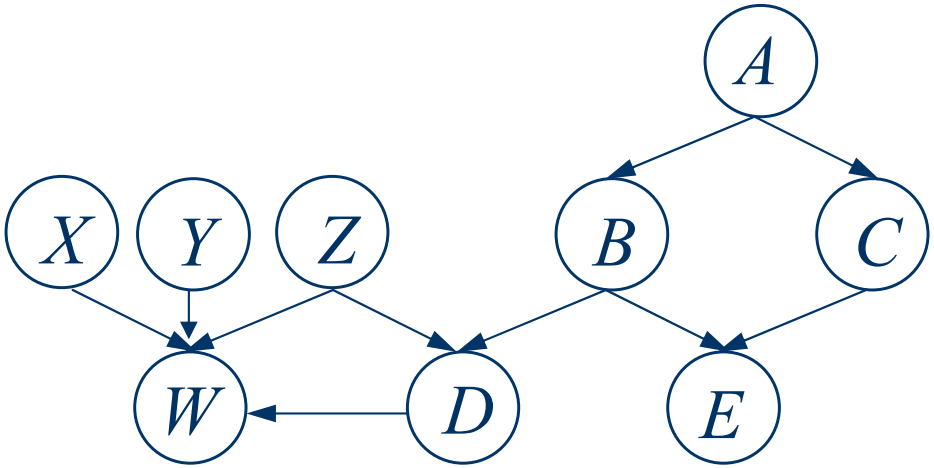
Bayesian networks (AI)

- Developed in the eighties for expert systems
  - Pathfinder (medicine)
  - Vista (space shuttle)
- Initially, attempt to capture human reasoning
- Also known as *probabilistic networks, belief networks, causal networks*

Bayesian networks are composed of…

- A directed acyclic graph
- A variable associated with each node
- A conditional distribution associated with each variable:

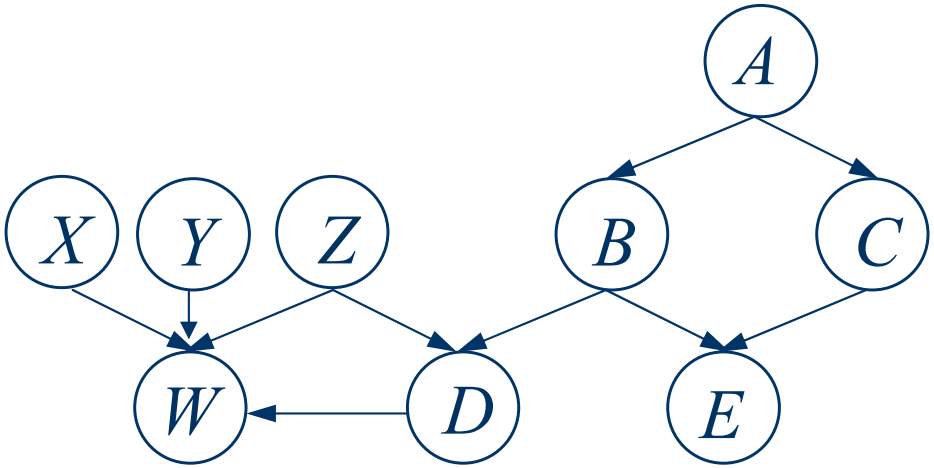
$$p(X_i|\mathrm{pa}(X_i))$$



Markov condition

- Semantics of Bayesian networks:
  Every node is independent of its nondescendants nonparents given its parents

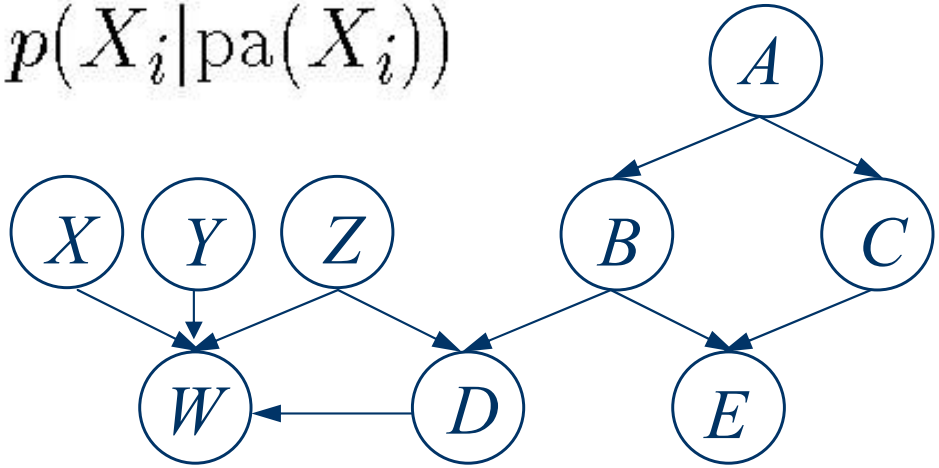
$$\begin{pmatrix} A \text{ independent } B \\ p(A|B,C) = p(A|C) \\ \text{when } p(B,C)>0 \end{pmatrix}$$



Markov condition

- Semantics of Bayesian networks:
  Every node is independent of its nondescendants nonparents given its parents
- Implies:

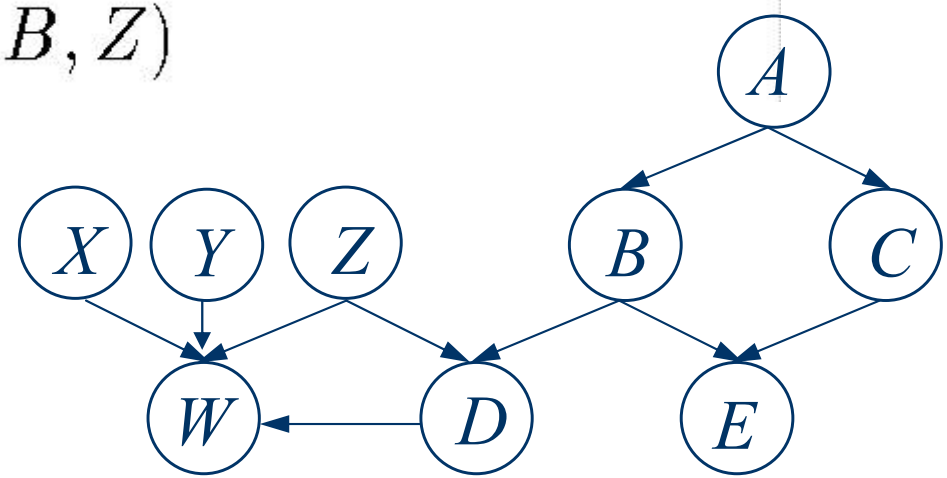
$$p(X_1, \ldots, X_n) = \Pi_i \quad p(X_i | \mathrm{pa}(X_i))$$



Completing the example

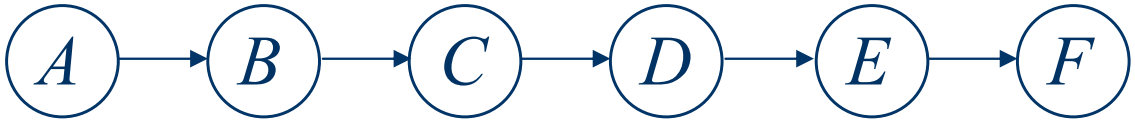
- Joint distribution:

$$p(A,B,C,D,E,W,X,Y,Z) = p(A)p(B|A)p(C|A)p(E|B,C)\,p(X)p(Y)p(Z)p(W|X,Y,Z)\,p(D|B,Z)$$



D-separation in chains

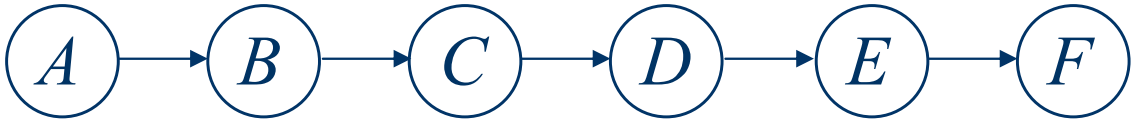
- Consider:



- A and F are *separated* by D
- A and F are independent given D

D-separation in chains

- Consider:



- A and F are *separated* by D
- A and F are independent given D

$p(ABCD|EF) = p(ABD|E) \rightarrow p(A|DEF) = p(A|E);$
$p(ABC|DE) = p(ABC|D) \rightarrow p(A|DE) = p(A|D);$
$p(A|DEF) = p(A|DE) = p(A|D); p(AE|DF) = p(AE|D); p(A|DF) = p(A|D)$

D-separation in general

- X and Y are *d-separated* by Z if along every path between X and Y, there is 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

