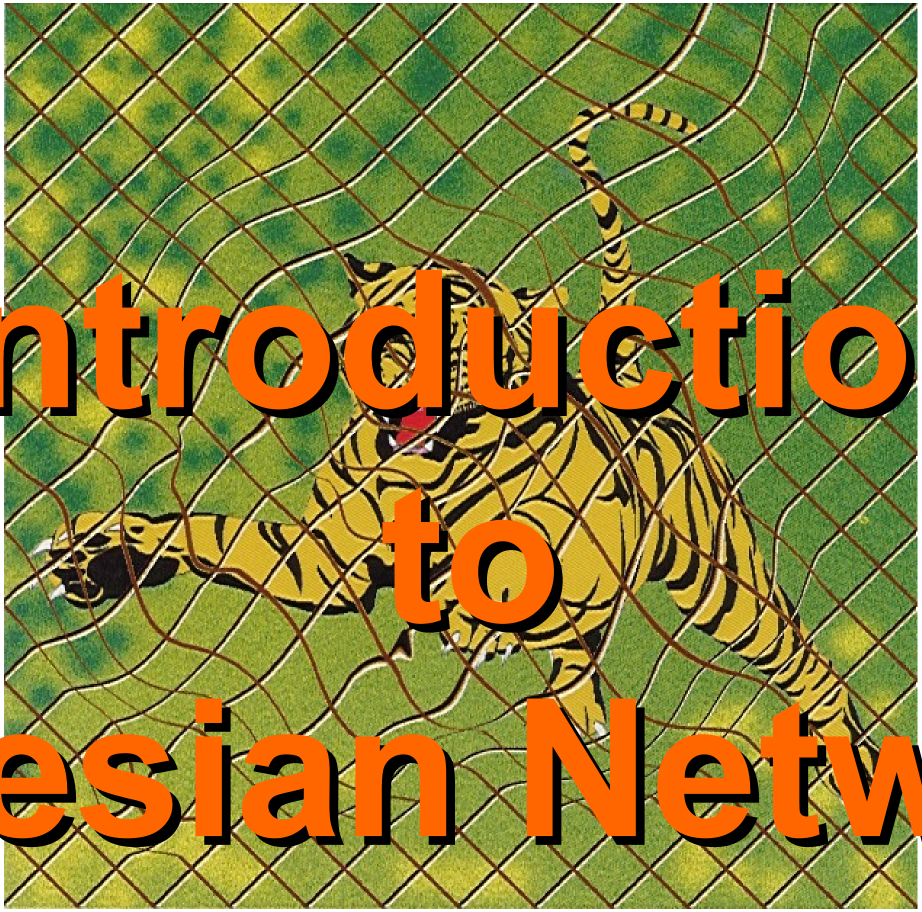


Introduction to Bayesian Networks

On learning and inference

- Assume n binary random variables $X_1,...,X_n$
- A joint probability distribution $P(X_1,...,X_n)$
- Inference:
  - compute the conditional probability distribution for the thing you want to know, given all that you know, marginalizing out all that you don’t know and don’t want to know
  - In pricinple exponential, requires $O(2^n)$ operations
  - Can be simplified if the joint distribution factorizes by indepecence: P(A,B)=P(A)P(B)
- Learning:
  - learn the model structure: what is (conditionally) independent of what
  - learn the parameters defining the ”local” distributions
- Supervised learning: construct directly a model for the required conditional distribution, without forming the joint distribution first

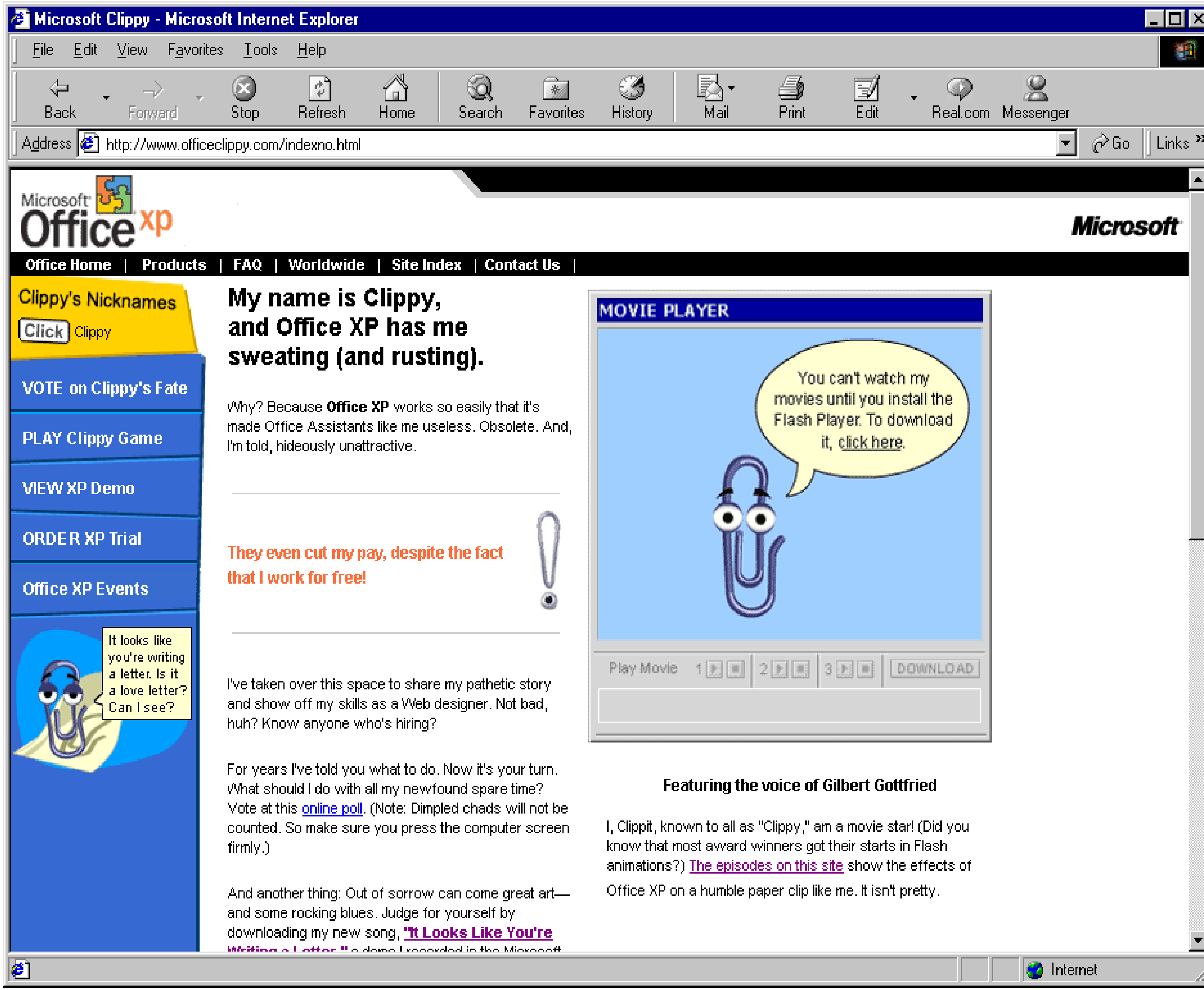
Probabilistic reasoning

- n (discrete) random variables $X_1,\ldots,X_n$
- joint probability distribution $P(X_1,\ldots,X_n)$
- Input: a partial value assignment Ω,
  $\Omega =< X_1, X_2=x_2, X_3, X_4=x_4, X_5=x_5, X_6,\ldots,X_n>$
- Probabilistic reasoning:
  - compute $P(X=x|\ \Omega)$ for all X not instantiated in Ω, and for all values of each X (the marginal distribution), OR:
  - find a MAP (maximum a posterior probability) assignment consistent with Ω
  - N.B. These are not the same thing!
- Bayesian networks: a family of probabilistic models and algorithms enabling computationally efficient probabilistic reasoning

