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

Ιωάννης Τσαμαρδίνος

Ioannis.tsamardinos@vanderbilt.edu

23 / 6 / 2004

Assistant Professor, Discovery Systems Laboratory
Department of Biomedical Informatics, Vanderbilt University

Acknowledgements

- Collaborators
  - Constantin F. Aliferis, Director DSL, Assistant Prof. DBMI
  - Douglas Hardin, Associate Prof. Mathematics
- Students
  - Laura E. Brown
  - Alexander Statnikov
- Support
  - NIH
  - Vanderbilt University

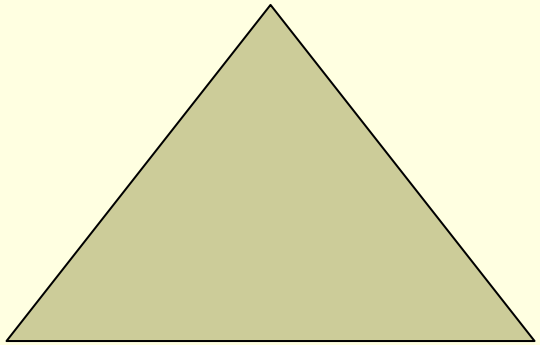
Outline

- Bayesian Networks and Bayesian Network Learning
  - Motivation/Problems
  - Theory
  - Algorithms
  - Results
- Variable Selection for Classification
- Causal Discovery

Connections

Bayesian Network Learning:

Learn the set of statistical dependencies and independencies



Variable Selection:

What is the minimal subset of variables, with the maximum predictive power

Causal Discovery:

What is causing (directly) what

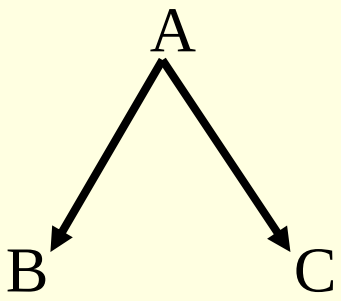
Bayesian Networks and Bayesian Network Learning

Bayesian Networks for Decision Support Systems

- A BN can provide the probability distribution of a variable given the values of any subset of other variables
- Also, it can provide the value of knowing the value of an additional variable
- BN with extensions can provide utility of decisions and help in decision making
- A number of Decision Support Systems are based on BNs.
- A domain expert used to provide the BN for the support system from his/her experience
- Modern Decision Support Systems:
  - Statistical data are gathered
  - A BN learning algorithm automatically constructs the BN from the 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/Condition
- Graph has to be DAG (directed acyclic) in the standard BN model



+

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

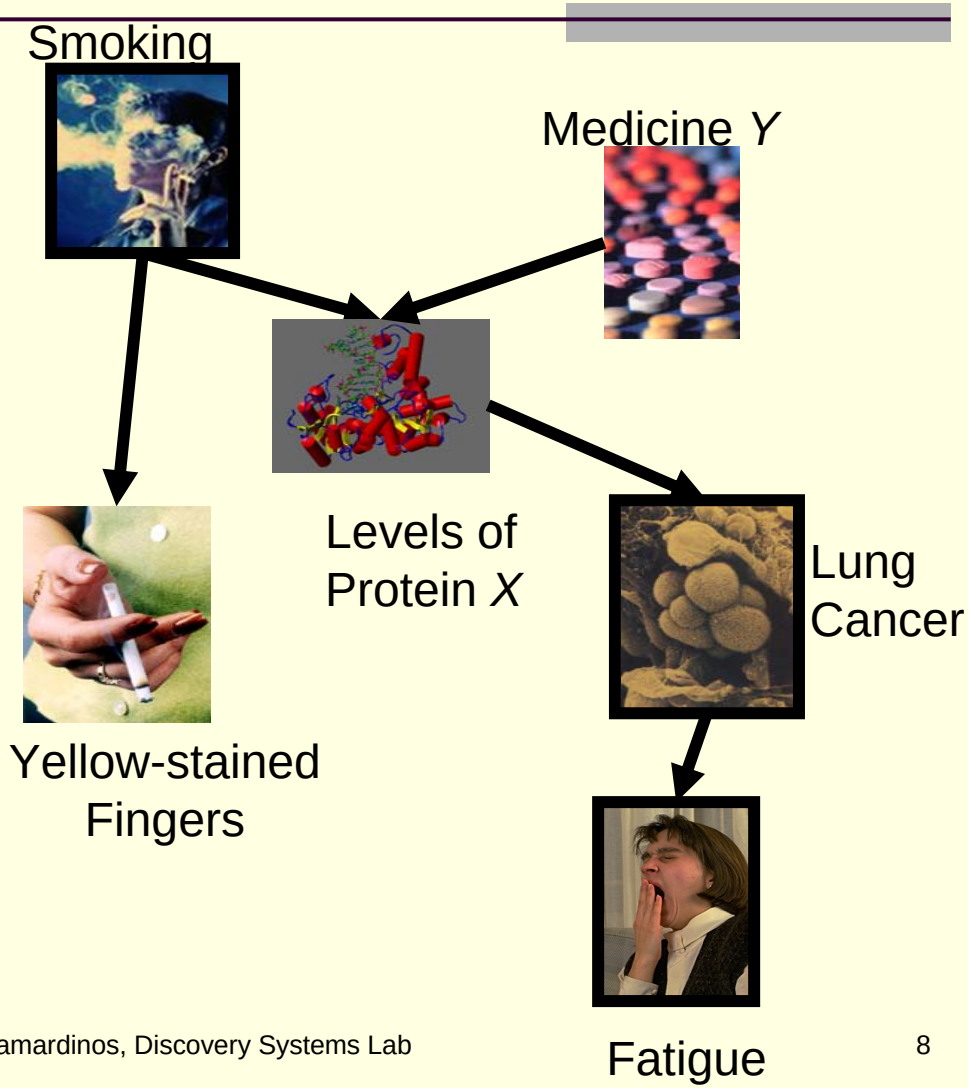
+

Markov Condition

- Any JPD can be represented in BN form

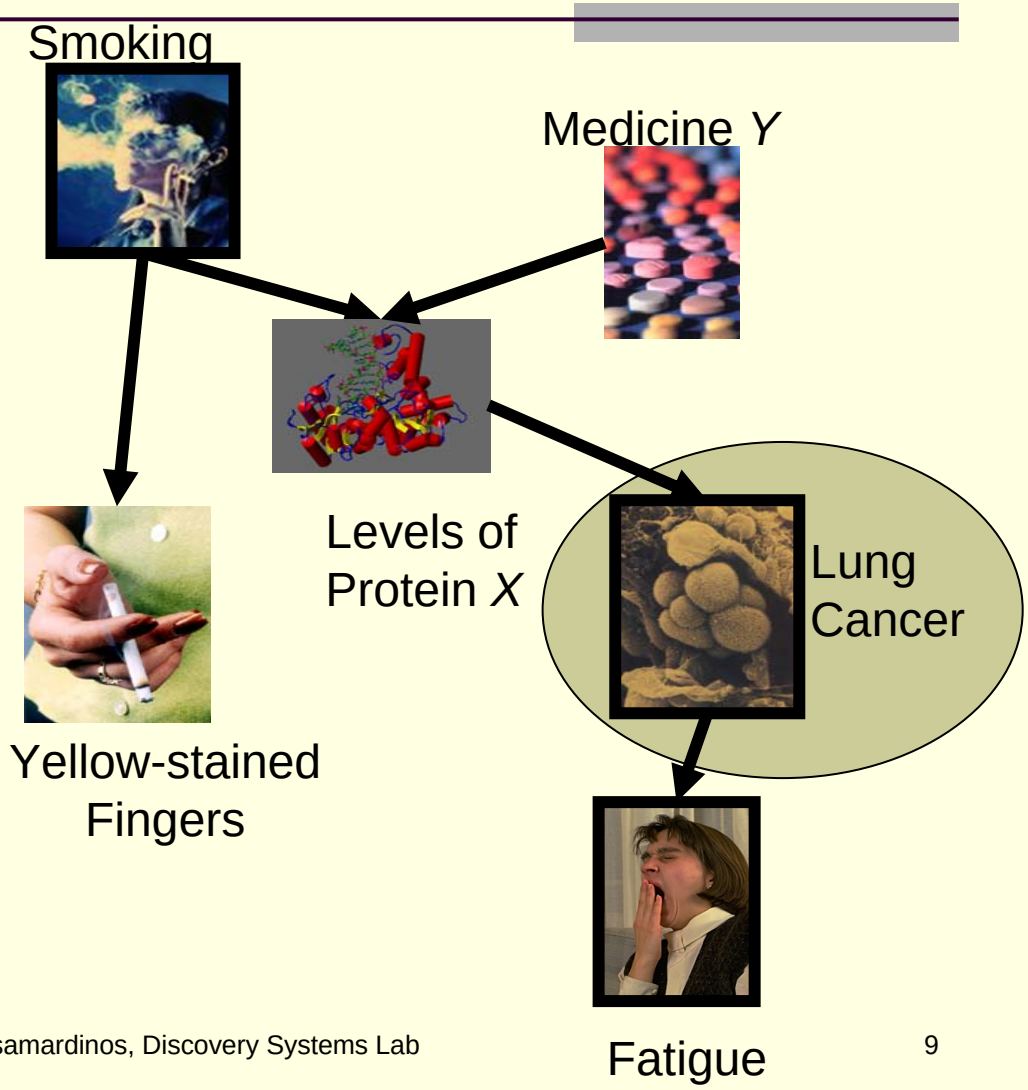
Bayesian Networks

- 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***



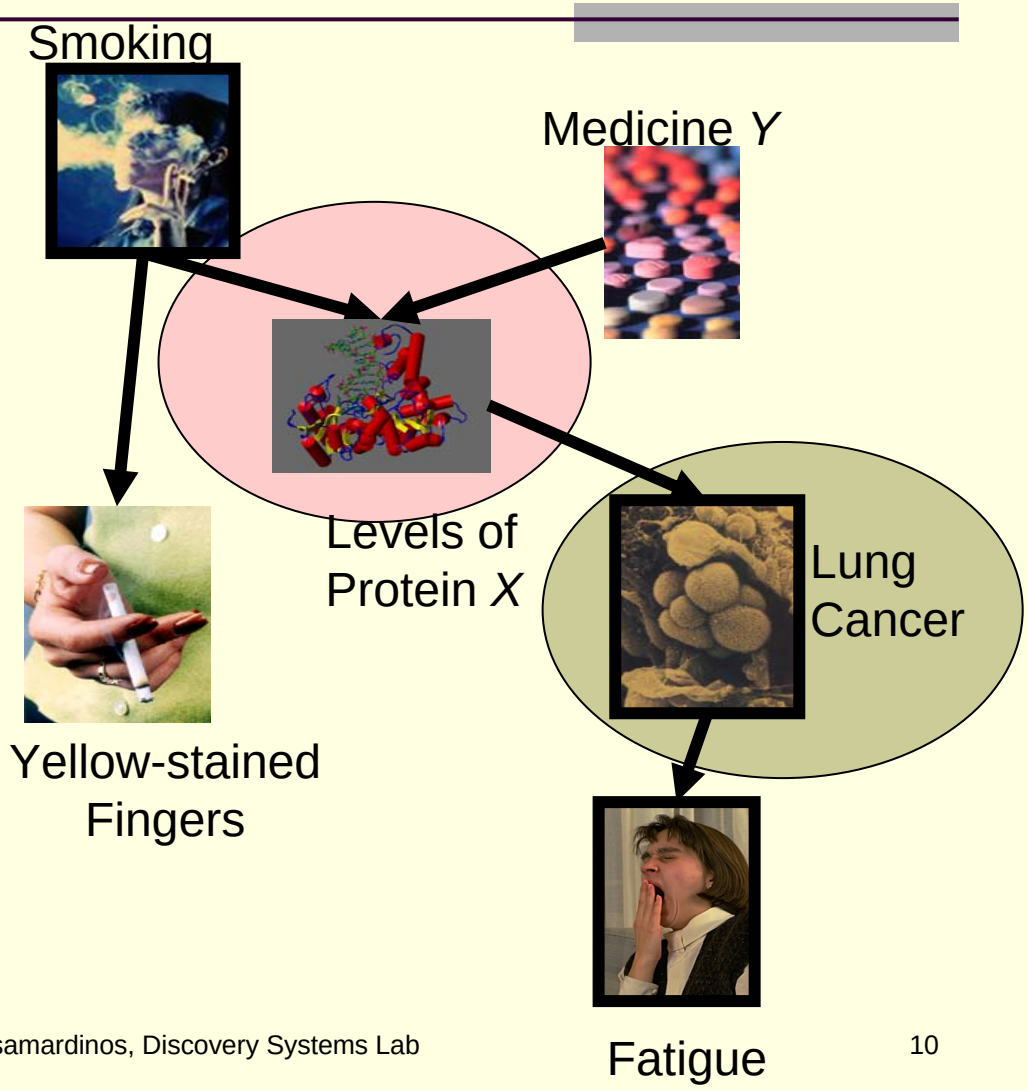
Bayesian Networks

- Markov Condition: the probability distribution of any node/variable ***N*** given its parents ***P*** is independent of any subset of the non-descendent nodes ***W*** of ***N***



Bayesian Networks

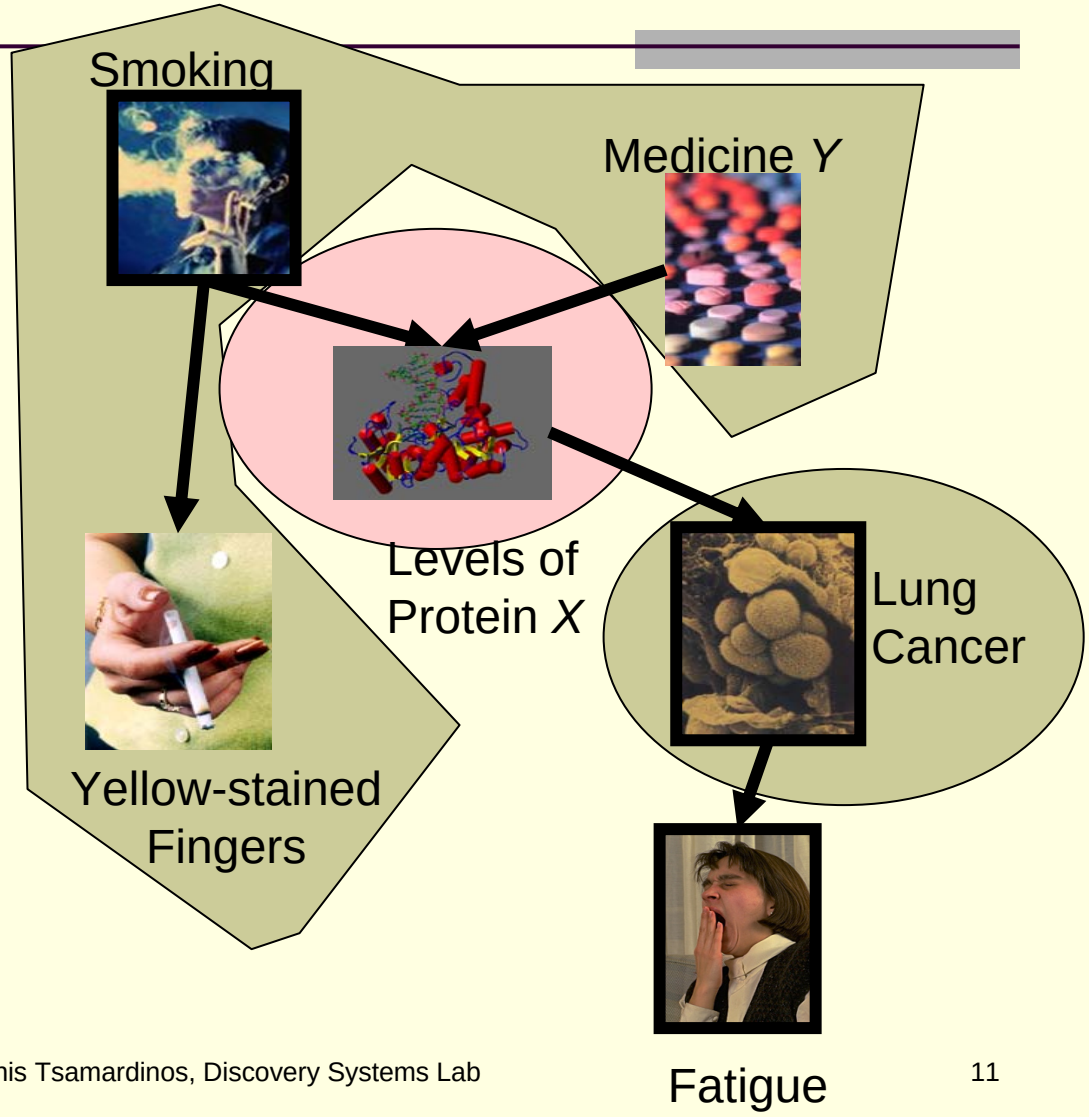
- Markov Condition: the probability distribution of any node/variable ***N*** given its parents ***P*** is independent of any subset of the non-descendent nodes ***W*** of ***N***