- In a Bayesian network, if two variables are d-separated, they are independent

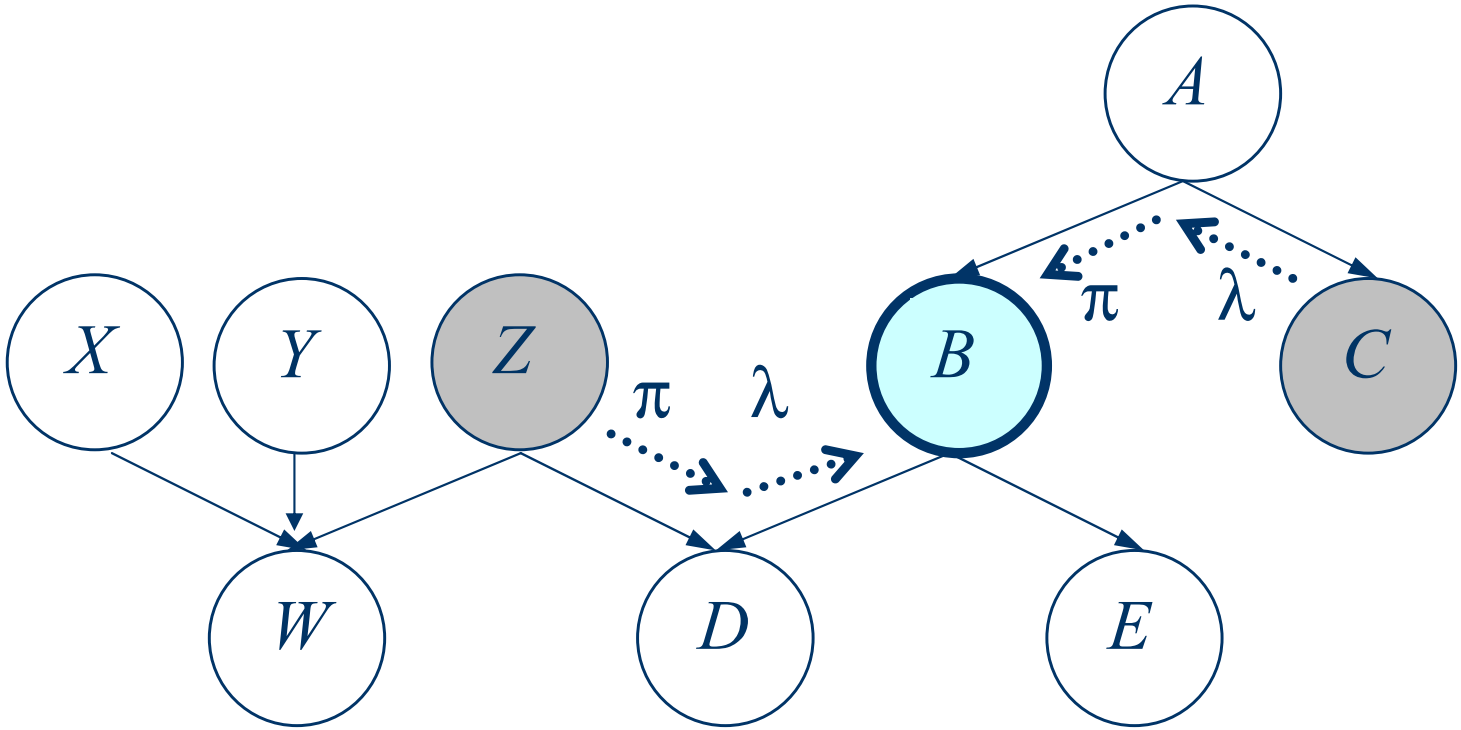
Inferences

- Typically an *inference* is a computation of posterior probability:

$$p(X_q|\mathbf{X}_E) = \frac{p(X_q, \mathbf{X}_E)}{p(\mathbf{X}_E)}$$

$$= \frac{\sum_{\mathbf{X}\setminus\{X_q,\mathbf{X}_E\}} \prod_i p(X_i|\mathrm{pa}(X_i))}{\sum_{\mathbf{X}\setminus\{\mathbf{X}_E\}} \prod_i p(X_i|\mathrm{pa}(X_i))}$$

Pearl's message propagation algorithm



Inference for generic networks

- Several *exact* algorithms:
  all essentially pass “messages” around

- Several *approximate* algorithms:
  - Monte Carlo and MCMC
  - Simplifications

Estimation/Learning

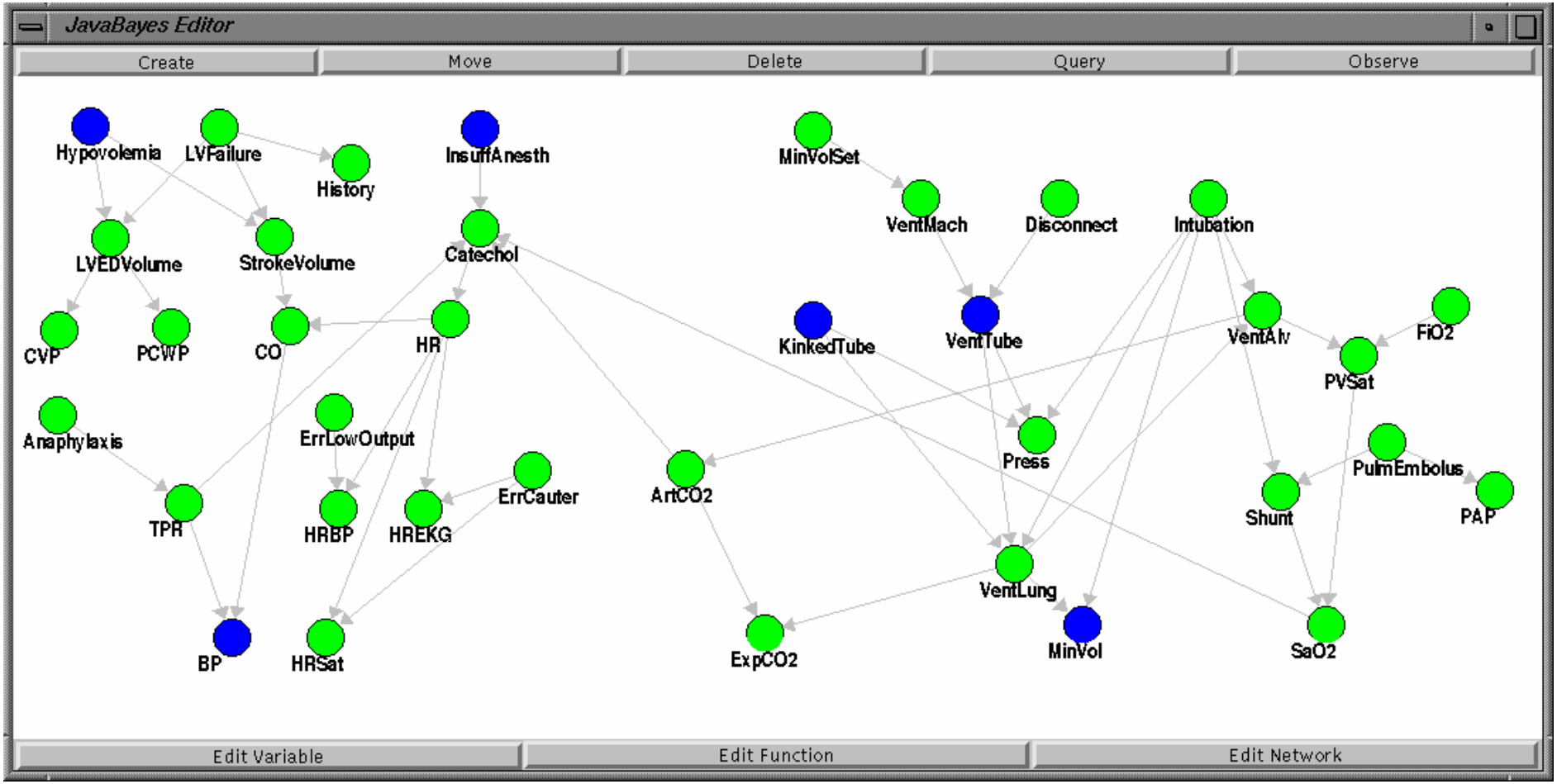
- For discrete variables, multinomial estimation
  - counting when data are complete
  - EM when data are incomplete

- *Structure learning* is much more difficult:
  - identification problems
  - several search strategies, no “winner” so far
  - exception: it is easy to learn a tree!

Applications

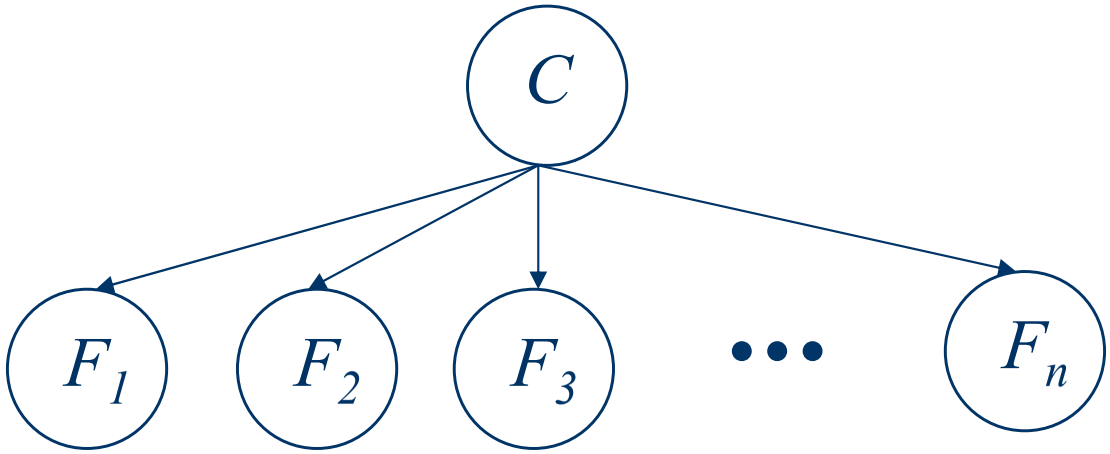
- Expert systems
- Classification
- Coding
- Dynamic systems identification and prediction
  - Dynamic Bayesian networks
- Used “inside” other models