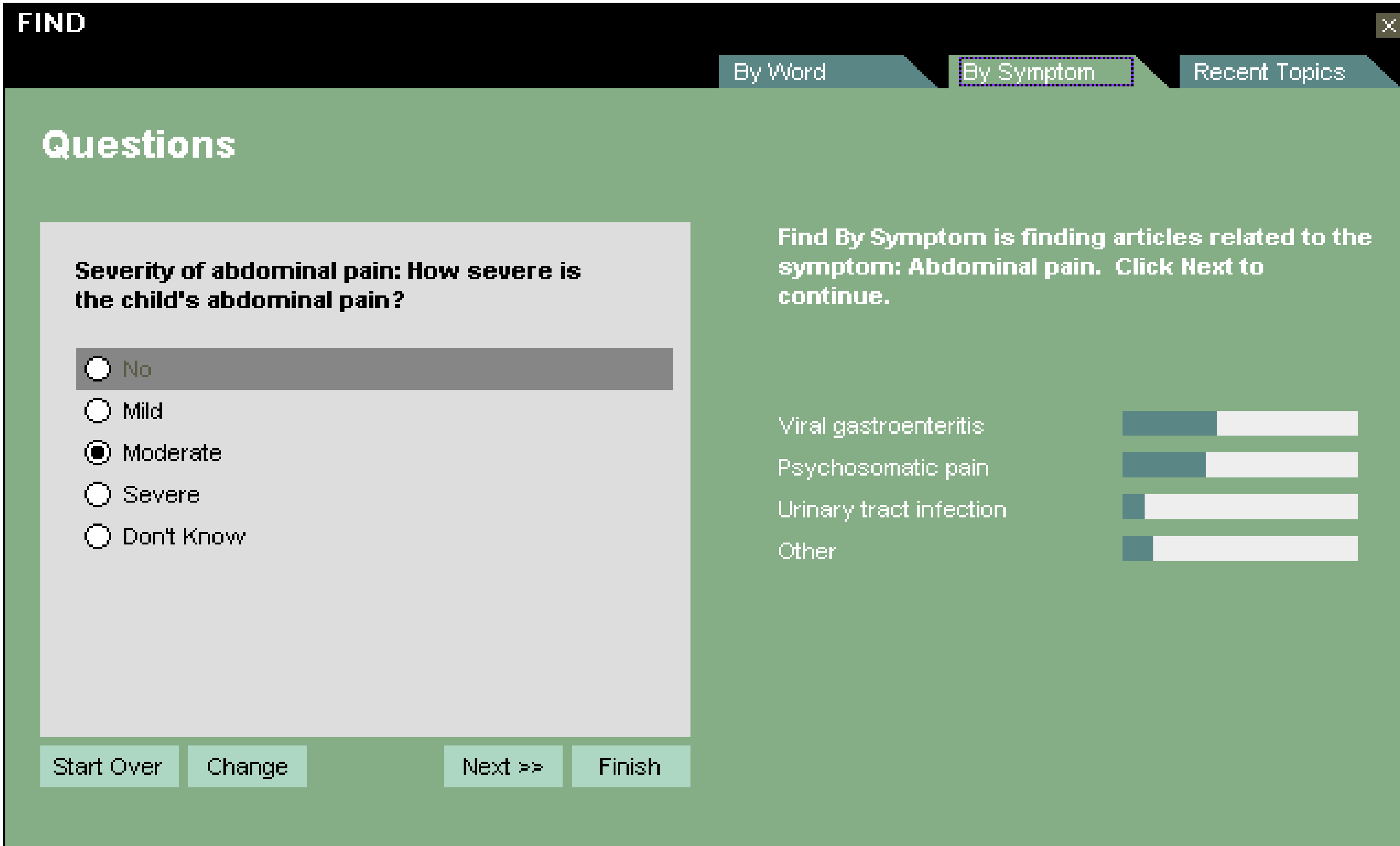
Bayesian networks: a ”Billion dollar” perspective

“Microsoft’s competitive advantage, he [Gates] responded, was its expertise in “Bayesian networks”. Ask any other software executive about anything “Bayesian” and you’re liable to get a blank stare. Is Gates onto something? Is this alien-sounding technology Microsoft’s new secret weapon?”

(Leslie Helms, Los Angeles Times, October 28, 1996.)


Microsoft Clippy - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit Real.com Messenger
Address http://www.officeclippy.com/indexno.html
Go Links
Microsoft Office xp
Microsoft
Office Home | Products | FAQ | Worldwide | Site Index | Contact Us |
Clippy's Nicknames
Click Clippy
VOTE on Clippy's Fate
PLAY Clippy Game
VIEW XP Demo
ORDER XP Trial
Office XP Events
It looks like you're writing a letter. Is it a love letter? Can I see?
My name is Clippy, and Office XP has me sweating (and rusting).
Why? Because Office XP works so easily that it's made Office Assistants like me useless. Obsolete. And, I'm told, hideously unattractive.
They even cut my pay, despite the fact that I work for free!
I've taken over this space to share my pathetic story and show off my skills as a Web designer. Not bad, huh? Know anyone who's hiring?
For years I've told you what to do. Now it's your turn. What should I do with all my newfound spare time? Vote at this online poll. (Note: Dimpled chads will not be counted. So make sure you press the computer screen firmly.)
And another thing: Out of sorrow can come great art—and some rocking blues. Judge for yourself by downloading my new song, "It Looks Like You're
MOVIE PLAYER
You can't watch my movies until you install the Flash Player. To download it, click here.
Play Movie 1 2 3 DOWNLOAD
Featuring the voice of Gilbert Gottfried
I, Clippit, known to all as "Clippy," am a movie star! (Did you know that most award winners got their starts in Flash animations?) The episodes on this site show the effects of Office XP on a humble paper clip like me. It isn't pretty.
Internet


Microsoft Pregnancy and Child Care
Home
Go To
Find
Options
Help
Microsoft Health Preview
Pregnancy and Child Care
Medical Advisory Board
What's New
Click here for this month's highlights in Microsoft Pregnancy and Child Care.
Library
To browse through illustrated articles on pregnancy, birth, and early child care, click here.
Find By Word
If you know what you're looking for, click here to search the Library by keywords.
Find By Symptom
Click here to find useful information in the Library related to children's symptoms.
Community Center
Have a story to share? Want to send us mail? Click here to access our community bulletin boards.

