

---

Advances in Bayesian Network  
Learning, Causal Discovery, and  
Variable Selection in Massive Datasets  
with Applications in Biomedicine

*Ioannis Tsamardinos*  
Discovery Systems Laboratory  
Department of Biomedical Informatics, Vanderbilt University

# Important Tasks in Biomedicine

---

## ■ ***Prevention***

- Knowing that “Smoking causes lung cancer” may convince people to stop smoking

## ■ ***Diagnosis***

- Knowing that “people with cancer often have yellow-stained fingers and feel fatigue”, diagnose lung cancer

## ■ ***Treatment***

- Knowing that “the presence of protein *X* causes cancer, inactivate protein *X*, using medicine *Y* that causes *X* to be inactive”

# Learning from Data in Biomedicine

---

- Types of Statistical Data
  - Experimental (manipulate a variable, observe results)
  - Observational
  - Mixed
- Problem Definition
- Given data and a target variable  $T$ 
  - Build a high quality predictive/diagnostic/classification model for  $T$
  - Build a highly probable causal model for  $T$

# Example Problems: Classification

---

- Predict cancer type given gene expression data
- Predict the gene expression level of a given gene given gene expression data
- Predict protein concentration level given mass-spectroscopy data
- Predict biochemical properties of drugs given structural properties
- Predict length of stay of patients given clinical data
- Determine patient diagnosis given clinical data
- Determine content and quality of medical journal papers

# Example Problems: Causal Discovery

---

- Discover the genes that (directly) cause cancer
- Discover the genes that (directly) affect (cause) a gene's expression level to change
- Discover the structural properties of a drug that directly cause it to exhibit certain biochemical properties

# Example

- Variables: gene expression levels for 15,000 genes measured in cancerous or normal tissue of subjects
- Target  $T$ : type of cancer
- Learn to classify/diagnose  $T$  of future patients
- Find which genes directly cause  $T$ .
- Find the minimal set of genes convey the most information for  $T$ .

Variables

Training  
Instances

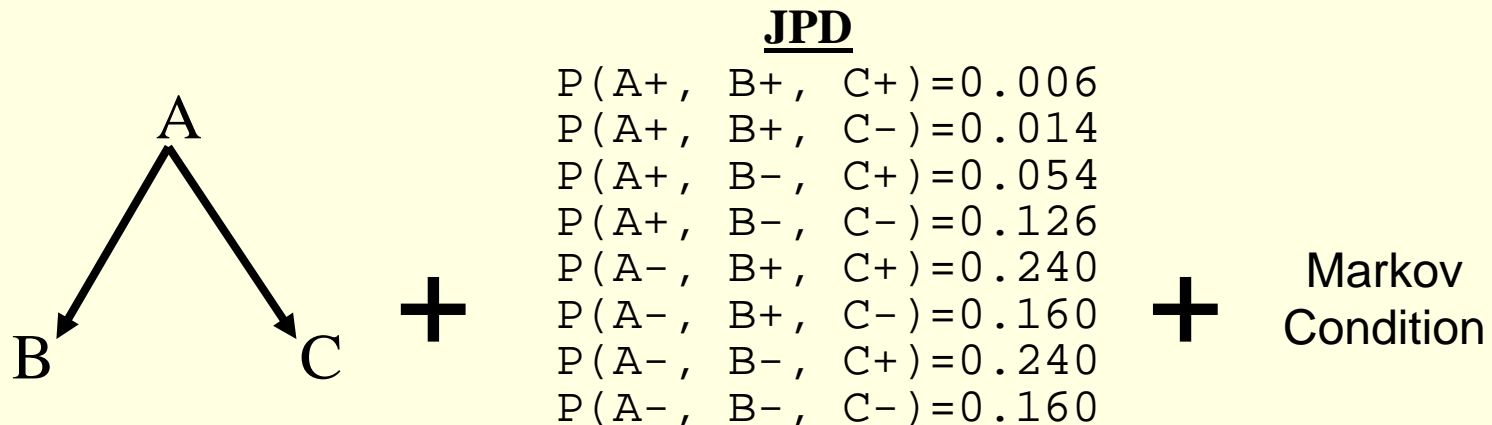
Cancer Type	Exp. Level G1	Exp. Level G2	...	Exp. Level G15000
A	0.03	5.4	...	-0.2
A	0.23	1.2	...	1.2
S	4.3	-0.4	...	3.2
...	...	...	...	...

# Data in Biomedicine: The Challenge

- Availability of extremely high dimensional data:
  - Because of mass throughput techniques
    - Gene expression microarray data: (range 10K-15K variables)
    - Mass-spectroscopy(60K-65K)
    - Chemical structural properties (140K)
    - Text-categorization (10K-20K)
  - Because of time-series measurements or representational issues
    - Variable A at times 1, 2, ..n, becomes  $A_1, \dots, A_n$
    - Variable A with values {disease<sub>1</sub>, ..., disease<sub>n</sub>} becomes binary variables  $A_{disease1}, \dots, A_{disease n}$
- Small number of training instances
- Causal Discovery with very high dimensional data

# Bayesian Networks: Definition

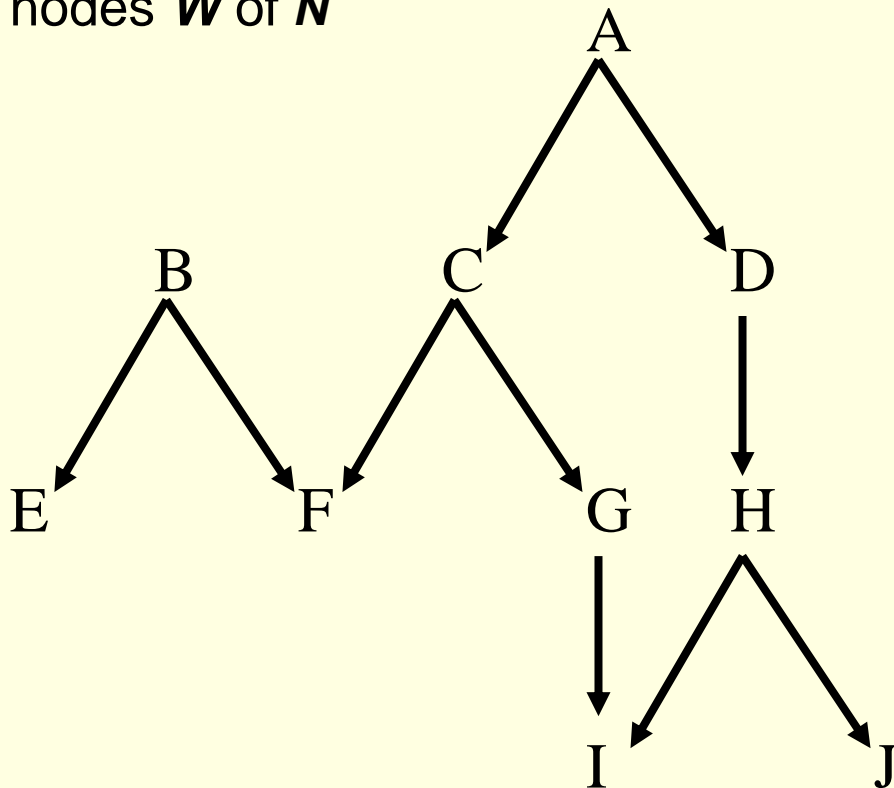
- Consider a set of variables  $V$  and their joint probability distribution JPD
- BN=Graph + Joint Probability Distribution connected by the Markov Property
- Graph has to be DAG (directed acyclic) in the standard BN model



- Any JPD can be represented in BN form

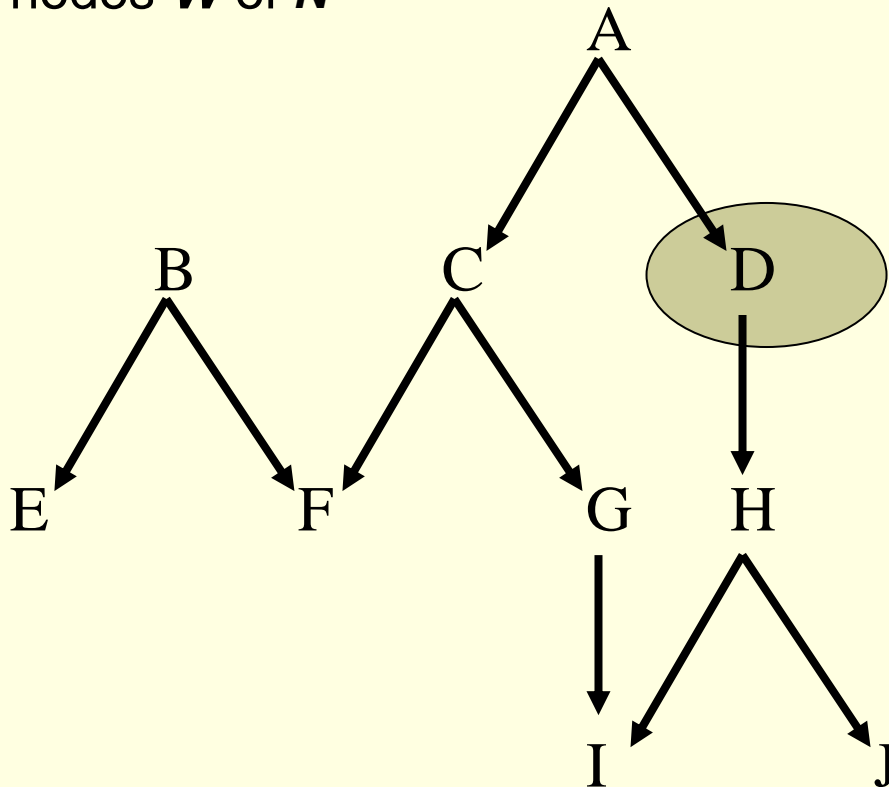
# Bayesian Networks

- Markov Property: the probability distribution of any node  $N$  given its parents  $P$  is independent of any subset of the non-descendent nodes  $W$  of  $N$



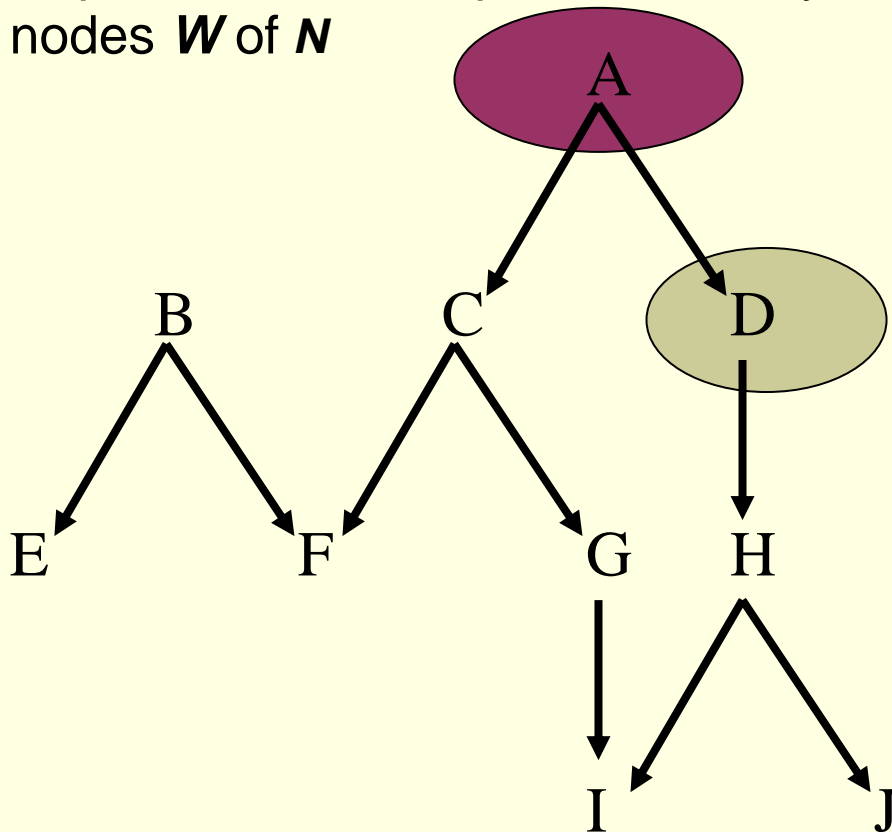
# Bayesian Networks

- Markov Property: the probability distribution of any node  $N$  given its parents  $P$  is independent of any subset of the non-descendent nodes  $W$  of  $N$



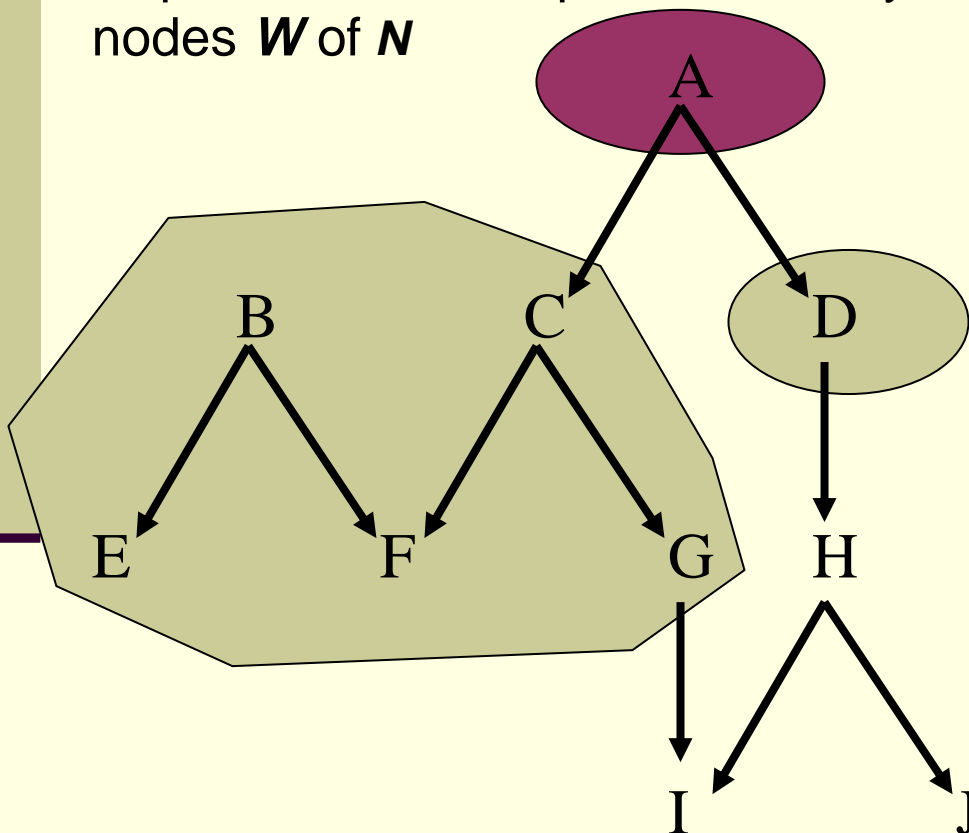
# Bayesian Networks

- Markov Property: the probability distribution of any node  $N$  given its parents  $P$  is independent of any subset of the non-descendent nodes  $W$  of  $N$



# Bayesian Networks

- Markov Property: the probability distribution of any node  $N$  given its parents  $P$  is independent of any subset of the non-descendent nodes  $W$  of  $N$

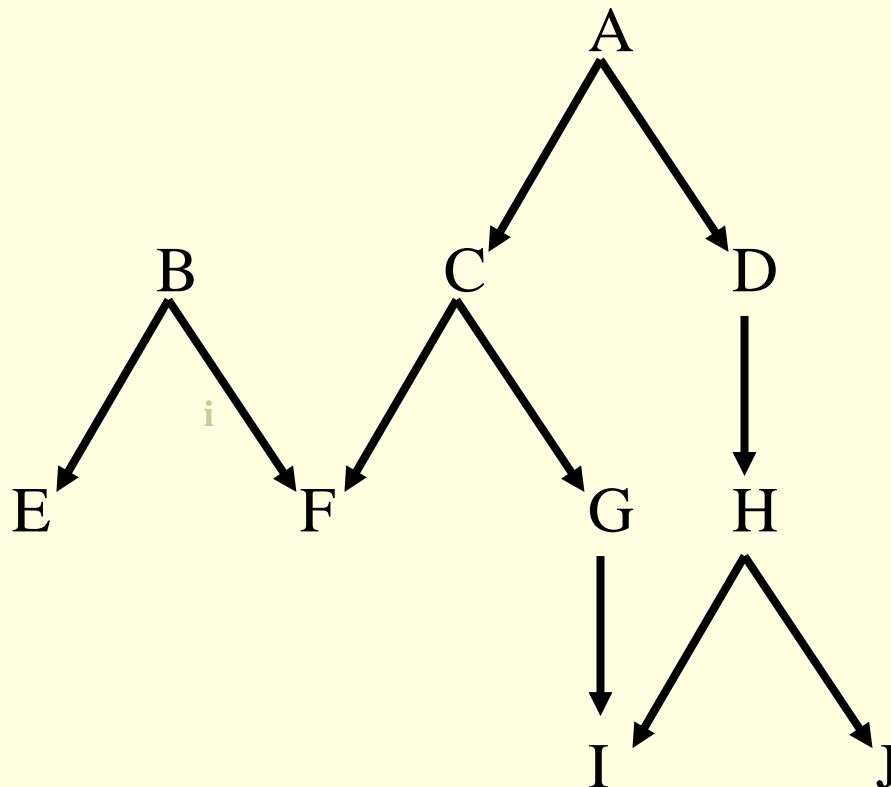


$$\text{Ind}(D, B, C, E, F, G | A)$$

$$\text{Ind}(D, B | A)$$

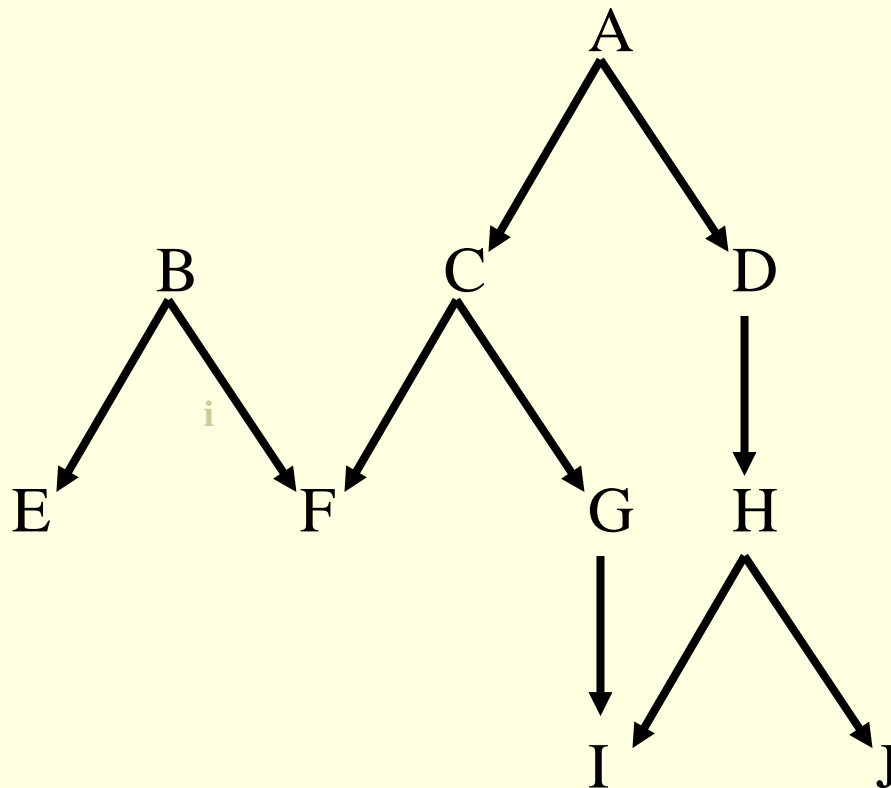
$$\text{Ind}\{F, E, G | B, C\}$$

# Bayesian Networks



$$\begin{aligned} P(V) = p(A,B,C,D,E,F,G,H,I,J) = & \\ & p(A) \times \\ & p(B|A) \\ & p(C|A,B) \times \\ & p(D|A,B,C) \times \\ & p(E|A,B,C,D) \times \\ & p(F|A,B,C,D,E) \times \\ & p(G|A,B,C,D,E,F) \times \\ & p(H|A,B,C,D,E,F,G) \times \\ & p(I|A,B,C,D,E,F,G,H) \times \\ & p(J|A,B,C,D,E,F,G,H,I) \end{aligned}$$