The Alarm network



Classification

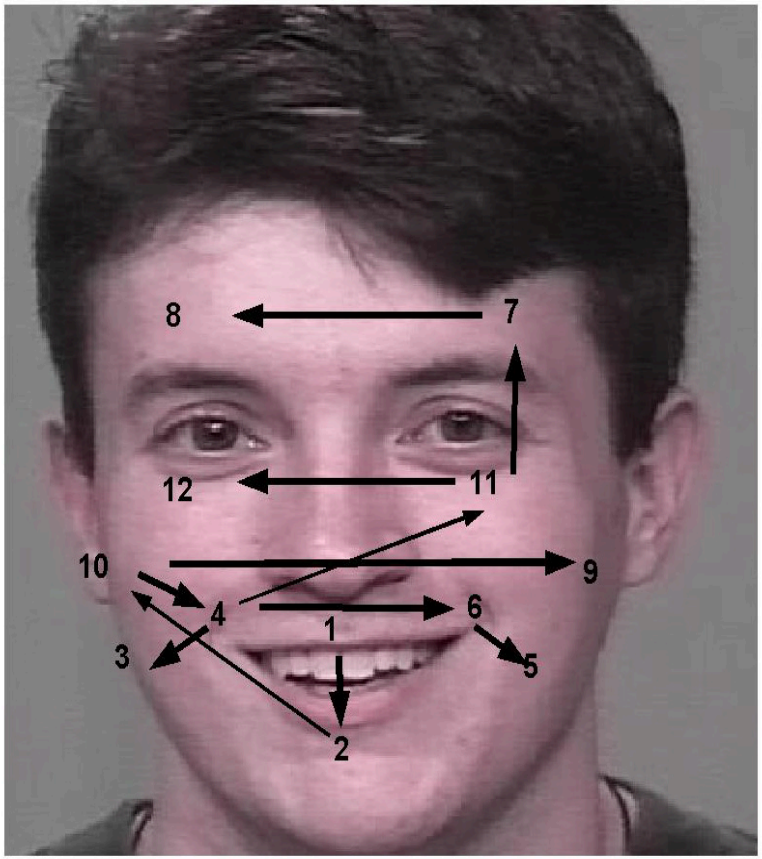
- A *classifier* receives a measurement of *features* and classifies these features
- Example: Naive Bayes



Classification of Facial Expressions

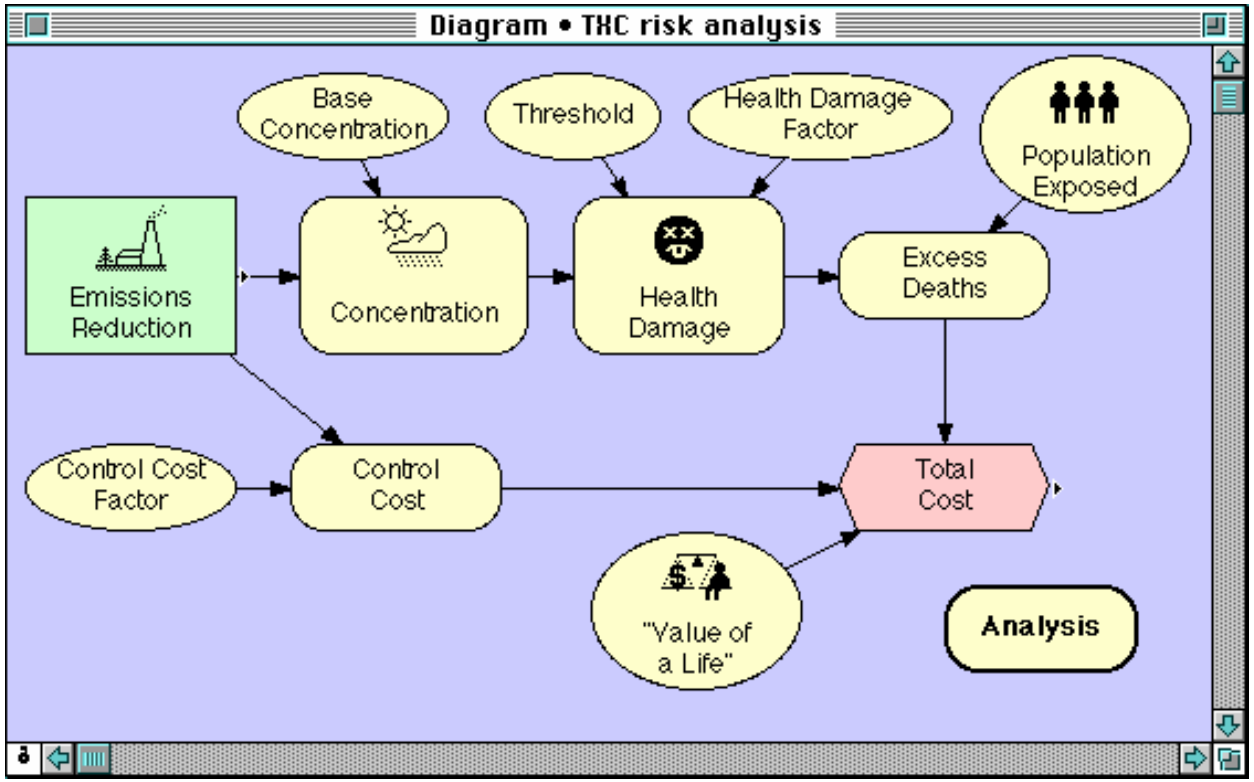
- Tree-like classifier estimated from data

- Work by Ira Cohen, UIUC and HP Labs



Influence diagrams

- Bayesian networks + decisions + values
- More compact than decision trees



Analytica

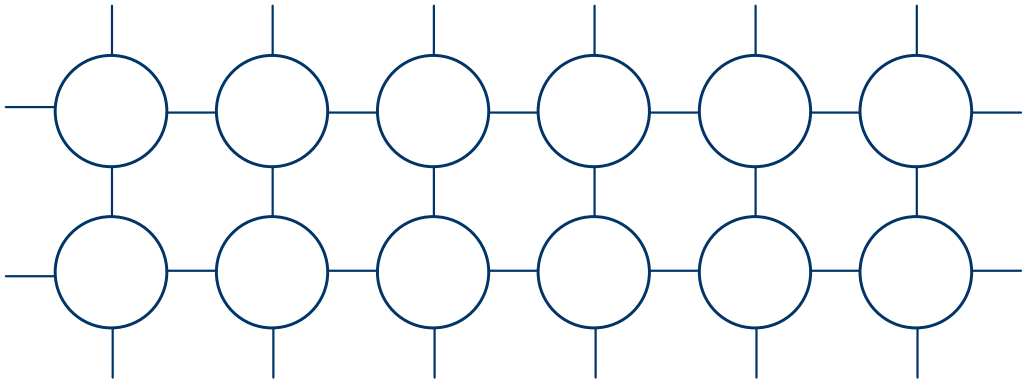
www.lumina.com

Lots of names:

- Decision trees
- Bayesian networks
- Influence diagrams
- *Markov random fields*
- *Chain graphs*
- *Markov Decision Processes*

Markov Random Fields

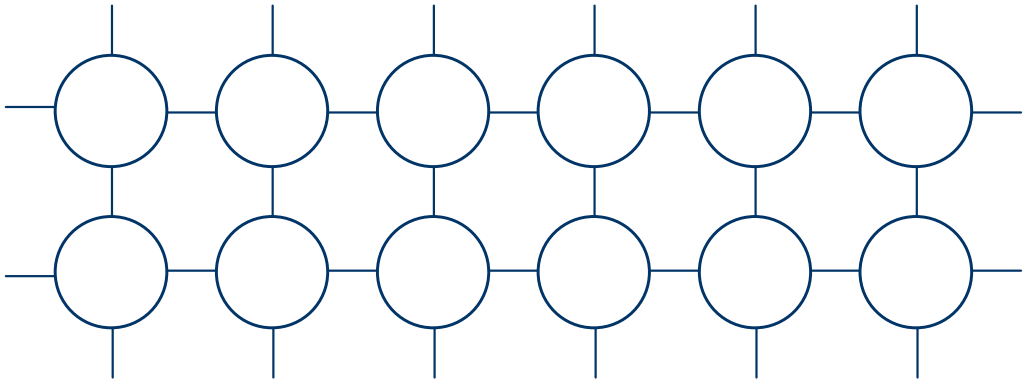
- Collection of *sites*
- Neighborhood system



- Markov condition:
  a variable is independent of all other variables (except its neighbors) given its neighbors
- Implies a unique Gibbs distribution defined by the neighbors

The Ising model

- Binary variables
  states +1 / -1

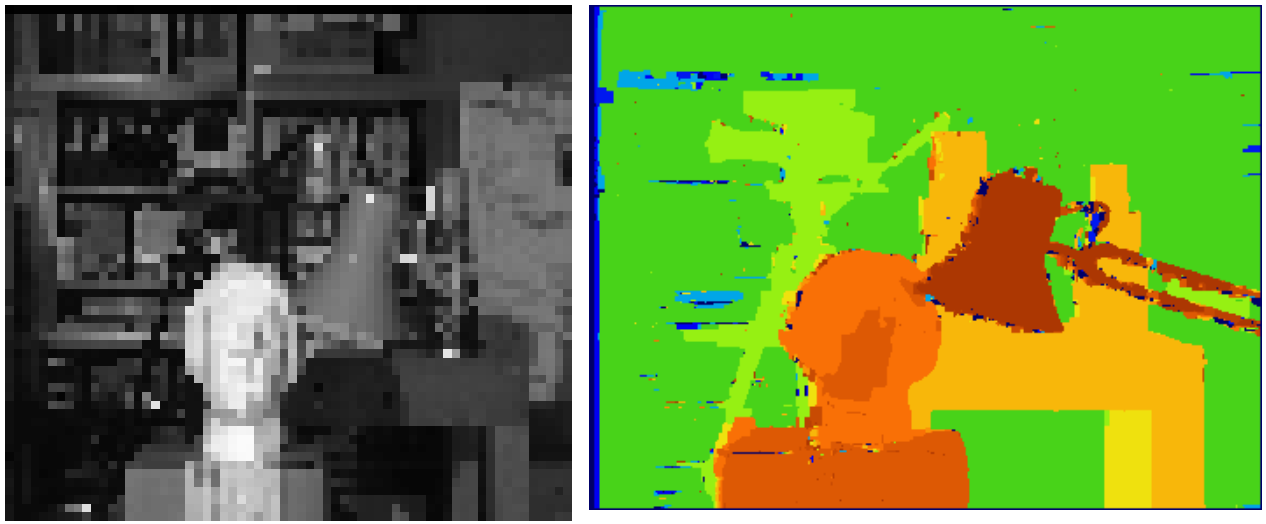


- Probability of
  configuration
  defined by functions on pairs of nodes

- Used in physics, also in neural networks

Applications

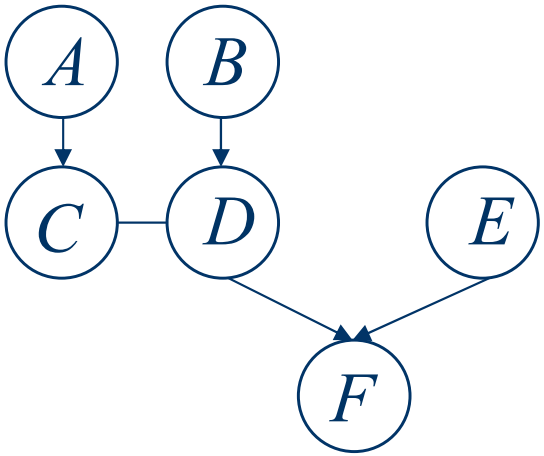
- Spatial statistics
- Image processing and computer vision