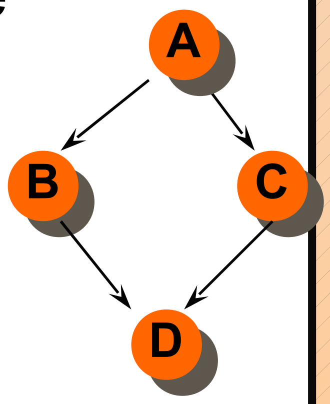

FIND
By Word
By Symptom
Recent Topics
Questions
Severity of abdominal pain: How severe is the child's abdominal pain?
No
Mild
Moderate
Severe
Don't Know
Start Over
Change
Next >>
Finish
Find By Symptom is finding articles related to the symptom: Abdominal pain. Click Next to continue.
Viral gastroenteritis
Psychosomatic pain
Urinary tract infection
Other

What do Bayesian networks have to offer?

- encoding of the covariation between “input” variables - BN can handle incomplete data sets
- allows one to learn about causal relationships (predictions in the presence of interventions)
- natural way of combining domain knowledge and data as a single model
- Computationally efficient inference algorithms for multi-dimensional domains

Bayesian networks: basics

- A Bayesian network is a model of probabilistic dependencies between the domain variables.
- The model can be described as a list of (in)dependencies, but is is usually more convenient to express them in a graphical form as a directed acyclic network.
- The nodes in the network correspond to the domain variables, and the arcs reveal the underlying dependencies, i.e., the hidden structure of the domain of your data.
- The ”quantitative strengths” of the dependencies are modeled as conditional probability distributions (not shown in the graph).



Bayesian networks?

- A very poor name, nothing Bayesian per se
- A parametric probabilistic model that
  - can be used for Bayesian inference (or not)
  - can be learned via Bayesian methods (or not)
  - is conveniently represented as a graph (a probabilistic graphical model)
- A better name: directed acyclic graph (DAG)
- (Even better: acyclic directed graph)

The two-variable case

- Assume two binary (Bernoulli distributed) variables A and B
- Two examples of the joint distribution P(A,B):

	B=1	B=0	P(A)
A=1	0.08	0.02	0.10
A=0	0.72	0.18	0.90
P(B)	0.80	0.20	

$$P(A,B)=P(A)P(B)$$

We only need the marginals P(A) and P(B)!

	B=1	B=0	P(A)
A=1	0.08	0.02	0.10
A=0	0.18	0.72	0.90
P(B)	0.26	0.74	

$$P(A,B) \neq P(A)P(B)$$

We need the full table (or: $P(A,B)=P(A)P(B|A)$)

Independence

- If P(A,B)=P(A)P(B), A and B are said to be **independent**
- Note that this also means that P(A | B) = P(A) (and: P(B | A) = P(B))
- If A and B are not independent, they are dependent
- Independe can be used to separate from all joint distributions P(A,B) the subset where the independence holds
- Independence simplifies (constrains) things:
  - $A \perp B$: *subset of distributions*
  - *not* $A \perp B$: *all distributions*
-

Types of independence

- if P(A=a,B=a) = P(A=a)P(B=b) for all a and b, then we call A and B (marginally) independent.
- if P(A=a,B=a | C=c) = P(A=a|C=c)P(B=b|C=c) for all a and b, then we call A and B conditionally independent given C=c.
- if P(A=a,B=a | C=c) = P(A=a|C=c)P(B=b|C=c) for all a, b and c, then we call A and B conditionally independent given C.
- $P(A,B)=P(A)P(B)$ implies

$$P(A|B)=\frac{P(A,B)}{P(B)}=\frac{P(A)P(B)}{P(B)}=P(A)$$

Examples

- Amount of Speeding fine $\perp$ Type of car | Speed
  - But: Amount of Speeding fine $\not\perp$ Type of car
- Lung cancer $\perp$ Yellow teeth | Smoking
  - But: Lung cancer $\not\perp$ Yellow teeth
- Child's genes $\perp$ Grandparent's genes | Parents' genes
  - But: Child's genes $\not\perp$ Grandparent's genes
- Ability of Team A $\perp$ Ability of Team B
  - But: Ability of Team A $\not\perp$ Ability of Team B | Outcome of A vs. B game

Independence saves space

- If A and B are independent given C:

  P(A,B,C) = P(C,A,B)

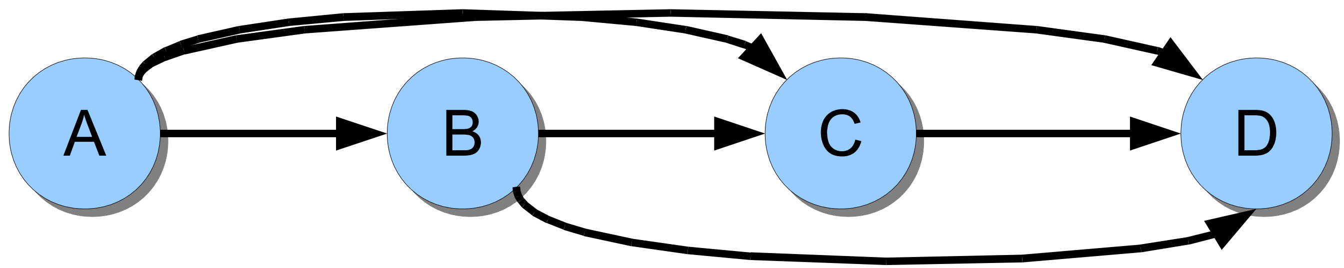
  = P(C)P(A|C)P(B|A,C)

  = P(C)P(A|C)P(B|C)

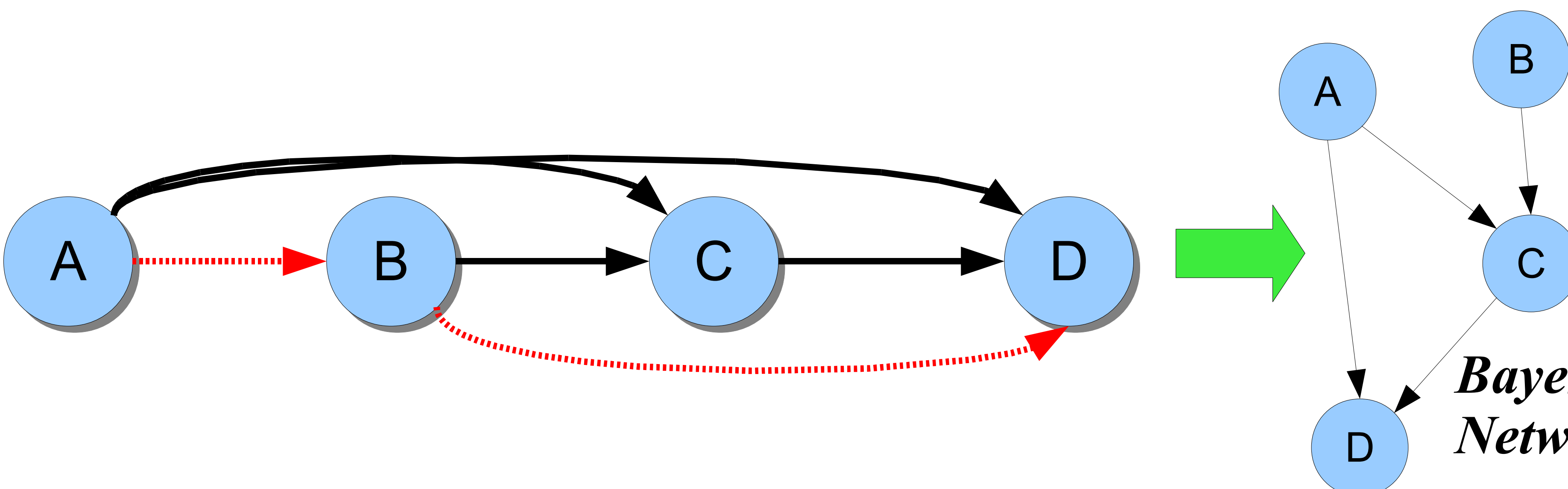
- Instead of having a full joint probability table for P(A,B,C), we can have a table for P(C) and tables P(A|C=c) and P(B|C=c) for each c.
  - Even for binary variables this saves space:
    - $2^3 = 8$ vs. 2 + 2 + 2 = 6.
  - With many variables and many independences you save **a lot**.