# Bayesian Networks



$$P(V) = p(A) \times$$

$$p(B|A)$$

$$p(C|A,B) \times$$

$$p(D|A,B,C) \times$$

$$p(E|A,B,C,D) \times$$

$$p(F|A,B,C,D,E) \times$$

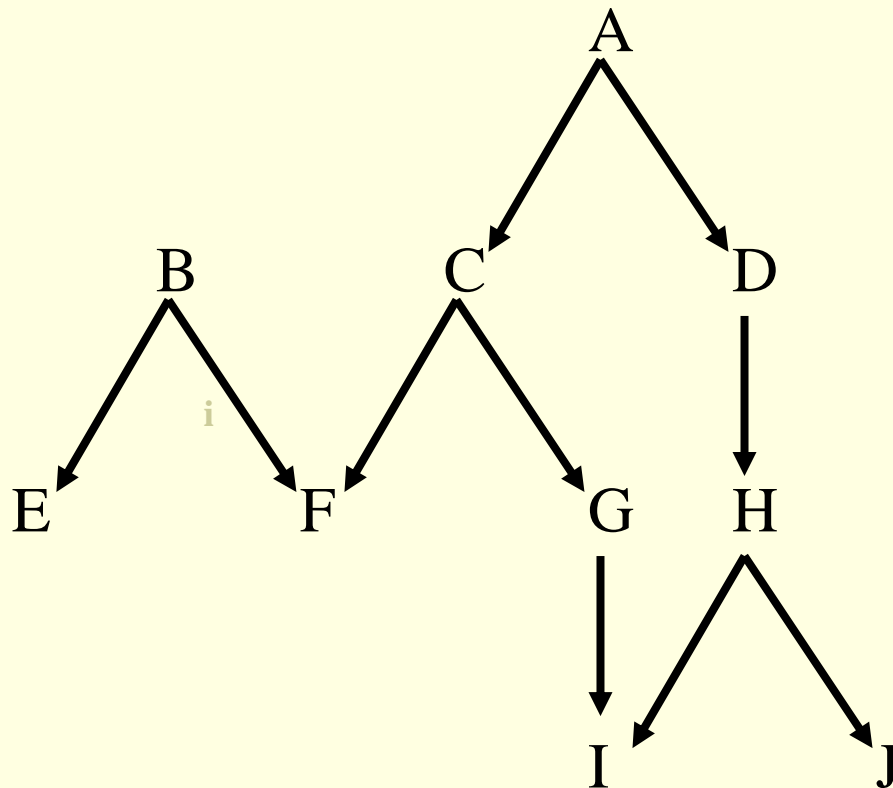
$$p(G|A,B,C,D,E,F) \times$$

$$p(H|A,B,C,D,E,F,G) \times$$

$$p(I|A,B,C,D,E,F,G,H) \times$$

$$p(J|A,B,C,D,E,F,G,H,I)$$

# Bayesian Networks



$$P(\mathbf{V}) = p(A | \text{Pa}(A))$$

$$p(B | \text{Pa}(B)) \times$$

$$p(C | \text{Pa}(C)) \times$$

$$p(D | \text{Pa}(D)) \times$$

$$p(E | \text{Pa}(E)) \times$$

$$p(F | \text{Pa}(F)) \times$$

$$p(G | \text{Pa}(G)) \times$$

$$p(H | \text{Pa}(H)) \times$$

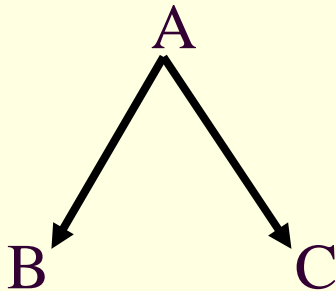
$$p(I | \text{Pa}(I)) \times$$

$$p(J | \text{Pa}(J)) =$$

$$\prod_i P(V_i | \text{Pa}(V_i))$$

# Bayesian Networks

Variables are binary:  $P(A,B,C)=P(A)P(B|A)P(C|A)$   
values {+, -}



The original JPD:

$P(A+, B+, C+) = 0.006$   
 $P(A+, B+, C-) = 0.014$   
 $P(A+, B-, C+) = 0.054$   
 $P(A+, B-, C-) = 0.126$   
 $P(A-, B+, C+) = 0.240$   
 $P(A-, B+, C-) = 0.160$   
 $P(A-, B-, C+) = 0.240$   
 $P(A-, B-, C-) = 0.160$

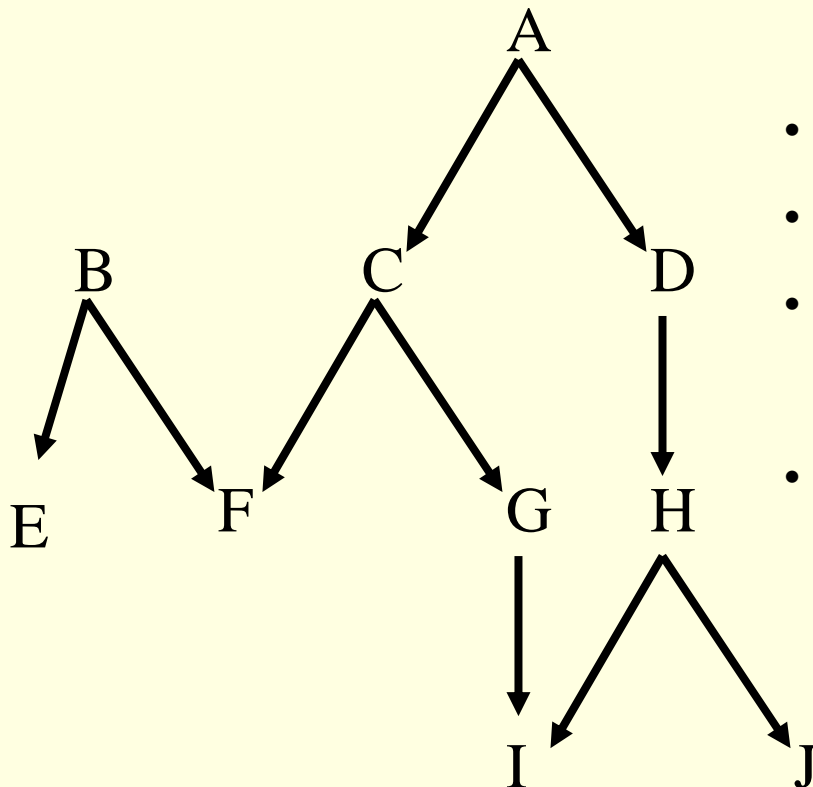
Becomes:

$P(A+) = 0.8$   
 $P(B+ | A+) = 0.1$   
 $P(B+ | A-) = 0.5$   
 $P(C+ | A+) = 0.3$   
 $P(C+ | A-) = 0.6$

A (potentially)  
exponential savings in  
parameters)

# Inference in Bayesian Networks

## ■ Known Bayesian Network:



- Forward:  $P(D+, I- | A+) = ?$
- Backward:  $P(A+ | C+, D+) = ?$
- Forward & Backward:  
 $P(D+, C- | I+, E+) = ?$
- Arbitrary  
predictors/predicted variables

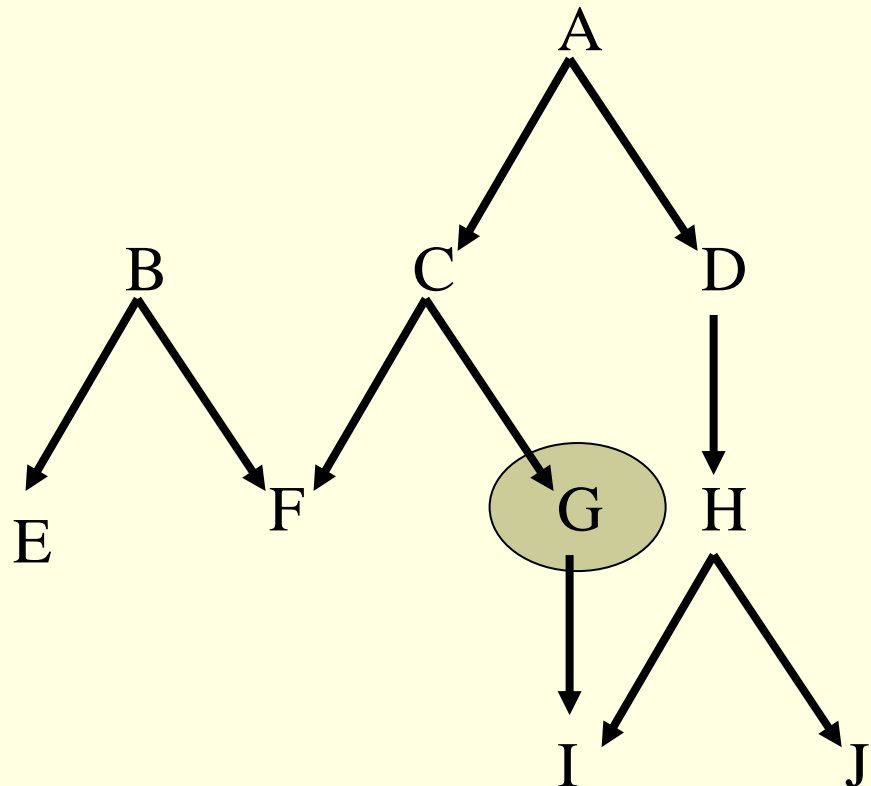
# Inference in Bayesian Networks

---

- Diagnosis/prediction/classification:
  - given any subset of the variables, calculate the probability distribution of any other variable (set) (e.g., given symptoms, diagnose patient): e.g.  $P(\text{Disease} \mid \text{Some Findings})$ ,  $P(\text{Gene Level}=\text{High} \mid \text{Gene Levels})$
- Algorithms exist for probabilistic inference
  - Exact
  - Approximate
  - NP-complete in the general case

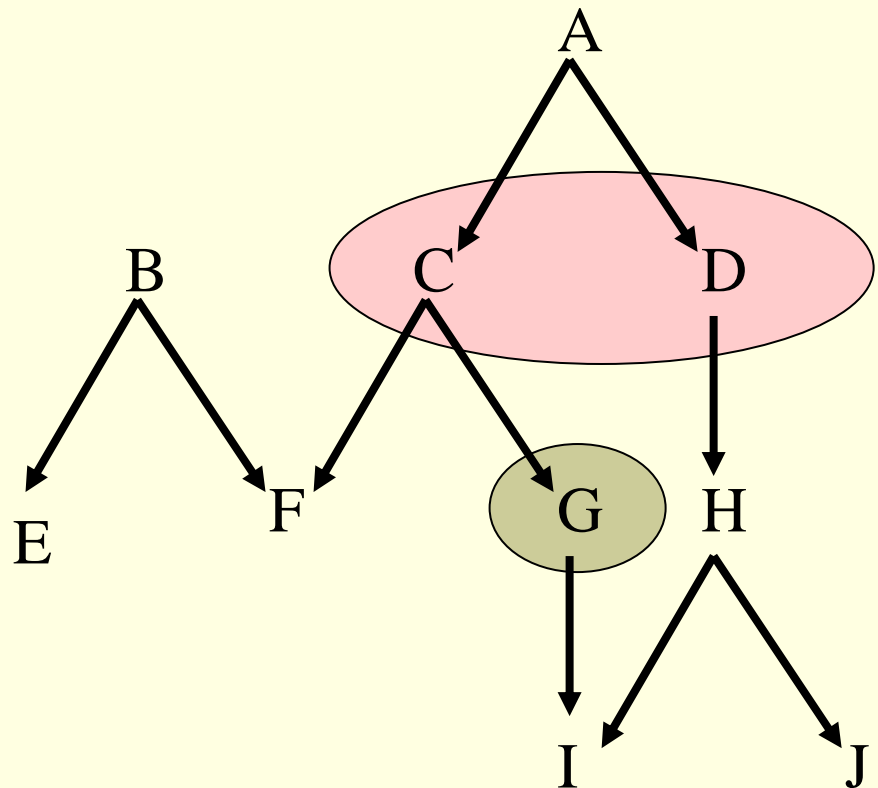
# The Markov Blanket

- $MB(T)$ : The minimal set of variables conditioned on which, all other variables are independent of  $T$
- $MB(T)$ : The set of parents, children, and spouses of  $T$



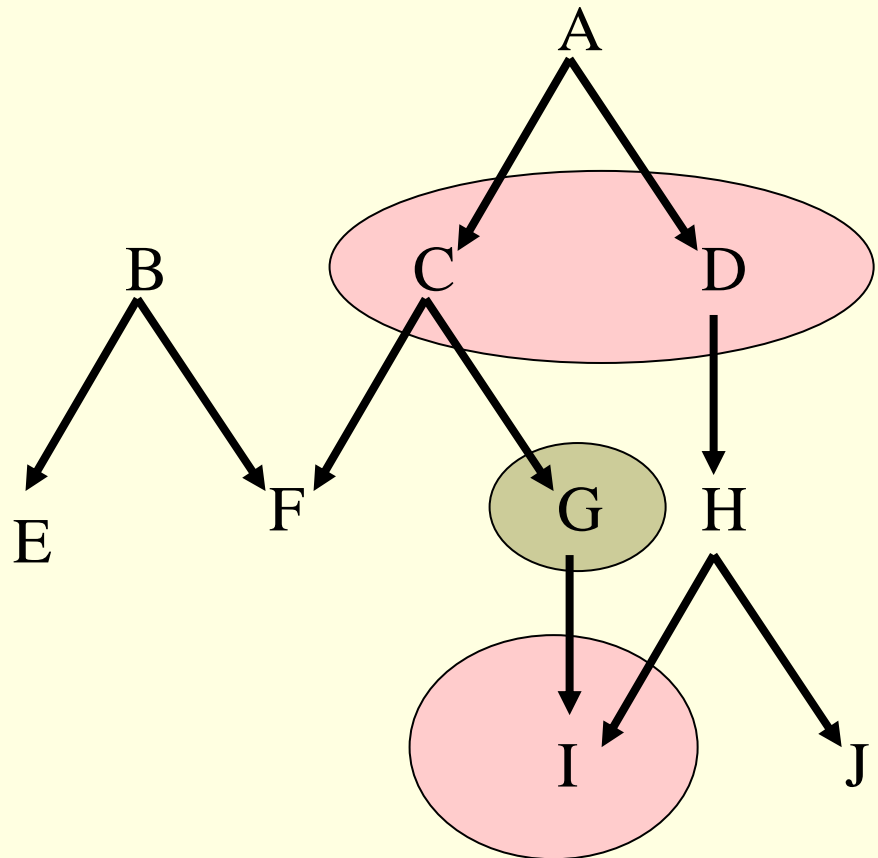
# The Markov Blanket

- $MB(T)$ : The minimal set conditioned on which, all other variables are independent of  $T$
- $MB(T)$ : The set of parents, children, and spouses of  $T$



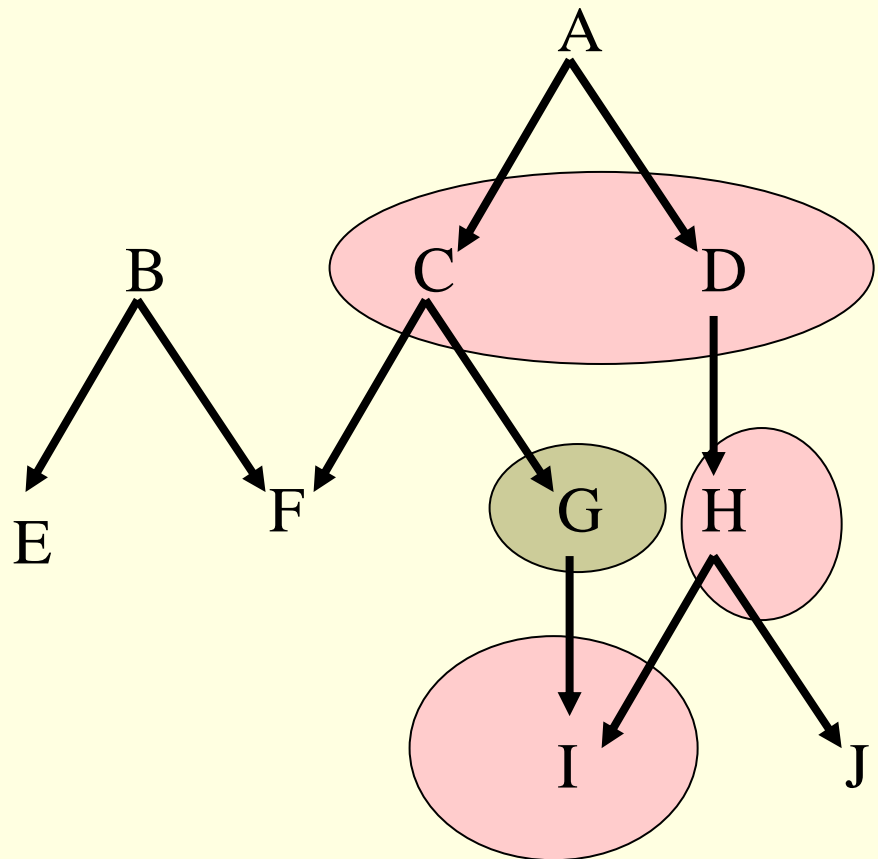
# The Markov Blanket

- $MB(T)$ : The minimal set conditioned on which, all other variables are independent of  $T$
- $MB(T)$ : The set of parents, children, and spouses of  $T$