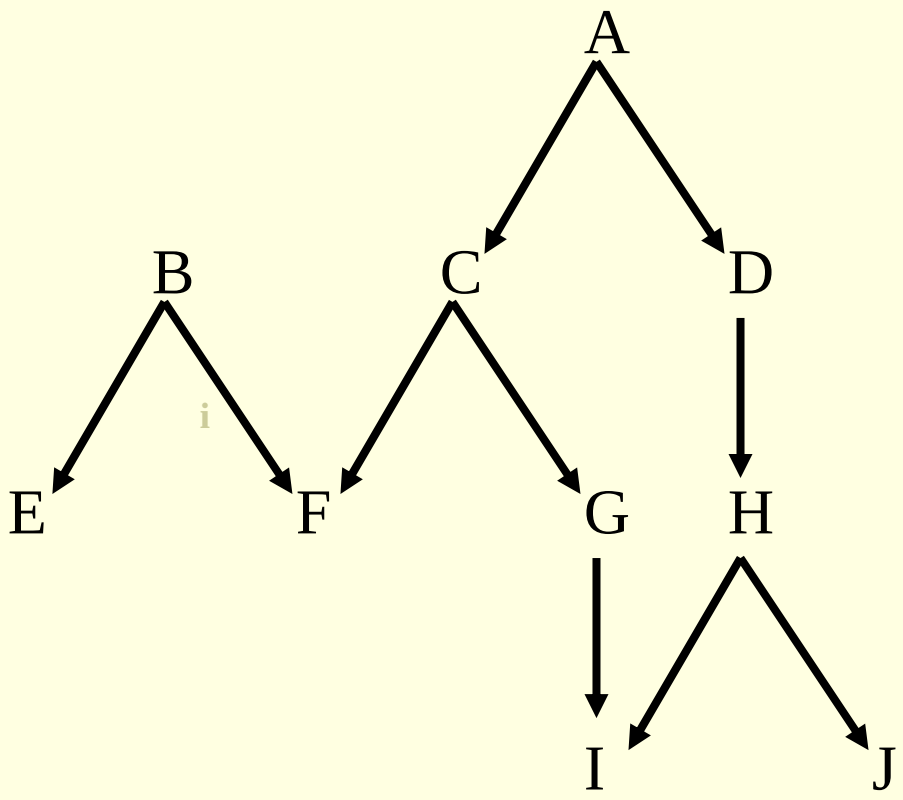
Bayesian Networks

- Markov Condition: the probability distribution of any node/variable ***N*** given its parents ***P*** is independent of any subset of the non-descendent nodes ***W*** of ***N***

Ind(Lung Cancer; Smoking | Levels of Protein X)

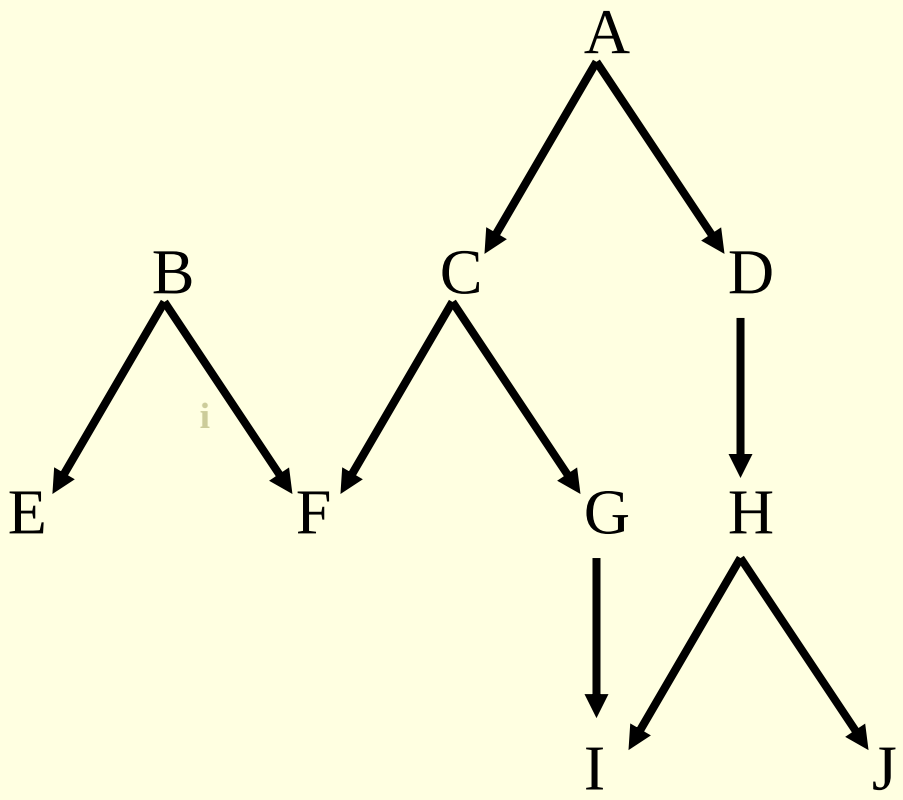


Bayesian Networks



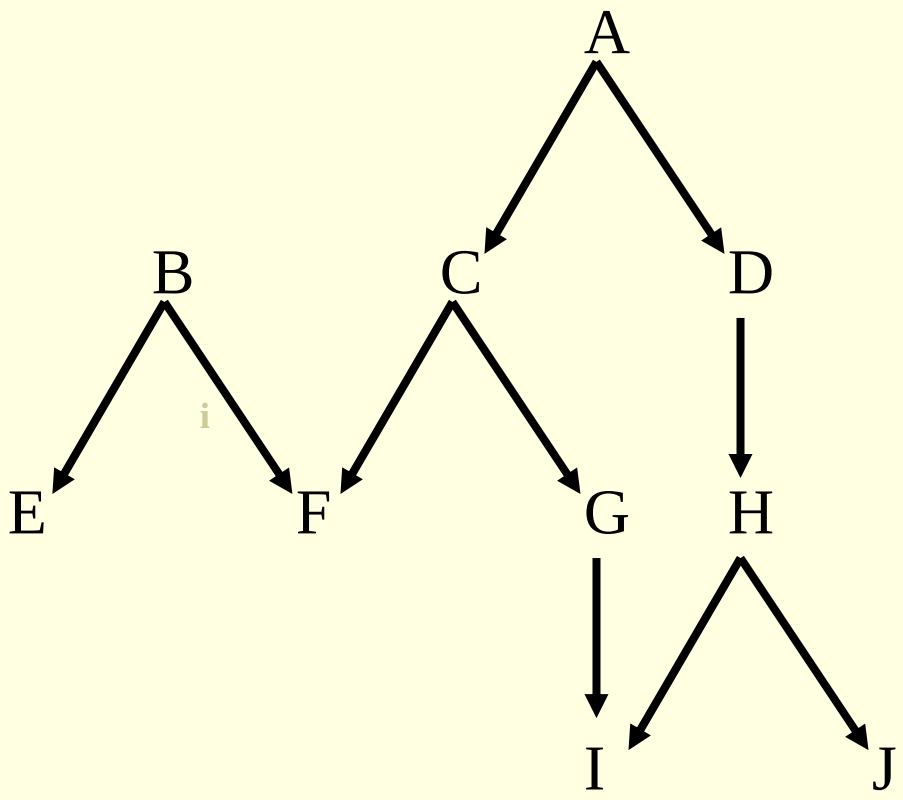
$$P(\boldsymbol{V}) = p(A,B,C,D,E,F,G,H,I,J) =$$
$$p(A) \times$$
$$p(B|A)$$
$$p(C|A,B) \times$$
$$p(D|A,B,C) \times$$
$$p(E|A,B,C,D) \times$$
$$p(F|A,B,C,D,E) \times$$
$$p(G|A,B,C,D,E,F) \times$$
$$p(H|A,B,C,D,E,F,G) \times$$
$$p(I|A,B,C,D,E,F,G,H) \times$$
$$p(J|A,B,C,D,E,F,G,H,I)$$

Bayesian Networks



$$P(\boldsymbol{V}) = p(A) \times p(B|\cancel{A}) \times p(C|A,\cancel{B}) \times p(D|A,\cancel{B},\cancel{C}) \times p(E|\cancel{A},\cancel{B},\cancel{C},\cancel{D}) \times p(F|A,B,C,D,E) \times p(G|A,B,C,D,E,F) \times p(H|A,B,C,D,E,F,G) \times p(I|A,B,C,D,E,F,G,H) \times p(J|A,B,C,D,E,F,G,H,I)$$

Bayesian Networks

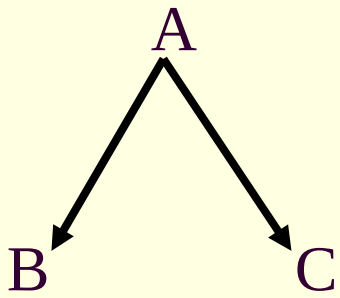


$$P(\boldsymbol{V}) = p(A|Pa(A)) \; p(B|Pa(B)) \times p(C|Pa(C)) \times p(D|Pa(D)) \times p(E|Pa(E)) \times p(F|Pa(F)) \times p(G|Pa(G)) \times p(H|Pa(H)) \times p(I|Pa(I)) \times p(J|Pa(J)) = \prod_i P(V_i \mid Pa(V_i))$$

Bayesian Networks

Variables are binary: values {+, -}

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



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)

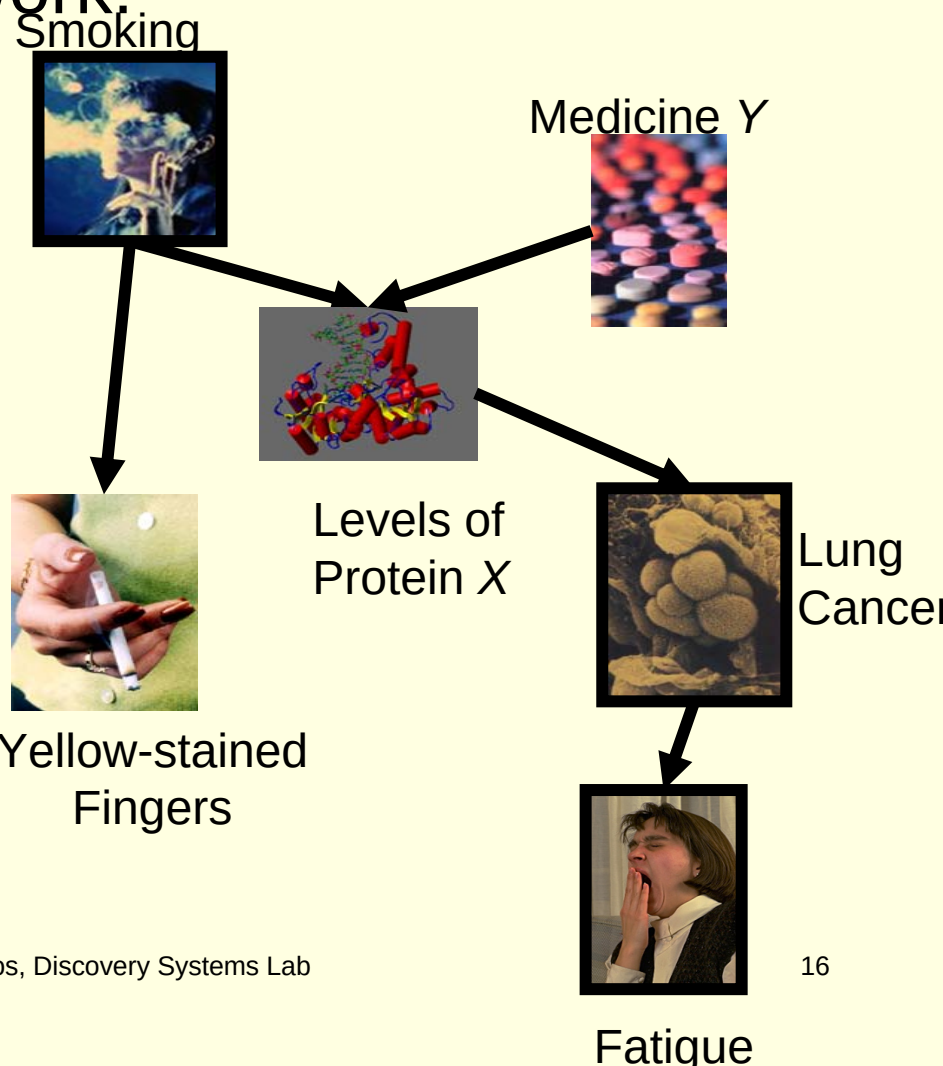
Inference in Bayesian Networks

- Known Bayesian Network:

P(Lung Cancer|Smoking, Medicine Y)?

P(Yellow Stained Fingers| Fatigue)?