Chain Rule – Independence - BN

$$Chain\,rule: P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)$$



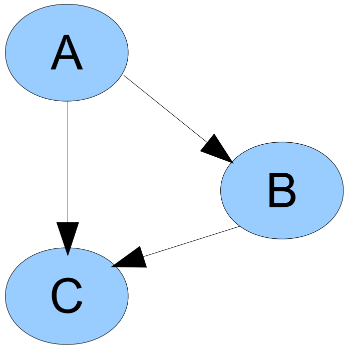
$$Independence: P(A,B,C,D) = P(A)P(B)P(C|A,B)P(D|A,C)$$

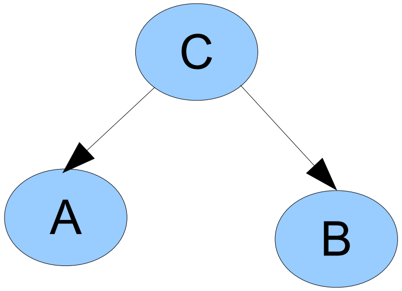


Bayesian Network

But order can matter

- P(A,B,C) = P(C,A,B)
  - P(A)P(B|A)P(C|A,B) = P(C)P(A|C)P(B|A,C)
  - And if A and B are conditionally independent given C:
    1. P(A,B,C) = P(A)P(B|A)P(C|A,B)
    2. P(C,A,B) = P(C)P(A|C)P(B|C)

1. 

2. 

Bayes net as a factorization

- Bayesian network structure forms a directed acyclic graph (DAG).
- If we have a DAG G, we denote the parents of the node (variable) X_i with $Pa_G(x_i)$ and a value configuration of $Pa_G(x_i)$ with $pa_G(x_i)$:

$$P(x_{1,} x_{2,} \ldots, x_n | G) = \prod_{i=1}^{n} P(x_i | pa_G(x_i)),$$

where $P(x_i|pa_G(x_i))$ are called local probabilities.

  - Local probabilities are stored in the conditional probability tables (CPTs).

A Bayesian network

P(Cloudy)

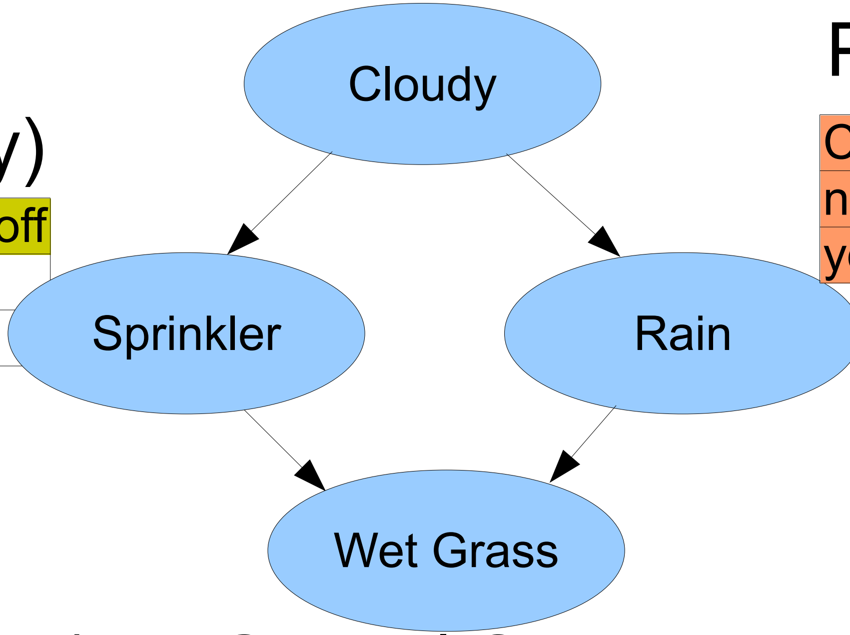
	Cloudy=no	Cloudy=yes
	0.5	0.5

P(Rain | Cloudy)

Cloudy	Rain=yes	Rain=no
no	0.2	0.8
yes	0.8	0.2

P(Sprinkler | Cloudy)

Cloudy	Sprinkler=on	Sprinkler=off
no	0.5	0.5
yes	0.9	0.1



P(WetGrass | Sprinkler, Rain)

Sprinkler	Rain	WetGrass=yes	WetGrass=no
on	no	0.90	0.10
on	yes	0.99	0.01
off	no	0.01	0.99
off	yes	0.90	0.10

Causal order recommended

- Causes first, then effects.
- Since causes render direct consequences independent yielding smaller CPTs
- Causal CPTs are easier to assess by human experts
- Smaller CPT:s are easier to estimate reliably from a finite set of observations (data)
- Causal networks can be used to make causal inferences too.

Inference in Bayesian networks

- Given a Bayesian network B (i.e., DAG and CPTs) , calculate P(**X**|**e**) where **X** is a set of query variables and **e** is an instantiaton of observed variables **E** (**X** and **E** separate).
- There is always the way through marginals:
  - normalize $P(\mathbf{x},\mathbf{e}) = \Sigma_{\mathbf{y} \in dom(\mathbf{Y})} P(\mathbf{x},\mathbf{y},\mathbf{e})$, where dom(**Y**), is a set of all possible instantiations of the unobserved non-query variables **Y**.
- There are much smarter algorithms too, but in general the problem is NP hard (more later).

Back to the two-variable case...

Model M1:

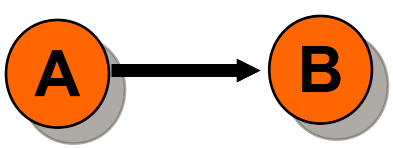
A and B independent

P(A,B) = P(A)P(B)

Model M2:

A and B dependent

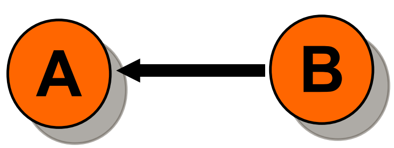
P(A,B) = P(A)P(B|A)



Model M3:

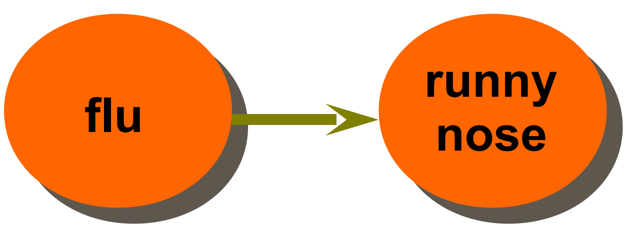
A and B dependent

P(A,B) = P(B)P(A|B)

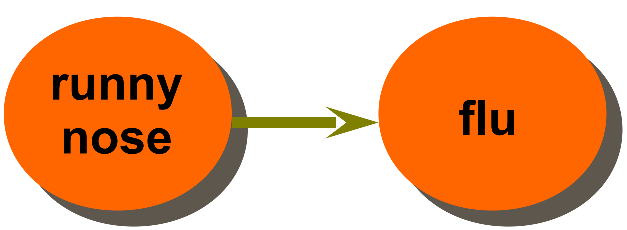


Equivalence classes

- Equivalence class = set of BN structures which can used for representing exactly the same set of probability distributions.
- The ”causally natural” version makes it easier to determine the conditional probabilities.