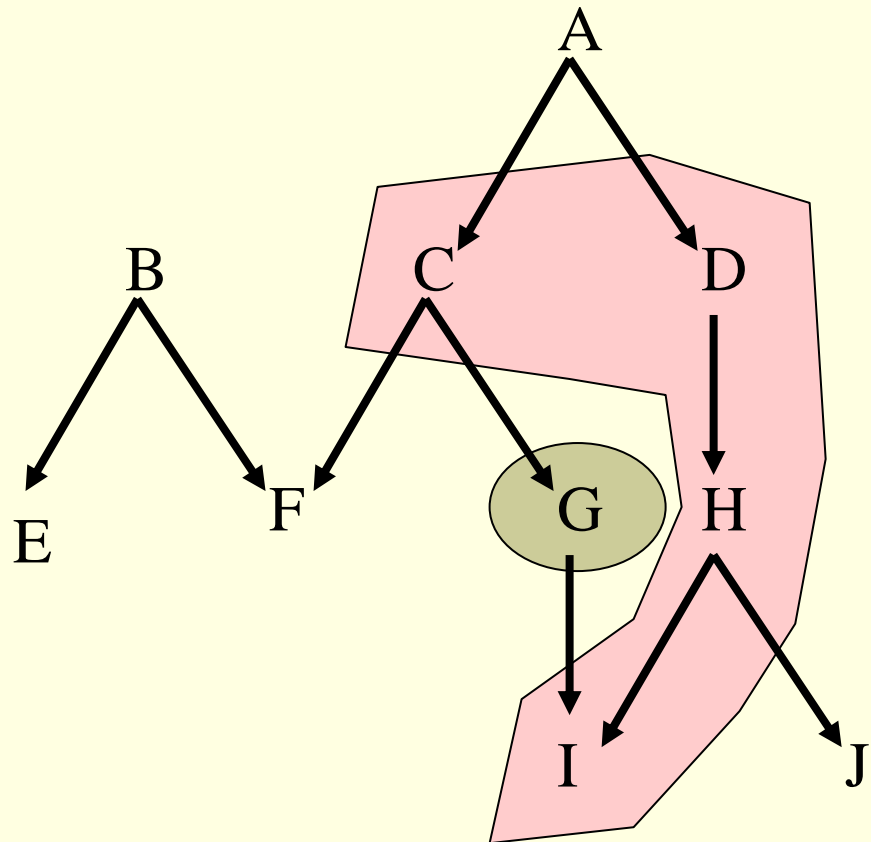
# The Markov Blanket

- $MB(T)$ : The minimal set conditioned on which, all other variables are independent of  $T$
- $MB(T)$ : The set of parents, children, and spouses of  $T$



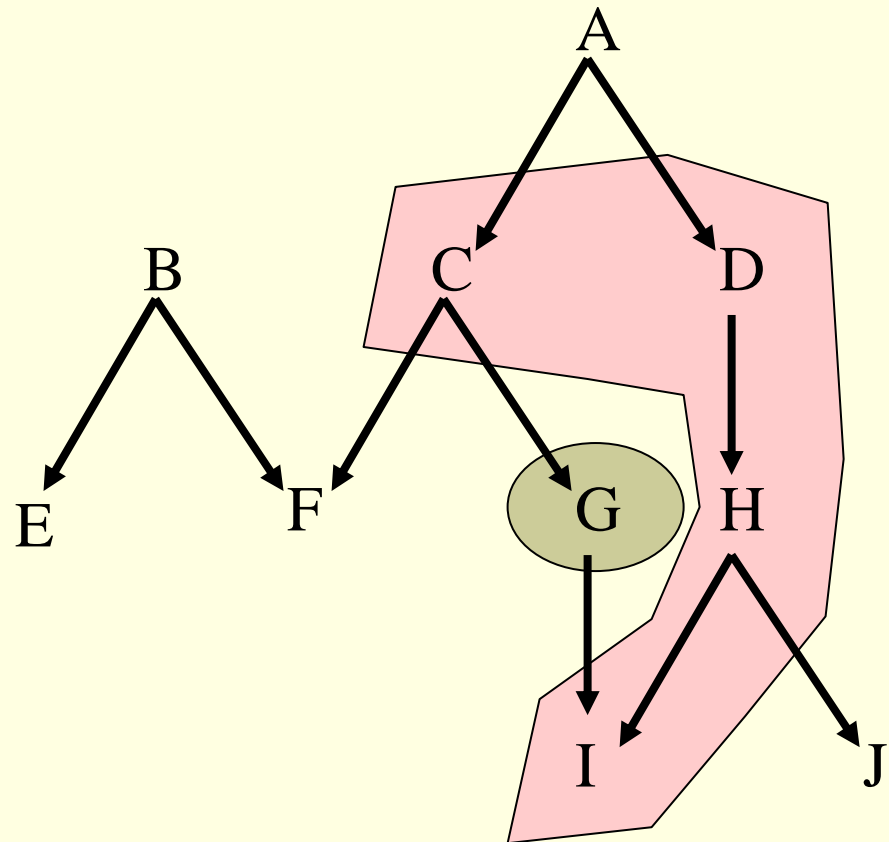
# The Markov Blanket

- $MB(T)$ : The minimal set conditioned on which, all other variables are independent of  $T$
- $MB(T)$ : The set of parents, children, and spouses of  $T$



# The Markov Blanket

- Knowing the values of the  $MB(T)$ , all other variables are irrelevant (no new information)



# Markov Blanket for Variable Selection

---

- $MB(T)$  is all the variables we need for optimal classification (if we have a powerful enough classifier/density estimator)
- If probability density of  $T$  is desired,  $MB(T)$  is the minimum set of variables required
- If maximum accuracy is desired (0/1 Loss function),  $MB(T)$  is a good approximation of the minimum set.
- [Tsamardinos, Aliferis, AI&Stats 2003]

# Bayesian Networks and Causal Discovery

---

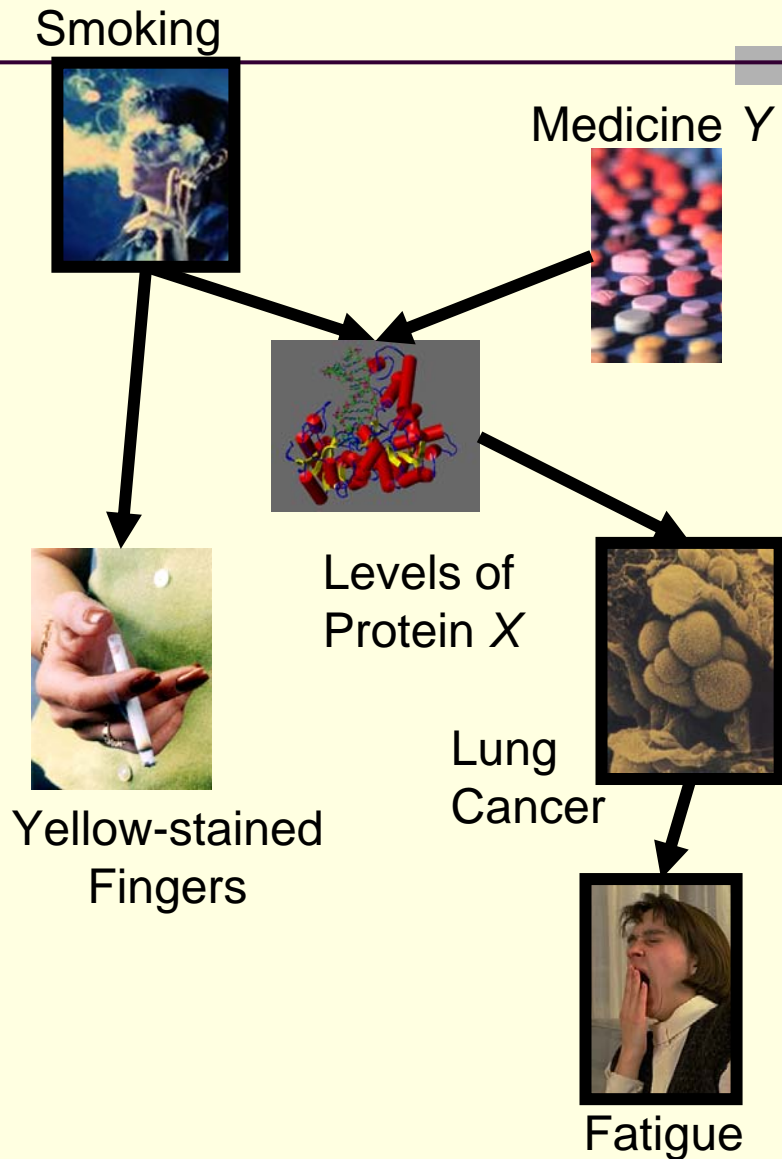
## Bayesian Networks

- Edges: probabilistic dependence
- Markov Condition: A node  $N$  is independent from non-descendants given its parents
- Probabilistic reasoning

## Causal Bayesian Networks

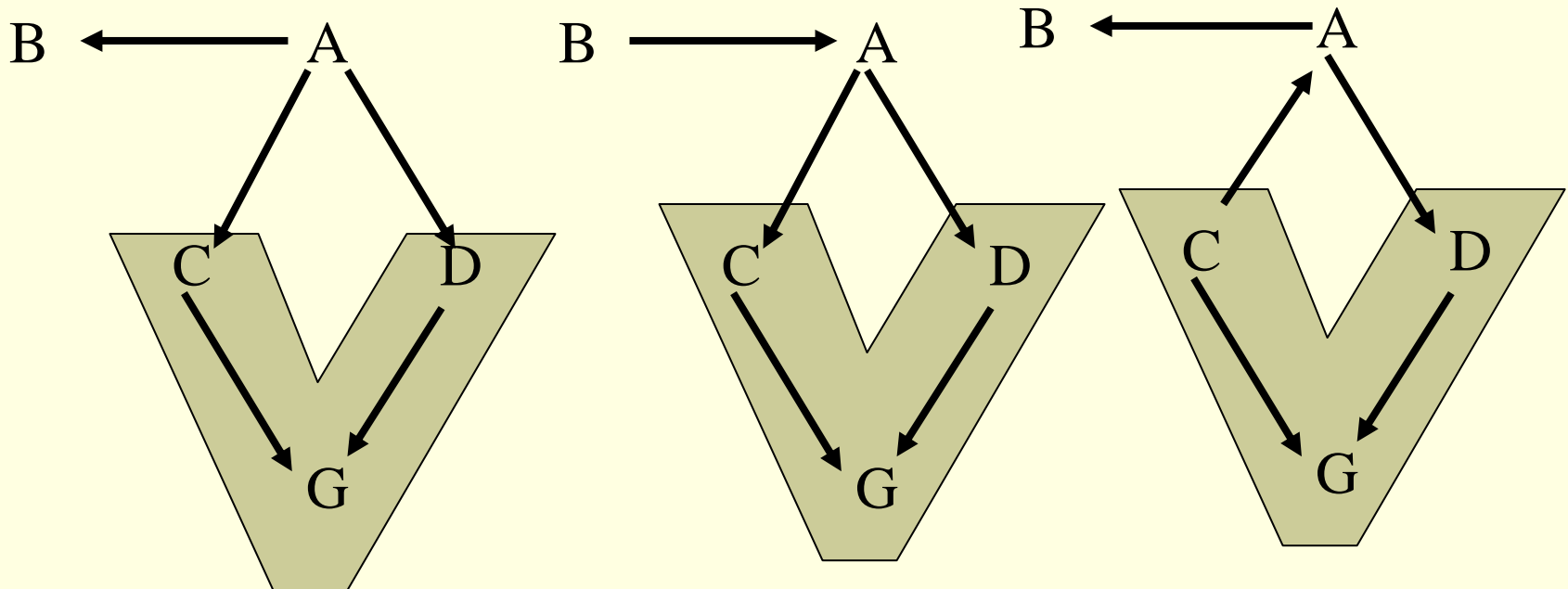
- Edges represent direct causal effects
- Causal Markov Condition: A node  $N$  is independent from non-descendants given its direct causes
- Probabilistic reasoning + causal inferences

# Bayesian Networks and Causal Discovery



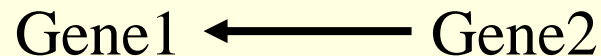
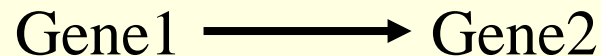
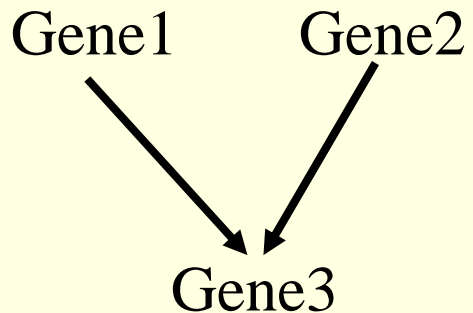
# Causal Bayesian Networks

- There may be many (non-causal) BNs that capture the same distribution.
- All such BNs have the same edges (ignoring direction) same v-structures
- Statistically equivalent



# Causal Bayesian Networks

- If there is a (faithful) Causal Bayesian Network that captures the data generation process, it has to have the same edges and same v-structures as any (faithful) Bayesian Network that is induced by the data.



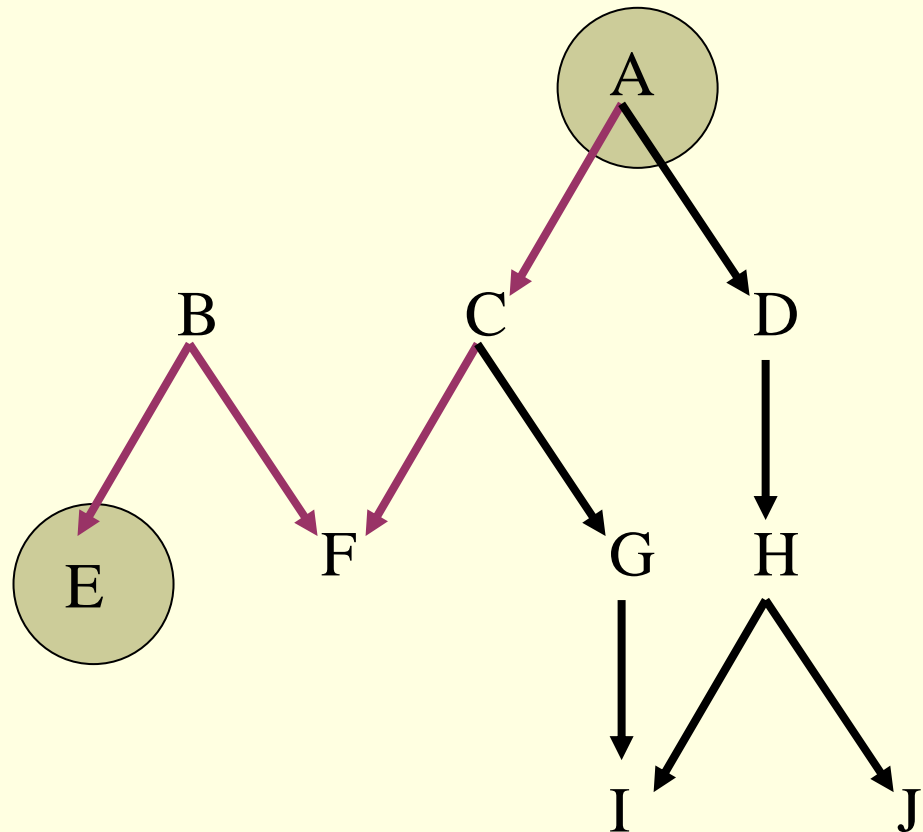
# Bayesian Networks are Useful

---

- Probabilistic inference, diagnosis, classification, prediction, value of information.
- Variable Selection (optimal, principled)
- Causal Inference
- Causal hypotheses generation

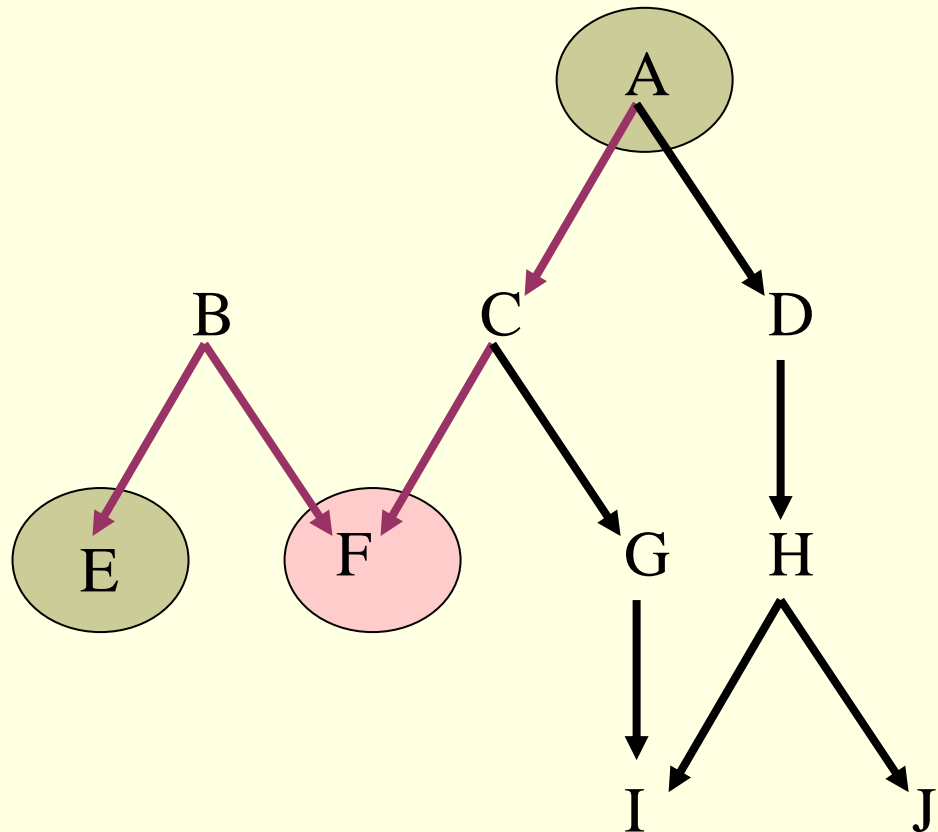
# *d*-separation

- Consider a (non-directed) path, e.g., E to A



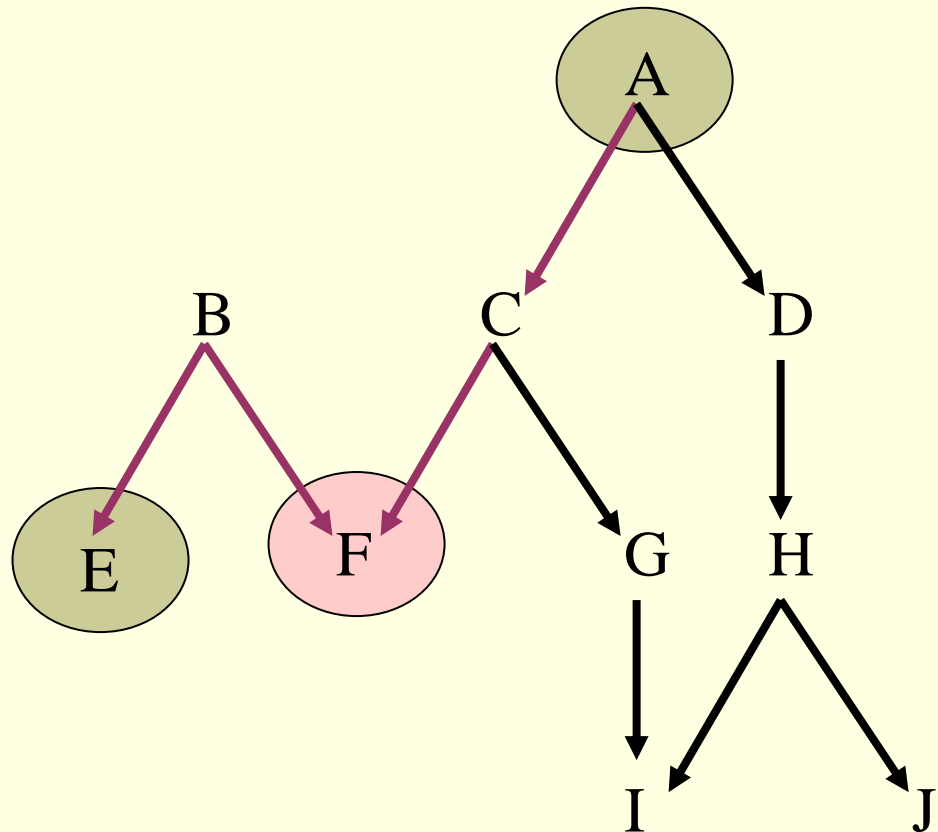
# *d*-separation: A criterion for independence