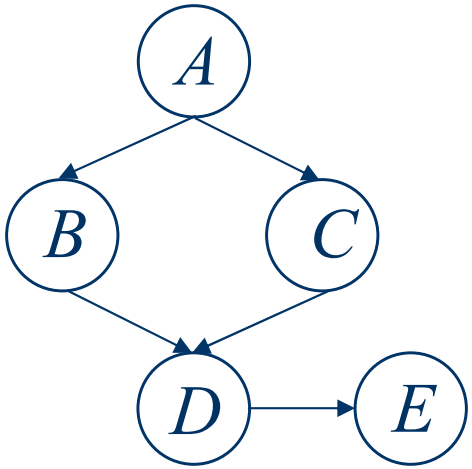
- Boykov, Veksler, Zabih 1998 - Cornell

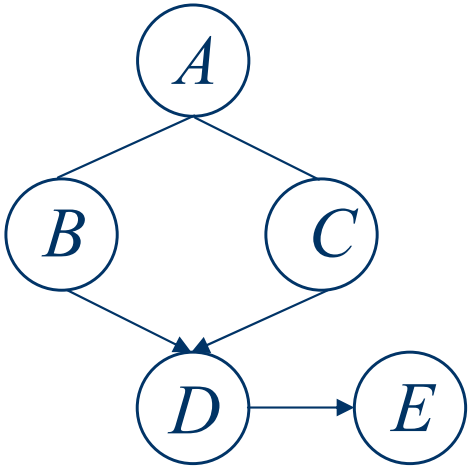
Chain graphs and other combinations

- Chain graph:



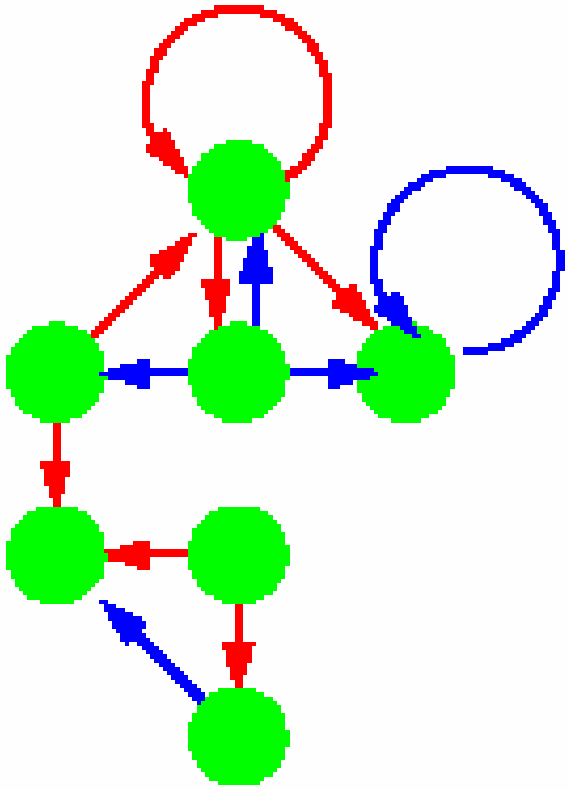
- IC* learning:





Markov Decision Processes

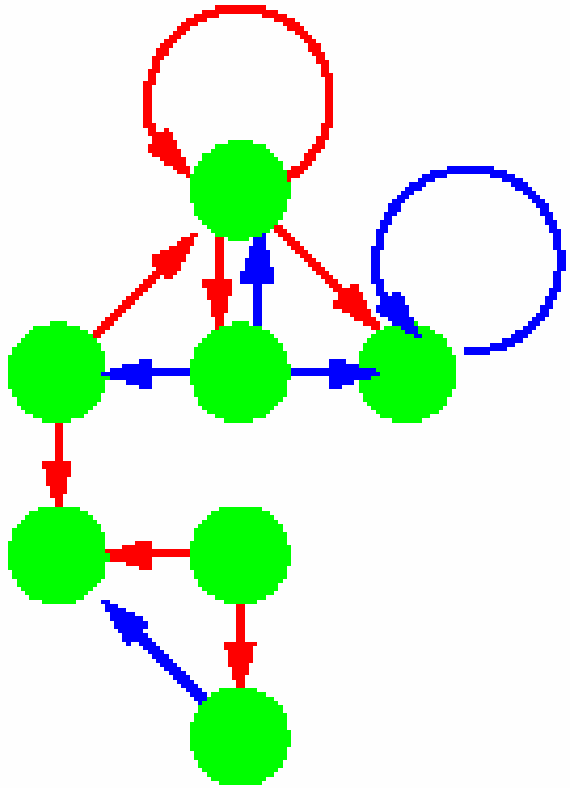
- Model for repetitive decisions
  - states S
  - actions A
  - transition probabilities
  - rewards $R(s,a)$
- Markov property:
  effects of action depend
  only on the current state



Processing Markov Decision Processes

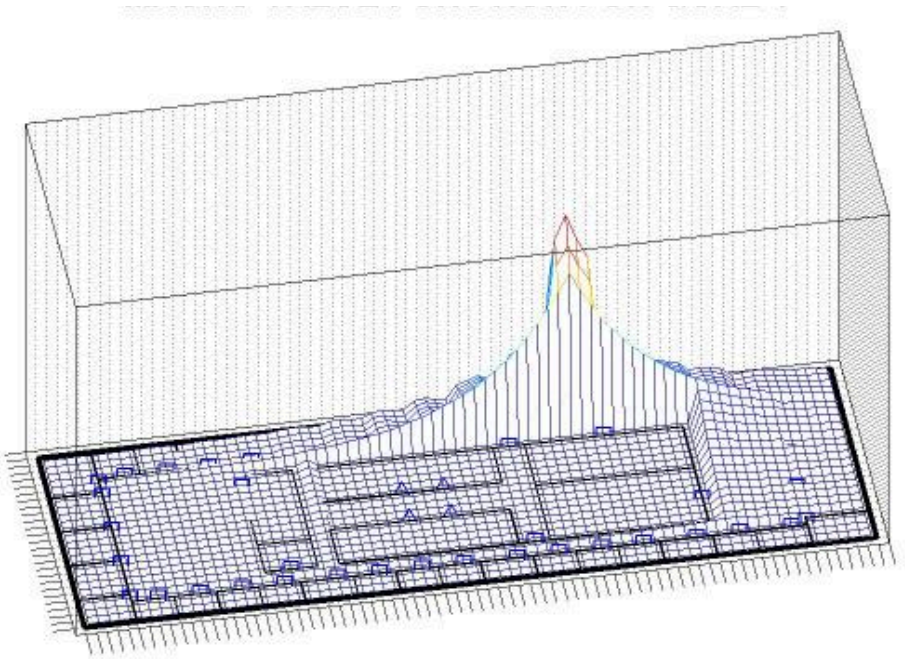
- Usually goal is to find actions to maximize "*expected*" *reward*

- Basic algorithms:
  - policy iteration
  - value iteration



Applications

- Model for planning in Operations Research and Artificial Intelligence
- Currently used in robotic planning



- Work by Terran Lane - MIT

Lots of names:

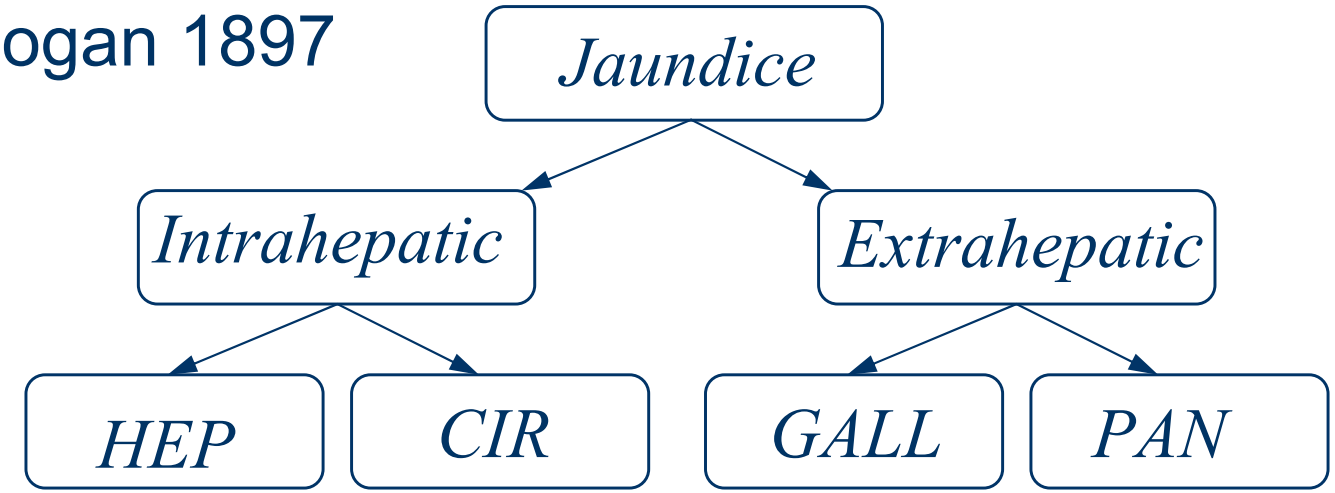
- Decision trees
- Bayesian networks
- Influence diagrams
- Markov random fields
- Chain graphs
- Markov Decision Processes
- *... POMDPs, HMMs ...*