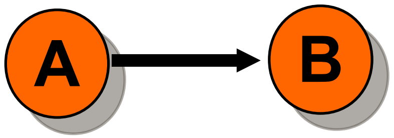
P(flu, ns) = P(flu)P(rn | flu)



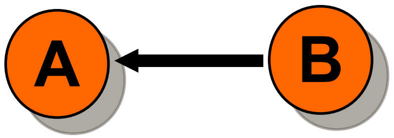
P(flu, rn) = P(rn)P(flu| rn)

The Bayes rule visualized

- $P_1(A,B)=P_1(A)P_1(B \mid A)$



- $P_2(A,B)=P_2(B)P_2(A \mid B)$



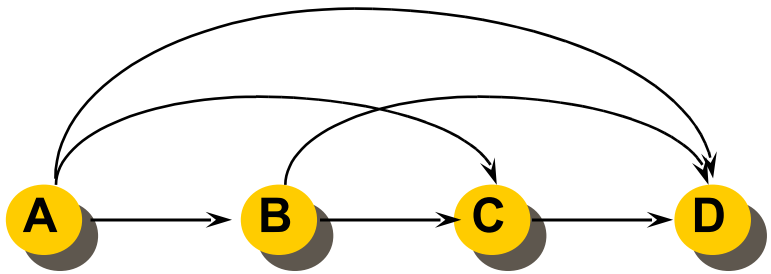
- Assume $P_1(A)$ and $P_1(B \mid A)$ fixed
- $P_2(A,B)=P_1(A,B)$ if:

$$P_2(A \mid B) = P_1(A)P_1(B \mid A)/P_2(B)$$

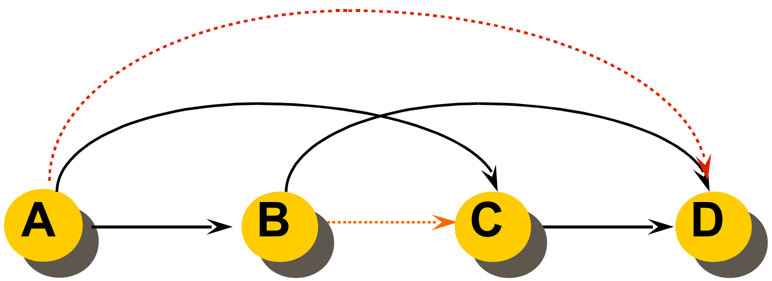
Another example

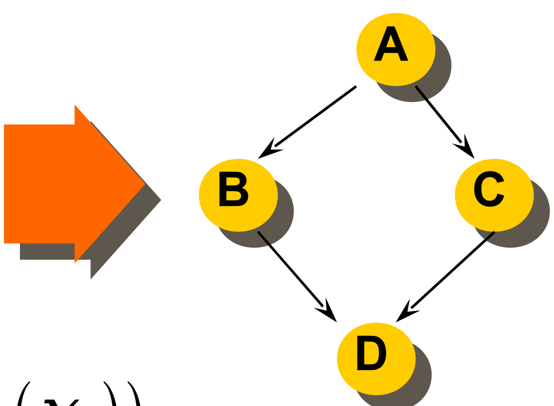
- From Bayes' rule, it follows that

P(A,B,C,D)=P(A)P(B|A)P(C|A,B)P(D|A,B,C)



Assume: P(C|A,B)=P(C|A) and P(D|A,B,C)=P(D|B,C)

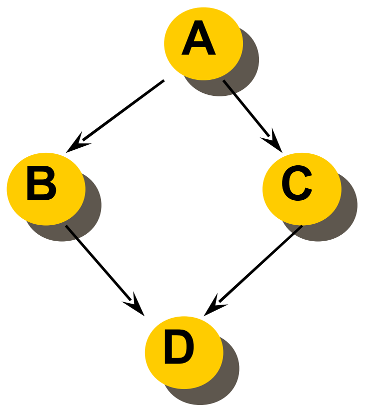




$$P(x_1, x_2, \ldots, x_n | G) = \prod_{i=1}^{n} P(x_i | pa_G(x_i))$$

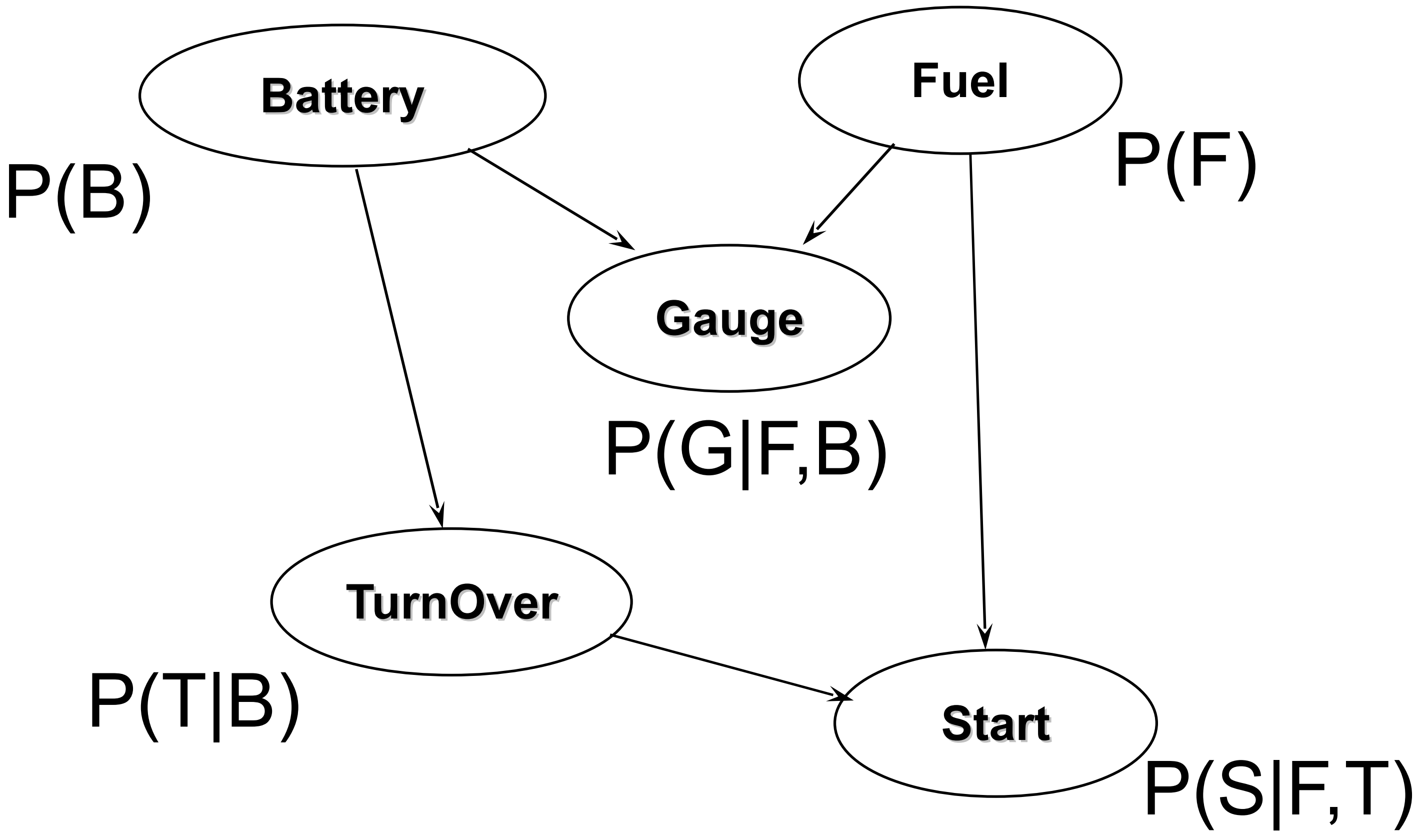
And the point is…?