- Collider: a node with two incoming edges



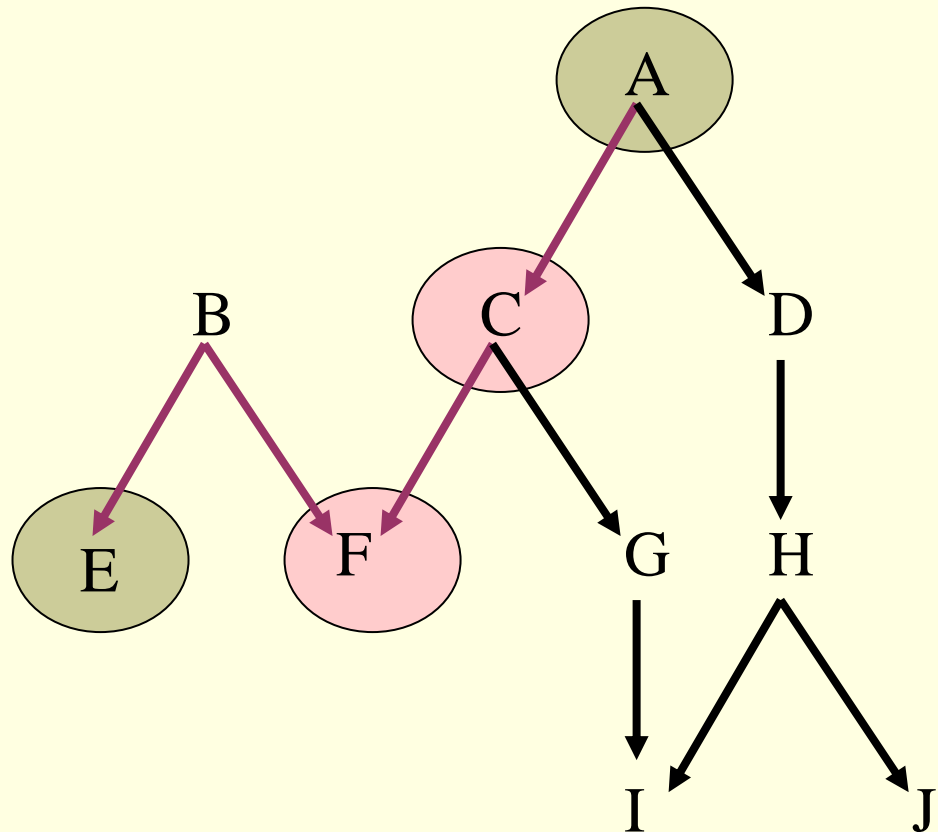
# *d*-separation: A criterion for independence

- Open path: information flows through it
- Colliders: open a path when conditioned on, close it otherwise
- Conditioned on F, path is open (A is giving information for E conditioned on F)



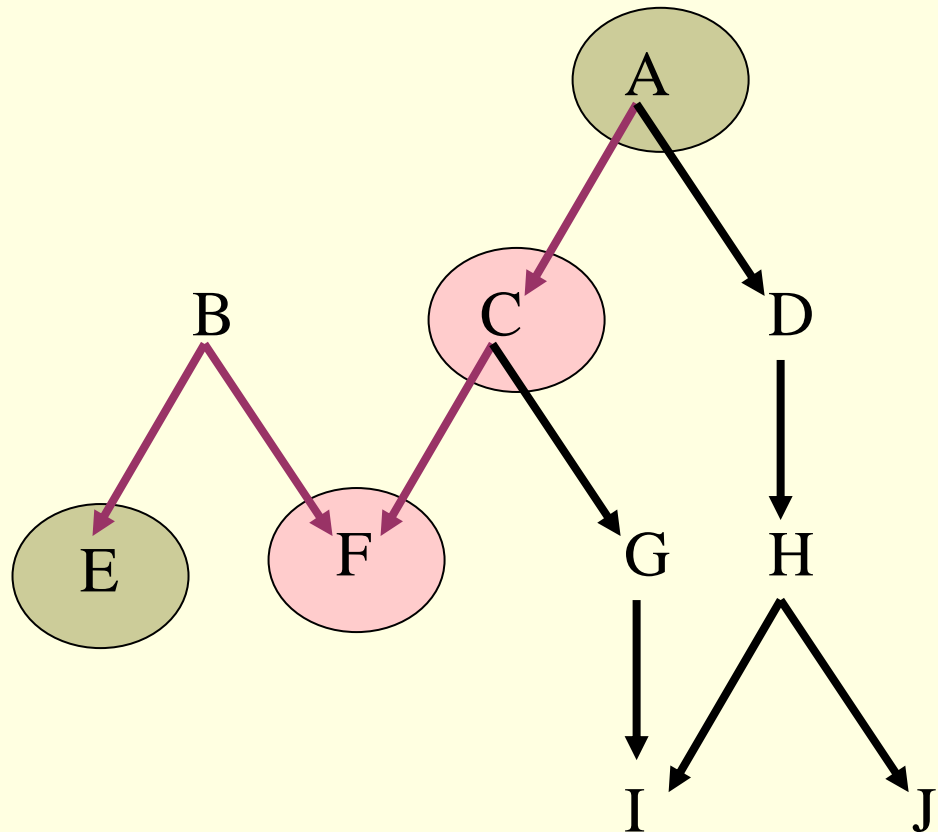
# *d*-separation: A criterion for independence

- Non-colliders: close the path when conditioned on, open it otherwise
- Conditioned on C the path is closed



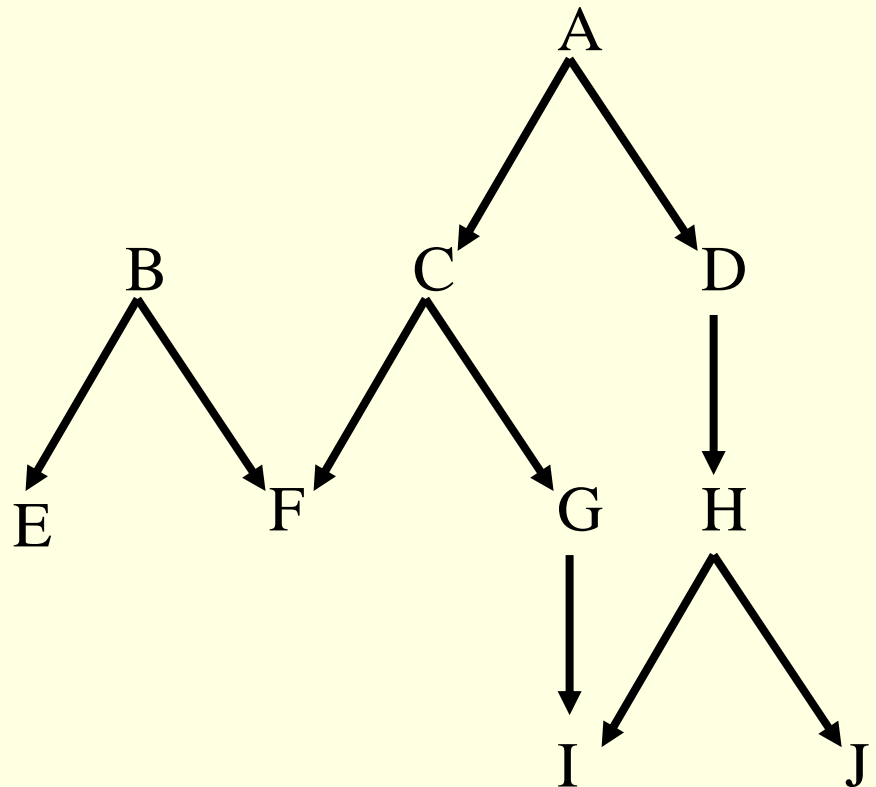
# *d*-separation: A criterion for independence

- Two nodes conditioned on a set of nodes:
  - *d-separated* if all paths are closed
- *d-separation*  $\Rightarrow$  Independence



# *d*-separation: A criterion for independence

- $\text{Dep}(E, A | F)$
- $\text{Ind}(E, A | \text{empty})$
- $\text{Ind}(A, J | D)$
- $\text{Dep}(A, J | D, I)$



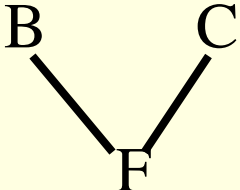
# Faithfulness

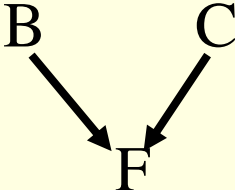
---

- When  $d$ -separation  $\Leftrightarrow$  independence
- Intuitively, an open path between  $A$  and  $B$  means there is association between them.
- Previous discussion holds for faithful BNs only

# Learning Bayesian Networks: Constraint-Based Approach

- An edge  $X - Y$  (of unknown direction) exists, if and only if for all sets of nodes  $S$ ,  $\text{Dep}(X, Y / S)$  (allows discovery of the edges)
- Test all subsets. If  $\text{Dep}(X, Y | s)$  holds, add the edge, otherwise do not.

■ If structure  and for every set  $S$  that

contains  $F$ ,  $\text{Dep}(X, Y / S)$ , then 

# Learning Bayesian Networks: Constraint-Based Approach

---

- Tests of conditional dependences and independencies from the data.
- Estimation using  $G^2$  statistic, conditional mutual-information, etc.
- Infer structure and orientation from results of tests.
- Based on the assumption these tests are accurate.
- The larger the number of nodes in the conditioning set, the more samples are required to estimate the dependence,  $\text{Ind}(A,B|C,D,E)$  more sample than  $\text{Ind}(A,B|C,D)$
- For relatively sparse networks, we can  $d$ -separate two nodes conditioned on a couple of variables (sample requirements in the low hundreds).

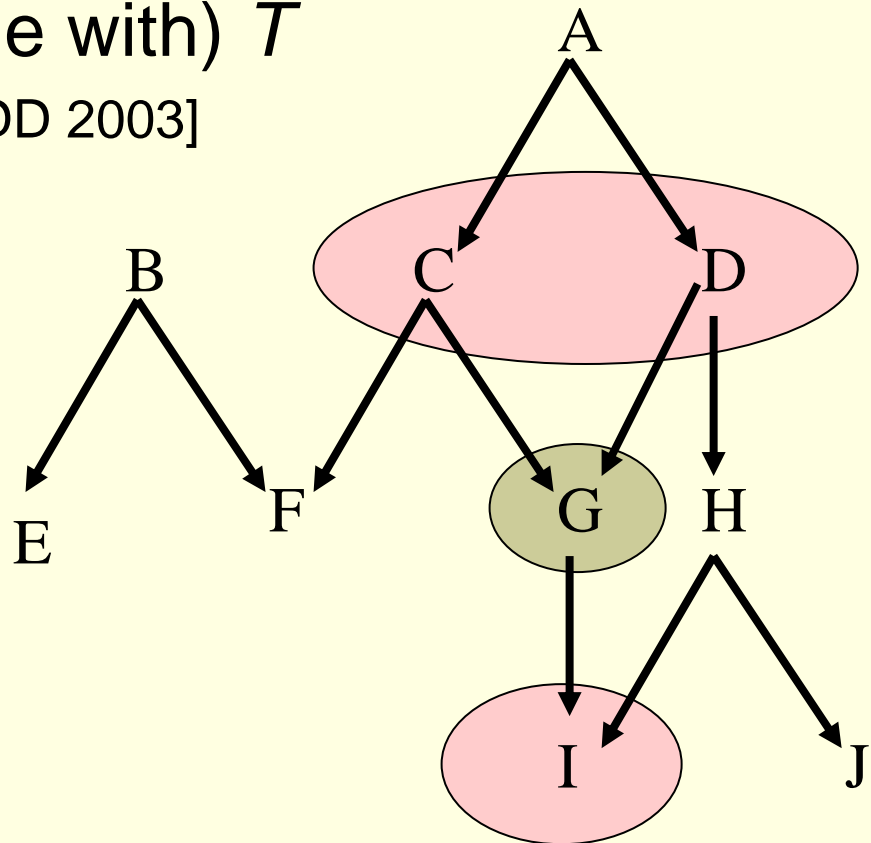
# Learning Bayesian Networks: Search-and-Score

---

- Score each possible structure
- Bayesian score:  $P(\text{Structure} \mid \text{Data})$
- Search in the space of all possible BNs structures to find the one that maximizes score.
- Search space too large. Greedy or local search is typical.
- Greedy search: add, delete, or reverse the edge that increases the score the most.

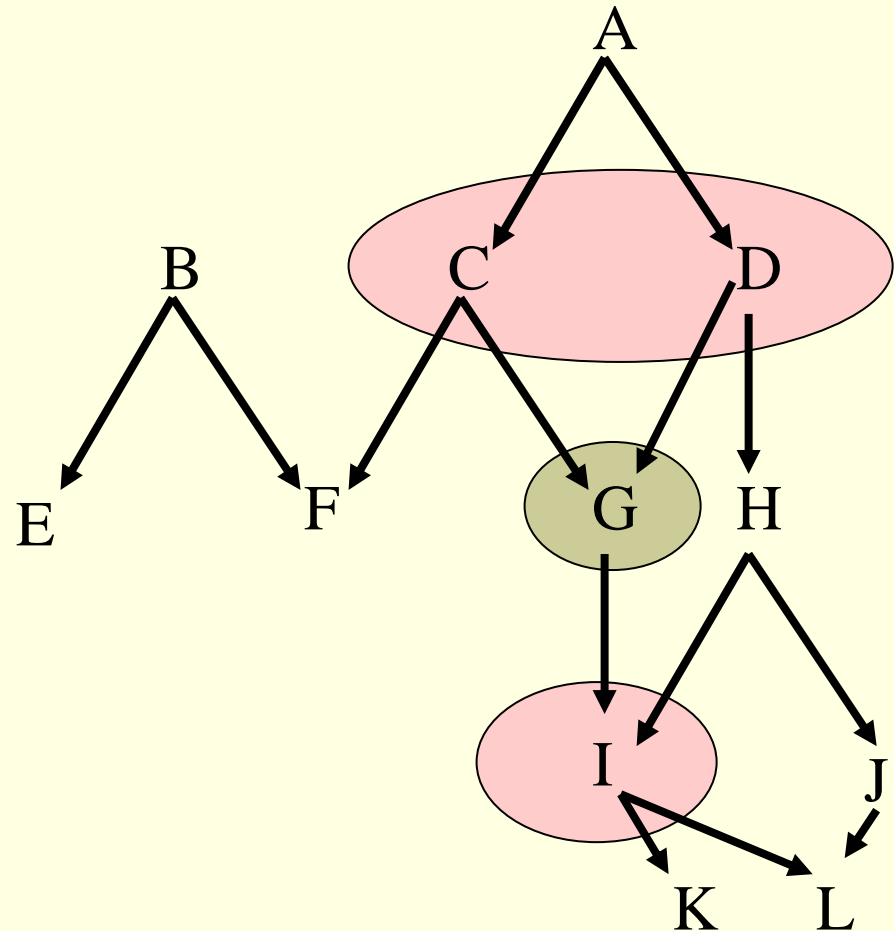
# Max-Min Parents and Children Algorithm

- Discovers the parents and children of (all nodes with an edge with)  $T$
- [Tsamardinos, Aliferis, KDD 2003]



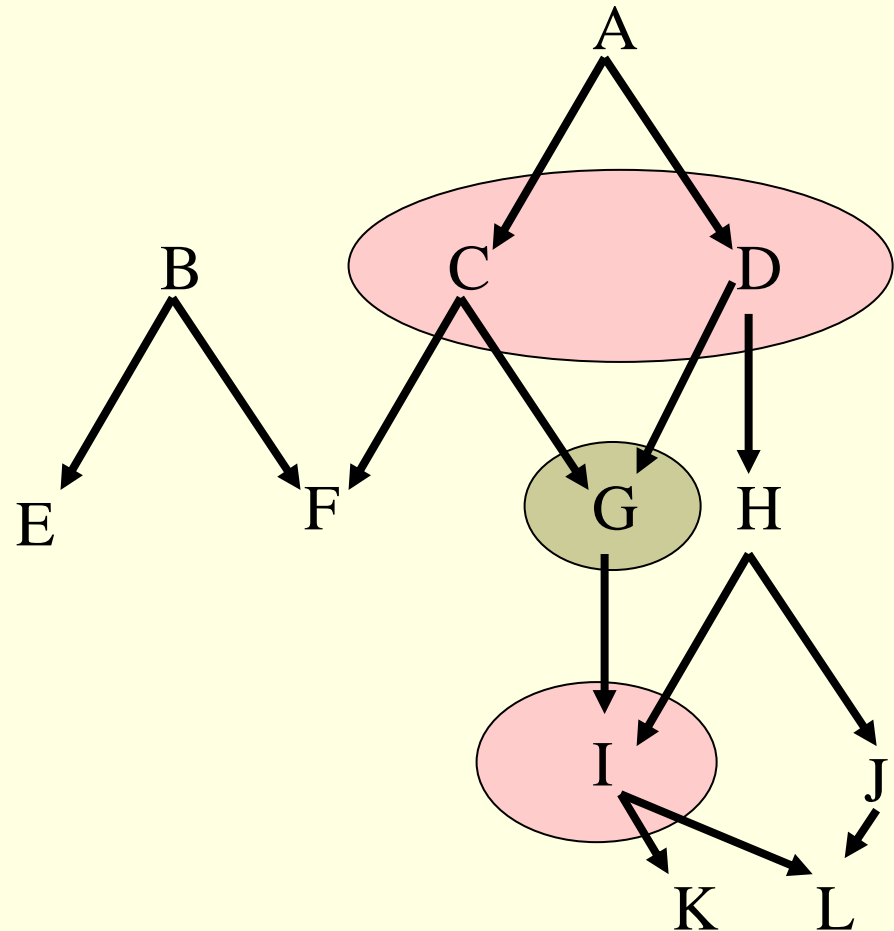
# Max-Min Parents and Children Algorithm

- Observation:  
conditioned on  
some subset of  
the parents and  
children any  
other node is  
independent  
with G



# Max-Min Parents and Children Algorithm

- Idea: if we quickly get a superset of the parents and children we can verify all other nodes become independent (conditioned on some subset)