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(Disease* | *Some Findings), P(Gene Level=High* | *Gene Levels)*
- Algorithms exist for probabilistic inference
  - Exact
  - Approximate
  - NP-complete in the general case

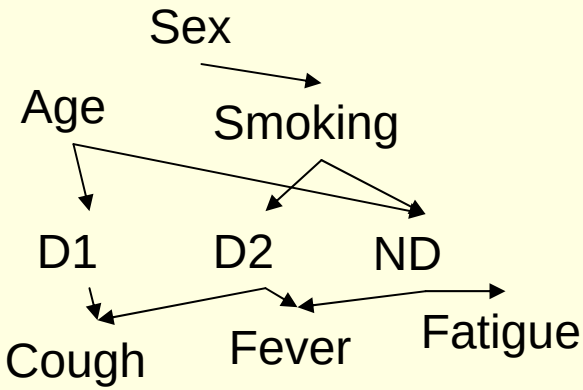
Learning Bayesian Networks

Learning Bayesian Networks from data

Given:

Patient	D1	D2	NoD	Age	Smoking	Sex	Fever	Cough	Fatigue
1	Y	N	N	24	Y	M	N	Y	N
2	N	N	Y	50	Y	F	N	N	N
…									
1000	N	N	Y	42	N	M	N	N	N

Find:

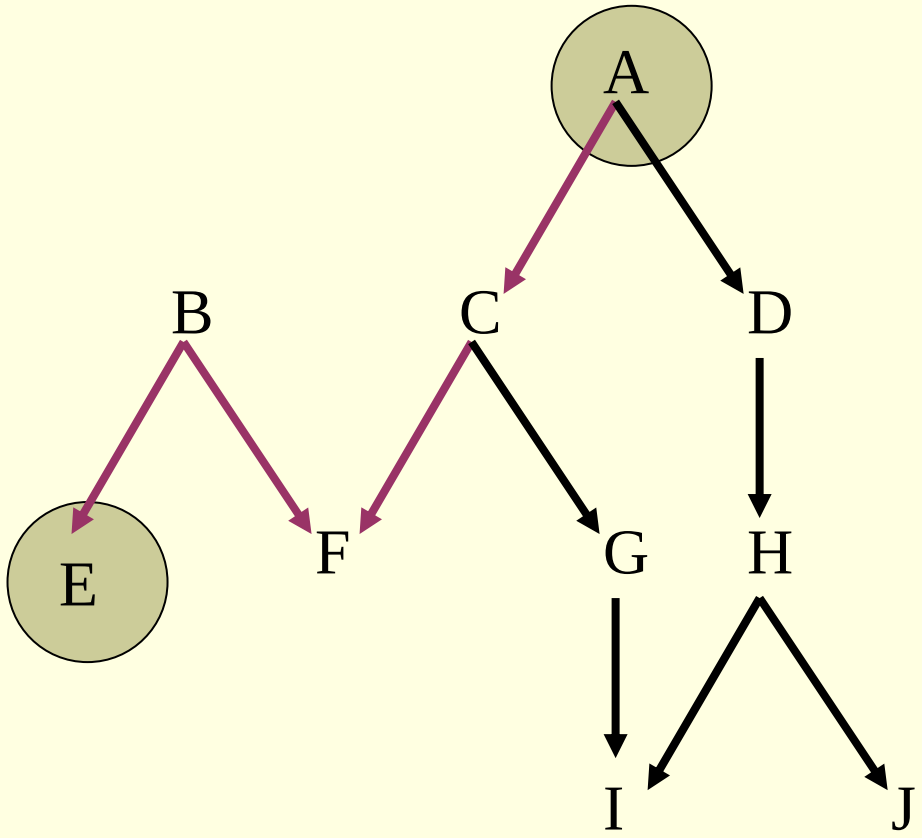


+

P(Sex=F) = 0.5
P(Sex=M)=0.5
P(Smoking=Y|Sex=M)=0.7
P(Smoking=Y|Sex=F)=0.6
….

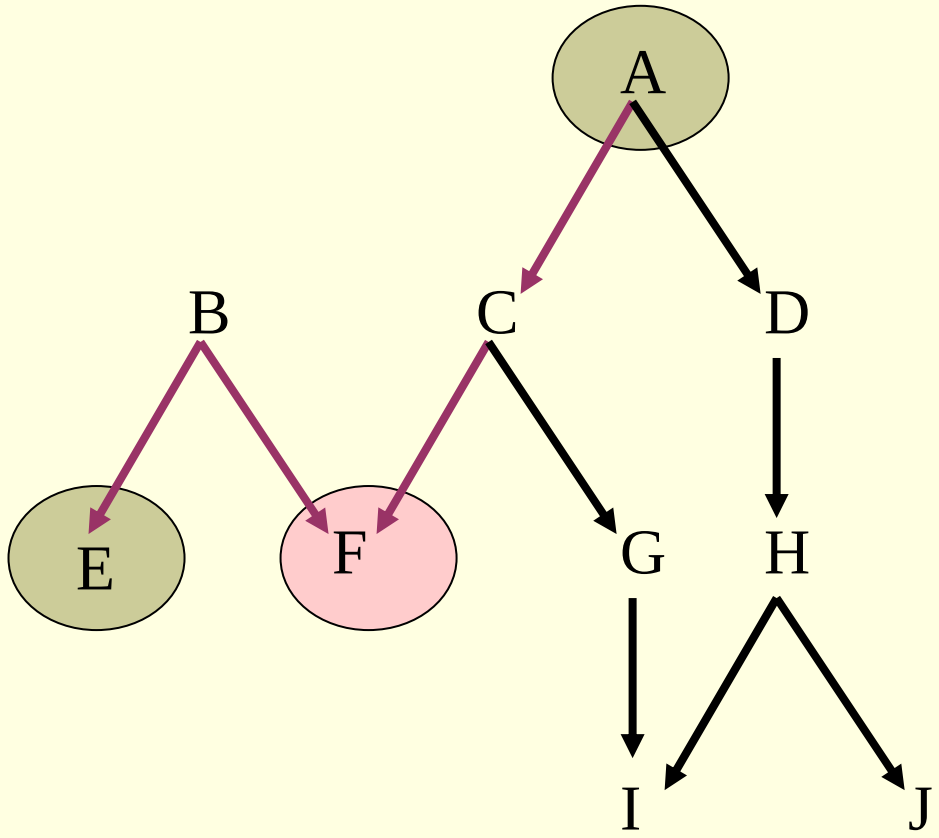
The *d*-separation criterion

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



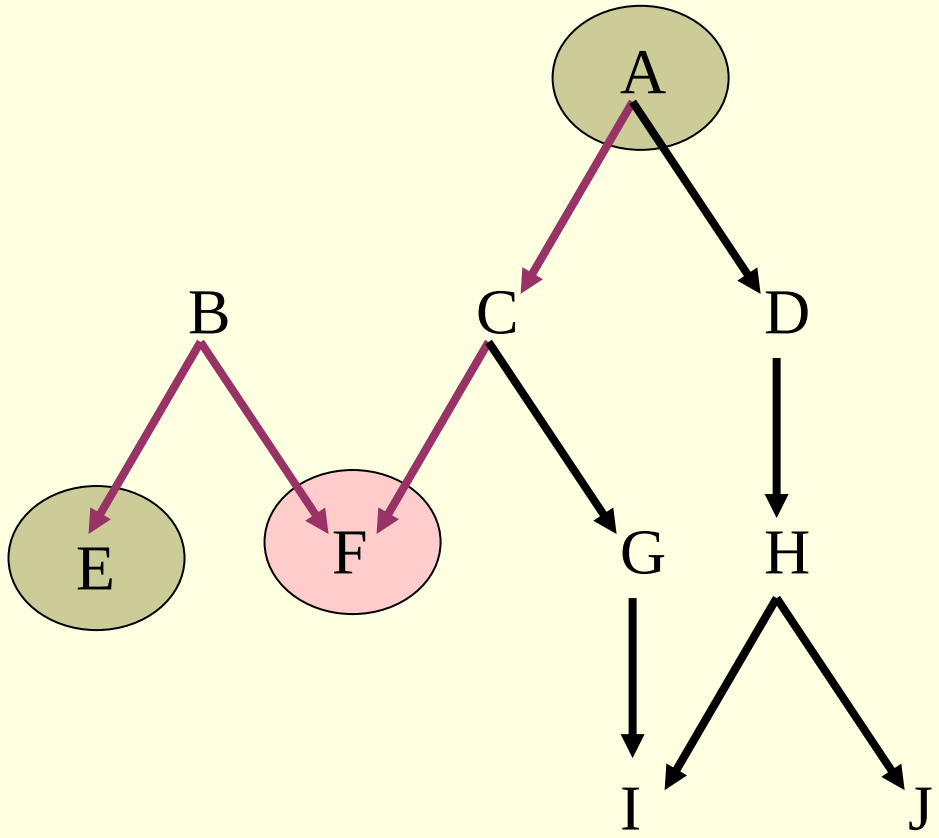
d-separation: A criterion for independence

- Collider: a node with two incoming edges



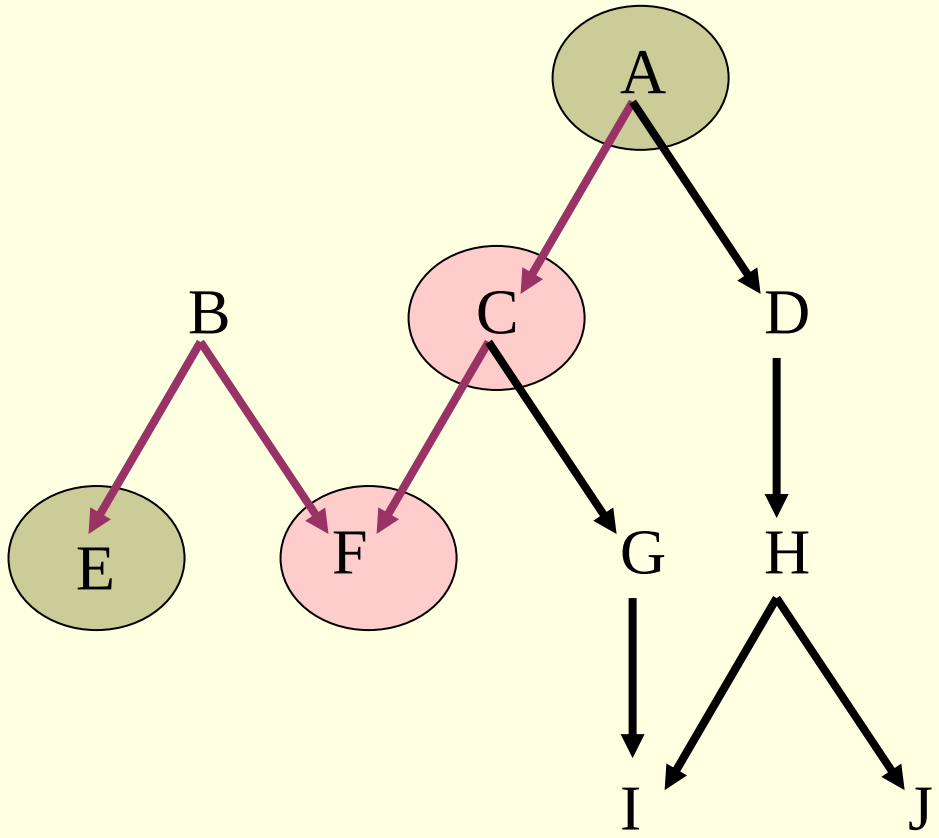
d-separation: A criterion for independence

- 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)



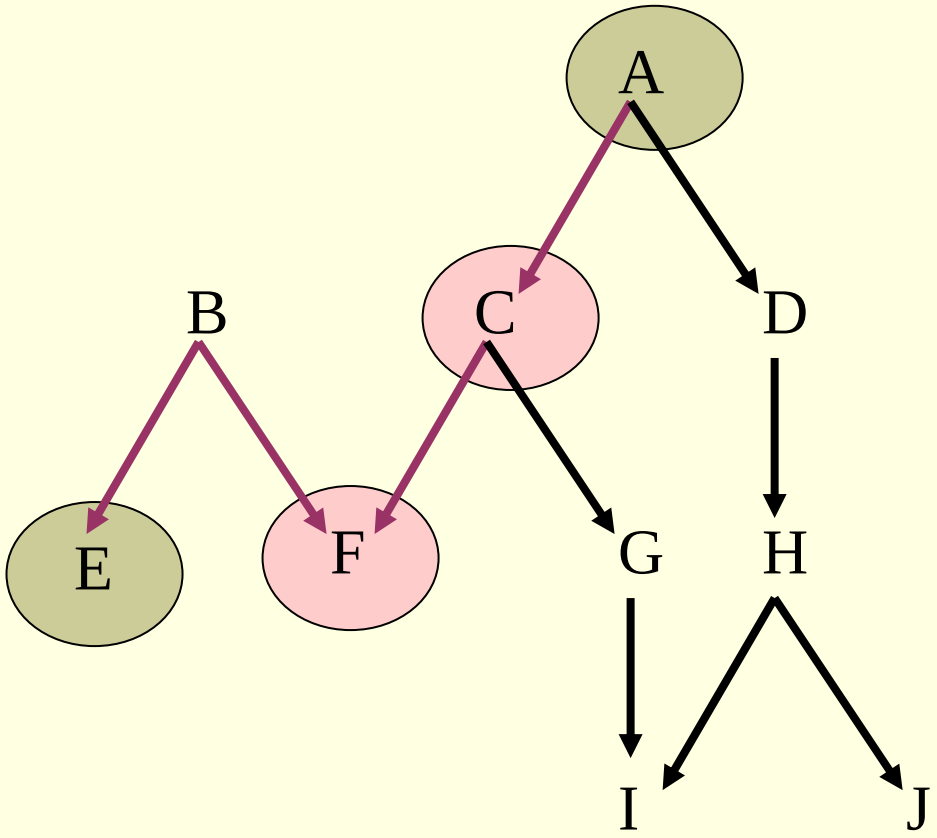
d-separation: A criterion for independence

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



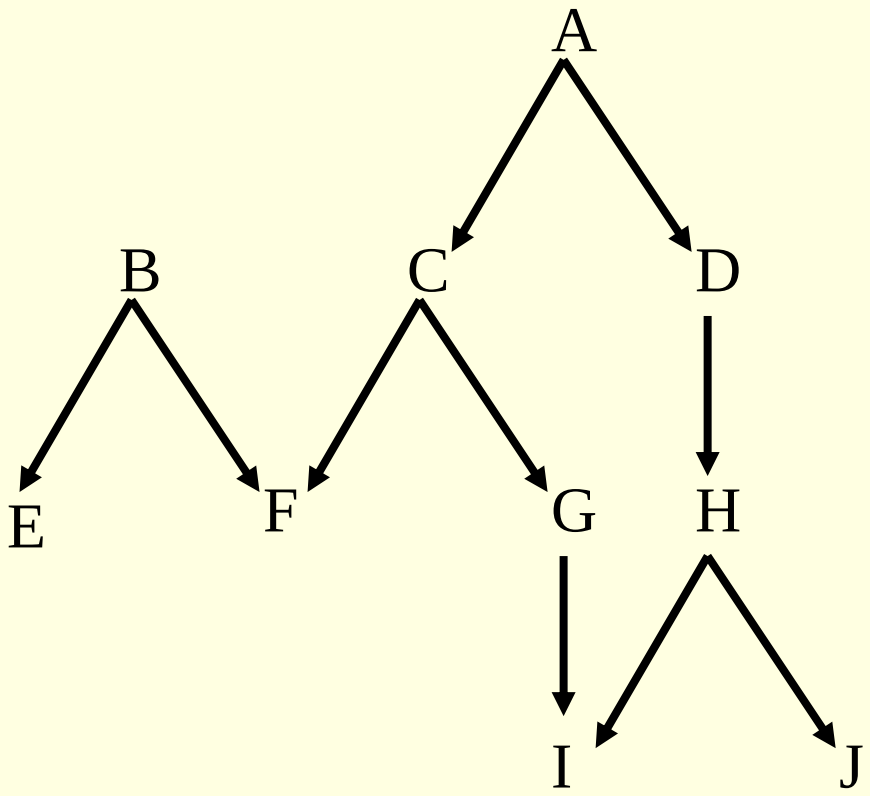
d-separation: A criterion for independence

- 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

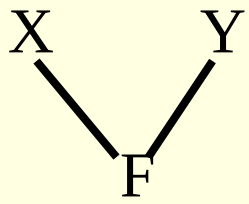
- Ind(E, A|empty)
- Ind(A,J|D)

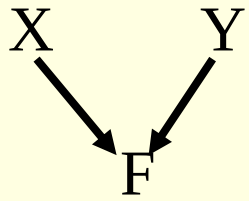


Faithfulness

- When *d*-separation ⇔independence
- Intuitively, an open path between *A* and *B* means there is association between them.