- simple conditional probabilities are easier to determine than the full joint probabilities
- in many domains, the underlying structure corresponds to relatively sparse networks, so only a small number of conditional probabilities is needed



P(+a,+b,+c,+d)=P(+a)P(+b|+a)P(+c|+a)P(+d|+b,+c)
P(–a,+b,+c,+d)=P(–a)P(+b|–a)P(+c|–a)P(+d|+b,+c)
P(–a,–b,+c,+d)=P(–a)P(–b|–a)P(+c|–a)P(+d|–b,+c)
P(–a,–b,–c,+d)=P(–a)P(–b|–a)P(–c|–a)P(+d|–b,–c)
P(–a,–b,–c,–d)=P(–a)P(–b|–a)P(–c|–a)P(–d|–b,–c)
P(+a,–b,–c,–d)=P(+a)P(–b|+a)P(–c|+a)P(–d|–b,–c)
. . .

A Bayesian Network



Building a Bayesian Network

P(T=none) = 0.003
P(T=click)= 0.001
P(T=normal)= 0.996

P(S=yes|T=none) = 0.0
P(S=no|T=none) = 1.0

P(S=yes|T=click) = 0.02
P(S=no|T= click) = 0.98

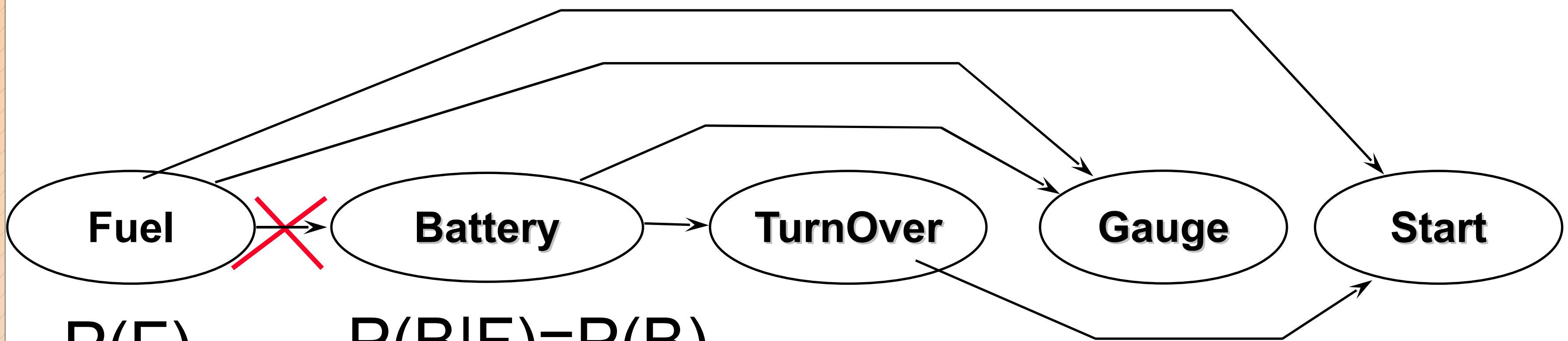
P(S=yes|T=normal) = 0.97
P(S=no|T=normal) = 0.03

Missing Arcs Encode Conditional Independence

p(T=none) = 0.003
p(T=click)= 0.001
p(T=normal)= 0.996

p(G=not empty) = 0.995
p(G=empty) = 0.005

A Modular Encoding of a Joint Distribution



P(F) P(B|F)=P(B)

P(T|B,F)=P(T|B)

P(G|F,B,T)=P(G|F,B)

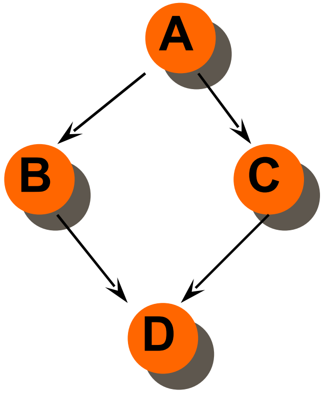
P(S|F,B,T,G)=P(S|F,T)

P(F,B,T,G,S)
= P(F) P(B|F) P(T|B,F) P(G|F,B,T) P(S|F,B,T,G)
= P(F) P(B) P(T|B) P(G|F,B) P(S|F,T)

Bayesian networks: the textbook definition

- A Bayesian (belief) network representation for a probability distribution *P* on a domain $(X_1,\dots,X_n)$ is a pair (G,Θ), where *G* is a directed acyclic graph whose nodes correspond to the variables $X_1,\dots,X_n$, and whose topology satisfies the following: each variable X is conditionally independent of all of its non-descendants in *G*, given its set of parents pa_X, and no proper subset of pa_X satisfies this condition. The second component Θ is a set consisting of all the conditional probabilities of the form $P(X|pa_X)$.