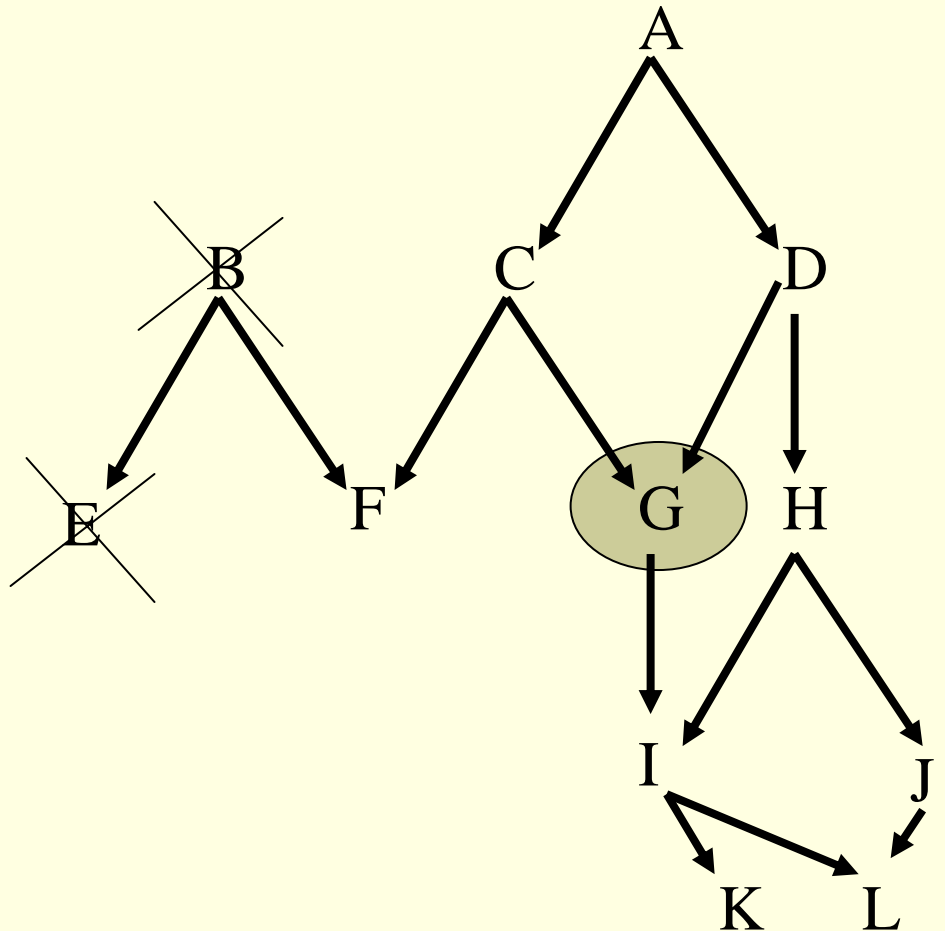
# Max-Min Parents and Children Algorithm

---

- Phase I: Forward
- Start with the empty set for  $PC(T) = \emptyset$
- Repeat
  - For each subset  $s$  of  $PC(T)$ 
    - Remove from further consideration all nodes  $X$  that  $\text{Ind}(X, T/s)$
  - Heuristic Step: Select variable  $X$  out of the remaining ones
  - $PC(T) = PC(T) \cup X$
- Until no change in  $PC(T)$  (all variables independent)

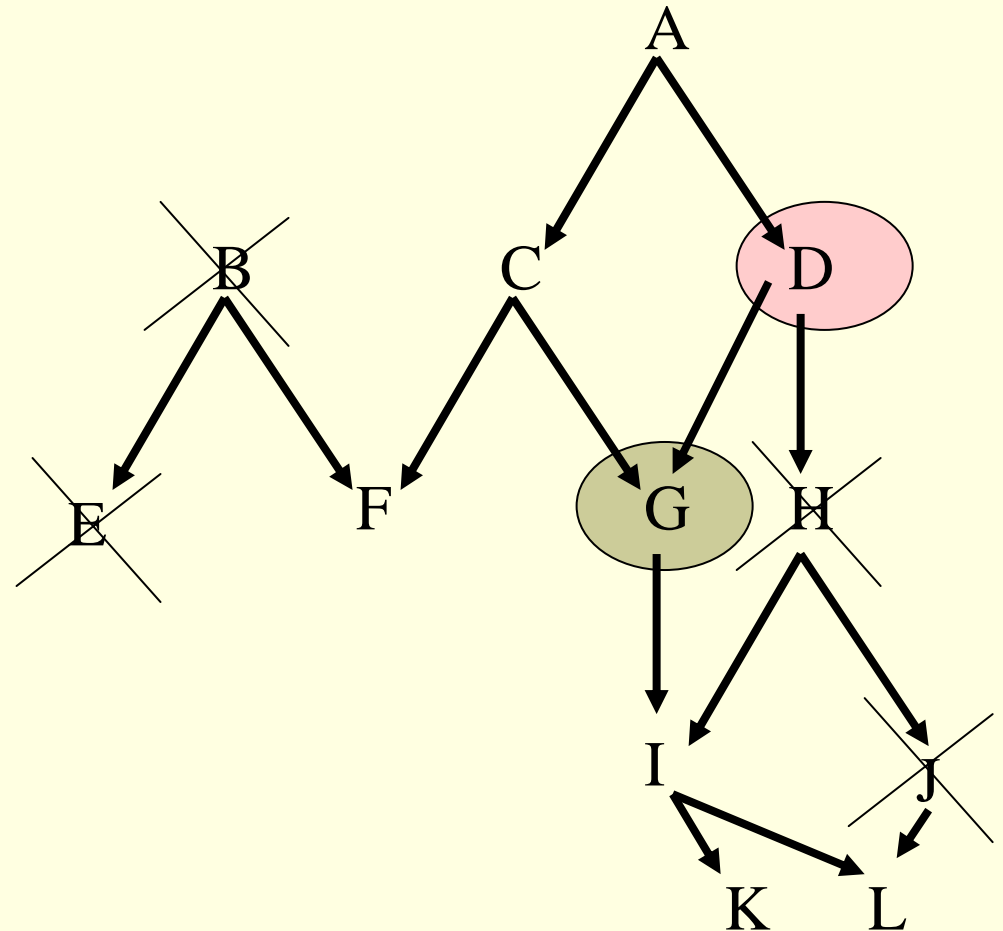
# Max-Min Parents and Children Algorithm

- $PC(T) = \emptyset$
- For each subset  $s$  of  $PC(T)$ 
  - Remove all nodes  $X$  that  $\text{Ind}(X, T/s)$
- Select variable  $D$  out of the remaining ones
- $PC(T) = \{D\}$



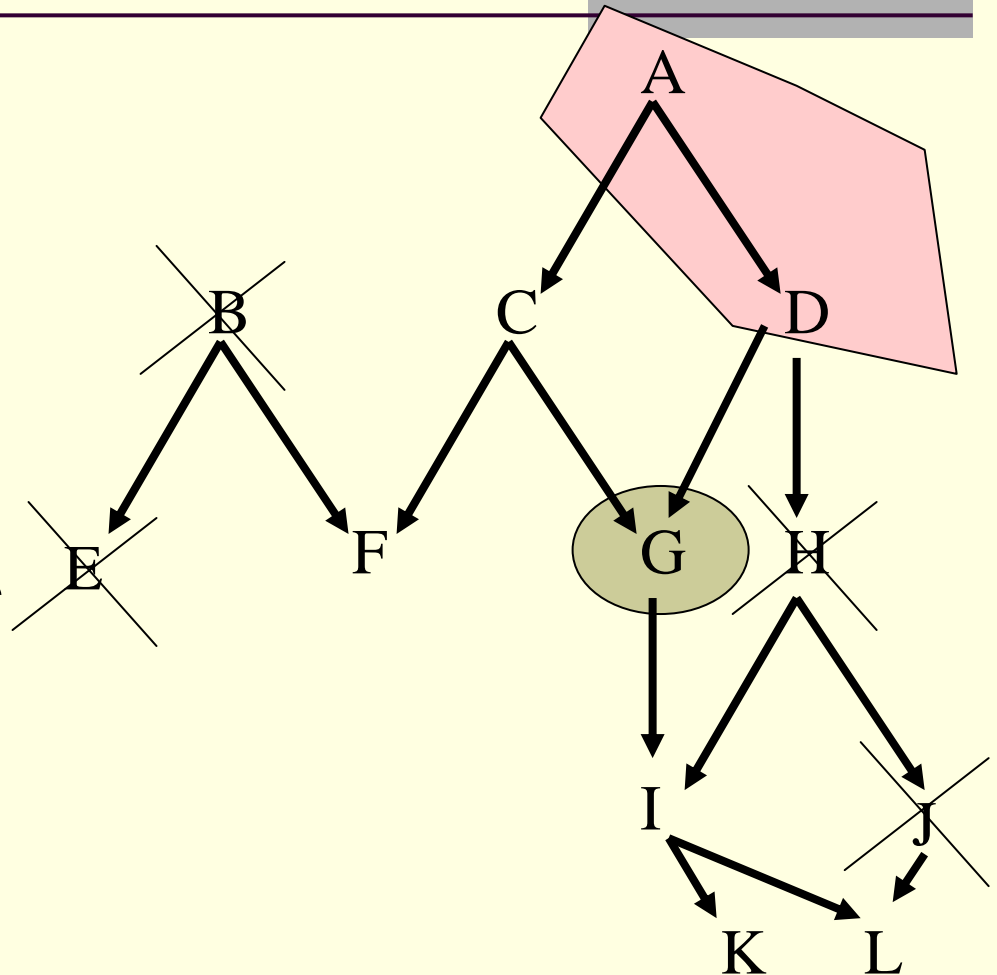
# Max-Min Parents and Children Algorithm

- $PC(T) = \{D\}$
- For each subset  $s$  of  $PC(T)$ 
  - Remove all nodes  $X$  that  $\text{Ind}(X, T/s)$
- Select variable  $A$  out of the remaining ones
- $PC(T) = \{D, A\}$



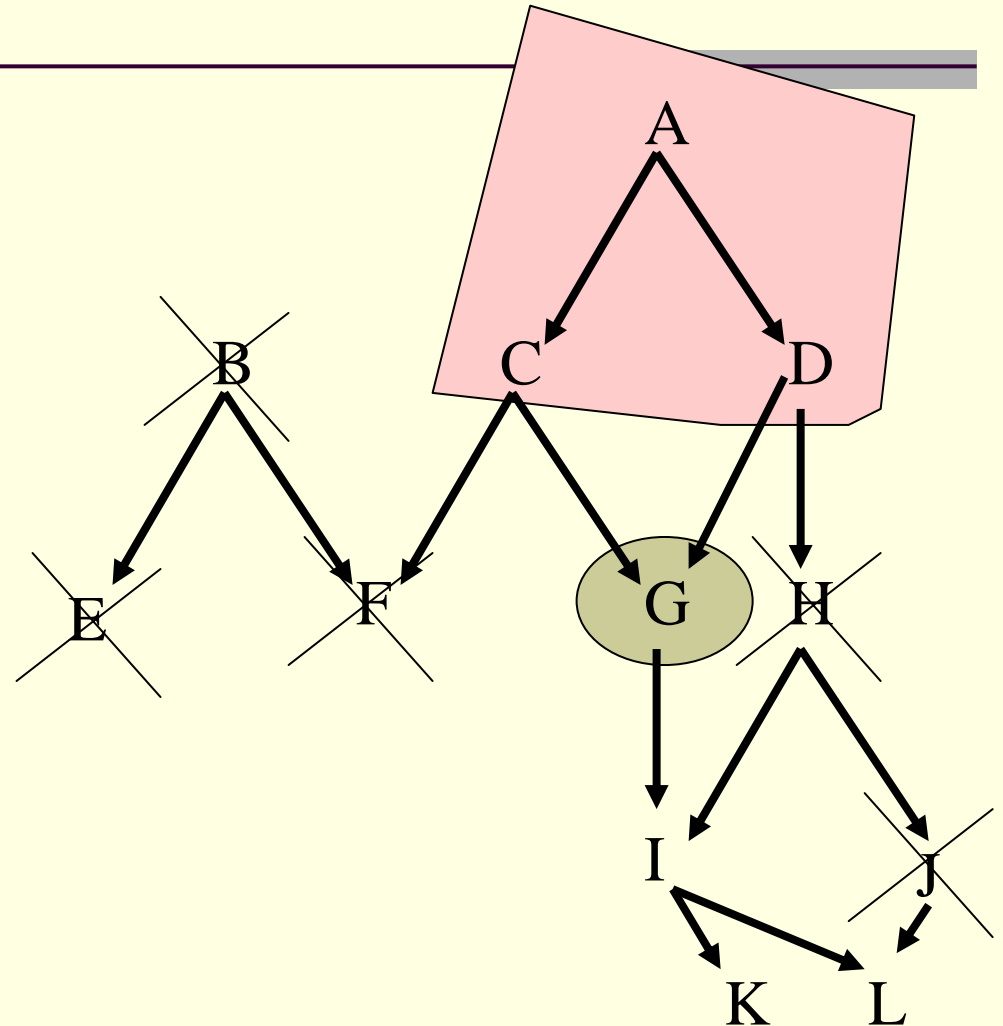
# Max-Min Parents and Children Algorithm

- $PC(T) = \{D, A\}$
- For each subset  $s$  of  $PC(T)$ 
  - Remove all nodes  $X$  that  $\text{Ind}(X, T/s)$
- Select variable  $C$  out of the remaining ones
- $PC(T) = \{D, A, C\}$



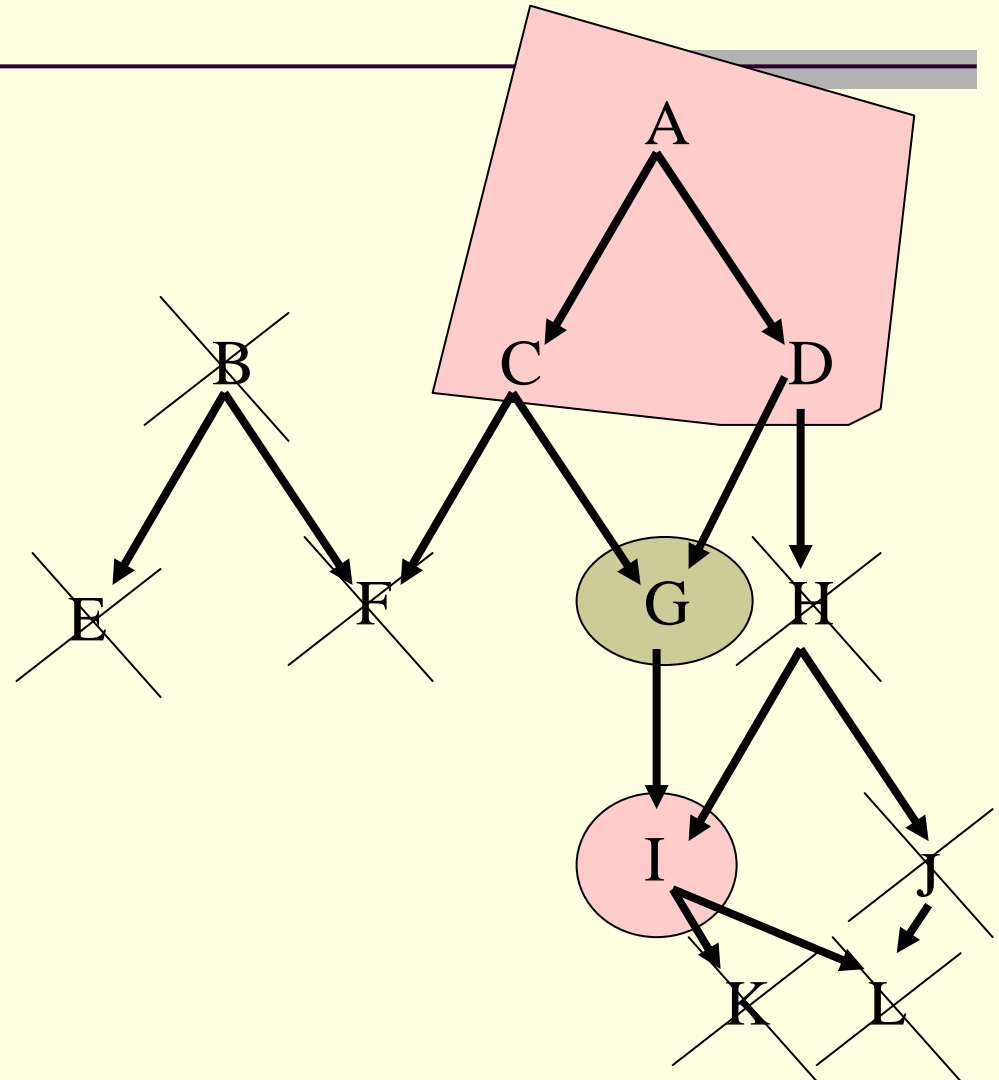
# Max-Min Parents and Children Algorithm

- $PC(T) = \{D, C\}$
- For each subset  $s$  of  $PC(T)$ 
  - Remove all nodes  $X$  that  $\text{Ind}(X, T/s)$
- Select variable  $I$  out of the remaining ones
- $PC(T) = \{D, C, I\}$



# Max-Min Parents and Children Algorithm

- $PC(T) = \{D, C, I\}$
- For each subset  $s$  of  $PC(T)$ 
  - Remove all nodes  $X$  that  $\text{Ind}(X, T/s)$
- No variable is left, so stop



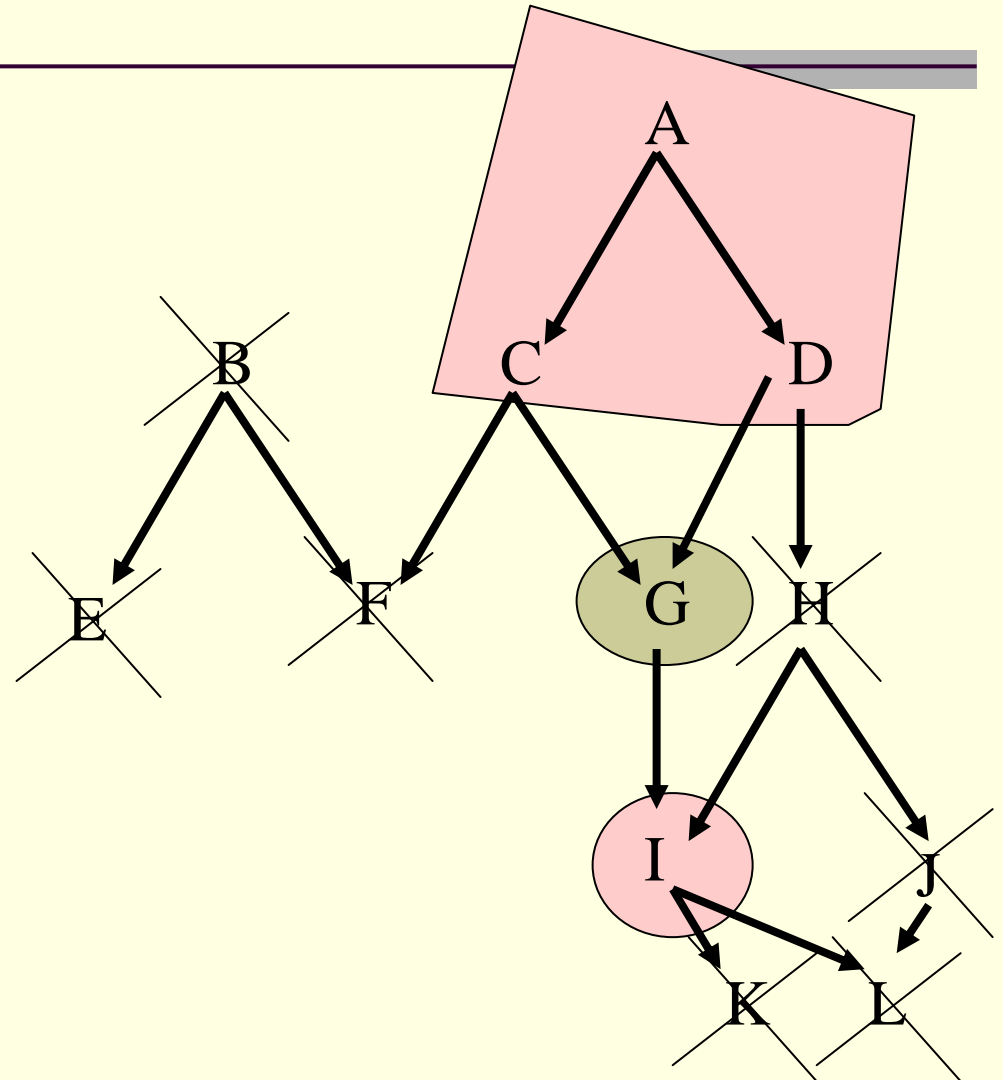
# Max-Min Parents and Children Algorithm