Learning Bayesian Networks: Constraint-Based Approach

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

and for every set *S* that contains F, Dep(*X, Y* | *S*), then 

Learning Bayesian Networks: Constraint-Based Approach

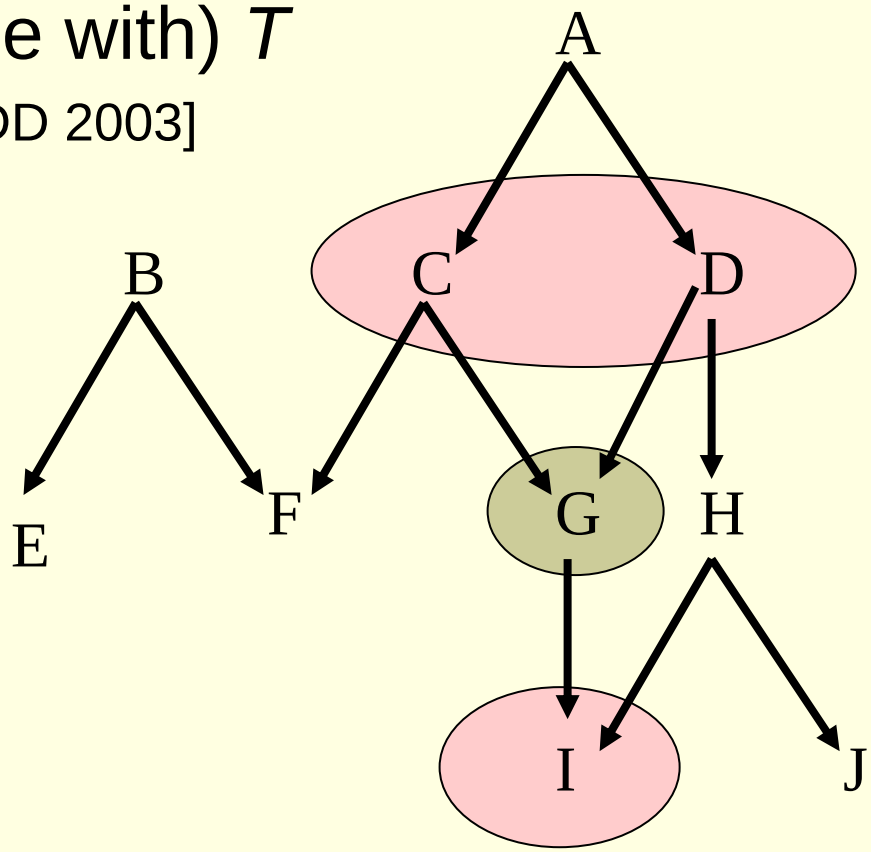
- Tests of conditional dependences and independencies from the data.
- Estimation using G^2 statistic, conditional mutual-information, etc.
- Infer structure and orientation from results of tests.
- Based on the assumption these tests are accurate.
- The larger the number of nodes in the conditioning set, the more samples are required to estimate the dependence, Ind(A,B|C,D,E) more sample than 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 (Cooper, Heckerman, et. al): P(Structure | 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]

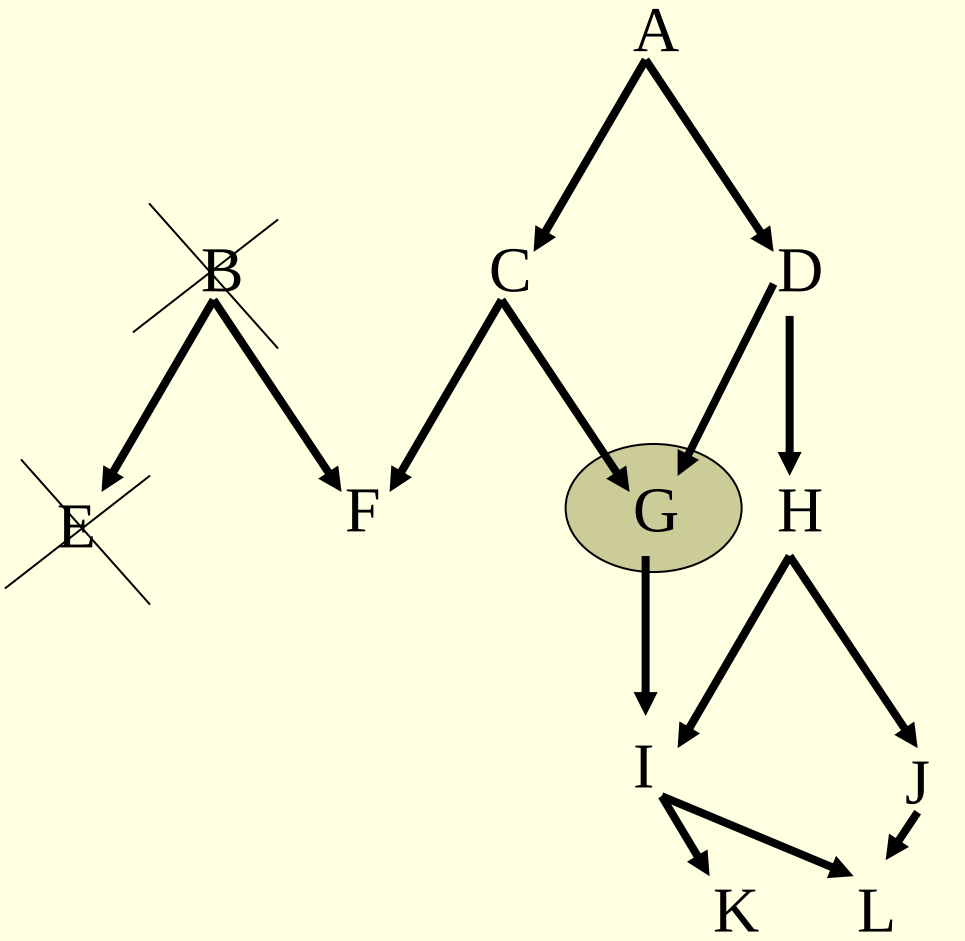


Max-Min Parents and Children Algorithm

- Given Data *D*, and a target variable *T*
- Phase I: Forward
- Start with the empty set for *PC(T)*=∅
- Repeat
  - For each subset *s* of *PC(T)*
    - Remove from further consideration all variables *X* that Ind*(X,T|s)*
  - Heuristic Step: Select variable *X* out of the remaining ones
  - *PC(T)= PC(T)* ∪ *X*
- Until no change in *PC(T)* (all variables independent)

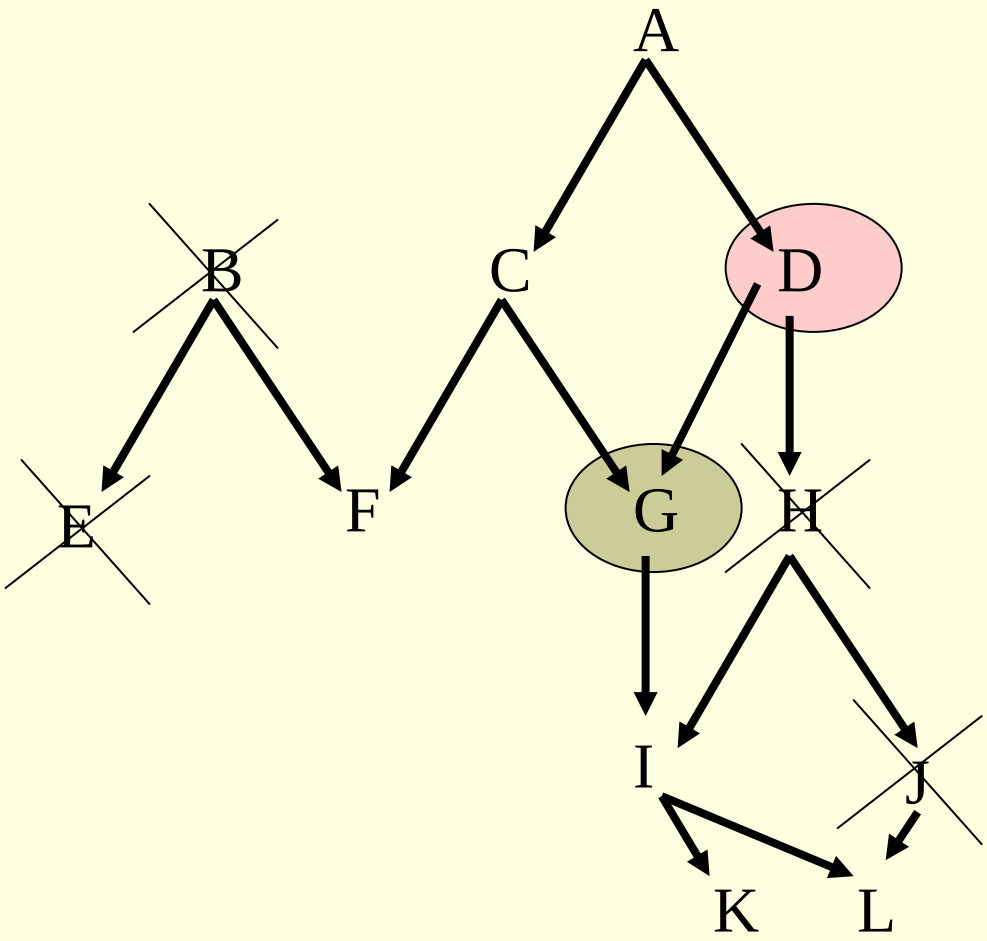
Max-Min Parents and Children Algorithm

- *PC(T)=∅*
- For each subset *s* of *PC(T)*
  - Remove all nodes *X* that Ind*(X,T|s)*
- Select variable *D* out of the remaining ones
- *PC(T) = {D}*