Back to imprecision/indeterminacy: Old roots in AI

- Old desire to find faithful representations of human perceptions about uncertainty
  - Nilsson's probabilistic logic (followed by many, in some cases represented by graphs)
  - MYCIN's certainty factors, NESTOR bounds (rules and rule chaining represented by graphs)

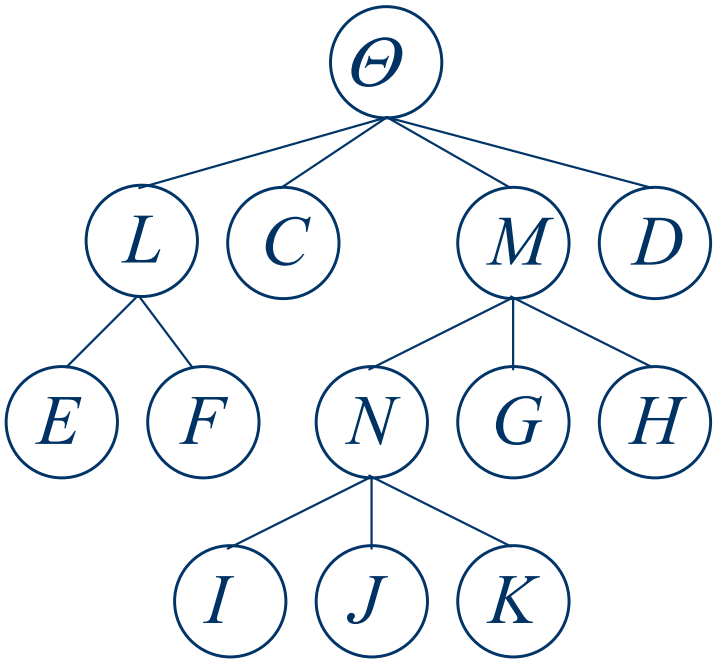
Hierarchical trees

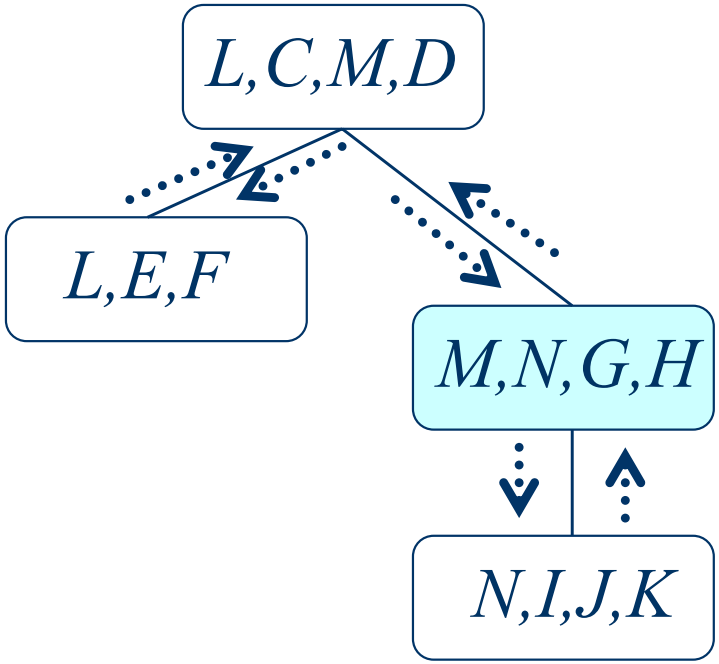
- Evidence introduced at any node (focal sets)
- Combination by Dempster rule
- Efficient algorithms available
  Gordon & Shortliffe 1985
  Shafer & Logan 1897



Shafer-Shenoy valuations

- Generalization of probability / belief functions:
  - Combination + Marginalization





  - Exact and approximate (Wilson & Moral 1996)

Closed-form inference

- Bayes and Dempter rules are easy to state

- For imprecise probabilities, what to do?
  A nice formula is

$$\underline{p}(A|B) = \frac{\underline{p}(A,B)}{\underline{p}(A,B) + \overline{p}(A^c,B)}$$

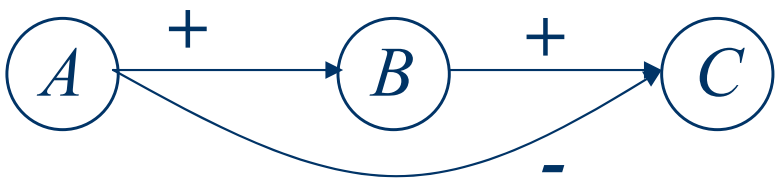
  but this only works for 2-monotone capacities

Early attempts (around 1990)

- Fertig-Breese interval influence diagrams
  - instead of probabilities, lower bounds on probabilities
  - model is 2-monotone, so locally it has closed-form
  - after each operation, go back to lower bounds
- van der Gaag's linear programming method
  - probabilistic linear inequalities
  - representation as undirected graph

Early attempts (around 1990)

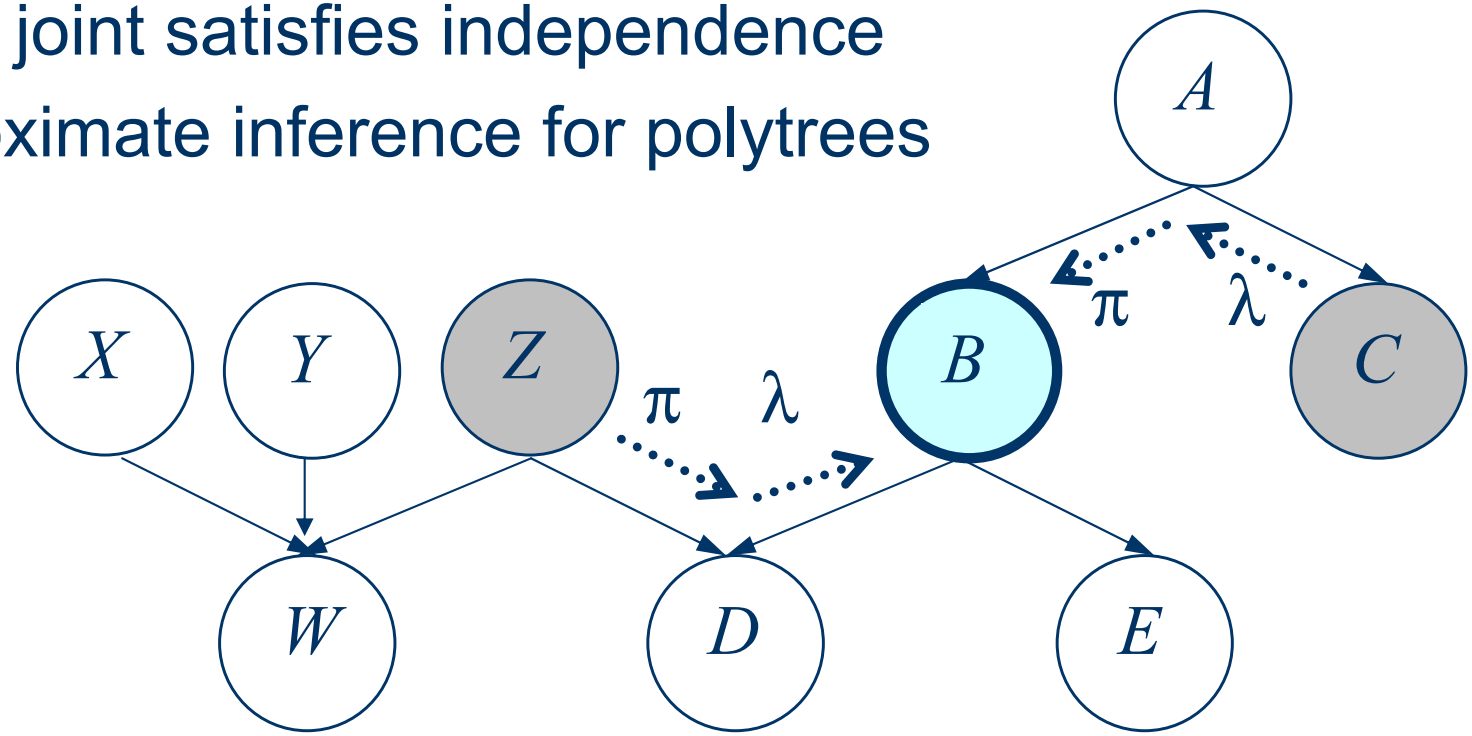
- Wellman's qualitative networks
  - instead of probabilities, just +/-
  - still very active area, with lots of extensions!



- "Order-of-magnitude" probabilities (kappa calculus, epsilon probabilities, ...)
  - still very active area!