G:



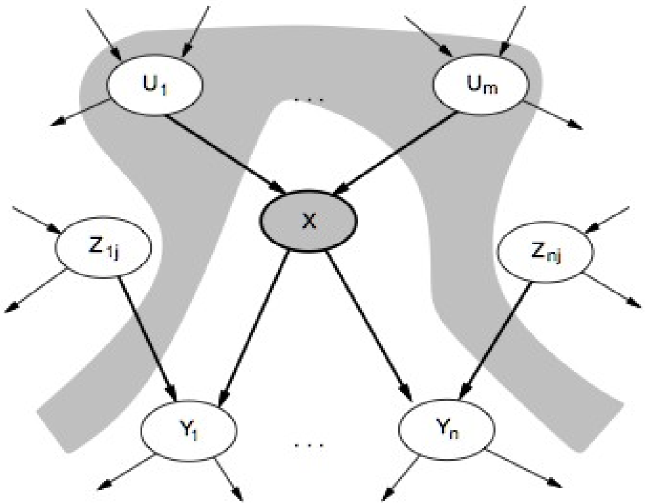
Θ = {P(+a), P(+b|+a), P(+b|-a), P(+c|+a), P(+c|-a), P(+d|+b,+c), P(+d|-b,+c), P(+d|+b,-c), P(+d|-b,-c)}

Markov conditions

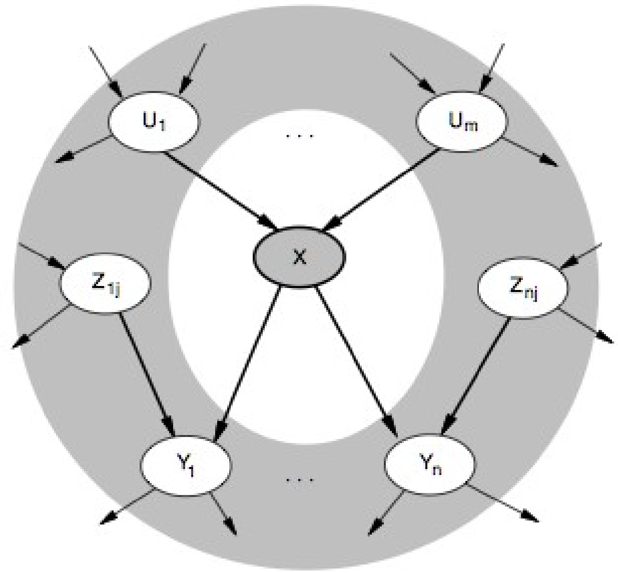
- Local (parental) Markov condition
  - X is independent of its non-descendants given its parents.
- Another local Markov condition
  - X is independent of any set of other variables given its parents, children and parents of its children (= Markov blanket)
- Global Markov Condition
  - X and Y are dependent given Z, iff they are d-separated by Z

Local Markov conditions visualized

- From Russell & Norvig's book:



”X is conditionally independent of its non-descendants, given its parents”



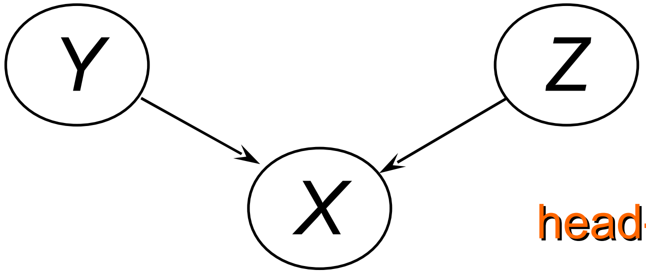
”X is conditionally independent of all the other variables, given its Markov blanket”

d-Separation (Pearl 1987)

- Theorem (Verma): $\boldsymbol{X}$ and $\boldsymbol{Y}$ are d-separated by Z implies $\boldsymbol{X} \perp \boldsymbol{Y} \mid Z$.
- Theorem (Geiger and Pearl): If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are not d-separated by Z, then there exists an assignment of the probabilities to the BN such that $(\boldsymbol{X} \perp \boldsymbol{Y} \mid Z)$ does not hold.

d-Separation

- A *trail* in a BN is a a cycle-free sequence (path) of edges in the corresponding undirected graph (the skeleton)
- A node x is a head-to-head node (a ”v-node”) along a trail if there are two consecutive arcs $Y \rightarrow X$ and $X \leftarrow Z$ on that trail:



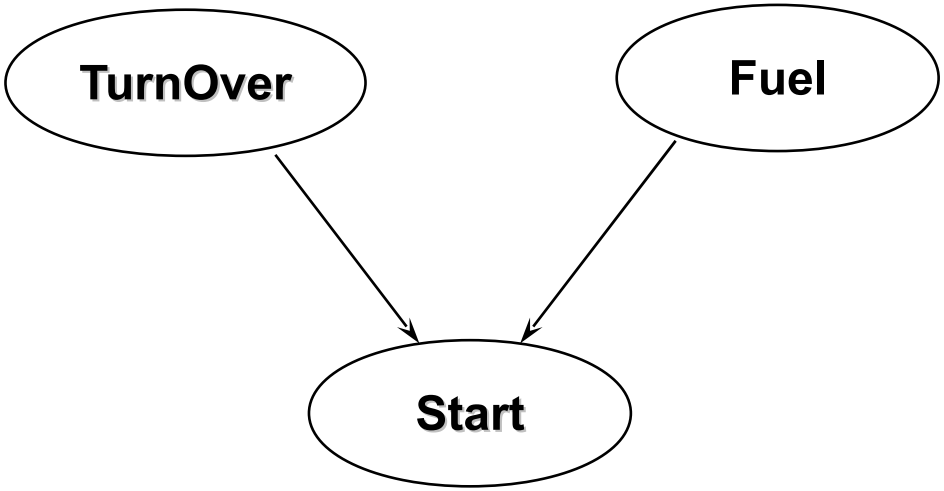
head-to-head node

d-Separation

- Nodes X and Y are d-connected by nodes Z along a trail from X to Y if
  - every head-to-head node along the trail is in Z or has a descendant in Z
  - every other node along the trail is not in Z

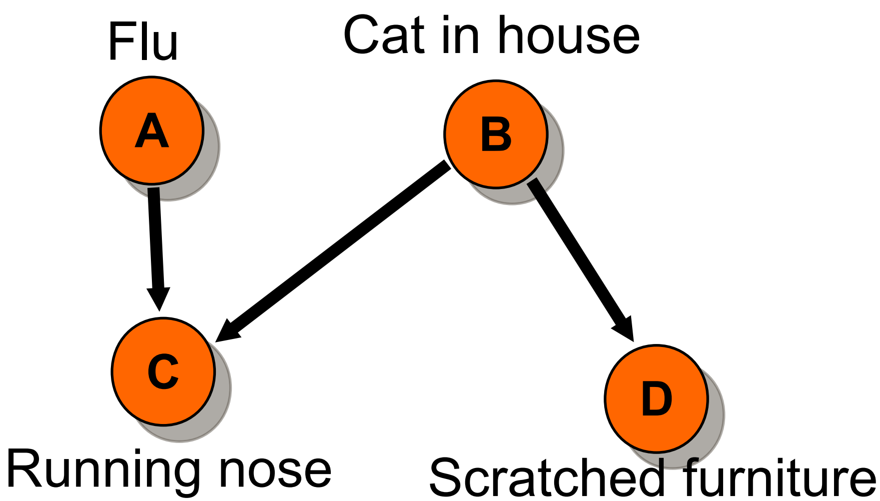
Nodes ***X*** and ***Y*** are d-separated by nodes Z if they are not d-connected by Z along any trail from ***X*** to ***Y***

Explaining Away (selection bias, Berkson's paradox)



If the car doesn't start, hearing the engine turn over makes no fuel more likely.