---

- Phase II: Backward
- For each variable  $X$  in  $PC(T)$ 
  - Remove  $X$ , if there exists subset  $s$  of  $PC(T)$  such that  $\text{Ind}(X, T | S)$

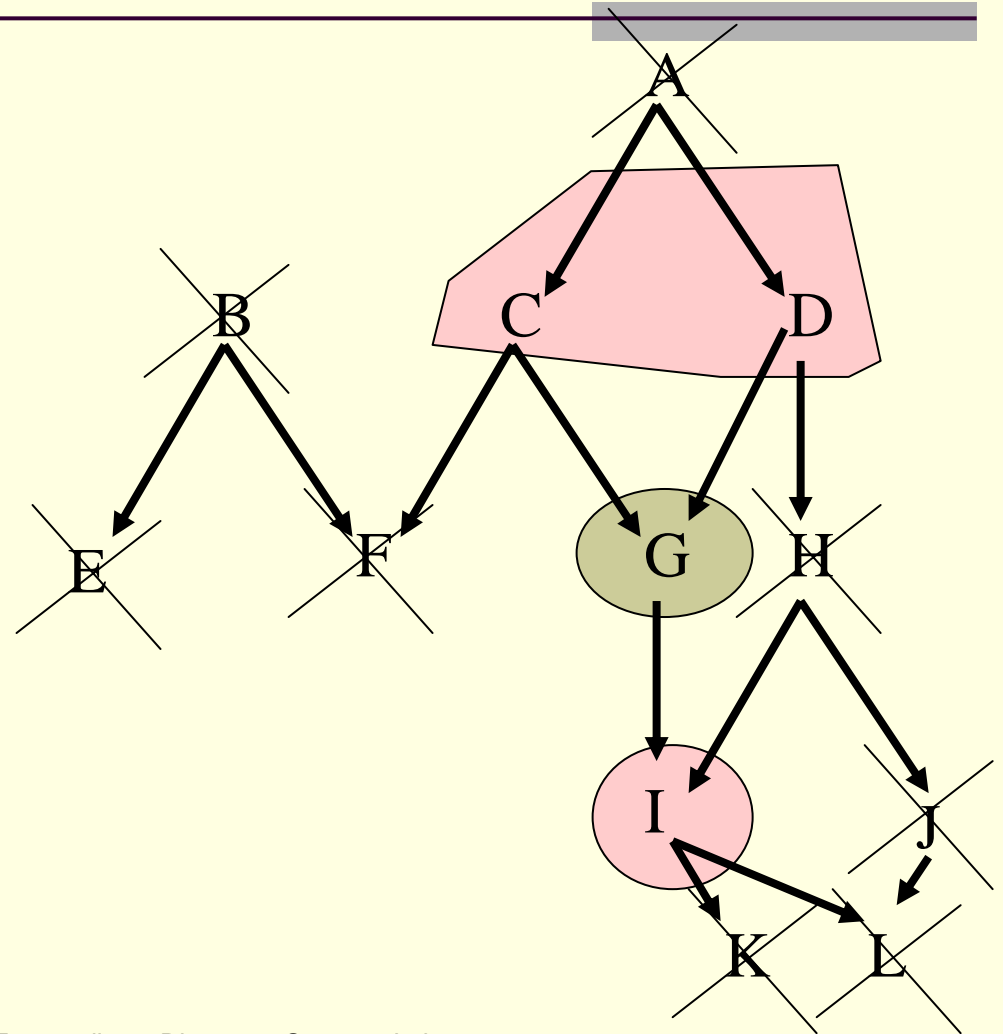
# Max-Min Parents and Children Algorithm

- $\text{Ind}(A, G/C, D)$
- So, remove  $A$



# Max-Min Parents and Children Algorithm

- $\text{Ind}(A, T/C, D)$
- So, remove  $A$



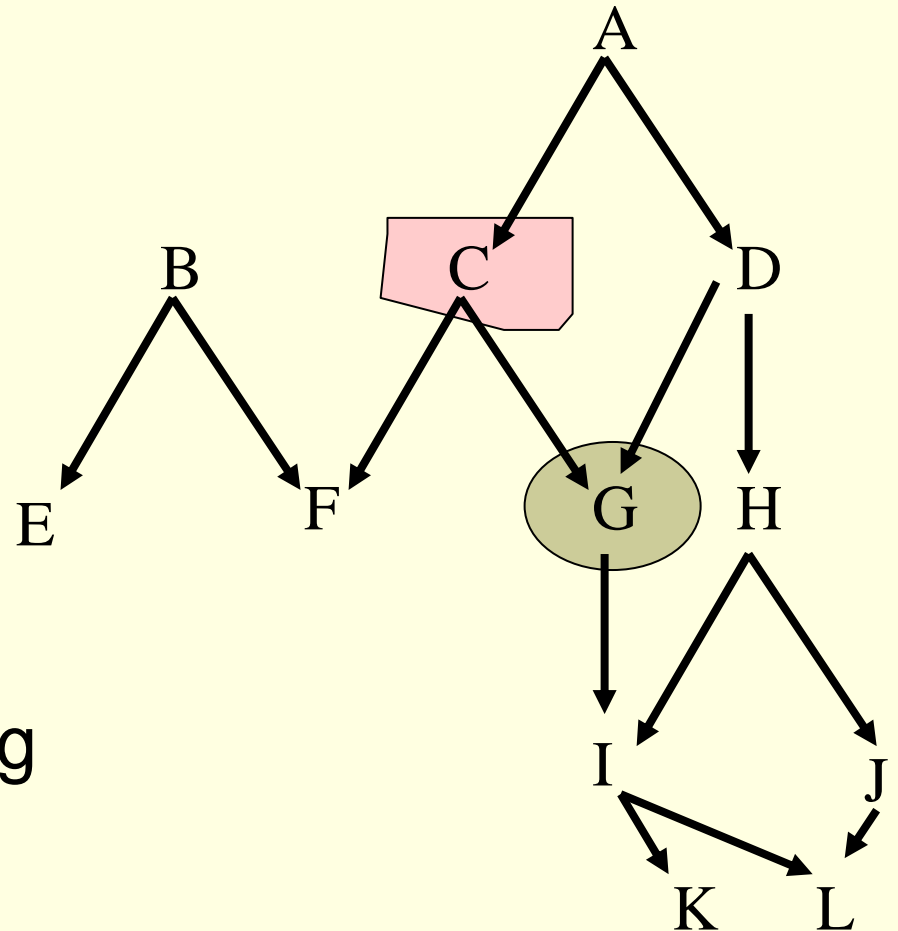
# Max-Min Parents and Children Algorithm

---

- The variable selection heuristic is essential
- Variables with high association are usually structurally close to the target.
- Max-Min heuristic:
  - Measure the association of each variable  $X$  with  $T$  condition on some subset  $s$ .
  - Select the variable that maximizes this association.
  - Subset  $s$  is the subset conditioned on which association of  $X$  with  $T$  is minimized.

# Max-Min Parents and Children Algorithm

- Assume conditioned on nothing:
- $\text{assoc}(A, G) > \text{assoc}(I, G)$
- But,  $\text{assoc}(A, G|C) < \text{assoc}(A, I|C)$
- So,  $I$  has better chances of entering  $PC(T)$  before  $A$



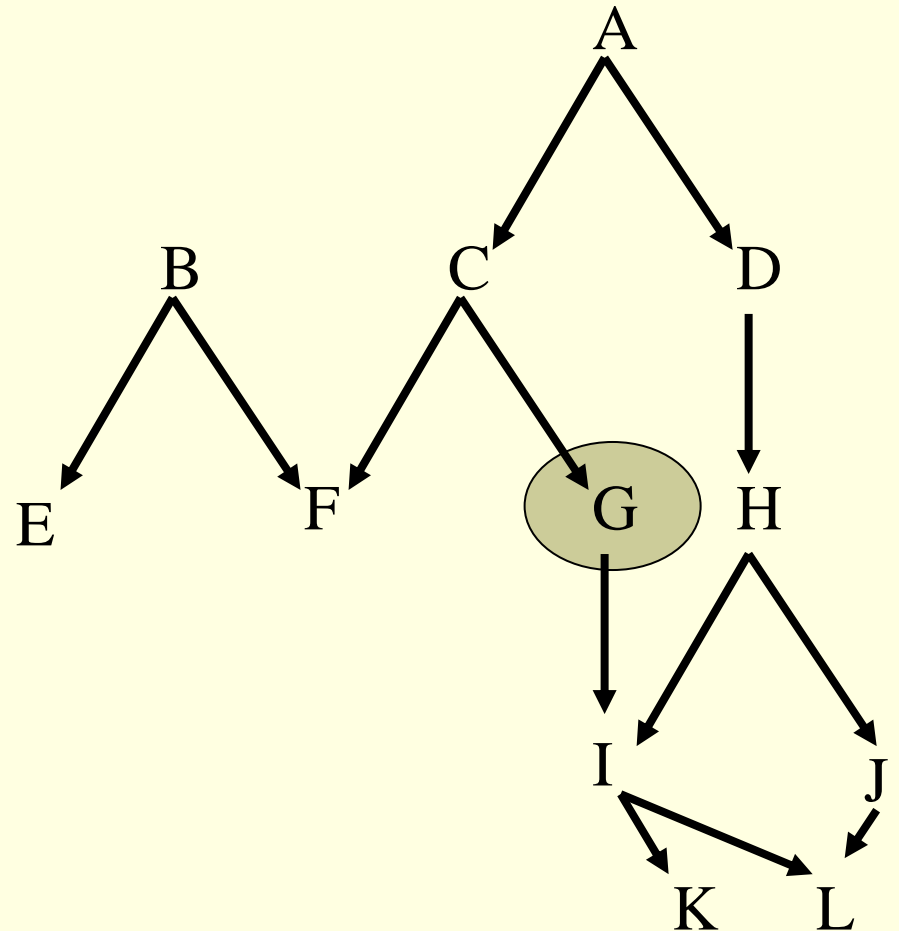
# Max-Min Parents and Children Algorithm

---

- Max-Min Parents and Children will find the true set of parents and children, provided there is enough sample for reliable statistics and the data generating procedure is faithful to some BN.