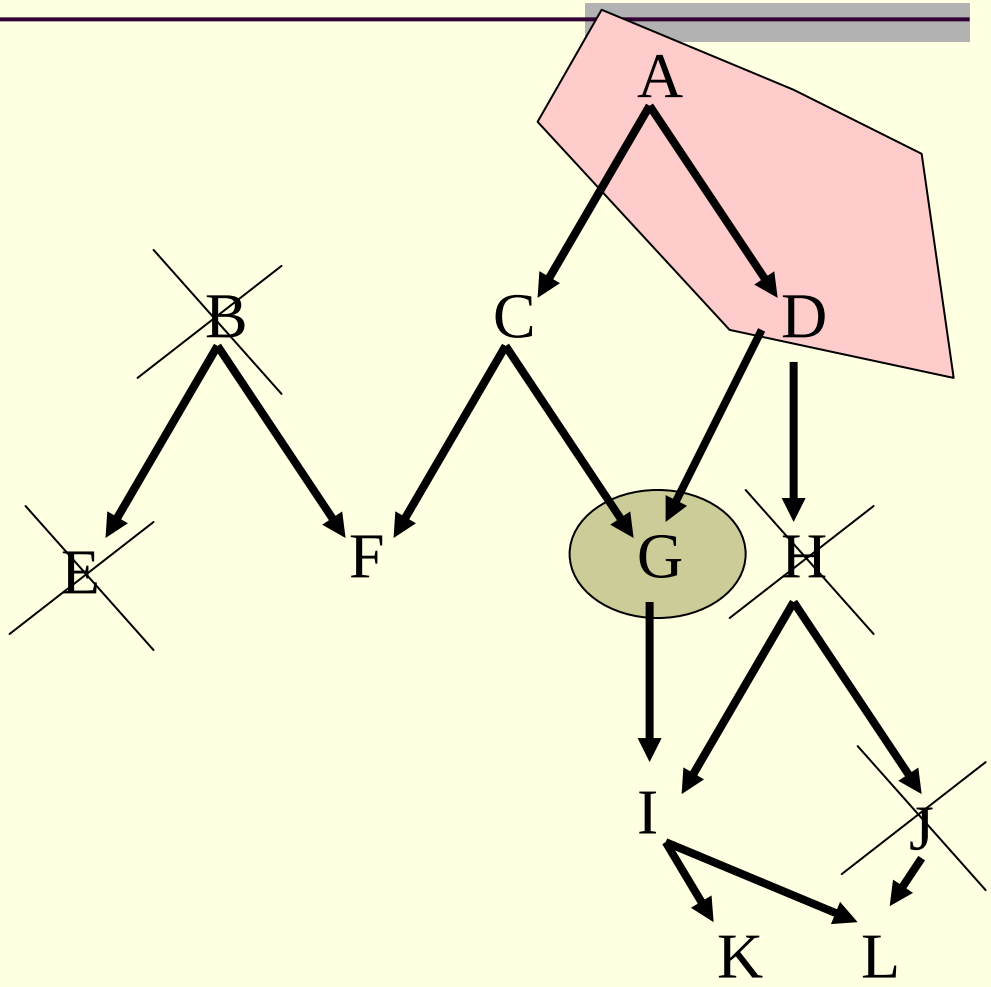
Max-Min Parents and Children Algorithm

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



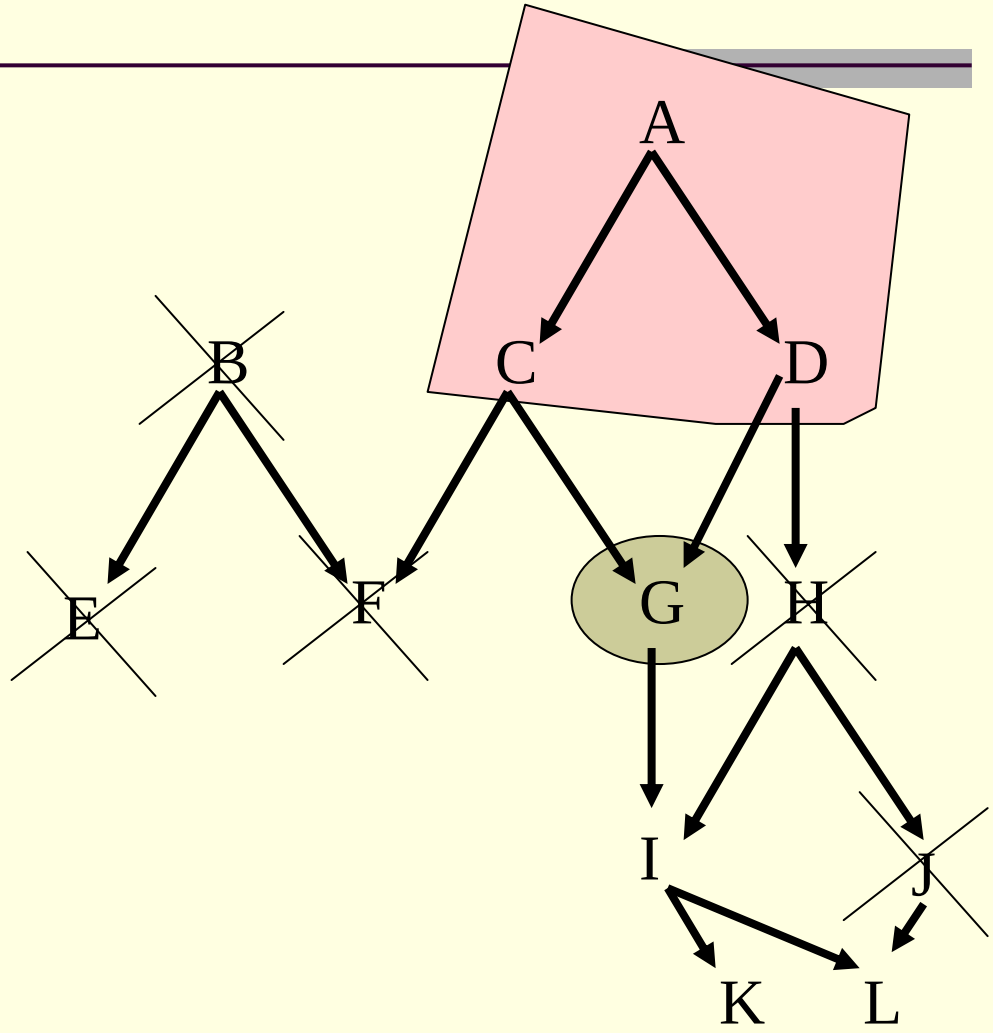
Max-Min Parents and Children Algorithm

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



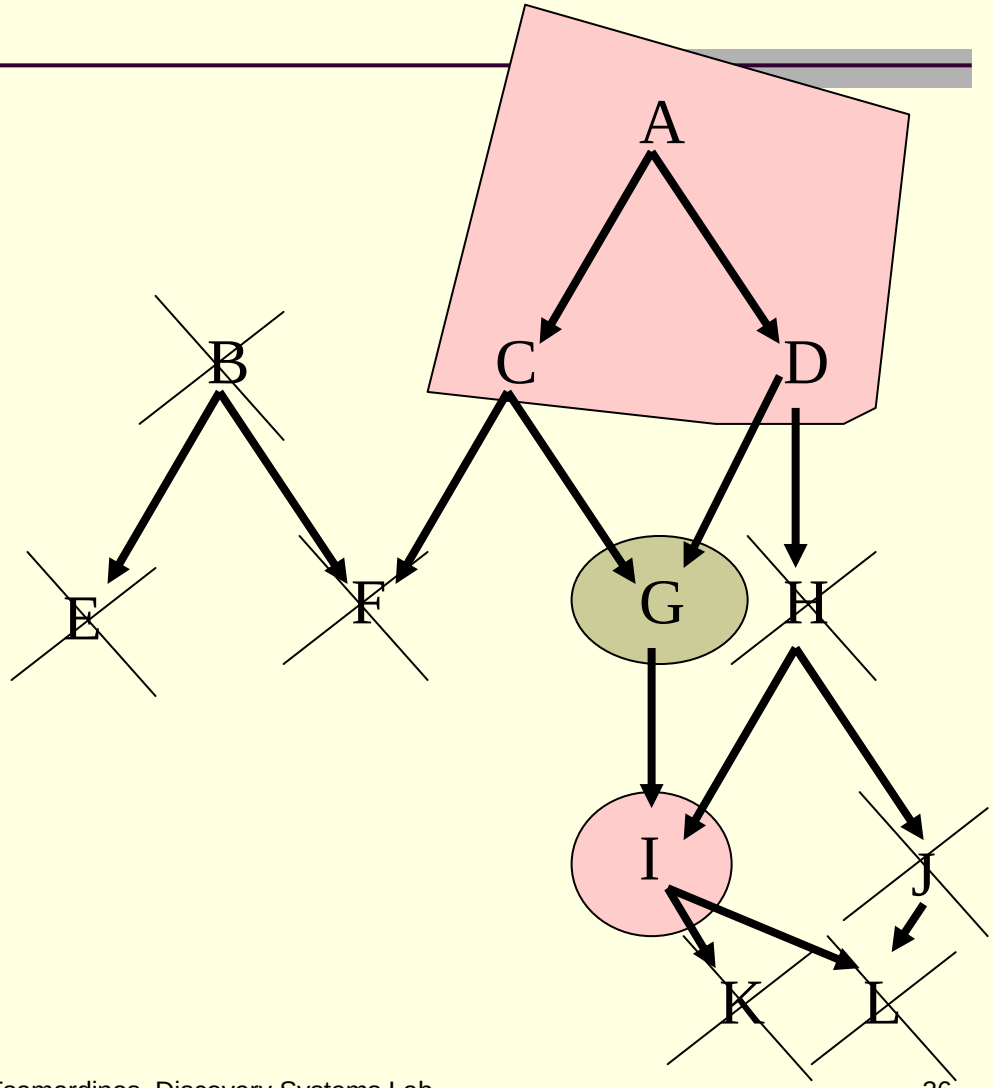
Max-Min Parents and Children Algorithm

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

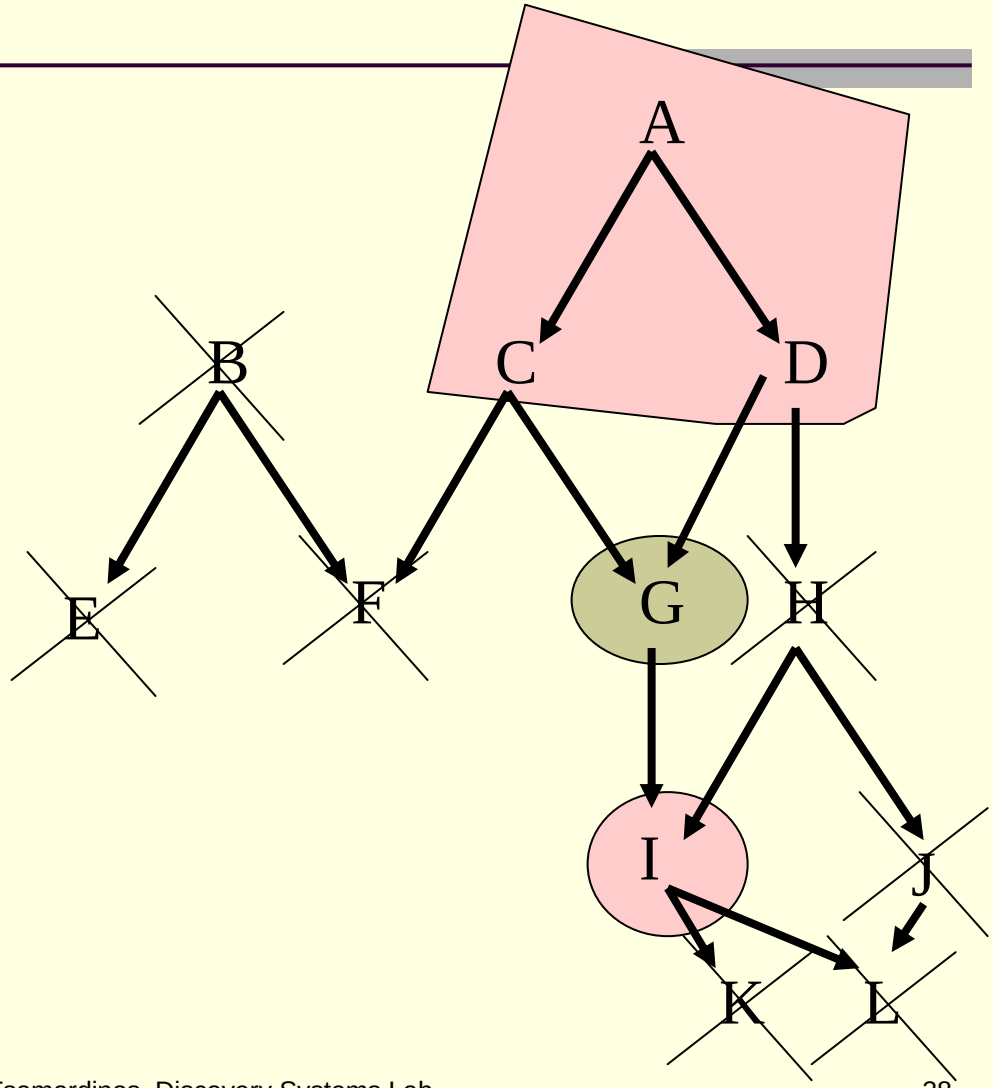


Max-Min Parents and Children Algorithm

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

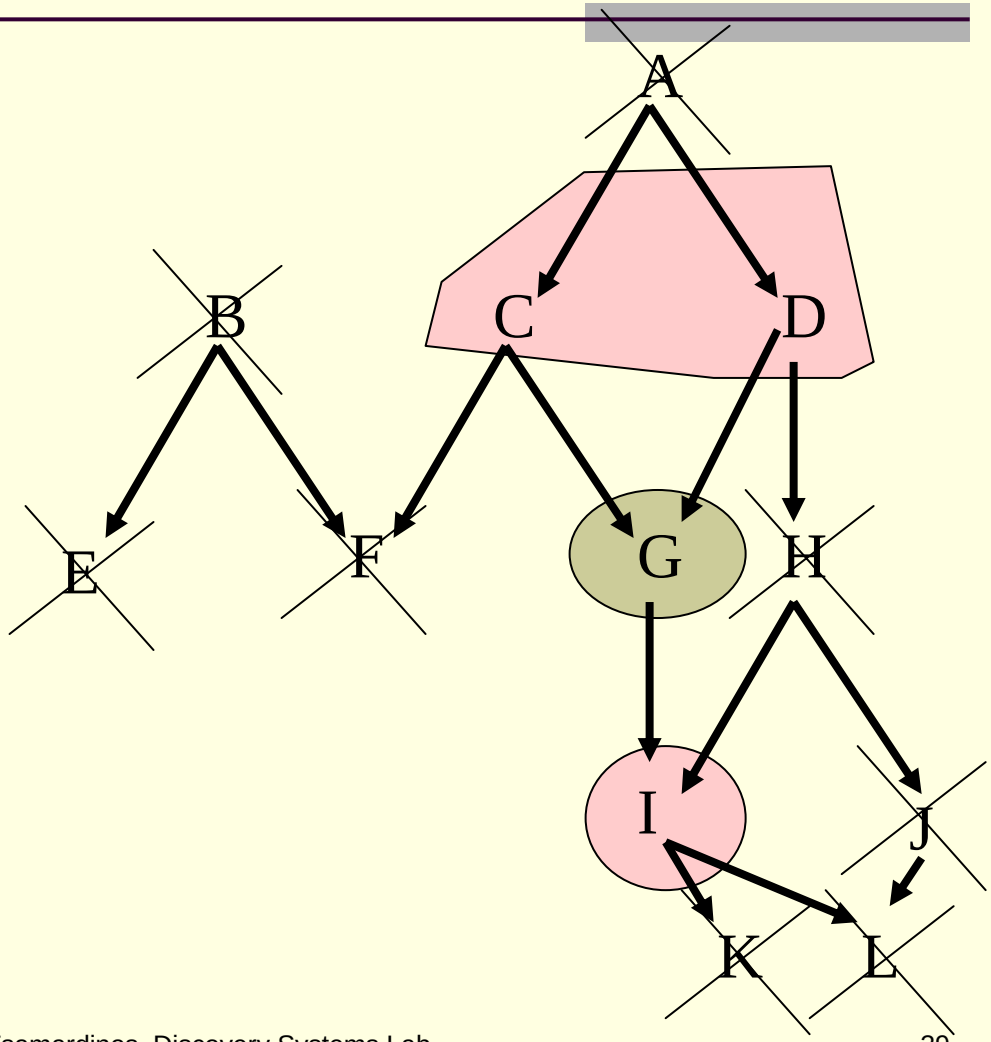
Max-Min Parents and Children Algorithm

- Ind(*A*,*G*|*C*,*D*)
- So, remove *A*



Max-Min Parents and Children Algorithm

- 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

- Max-Min Parents and Children (with minor modifications) 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.

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, constraining search to all the edges discovered in the previous step

Experiments for Bayesian Network Learning

- Start with a known Bayesian Network
- Sample data from the distribution of the network
- Try to reconstruct the network from the data

Comparison with Previous State-of-the-Art Algorithms

- Comparison with the **Sparse Candidate** (similar idea of constraining the search; one of the most prominent BN learning algorithm that scales up to hundreds of variables), the **PC**, and the **TPDA**.
- Measures of Comparison: the Structural Hamming Distance (similar to number of structural errors) and computation time
- Datasets simulated from the distribution of BNs used in real Decision Support Systems and tiled versions of them.

Reconstructing Bayesian Networks: MMHC vs rest in terms of structural errors

Table 2: Structural Hamming Distance (**SHD**) Results. A lower **SHD** implies less structural errors made. *Abs* columns report the actual **SHD** for **MMHC** and all other columns the normalized statistic by **MMHC**'s **SHD** (e.g., a ratio greater than 1 implies a *worse* performance than **MMHC**).

	Sample Size 500					Sample Size 5000				
	MMHC	**PC**	**TPDA**	**SCA** k=5	**SCA** k=10	**MMHC**	**PC**	**TPDA**	**SCA** k=5	**SCA** k=10
	Abs	Ratio with **MMHC**'s **SHD**				Abs	Ratio with **MMHC**'s **SHD**			
Alarm1	29	3.38	2.33	1.92	1.95	8	1.68	1.17	4.17	2.49
Alarm3	107	2.12	2.87	1.72	1.72	73	1.79	1.18	2.10	2.07
Alarm5	220	2.19	2.58	1.56	1.59	148	1.58	1.21	2.05	1.96
Alarm10	546	2.06	–	1.40	1.43	345	1.55	1.29	1.88	1.75
Child	16	2.78	4.57	1.38	1.42	0.00	20*	32.80*	11.00*	11.00*
HailFinder	178	2.24	1.45	0.93	–	200	1.77	1.18	0.97	–
Munin	532	–	1.34	1.19	–	462	–	1.35	1.26	–
Pigs	153	5.84	–	0.12	0.05	9.40	112	20.82	2.62	1.00
Gene	314	2.01	–	0.61	–	139	1.62	–	1.71	–
Average		**2.83**	**2.52**	**1.20**	**1.36**		**17.46**	**4.03**	**2.10**	**1.85**

Reconstructing Bayesian Networks: MMHC vs rest in terms of computation time

Table 3: Time Results. *Abs* columns report the actual **Time** in seconds for **MMHC** and all other columns the normalized statistic by **MMHC**'s **Time** (e.g., a ratio greater than 1 implies **MMHC**) was faster). **SC** becomes relatively slower than **MMHC** as the number of variables increases in Alarm1-Alarm10.

	Sample Size 500					Sample Size 5000				
	MMHC	**PC**	**TPDA**	**SCA** k=5	**SCA** k=10	**MMHC**	**PC**	**TPDA**	**SCA** k=5	**SCA** k=10
	Abs	Ratio with **MMHC**'s Time				Abs	Ratio with **MMHC**'s Time			
Alarm1	6	1.37	15.41	1.43	12.21	17	1.32	1.90	3.53	7.30
Alarm3	33	1.26	64.35	6.14	9.17	93	1.58	1.94	20.19	16.00
Alarm5	85	100.07	91.68	9.90	10.17	234	1.20	2.08	37.43	32.20
Alarm10	391	401.50	–	15.65	15.86	990	0.93	2.31	76.55	71.12
Child	3	0.55	24.61	0.59	7.03	14	1.08	1.32	0.61	1.77
HailFinder	13	1.54	50.98	2.56	–	94	5.73	2.13	2.61	–
Munin	351	–	23.00	3.79	–	1330	–	5.30	9.73	–
Pigs	1888	0.42	–	3.75	5.21	60608	0.28	0.26	1.26	1.06
Gene	4966	1.15	–	11.91	–	17250	0.47	–	36.27	–
Average		**63.48**	**45.00**	**6.19**	**9.94**		**1.57**	**2.16**	**20.91**	**21.58**

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

Never Say Never

- “In our view, inferring *complete* causal models (i.e., causal Bayesian Networks) is essentially impossible in large-scale data mining applications with thousands of variables”, Silverstein, Brin, Motwani, Ullman 2000

Variable Selection for Classification

Example Problems: Classification

- Classify 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

Machine Learning for Classification

- Data are fed to a learning algorithm such as Neural Networks, Support Vector Machines, K-Nearest Neighbors, Decision Trees, etc.
- The learning algorithms produces a classifier (i.e., a predictive/diagnostic/classification model).
- The model is used to classify future (unseen) instances.