Explaining away: another example



P(A=1)=0.05
P(B=1)=0.05
P(C=1|A=0,B=0)=0.001
P(C=1|A=1,B=0)=0.95
P(C=1|A=0,B=1)=0.95
P(C=1|A=1,B=1)=0.99
P(D=1|B=1)=0.99
P(D=1|B=0)=0.1

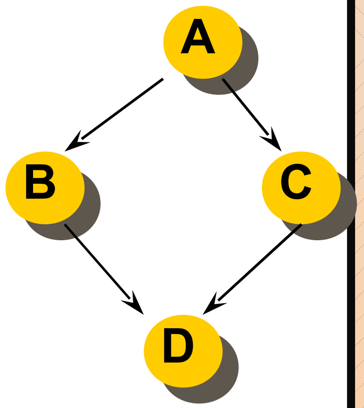
- Given C=1**,** the probability of A=1 is about 51%, and the probability of B=1 is also about 51%
- Given C=1 **and** D=1, the probability of A=1 goes down to 13% while the probability of B=1 goes up to 91%
- Details: see pages 53-56 of the report *Bayes-verkkojen mahdollisuudet*

Types of connections

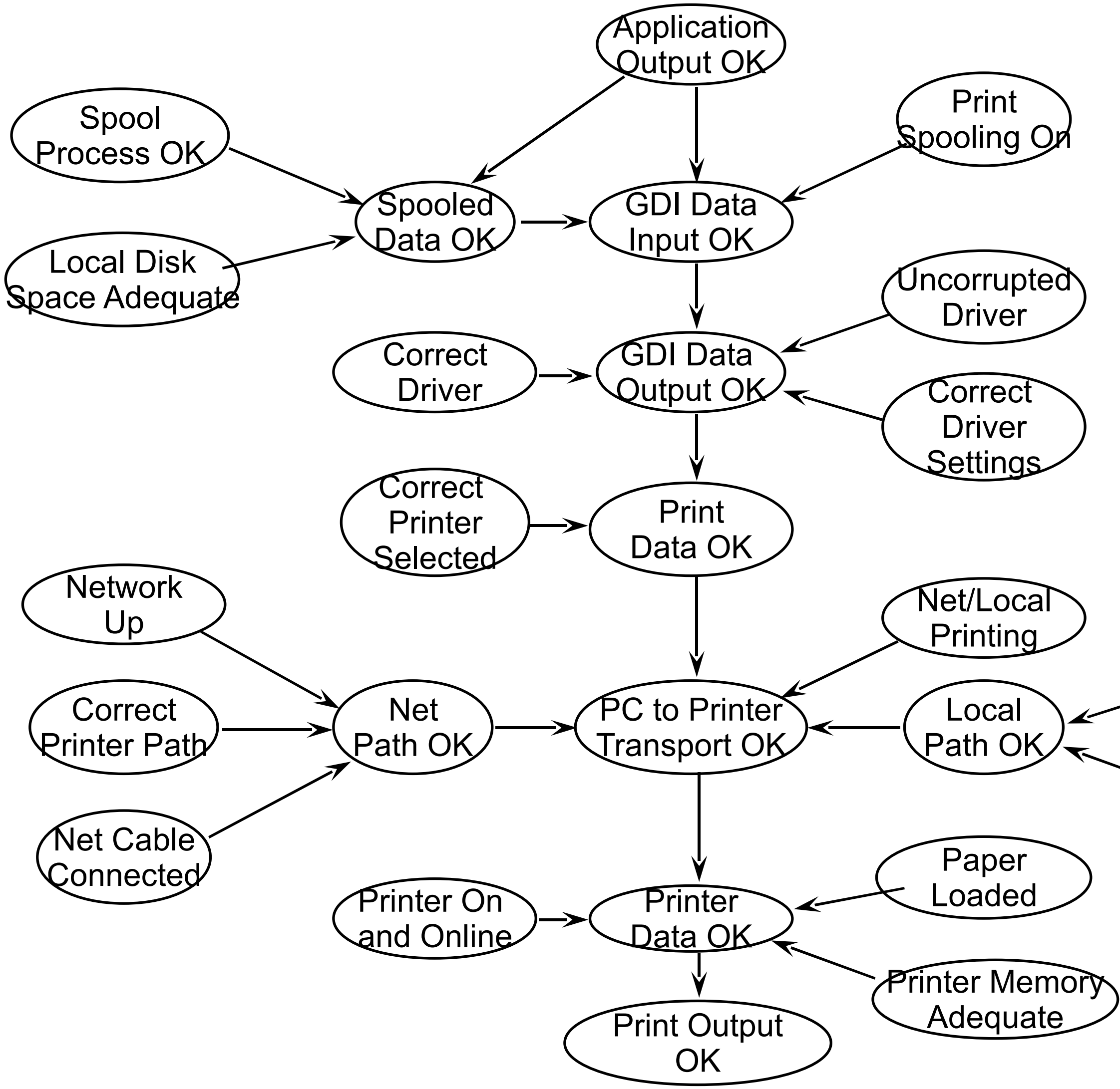
- There can be three types of connections on a trail:
  - Serial: X→Z→Y
    - Blocked at Z if Z known
  - Diverging: X←Z→Y
    - Blocked at Z if Z known
  - Converging (head-to-head): X→Z←Y
    - Blocked at Z UNLESS Z or any of its descendants known

Reading out the dependencies

- The Bayesian network on the right represents the following list of dependencies:
  - A and B are dependent on each other no matter what we know and what we don't know about C or D (or both).
  - A and C are dependent on each other no matter what we know and what we don't know about B or D (or both).
  - B and D are dependent on each other no matter what we know and what we don't know about A or C (or both).
  - C and D are dependent on each other no matter what we know and what we don't know about A or B (or both).
  - A and D are dependent on each other if we do not know both B and C.
  - B and C are dependent on each other if we know D or if we do not know D and also do not know A.



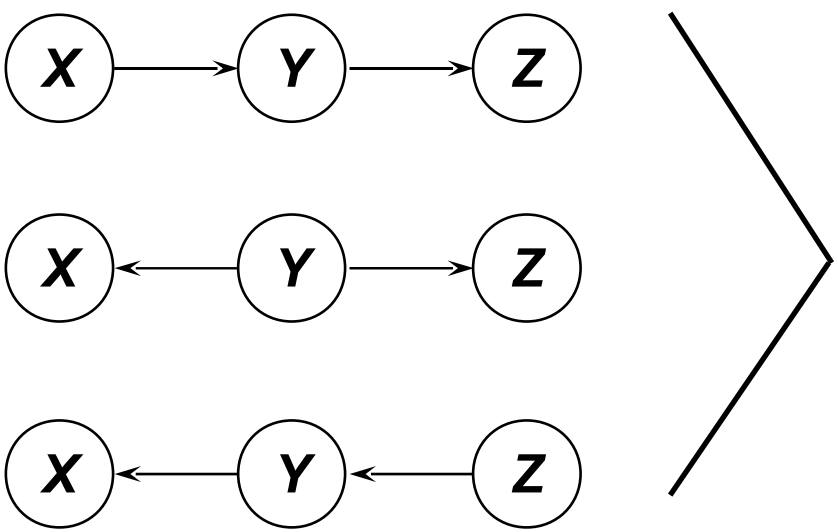
Printer Troubleshooter (W '95)



Equivalent Network Structures

Two network structures for domain X are independence equivalent if they encode the same set of conditional independence statements

Example:



$$X \perp Z \mid Y$$

Equivalent network structures

- Verma (1990): Two network structures are independence equivalent if and only if:
  - They have the same skeleton
  - They have the same v-structures