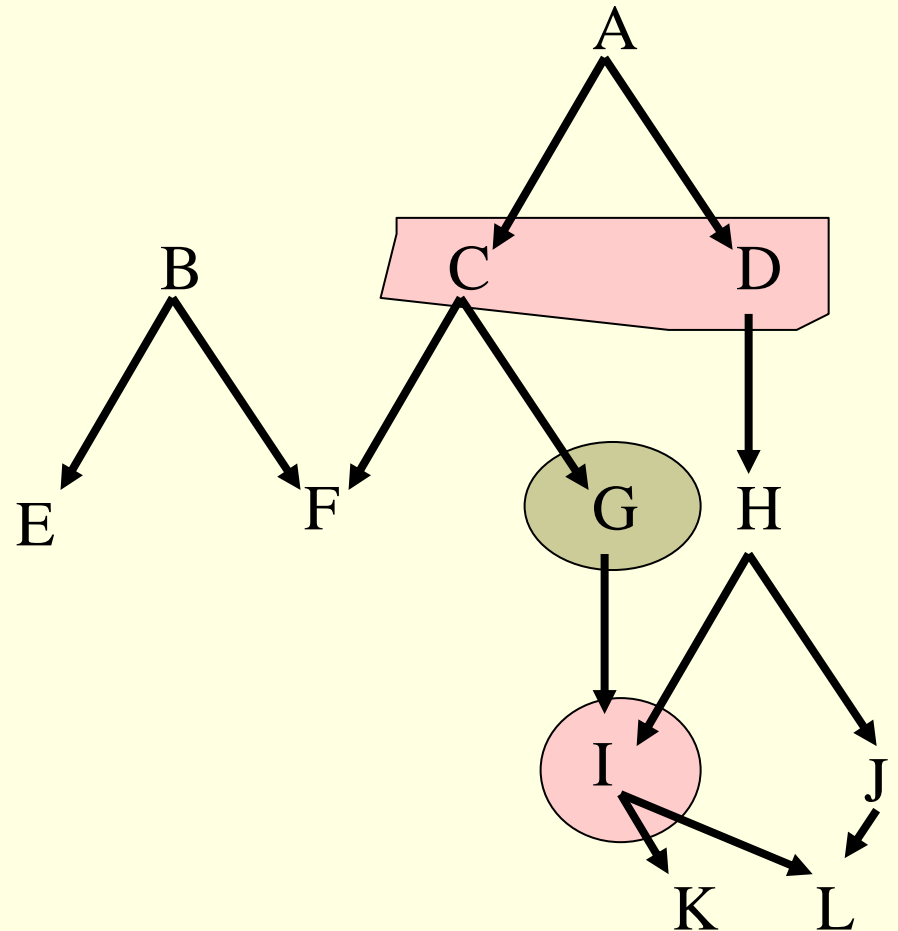
# Discovering the Markov Blanket

- Find  $PC(G)$



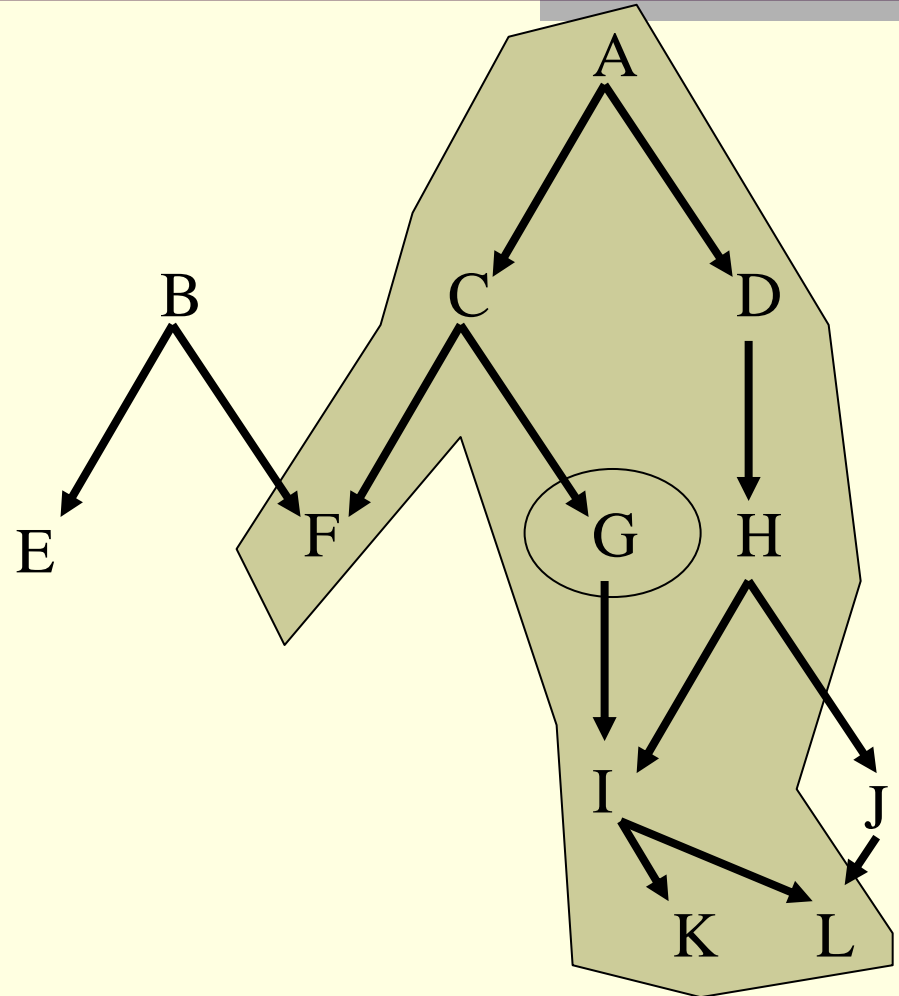
# Discovering the Markov Blanket

- Find  $PC(G)$
- Find  $PC(X)$ , for every  $X$  in  $PC(G)$



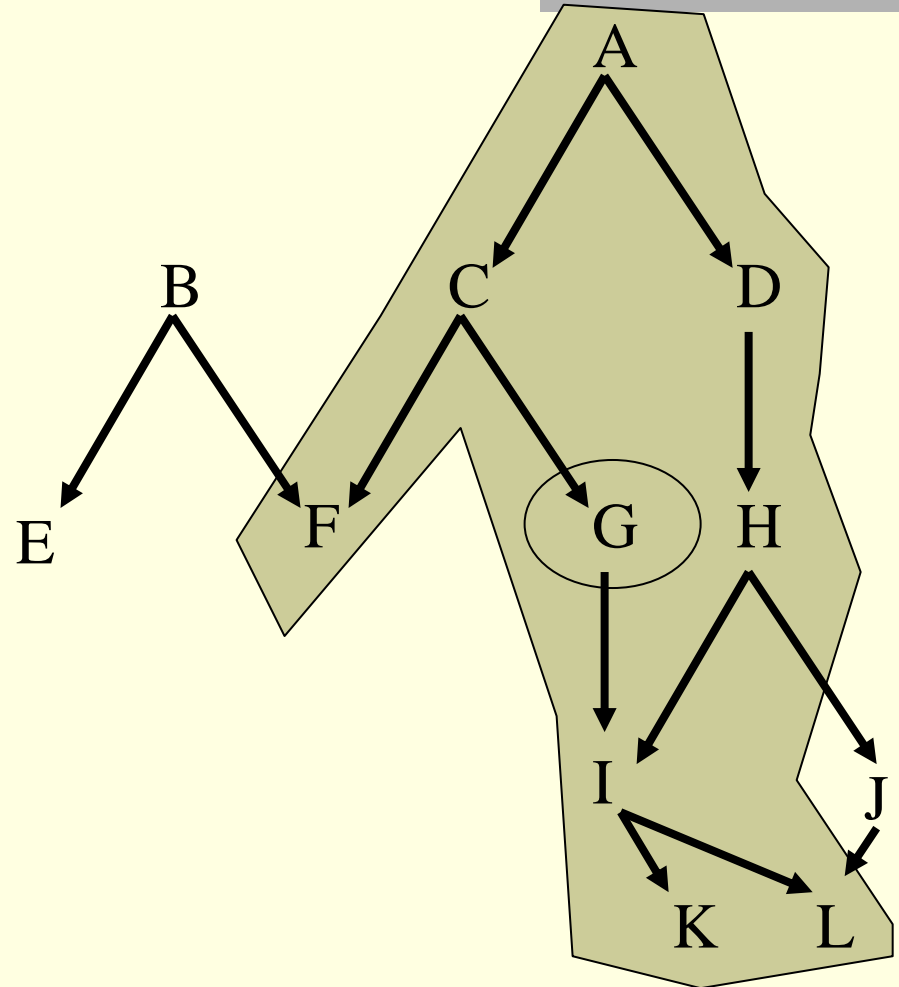
# Discovering the Markov Blanket

- Find  $PC(G)$
- Find  $PC(X)$ , for every  $X$  in  $PC(G)$



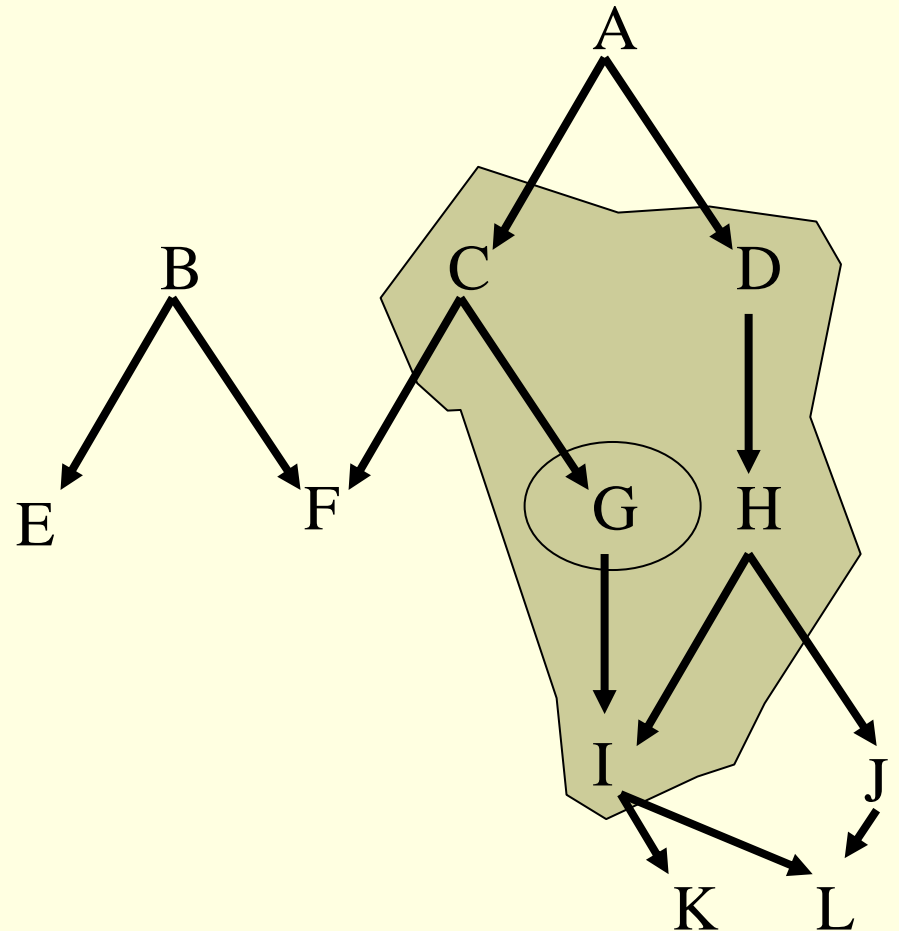
# Discovering the Markov Blanket

- Find  $PC(G)$
- Find  $PC(X)$ , for every  $X$  in  $PC(G)$
- Need to remove  $A, F, K, L$
- Keep just  $H$



# Discovering the Markov Blanket

- $H$  is the only node which is dependent with  $G$  conditioned with  $C$  on any subset of  $PC(T)$  that contains  $I$ .
- For each node  $X$  look for nodes  $Y$  (such as  $I$  in this example) such that  $Dep(X, T / Y \cup s)$ ,  $s$  subset of  $PC(T)$
- Max-Min Markov Blanket (MMMB)



# Experiments for Bayesian Network Learning

---

- Start with a known Bayesian Network
- Sample data from the distribution of the network
- Try to reconstruct
  1. The Markov Blanket of some variables.
  2. The parents and children sets of different variables
  3. The skeleton (the edges with no orientation)
  4. The full network

# Discovering the Markov Blanket

---

- Small networks
  - ALARM, 37 vars
  - Hailfinder, 56 vars
  - Pigs, 441 vars
  - Insurance, 27 vars
  - Win95Pts, 76 vars
- Large networks (tiled versions)
  - ALARM-5K (5000 vars)
  - Hailfinder-5K
  - Pigs-5K
- All variables act as targets in small networks, 10 in large networks

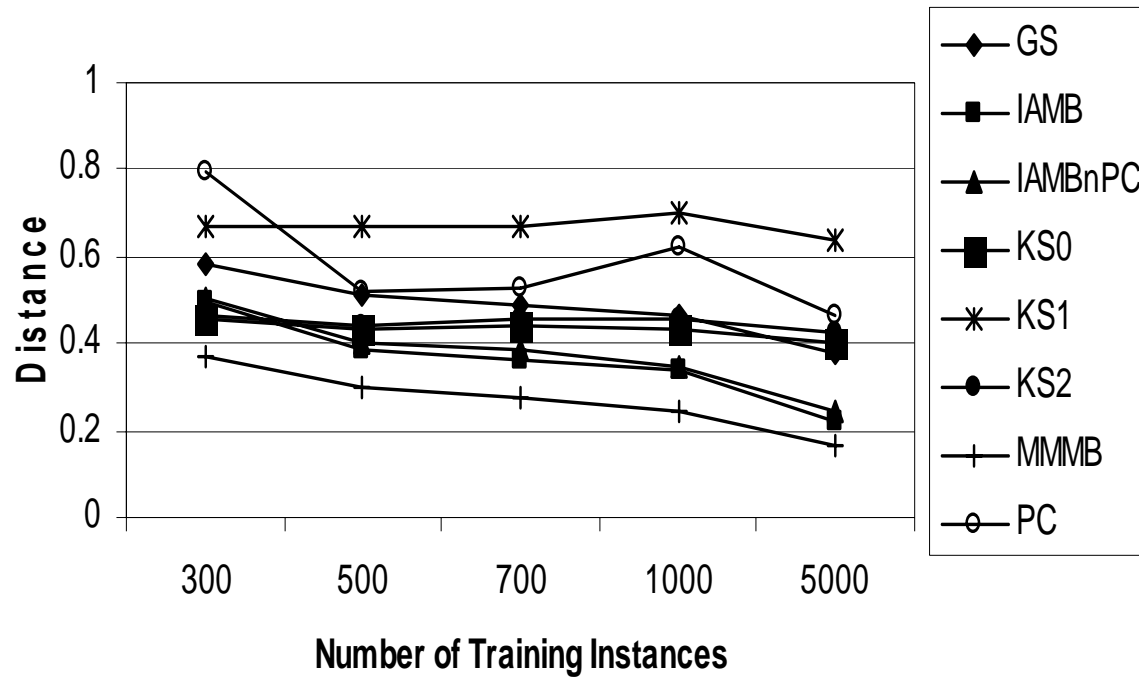
# Discovering the Markov Blanket

---

- Performance measure: Euclidean distance from perfect sensitivity and specificity in discovering  $MB(T)$  members.
- Sensitivity: percentage of True Positives (i.e., members of  $MB(T)$ ) identified as positives
- Specificity: percentage of True Negatives (i.e., non-members of  $MB(T)$ ) identified as negatives
- Algorithms PC [Spirtes, Scheines, Glymour], Koller-Sahami, Grow-Shrink [Margaritis, Thrun], Incremental Association Markov Blanket [Tsamardinos, Aliferis]

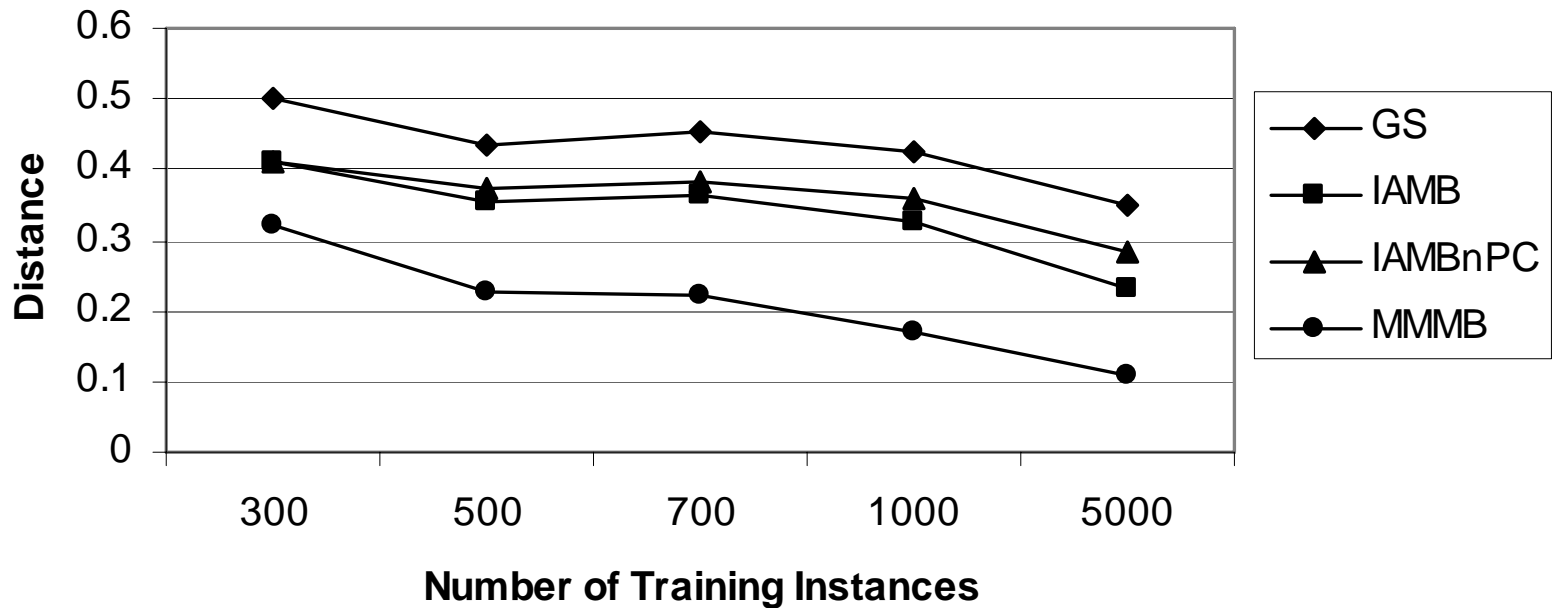
# Discovering the Markov Blanket

Comparison of MB(T) algorithms  
on the small BNs



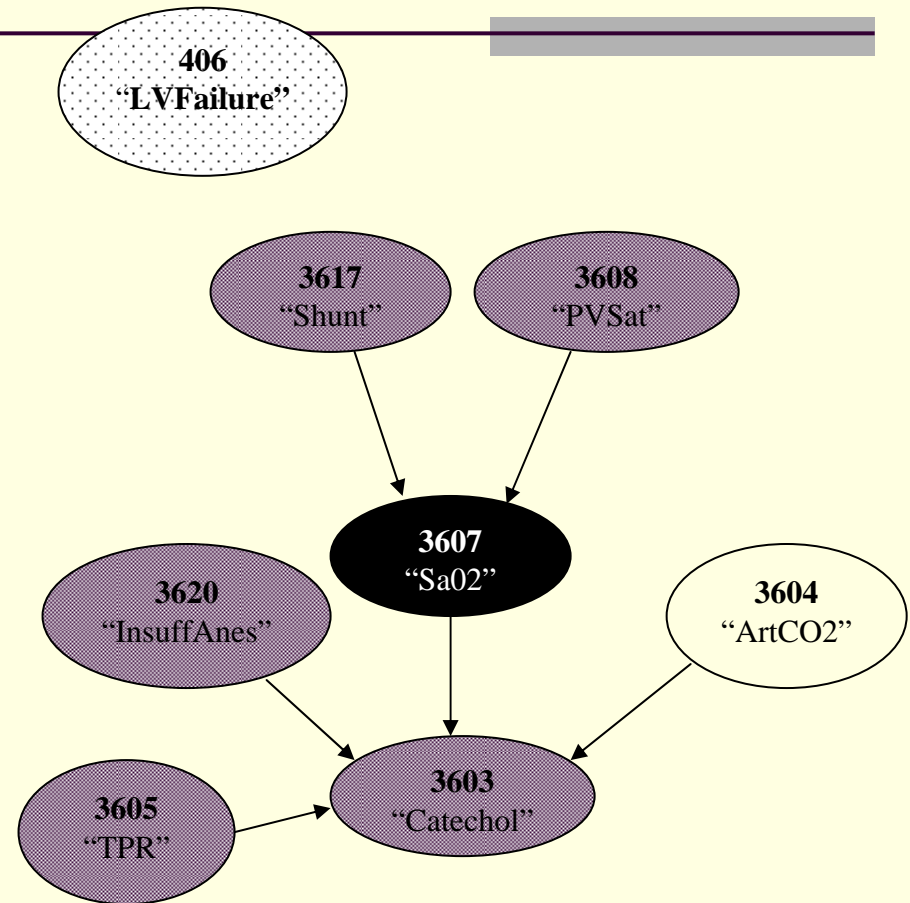
# Discovering the Markov Blanket

Comparison of MB(T) algorithms  
on the large BNs



# Discovering the Markov Blanket

- Distance of 0.1:  
sensitivity, specificity = 93%
- Distance of 0.2:  
sensitivity, specificity = 86%
- Average Distance of MMB with 5000 samples: 0.1
- Average Distance of MMB with 500 samples 0.2
- Example:  
distance=0.16, ALARM-5K, 5000 sample size



# Experiments for Variable Selection

---

- HITON:
  - Find  $MB(T)$  (in a similar fashion with  $MMPC$  but using a simpler heuristic)
  - Then start removing variables as long as they do not affect classification performance.
  - Uses different classifiers to assess performance.
- [Aliferis, Tsamardinos, AMIA 2003]

# Experiments with HITON: Datasets

---

- Drug discovery: classification of biomolecules as binding to thrombin (hence having potential or not as anti-clotting agents) on the basis of molecular structural properties
- Clinical diagnosis of arrhythmia into 8 possible disease categories on the basis of clinical and EKG data.
- Categorization of text (Medline documents) from the OHSUMED corpus as relevant to neonatal diseases or not
- Diagnosis of squamous vs. adenocarcinoma in patients with lung cancer using oligonucleotide gene expression array data
- Diagnosis of prostate cancer from analysis of mass-spectrometry signal peaks obtained from human sera

# Experiments with HITON: Datasets

<b>Dataset</b>	Thrombin	Arrhythmia	OHSUMED	Lung Cancer	Prostate Cancer
<b>Problem Type</b>	Drug Discovery	Clinical Diagnosis	Text Categorization	Gene Expression Diagnosis	Mass-Spec Diagnosis
<b>Variables #</b>	139,351	279	14,373	12,600	779
<b>Variable Types</b>	binary	nominal/ordinal/continuous	continuous	continuous	continuous
<b>Target</b>	binary	Nominal	binary	binary	binary
<b>Sample</b>	2,543	417	5000	160	326
<b>Evaluation metric</b>	ROC AUC	Accuracy	ROC AUC	ROC AUC	ROC AUC
<b>Design</b>	1-fold c.v.	10-fold c.v.	1-fold c.v.	5-fold c.v.	10-fold c.v.

# Experiments with HITON

---

- Classifiers used: linear and poly SVMs, KNN, Neural Networks, Decision Trees, Simple Base Classifier
- Variable Selection Baselines: Univariate Association Filtering, Recursive Feature Elimination, Specialized methods for text categorization, Backward/Forward Wrapping
- Evaluation Metric: Area Under the ROC curve or accuracy

# Drug Discovery: Thrombin

	UAF*	RFE	HITON	ALL
SVM	96.12%	93.29%	93.23%	93.69%
KNN	87.25%	89.71%	92.23%	88.21%
NN	<i>N/A</i>	92.04%	92.65%	<i>N/A</i>
Average	91.69%	91.68%	<b>92.7%</b>	90.95%
# of variables	34837	8709	<b>32</b>	139351

# Clinical Diagnosis: Arrhythmia

	UAF*	B/F*	HITON*	ALL*
DTI	73.94%	72.85%	71.87%	73.94%
KNN	63.22%	63.45%	65.30%	63.22%
NN	58.29%	60.90%	60.38%	58.29%
Average	65.15%	65.73%	<b>65.85%</b>	65.15%
# of variables	279	96	<b>63</b>	279

# Text Categorization: OHSUMED

	IG	$\chi^2$	HITON	ALL*
SVM	82.43%	85.91%	82.85%	90.50%
SBCtc	84.18%	86.23%	85.10%	84.25%
KNN	75.55%	81.76%	80.25%	77.56%
NN	82.47%	85.27%	83.97%	N/A
Average	81.16%	<b>84.79%</b>	83.04%	84.10%
# of variables	224	112	<b>34</b>	14373

# Gene Expression Diagnosis: Lung Cancer

	UAF*	RFE*	HITON*	ALL*
SVM	99.32%	98.57%	97.83%	99.07%
NN	99.63%	98.70%	98.92%	N/A
KNN	95.57%	91.49%	96.06%	97.59%
Average	98.17%	96.25%	97.60%	<b>98.33%</b>
# of variables	330	19	<b>16</b>	12,600

# Mass Spectrometry Diagnosis: Prostate Cancer

	UAF*	RFE*	HITON*	ALL*
SVM	98.50%	98.95%	99.10%	99.40%
NN	98.62%	98.78%	97.95%	99.27%
KNN	77.52%	86.53%	91.36%	76.94%
Average	91.55%	94.75%	<b>96.14%</b>	91.87%
# of variables	706	87	<b>16</b>	779

# Averages Over All Tasks

	Av. over Baseline Algorithms	HITON	ALL
Av. Perf. over classifiers	86.1%	<b>87.1%</b>	86.1%
Av. variable #	4,540	<b>32.3</b>	33,476
Av. reduction	x 8	x <b>1124</b>	x 1

# Discovering the Skeleton of a Bayesian Network

---

- Dataset: Tiled ALARM with 10,000 variables
- Training size: 1000 instances
- Algorithm: MMPC for each variable
  
- Results
- Sensitivity: 81%
- Specificity: 99%
- Time: 62hours, 2.4GHz Pentium IV
- Largest BN ever reconstructed
- Nothing to compare with on such a large dataset

# Reconstructing the Full Bayesian Network

---

- Algorithm Max-Min Hill Climbing:
  - MMPC with target every node to discover the edges of the network.
  - Then, search-and-score with hill-climbing to orient the edges found.
- Comparison with the Sparse Candidate (similar idea of constraining the search). The most prominent BN learning algorithm that scales up to hundreds of variables
- Measures of Comparison: BDeu score (probability of the BN given the data), number of structural errors