Data in Biomedicine: The Challenge

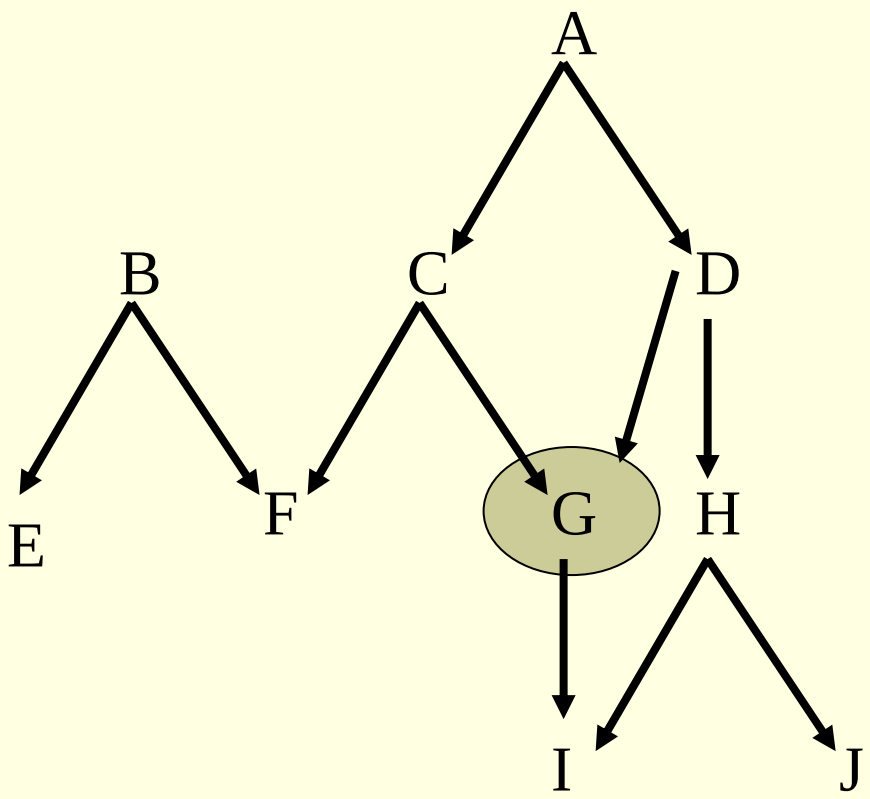
- Availability of extremely high dimensional data:
  - Because of mass throughput techniques
    - Gene expression microarray data: (range 10K-15K variables)
    - Mass-spectroscopy(60K-65K)
    - Chemical structural properties (140K)
    - Text-categorization (10K-20K)
  - Because of time-series measurements or representational issues
    - Variable A at times 1, 2, ..*n,* becomes A_1, …, A_n
    - Variable A with values {$disease_1$, …, $disease_n$} becomes binary variables $A_{disease1}$, …, $A_{disease\ n}$
- Small number of training instances
- Too many variables available

Example Problems: Variable Selection

- Select the smallest subset of variables with the highest predictive power
- E.g., select the minimum number of medical tests to diagnose a set of diseases
- It reduces the cost of observing the variables, risk to the patients
- It may increase the predictive power (treats the curse of dimensionality)
- It is easier to explain and trust a smaller model
- Bayesian Network theory to the rescue

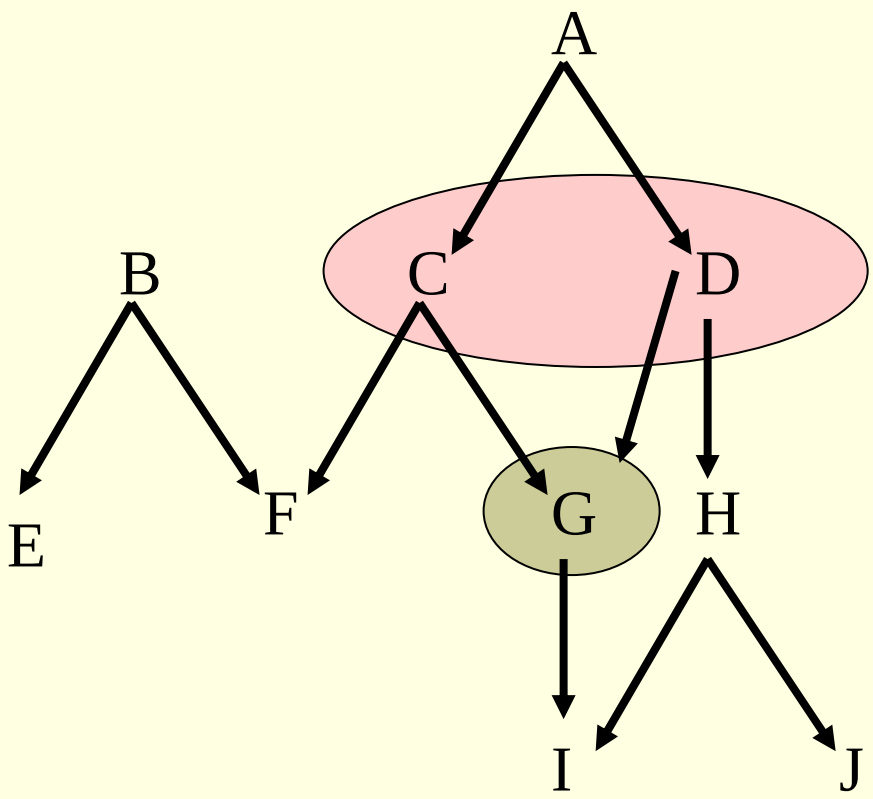
The Markov Blanket

- *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*



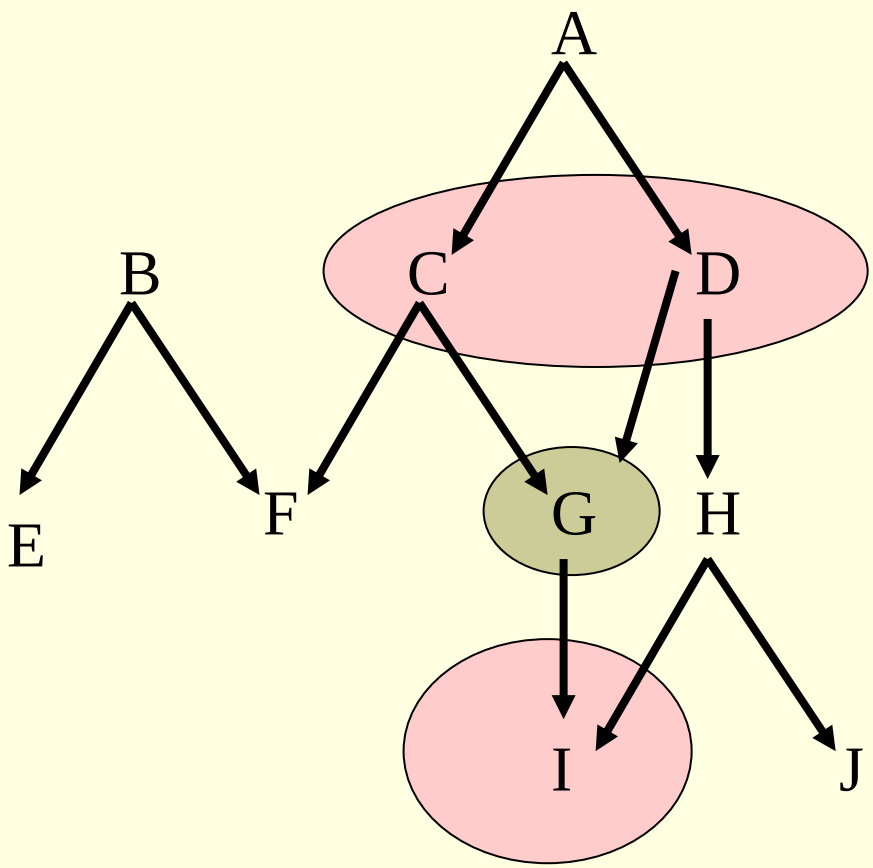
The Markov Blanket

- *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*



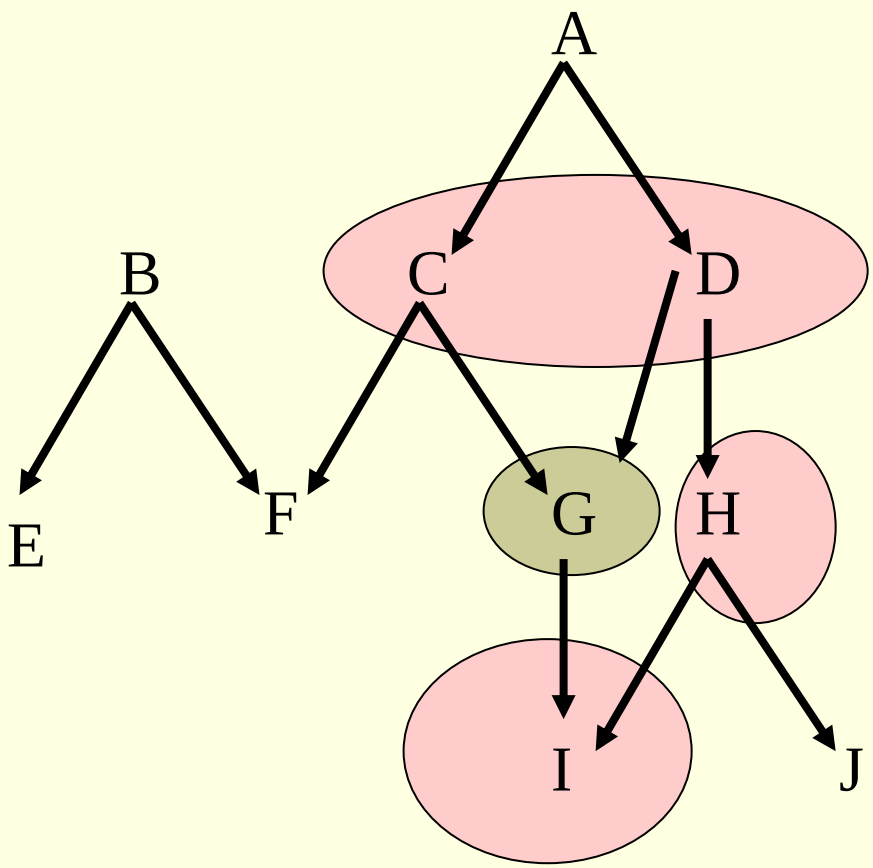
The Markov Blanket

- *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*



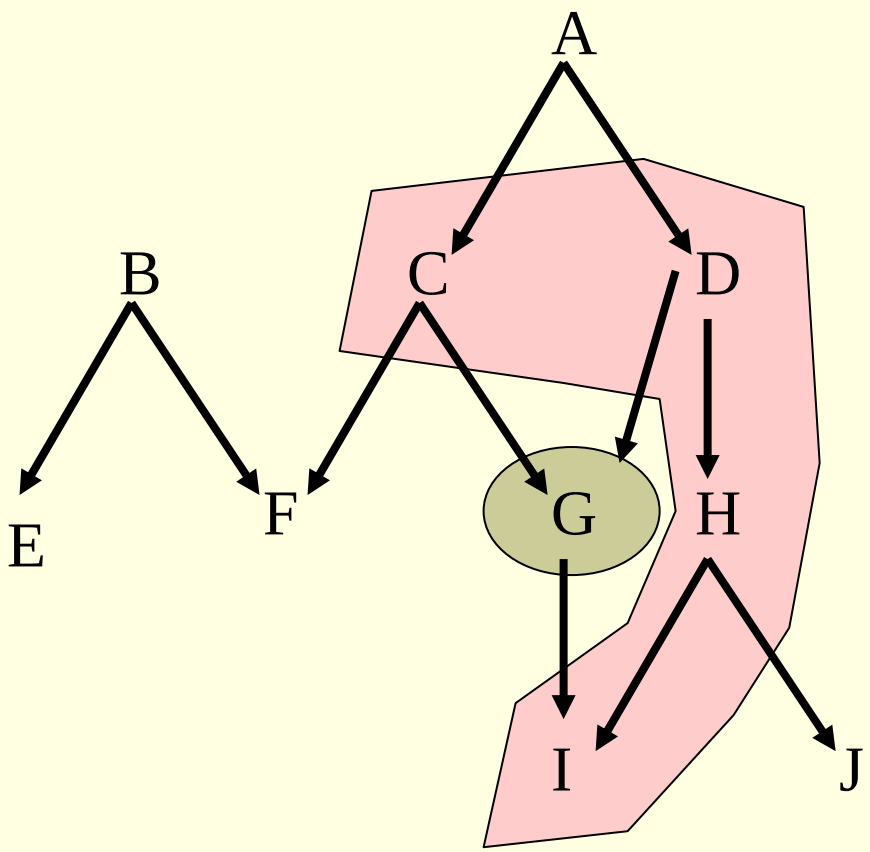
The Markov Blanket

- *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*



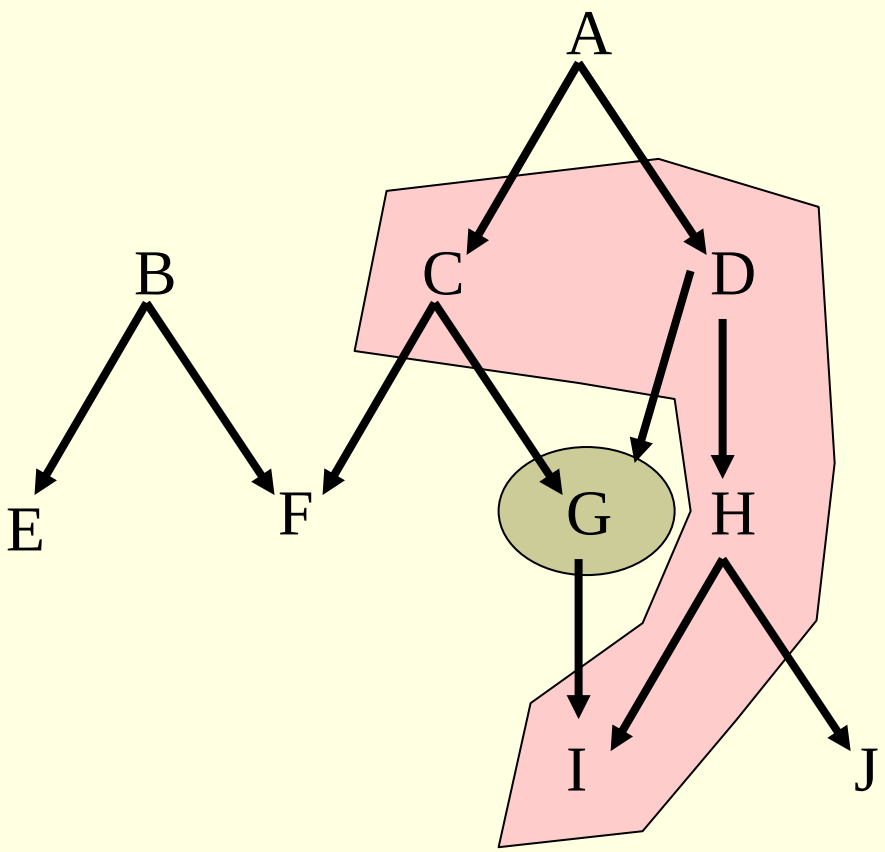
The Markov Blanket

- *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)

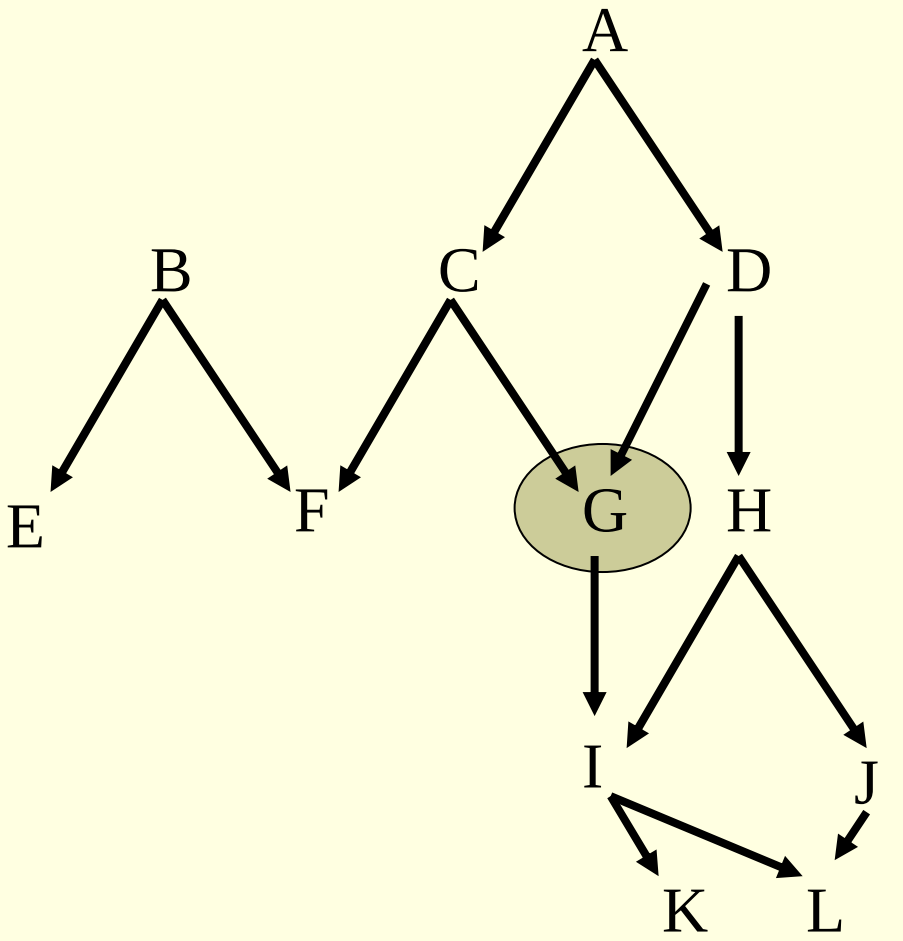


Markov Blanket for Variable Selection

- [Tsamardinos, Aliferis, AI&Stats 2003]
- *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.