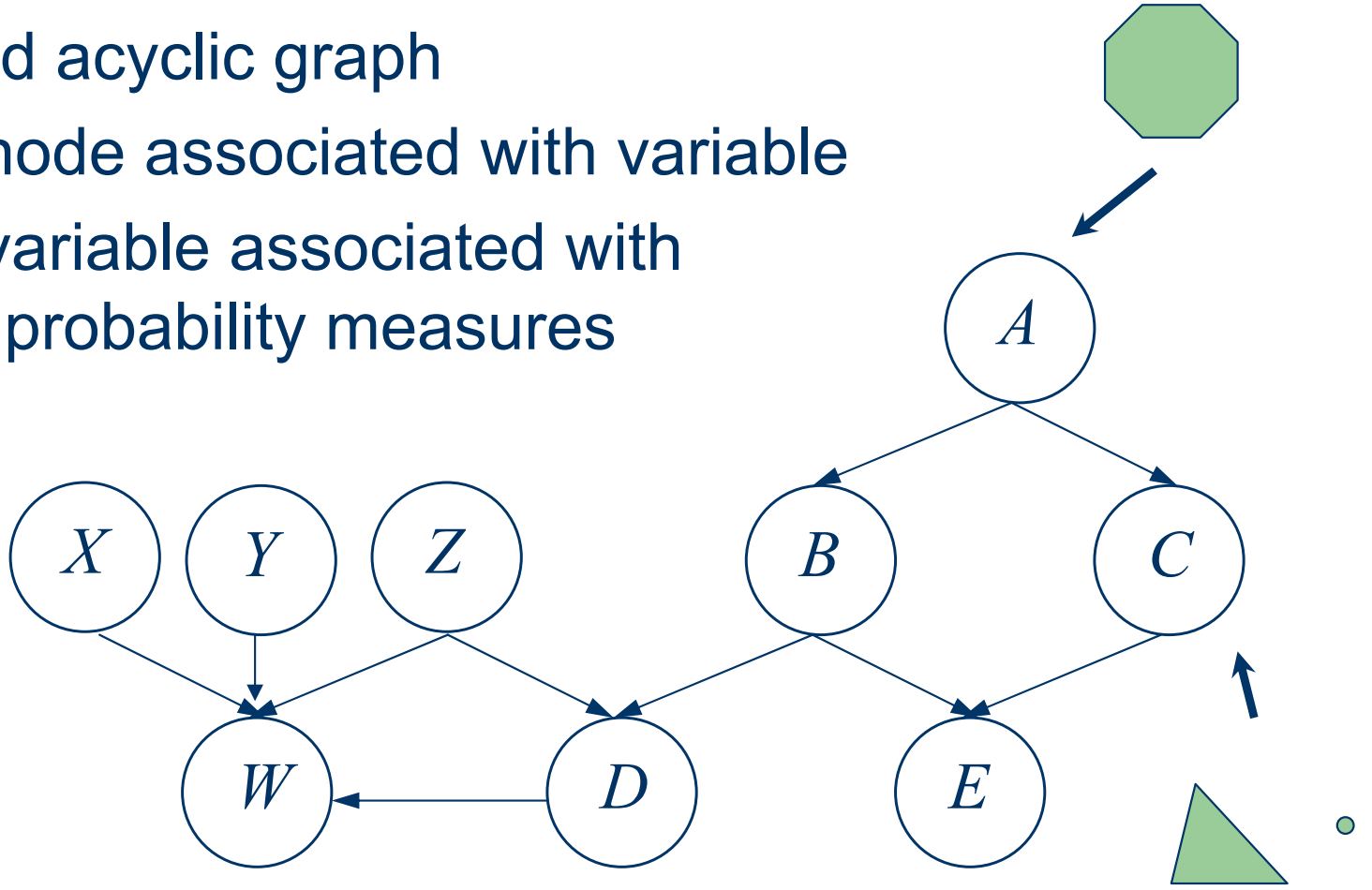
Early attempts (around 1990)

- Tessem's scheme
  - every node is associated with a set of probabilities
  - every joint satisfies independence
  - approximate inference for polytrees



Credal networks

- Directed acyclic graph
- Every node associated with variable
- Every variable associated with sets of probability measures



Semantics: Markov condition

- We would like to define a Markov condition for credal networks
- But, what is the concept of independence here?

The problem with independence

- Many concepts of independence:
  - *Epistemic independence*:
    $E[f(X)|Y] = E[f(X)]$ and $E[g(Y)|X] = E[g(Y)]$
  - *Strong independence*:
    measures in the set of probabilities satisfy "standard" independence
- There are many more!
- Even more concepts of *conditional* independence!

Epistemic independence

- If we use epistemic independence, then:
  - only one (not very efficient) algorithm to produce inferences (Cozman 2000)
  - the set of all joint measures is large: example with 4 binary variables has more than 6 million vertices!
  - little is known about d-separation

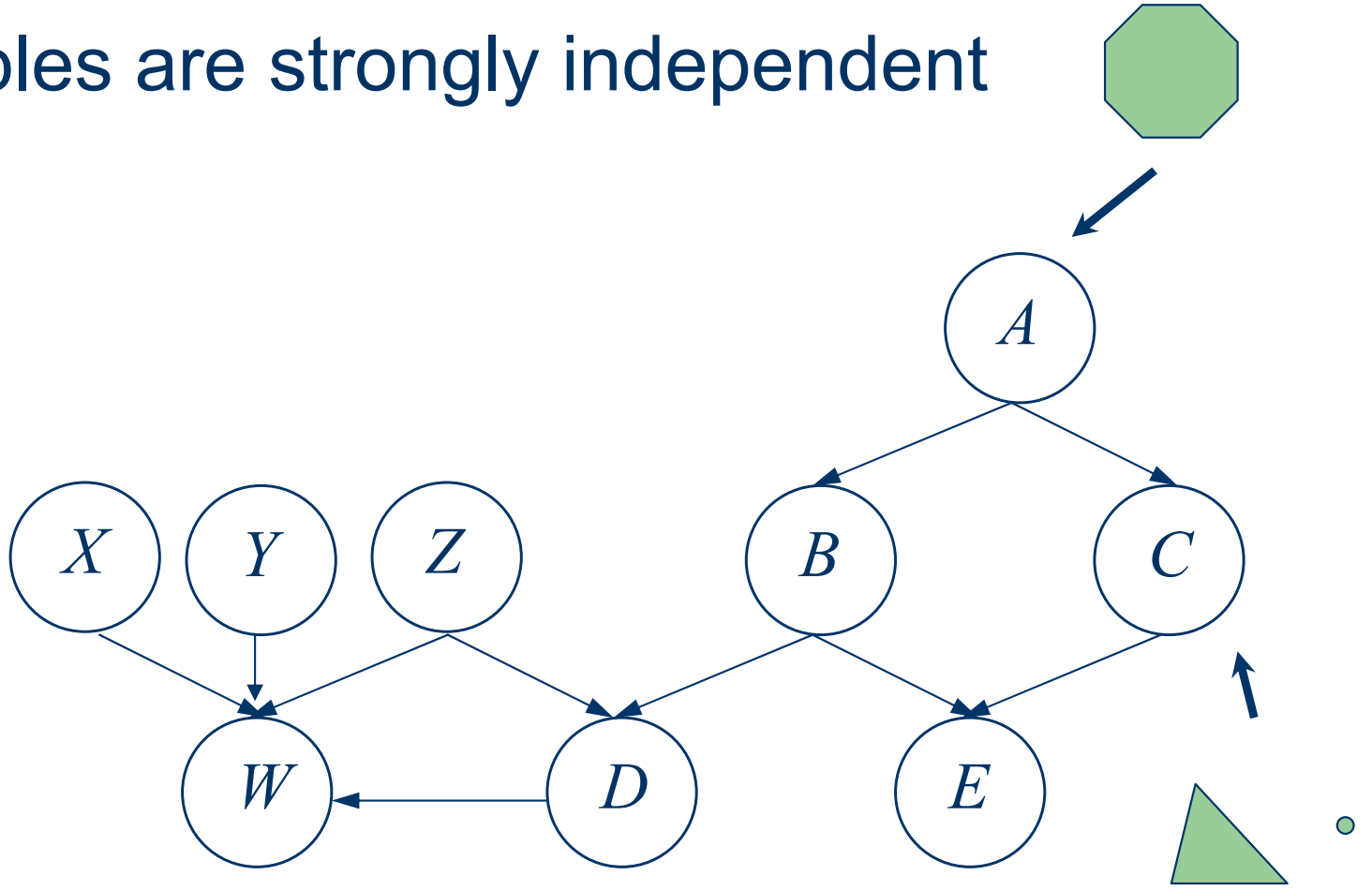
The case for strong independence

- Easy to relate to standard models
- d-separation applies
- Inference seems to be less complex

- Recent efforts by Cozman and by Moral & Cano provide solid foundation for this concept

So, we have

- Variables are strongly independent



Inference

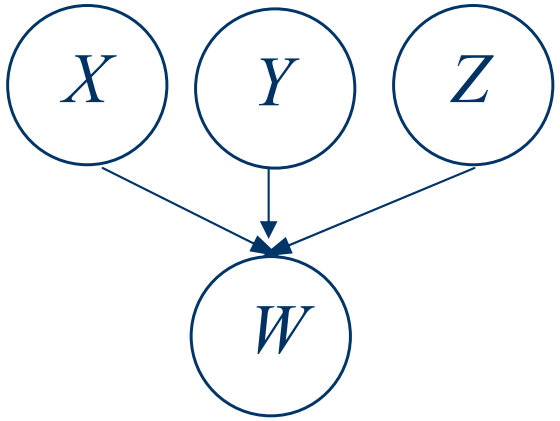
- Lower probability:

$$\underline{p}(X_q|\mathbf{X}_E) = \min \frac{\sum_{\mathbf{X}\setminus\{X_q,\mathbf{X}_E\}} \prod_i p(X_i|\mathrm{pa}(X_i))}{\sum_{\mathbf{X}\setminus\{\mathbf{X}_E\}} \prod_i p(X_i|\mathrm{pa}(X_i))}$$

- This is a *multilinear* program (a *signomial* program)
- Minimizing measures occur at vertices