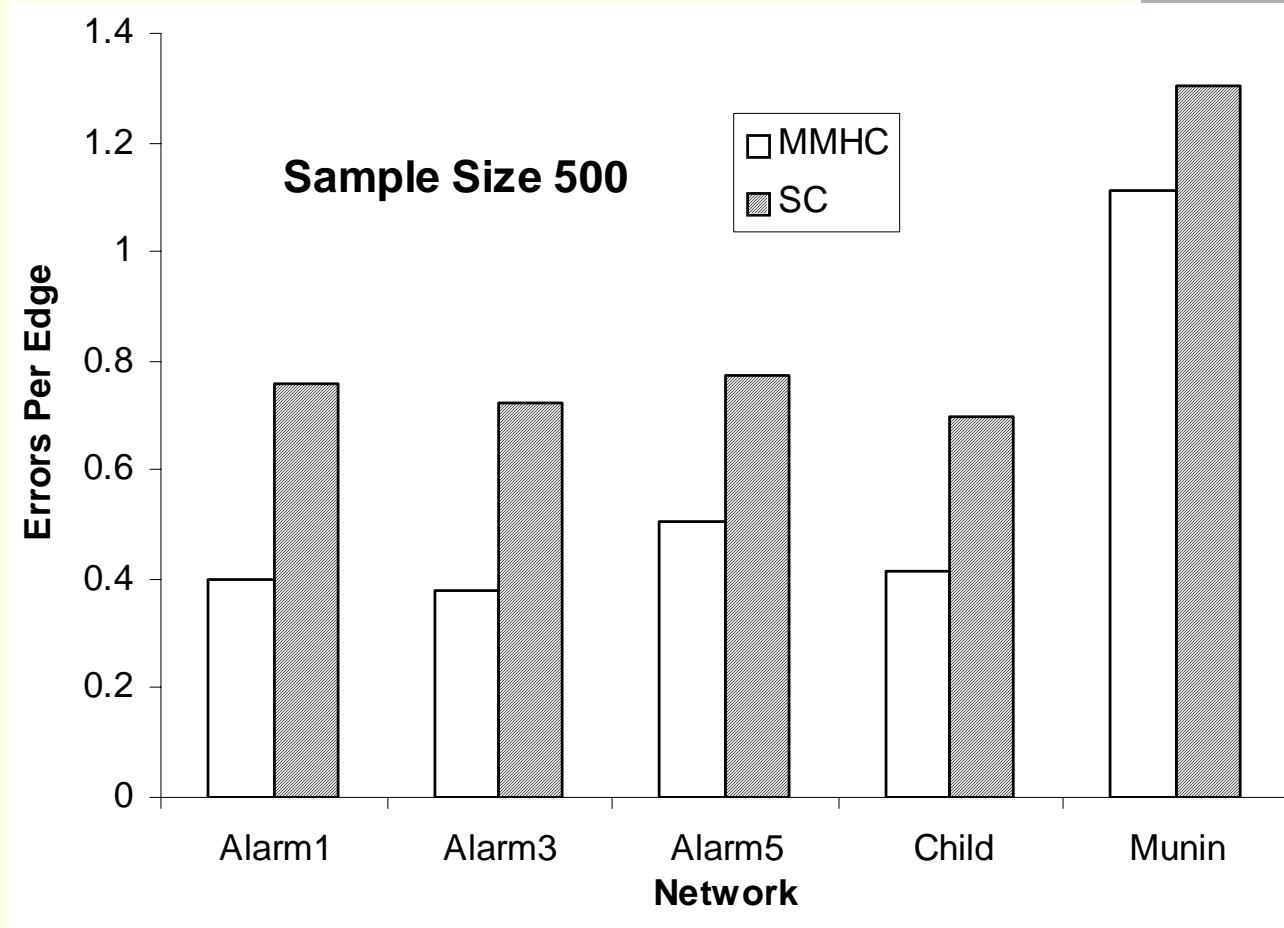
# Bayesian Networks

<b>Name</b>	<b># Vars</b>	<b># Edges</b>	<b>Description</b>
Alarm	37	46	Intensive Care Monitoring
Alarm3	111	149	Tiled version of Alarm
Alarm5	185	265	Tiled version of Alarm
Child	20	25	Diagnosis of “Blue babies”
Munin	189	282	electromyography assistant application
Gene	801	972	Learned by Sparse Candidate from gene expression data

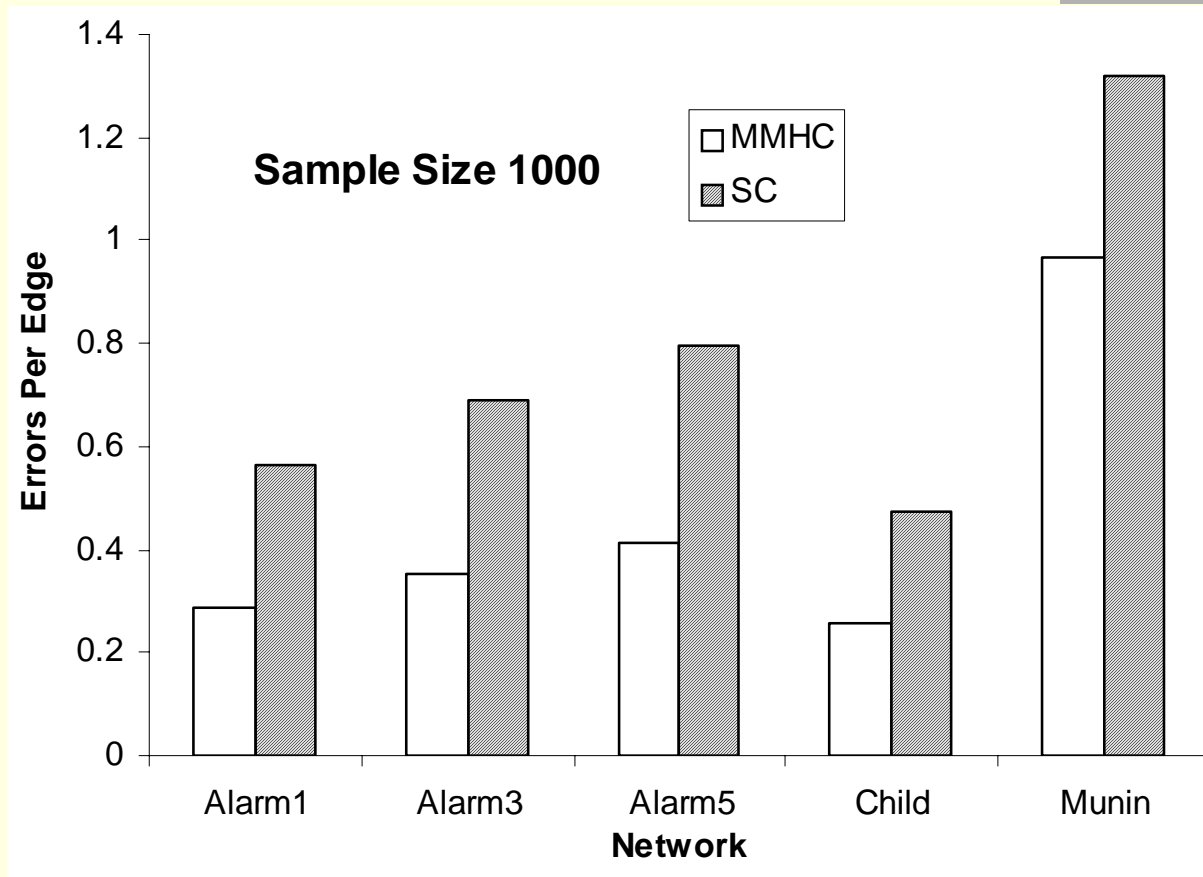
# MMHC versus Sparse Candidate

		BDeu Score		Structural Errors		Time in Seconds		
Network		MM-HC	Best of SC	MM-HC	Best of SC	MM-HC	SC k=5	SC k=10
500 Sample	Alarm	<b>-17.6</b>	-17.8	<b>18.4</b>	34.8	<b>5.8</b>	8.2	78.4
	Alarm3	<b>-59.7</b>	-60.3	<b>56.2</b>	107	<b>31.1</b>	175.2	272.3
	Alarm5	<b>-100.7</b>	-102.1	<b>133.8</b>	205	<b>77.6</b>	779.8	874.3
	Child	<b>-19.1</b>	-21.6	<b>10.4</b>	17.4	6.7	<b>2.2</b>	26.8
	Munin	-91.1	<b>-91</b>	<b>313.6</b>	367	4.4K	<b>1.3K</b>	N/A
5000 Sample	Alarm	<b>-14.2</b>	-14.3	<b>5.4</b>	22.2	<b>17</b>	42.4	110
	Alarm3	<b>-51</b>	-51.6	<b>39</b>	91	<b>92.5</b>	1.4K	1416
	Alarm5	<b>-86.7</b>	-87.5	<b>78</b>	172.3	<b>222.2</b>	6.6K	6.3K
	Child	<b>-17.7</b>	-21	<b>3.8</b>	10	47.5	<b>13</b>	43.6
	Munin	<b>-64.7</b>	-66.2	<b>238</b>	349.8	<b>4.9K</b>	12K	N/A
	Gene	<b>-634.8</b>	-640.7	<b>62.5</b>	93	<b>14K</b>	682K	N/A

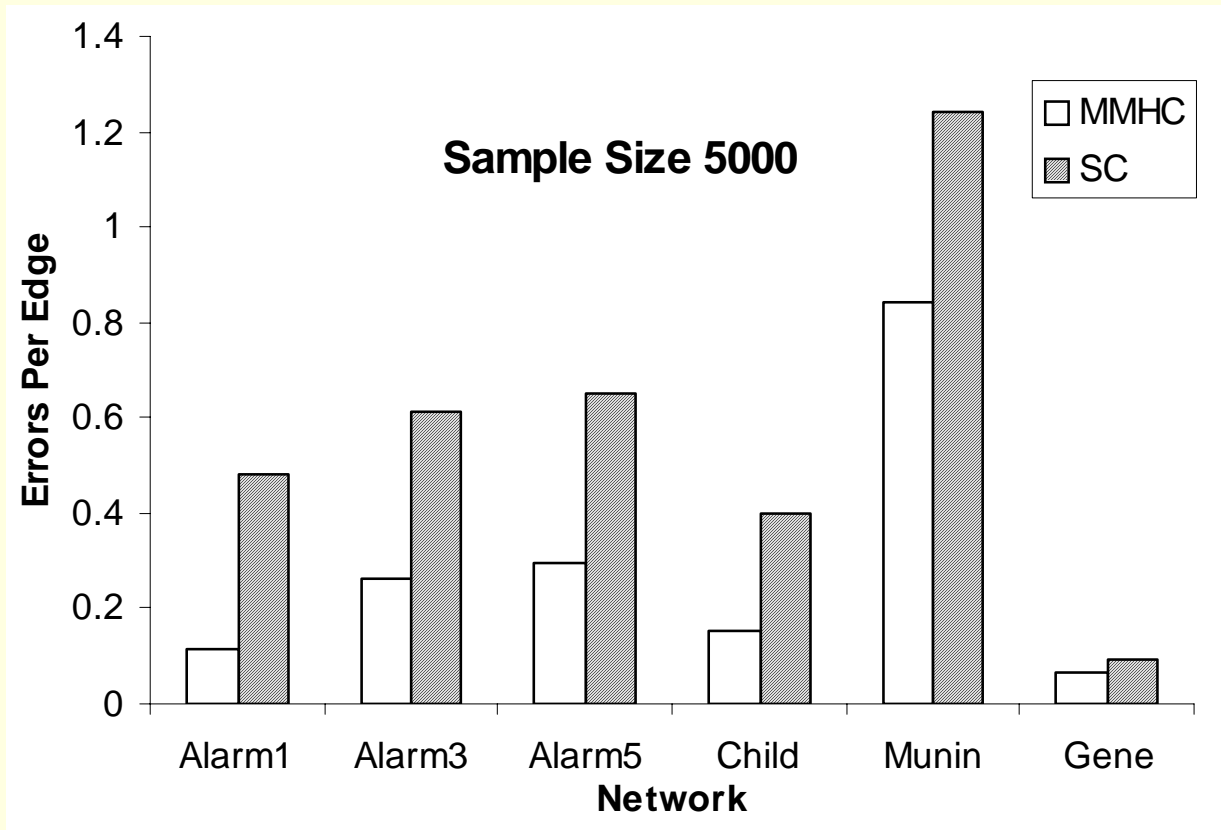
# MMHC versus Sparse Candidate



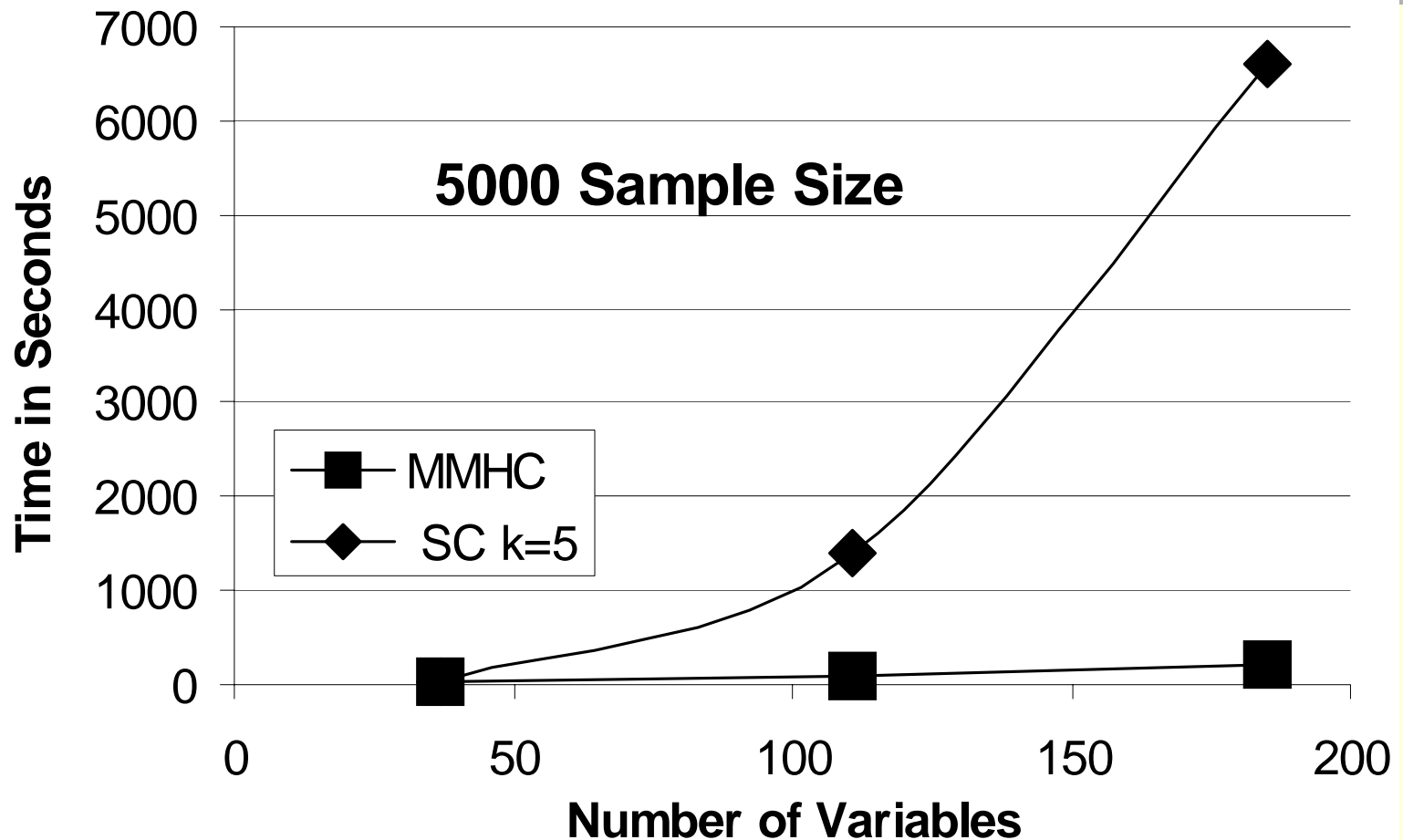
# MMHC versus Sparse Candidate



# MMHC versus Sparse Candidate



# MMHC versus Sparse Candidate



# Discussion

---

- Bayesian Networks: intuitive language for reasoning and representing causality.
- We can learn parents and children of a node (i.e., everything directly related to the node)
- We can learn Markov Blankets (minimal sets of predictors)
- We can learn the full structure efficiently and accurately.
  
- Causal discovery has not been verified with experiments.
- Causal discovery has incredible potential!
- Many theoretical problems in causal discovery still open:
  - Feedback cycles, time-stamped data.
  - Faithfulness: theoretically almost surely holds in the sample limit. What about finite sample?

# Future Work

---

- Improve efficiency. Goal: fastest BN learning, increase speed by 2 orders of magnitude.
- Improve quality
  - Use better statistics for the conditional independence tests
  - Use statistical theory for multiple tests adjustment (False Discovery Rate)
  - Discovery without assuming faithfulness
- Discovery with hidden variables
- Discovery with timed-stamped data
- Discovery with more expressive models (e.g. cyclic models)

# People in the Discovery Systems Lab

---

## Faculty

- Constantin Aliferis, M.D.,  
Ph.D., Assistant Professor,  
Director of DSL

## Students

- Yin Aphinyanaphongs
- Laura E. Brown
- Nafeh Fananapazir
- Lawrence Fu
- Alexander Statnikov  
(student/programmer)
- Firas Wehbe

## ■ Collaborators

- Douglas Fisher,
  - Ph.D. CS
- Douglas Hardin,
  - Ph.D., Math
- Pierre Massion,
  - M.D, medicine
- Trent Rosenbloom,
  - M.D., M.P.H., Biomedical Informatics

# Papers, Software, Information

---

- <https://discover1.mc.vanderbilt.edu/discover/public/>
- (or from Google: “Discovery Systems Laboratory”)
- (or from the DBMI web site under “projects”)

# Software Available

---

- Causal Explorer: Standard and early algorithms for causal discovery.
- MultiCat SVM library: a library with all major multiclass SVM methods + scripts for automating large scale experimentation
- Tiling Tool: algorithm for tiling copies of a small BN to create larger BNs sharing the same structural and probabilistic properties.
- By request: all algorithms described in our papers

# Other Projects in the Discovery Systems Laboratory

---

- Learning with Mixtures of Observational and Experimental Data (Tsamardinos, Aliferis)
- Improve efficiency and quality of BN learning (Tsamardinos, Aliferis)
- Learn clinical guidelines from data, merging, execution, optimization of clinical guidelines using AI planning (Tsamardinos)



Lawrence Fu

Laura Brown

Firas Webhe

# Other Projects in the Discovery Systems Laboratory

---

- Compare efficiency and quality of Multi-Category Support Vector Machines (Tsamardinos, Aliferis)
- Develop models for predicting clinical laboratory values from previous lab results and other clinical data (Aliferis, Tsamardinos, Rosenbloom)



Alexander  
Statnikov

# Other Projects in the Discovery Systems Laboratory

---

- Develop novel methods for protein marker selection from mass spectrometry data (prostate cancer) (Aliferis).



Nafeh  
Fananapazir

- Construct computer filters that automatically identify PubMed documents that belong to specific content categories (treatment, diagnosis, etiology, etc.)(Aliferis)



Yin  
Aphinyanaphongs

# Other Projects in the Discovery Systems Laboratory

---

- Analysis for prediction and causal discovery of lung cancer gene expression data (Aliferis, Statnikov, Tsamardinos, Massion)
- Connections between variable selection Support Vector Machines and Bayesian Network methods (Hardin, Tsamardinos, Aliferis)
- Variable selection with variable cost (Aliferis, Hardin, Tsamardinos)