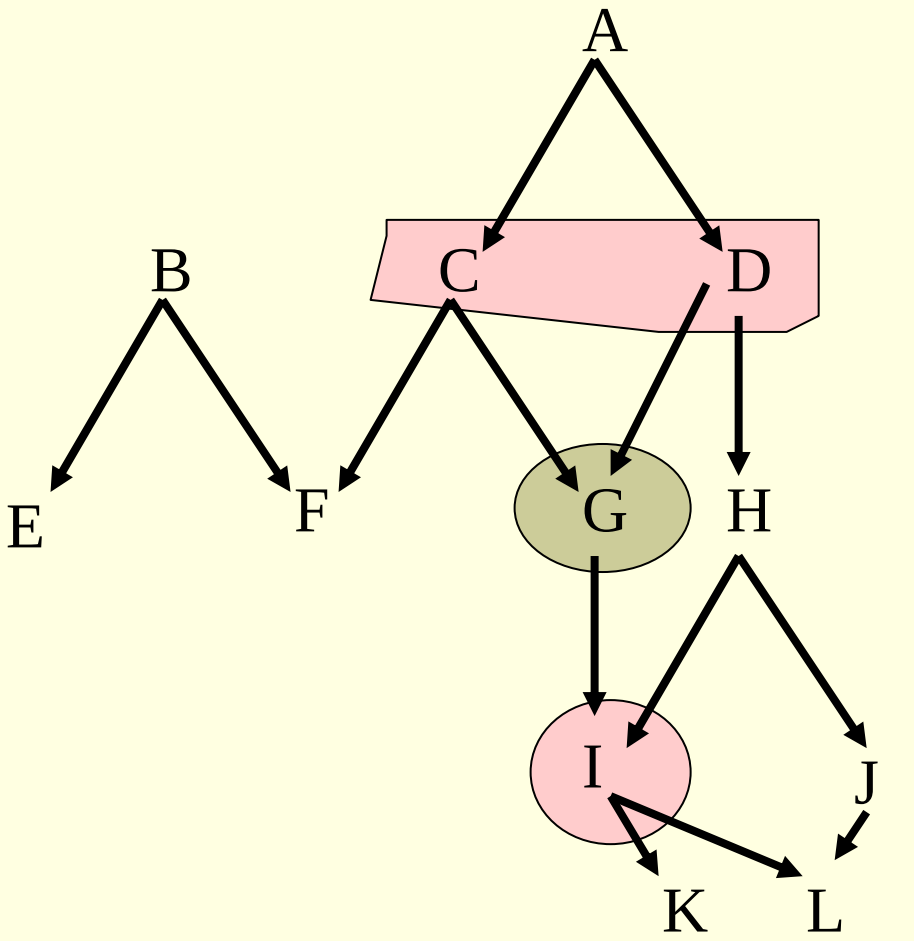
Discovering the Markov Blanket

- Find *PC(G)*



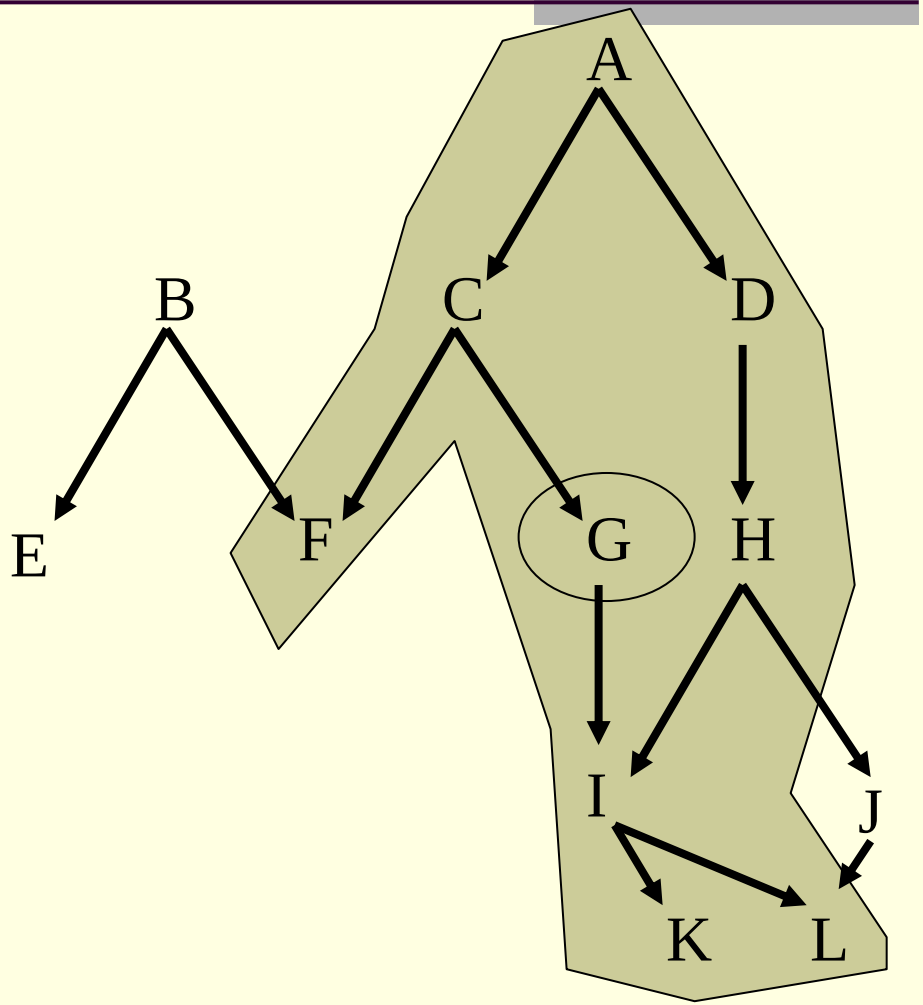
Discovering the Markov Blanket

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



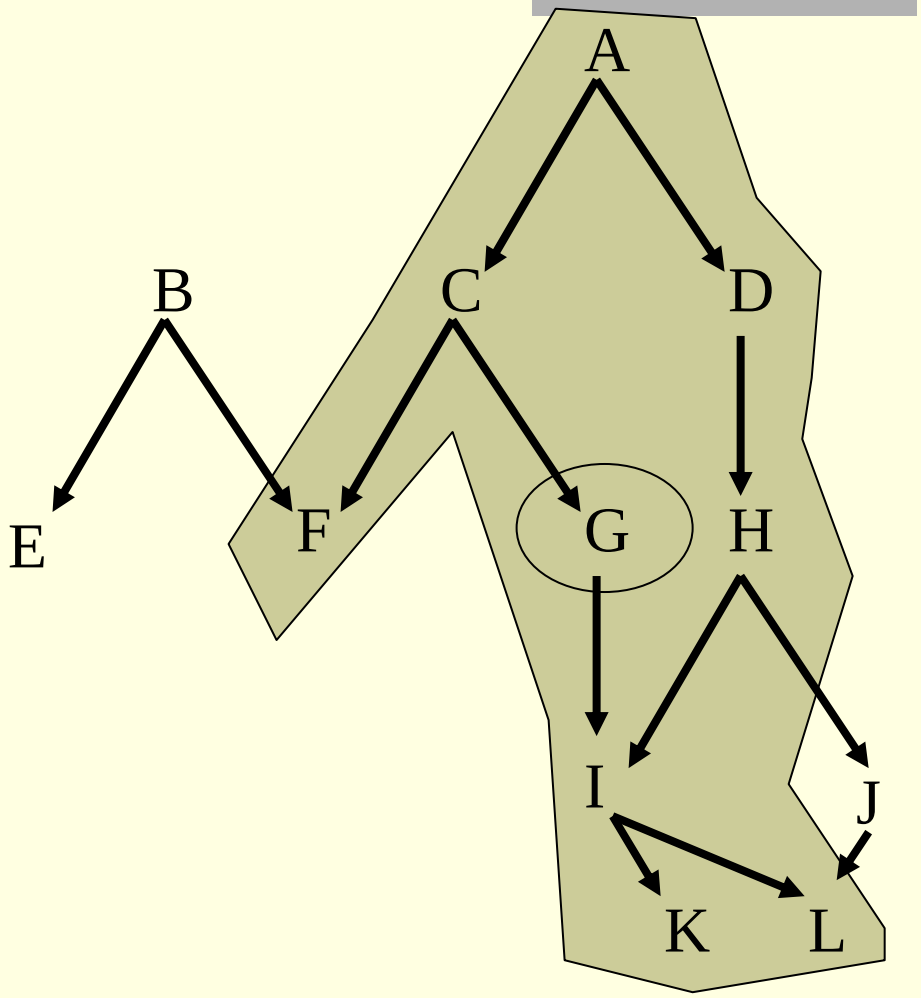
Discovering the Markov Blanket

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



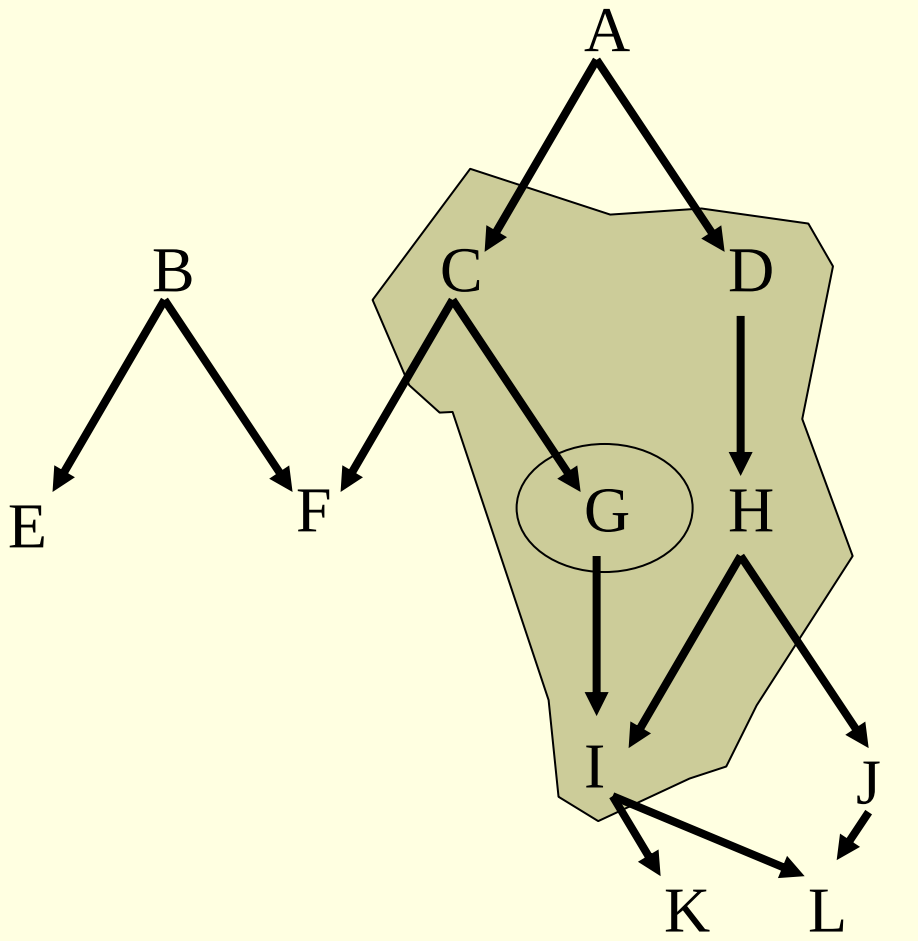
Discovering the Markov Blanket

- 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 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*∪*s*), *s* subset of *PC(T)*



Experiments for Variable Selection

- The HITON algorithm:
  - Find *MB(T)*
  - 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 squamus 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

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

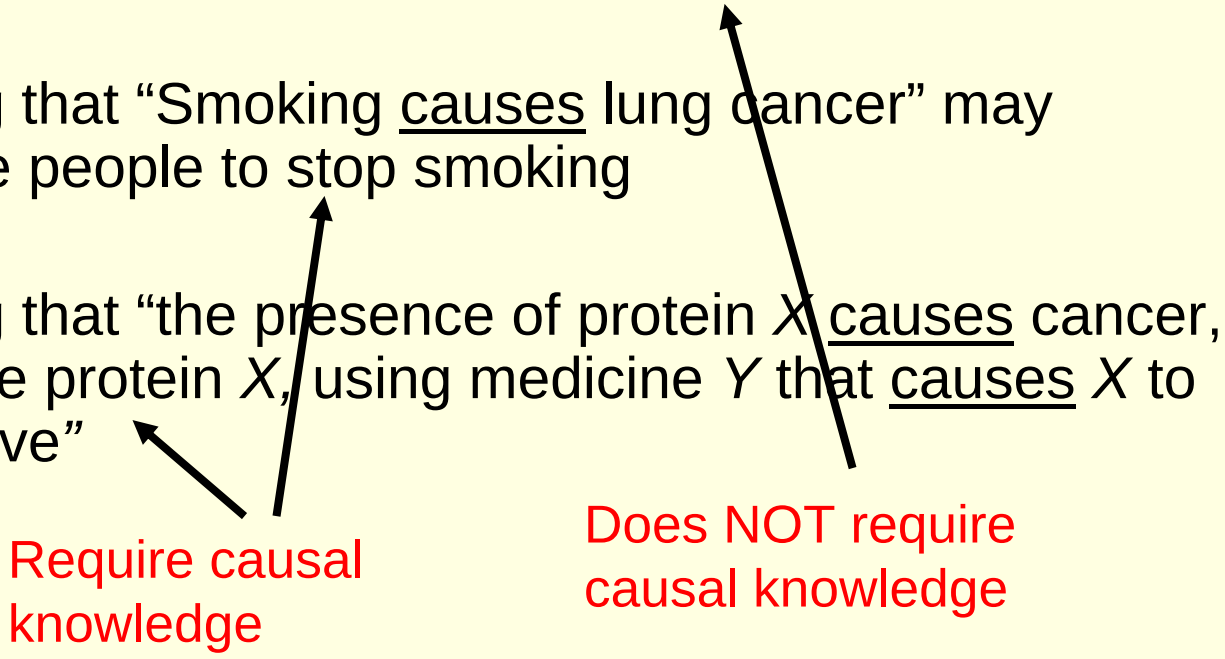
Averages Over All Tasks

	Av. over Baseline Algorithms	HITON	ALL No Variable Selection
Av. Perf. over classifiers	86.1%	**87.1%**	86.1%
Av. variable #	4,540	**32.3**	33,476
Av. reduction	x 8	x **1124**	x 1

Causal Discovery

Important Tasks in Biomedicine

- ***Diagnosis***
  - Knowing that “people with cancer often have yellow-stained fingers and feel fatigue”, diagnose lung cancer
- ***Prevention***
  - Knowing that “Smoking causes lung cancer” may convince people to stop smoking
- ***Treatment***
  - Knowing that “the presence of protein *X* causes cancer, inactivate protein *X*, using medicine *Y* that causes *X* to be inactive”