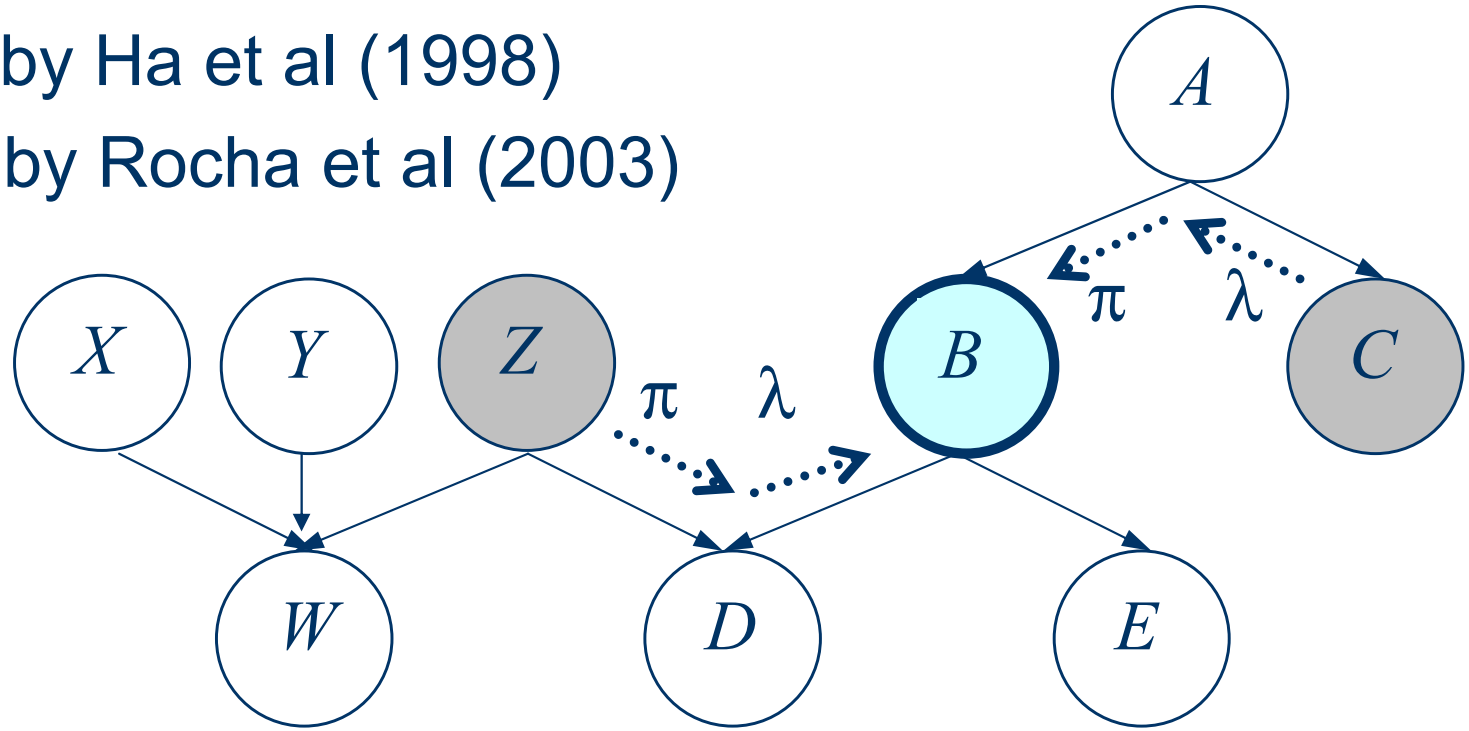
Difficulty: Too many potential vertices!!

- Example: variables with 3 values, 3 vertices per “local” set of probabilities: 3^{30} potential vertices!



Tessem's approximate inference

- every node is associated with a set of probabilities
- after every operation,
  transform to probability intervals
- extended by Ha et al (1998)
- Improved by Rocha et al (2003)



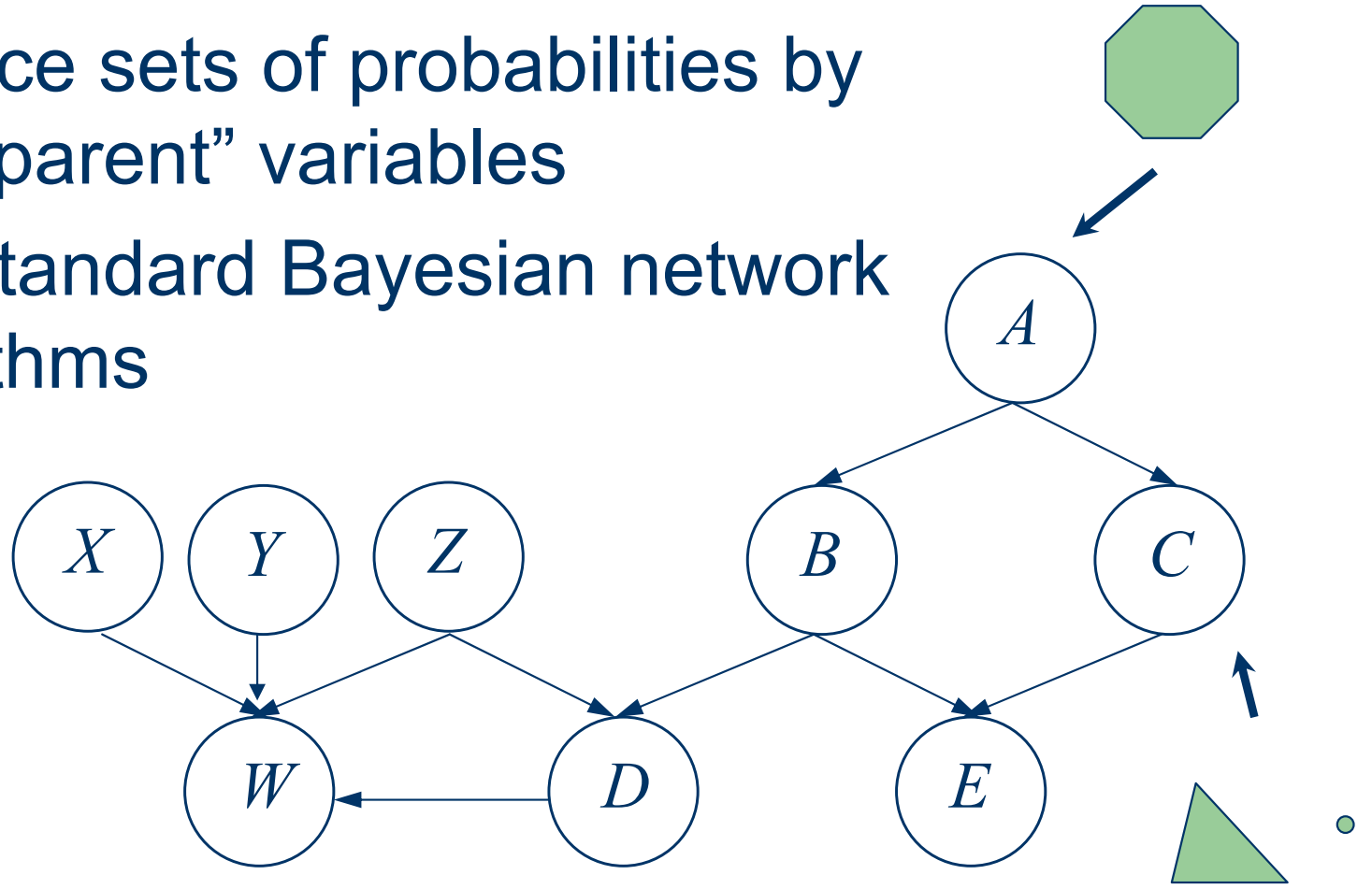
The 2U algorithm

- Fantastic observation:
  if we have a polytree and only binary variables,
  then we can quickly obtain lower probabilities!

- Running “essentially” Tessem’s algorithm, we actually get exact lower probabilities
  (Zaffalon 1997)

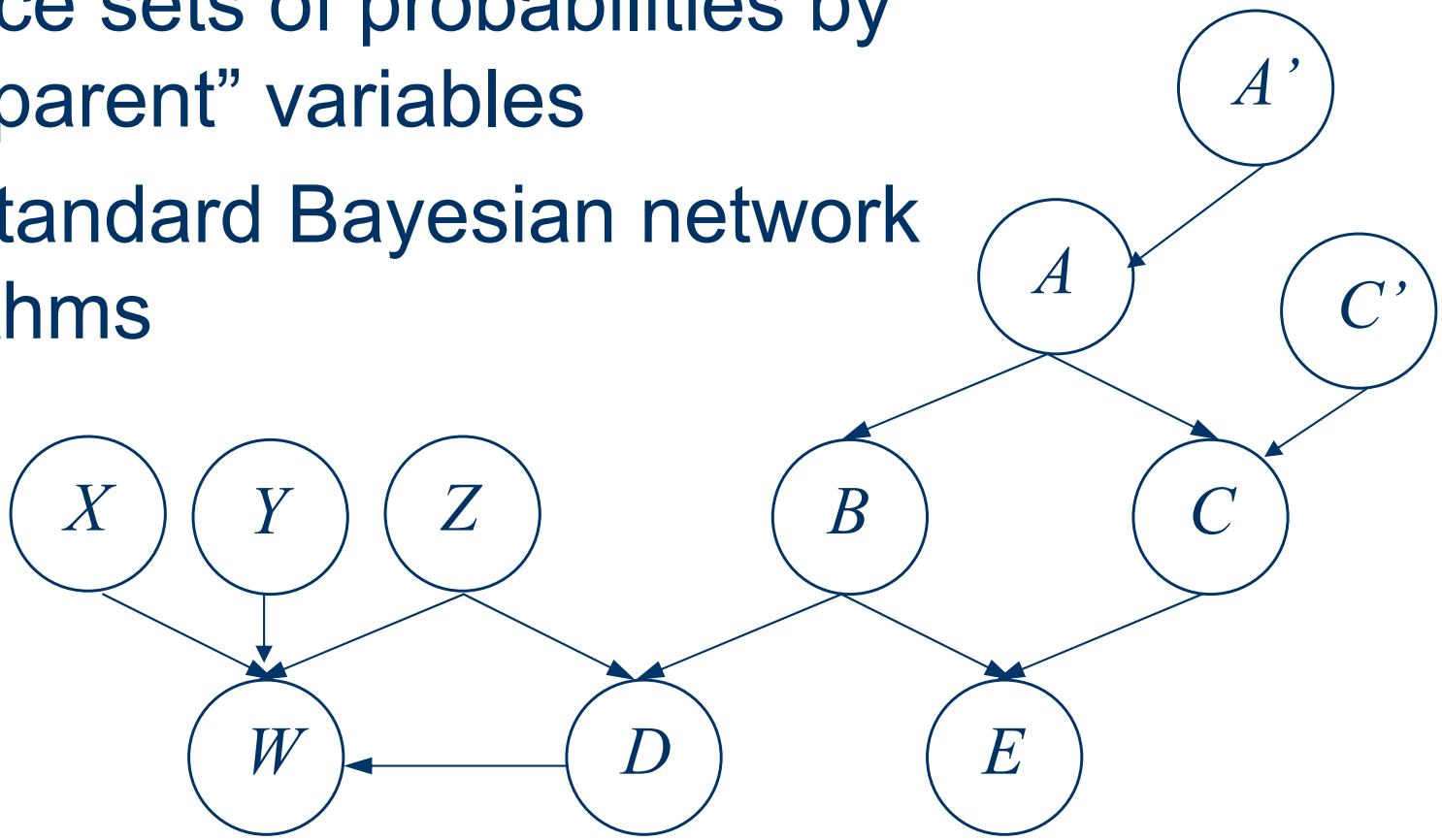
Strong independence:
Cano-Cano-Moral transformation

- Replace sets of probabilities by "transparent" variables
- Run standard Bayesian network algorithms



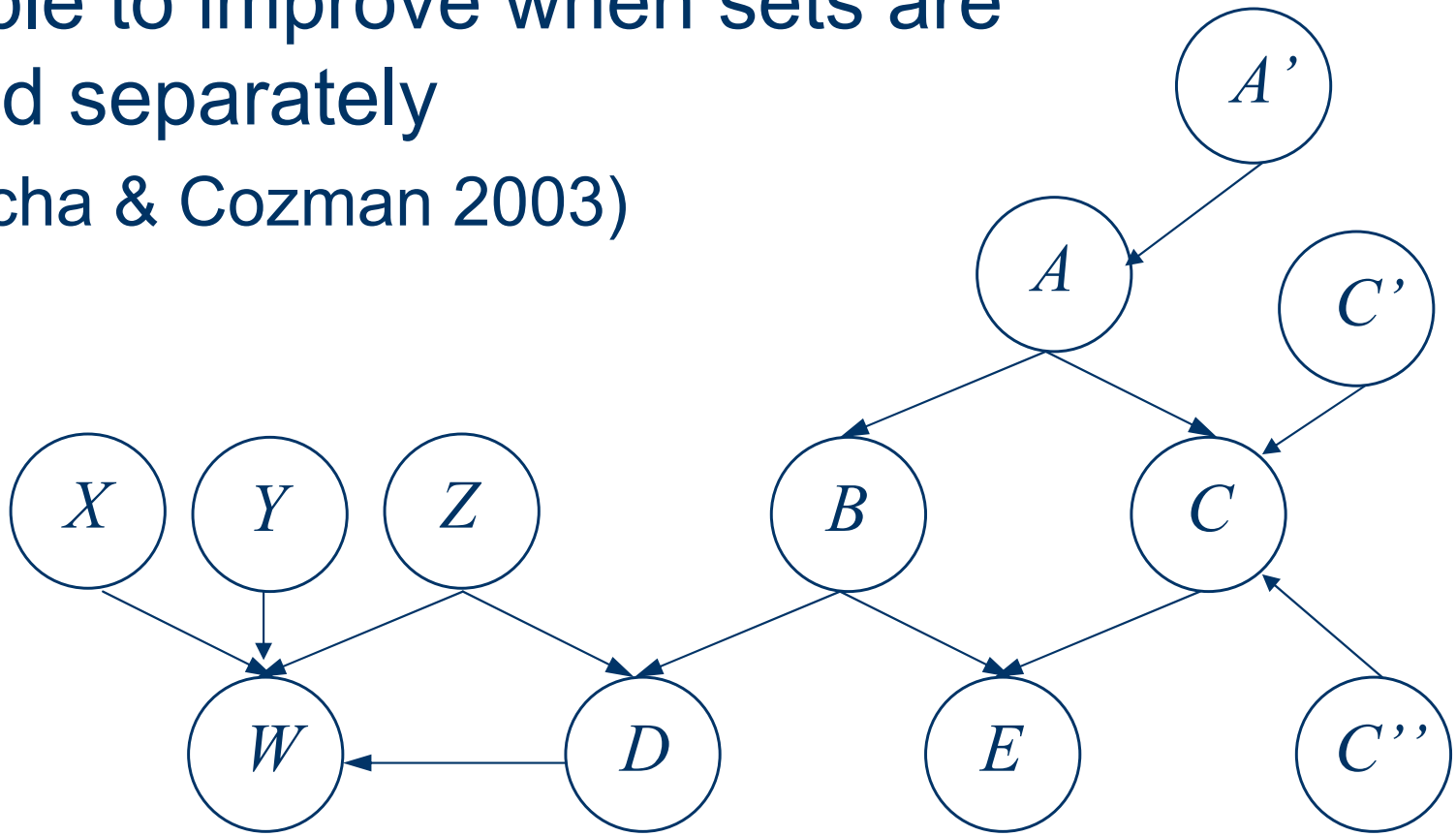
Strong independence:
Cano-Cano-Moral transformation

- Replace sets of probabilities by “transparent” variables
- Run standard Bayesian network algorithms



Strong independence:
Cano-Cano-Moral transformation

- Possible to improve when sets are defined separately
  - (Rocha & Cozman 2003)



Facing the optimization problem

- Andersen & Hooker (1994):
  - Coupling signomial programming with linear programming
- Cano-Cano-Moral (1994-):
  "local" search, leading to "inner" approximations
  - Importance sampling
  - Genetic algorithms
- Rocha & Cozman (2003):
  - Branch-and-bound (Tessem's and Cano-Moral's bounds)
  - Multilinear gradient search (Lukatskii-Shapot)

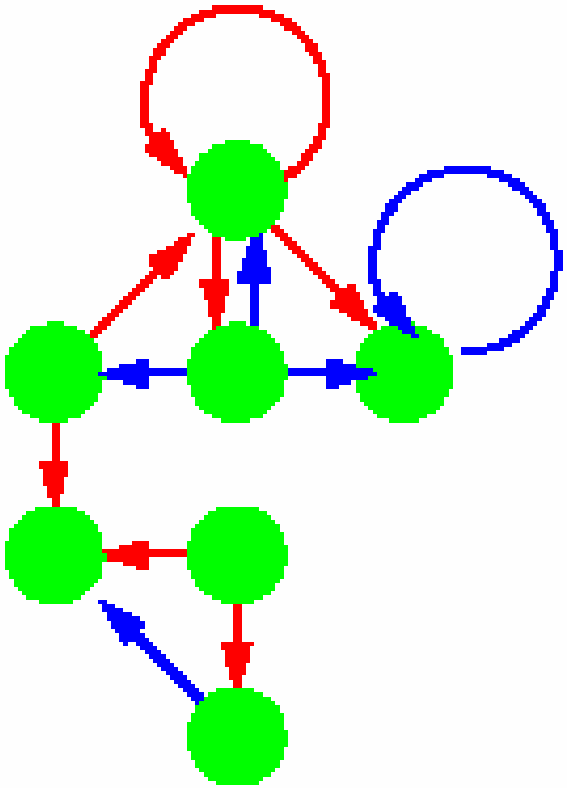
Learning

- Credal classification (Zaffalon):
  - Extending the Naïve Bayes classifier and tree-like classifiers
  - Using the Imprecise Dirichlet model
- Learning with missing data
  - Ramoni & Sebastiani's Bound and Collapse method
  - In general, missing data induces belief functions (Zaffalon)
- Pearl's causal inference and lack of identifiability

Imprecise Markov Decision Processes

- Obvious question:
  What happens when transition probabilities are imprecise?

- Not a new problem:
  - Satia & Lave (1970), White III & Eldeib (1984, 1986)
- Interesting twist in AI:
  "abstract" a very complex Markov Decision Process
  (Givan et al 2000)



Producing actions in Imprecise Markov Decision Processes

- Goal is object of debate:
  - Find *a policy* that is admissible?
  - Find *all* admissible policies?
  - Find a *min-max* policy? A *max-max* policy?
- There are versions of policy iteration and value iteration for imprecise Markov Decision Processes

Other models

- Chrisman (1997): undirected graphs
  - Exact propagation, subject to many conditions
- Thöne et al (1996), Lukasiewicz (1999-): interval constraints on probability of sentences
- Vantaggi (2001,2003): asymmetric graphoids
  - With proper consideration of zero probabilities

Conclusion: Hopefully this tutorial...

- Convinced that models based on graphs are
  - extensively used in practice
  - efficient, compact, flexible
  - theoretically interesting
- Reviewed existing efforts that deal with imprecision (sorry for omissions!!)

Conclusion: Ongoing research

- Fair amount of material on belief functions
- Intense effort to handle credal networks
  - Optimization is the *right* tool (?)
  - Still missing the “Gibbs sampler” for probability sets
  - Approximate inferences are (probably) the answer
- Effort on imprecise Markov Decision Processes

Conclusion: Existing material

- Significant amount of material in AI that uses graphs and imprecise probabilities:
  - Sensitivity analysis (Linda van der Gaag)
  - Default reasoning
  - Qualitative reasoning
  - Causal reasoning, identification problems
  - Learning with incomplete databases and incomplete information
  - Group decision making and consensus (multi-agents)
- Imprecise probabilities are not always recognized

Conclusion: Open topics

- Selection of independence concepts
- Properties of epistemic independence
- Markov random fields and chain graphs
- Learning (in particular in discrete models)
- Imprecise POMDPs, imprecise HMMs
- General-purpose software for such models
- **Practical applications...**

Conspicuously absent

- Possibility theory, fuzzy logic and related formalisms
- Discussion of zero probability and zero lower probability (Barbara Vantaggi's work)