Example Problems: Causal Discovery

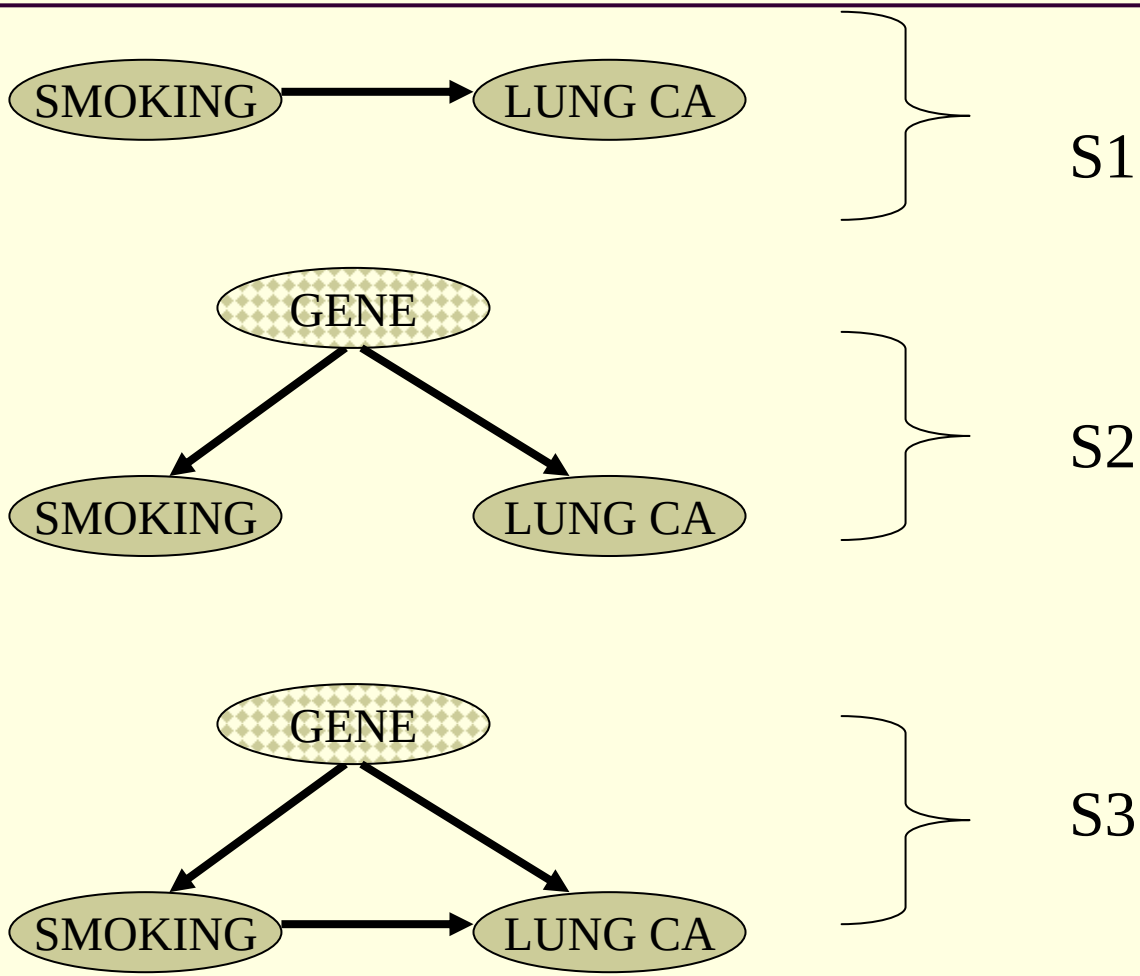
- Discover the genes that (directly) cause cancer
- Discover the genes that (directly) affect (cause) a gene's expression level to change
- Discover the structural properties of a drug that directly cause it to exhibit certain biochemical properties
- What SNP combination causes what disease?
- How genes and proteins are organized in complex causal regulatory networks?
- How genotype causes differences in response to treatment?

Causation and Association

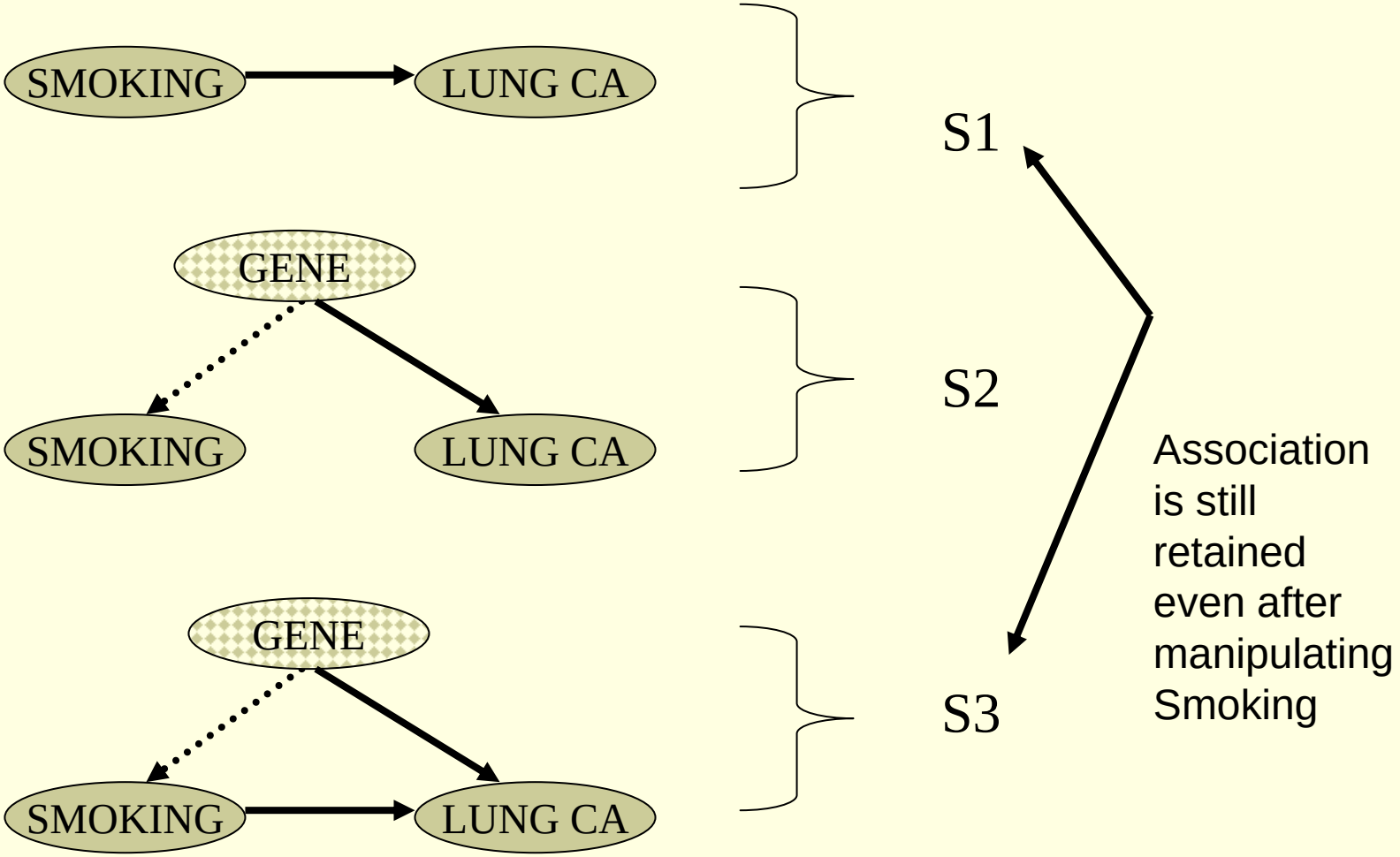
- Causation: A causes B means changing the value of A changes the probability distribution of B
- What is the relationship between the causation and association?
- If A causes B, are A and B always associated?
- If A is associated with B are they always causes or effects of each other? (directly?, indirectly?, conditionally, unconditionally?)

- Need to know causation to manipulate a system

Statistical Indistinguishability



RANDOMIZED CONTROLLED TRIALS



RCTs Are *not* always feasible!

- Unethical (smoking)
- Costly/Time consuming (gene manipulation, epidemiology)
- Impossible (astronomy)
- Extremely large number

Formal Computational Causal Discovery from Observational Data

- Formal algorithms exist! Two Nobel prizes in economics in the last 5 years!
- Most are based on a graphical-probabilistic language called “Causal Probabilistic Networks (a.k.a. “Causal Bayesian Networks”)
- Well-characterized properties of
  - What types of causal relations they can learn
  - Under which conditions
  - What kind of errors they may make

Types of Causal Discovery Questions

- What will be the effect of a manipulation to the system
- Is *A* causing *B*, *B* causing *A*, or neither?
- Is *A* causing *B* directly (no other observed variables interfere)?
- What is the smallest set of variables for optimally effective manipulation of *A*?
- Can we infer the presence of hidden confounder factors/variables?

Bayesian Networks and Causal Discovery

Bayesian Networks

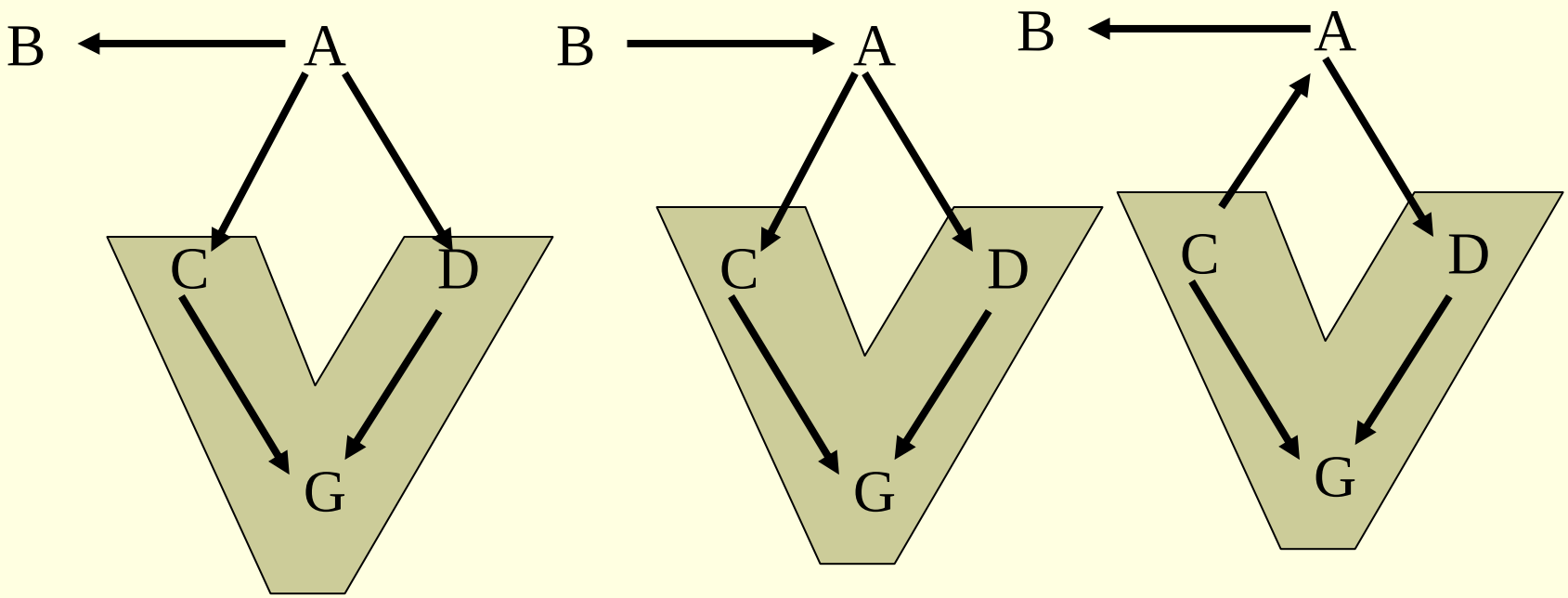
- Edges: probabilistic dependence
- Markov Condition: A node *N* is independent from non-descendants given its parents
- Probabilistic reasoning

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

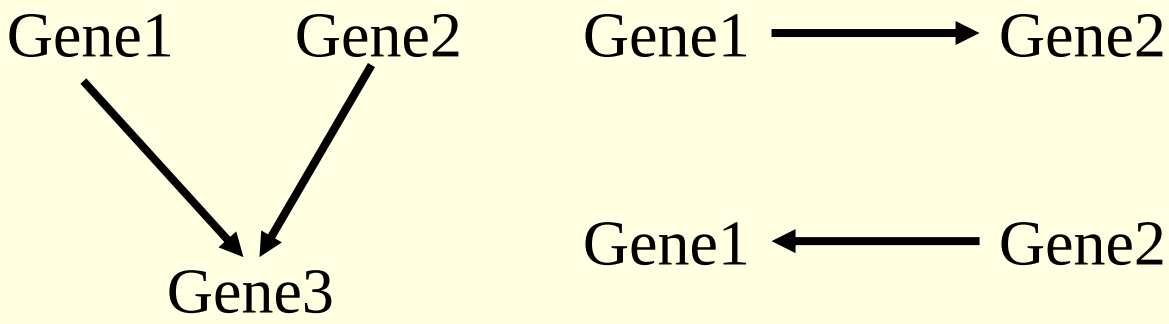
Causal Bayesian Networks

- There may be many (non-causal) BNs that capture the same distribution (i.e., they are statistically equivalent)
- All such BNs have the same edges (ignoring direction) same v-structures



Causal Bayesian Networks

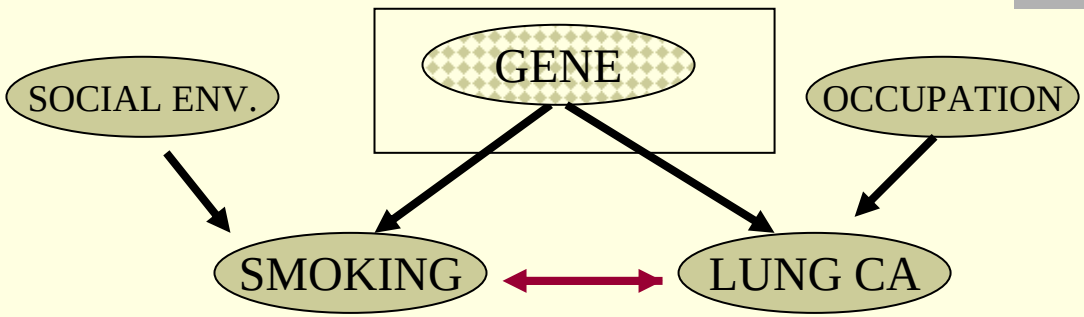
- If there is a (faithful) Causal Bayesian Network that captures the data generation process, it has to have the same edges and same v-structures as any (faithful) Bayesian Network that is induced by the data.



Assumptions for Causal Discovery Using Bayesian Networks

- Faithfulness: independencies occur because of structure, not because of fine tuned parameters or lack of sample or unmeasured variables
  - May cause false negatives
- Acyclicity: no feedback is allowed
  - ?
- Causal Sufficiency: no unmeasured confounders
  - May cause false positives
- Data are atemporal

FCI – Causal Discovery with Hidden Confounders



- Ind(SE,LC|∅)
- Dep(SE,LC|SM)
- Dep(SE,LC|SM, OC)
- Ind(SM,OC| ∅)
- Dep(SM,OC|LC)
- Dep(SM, OC|LC, SE)

- The only consistent model with all tests is one that has a hidden confounder

Summary, Conclusions, and Future Directions

Summary and Conclusions

- Bayesian Networks useful (among others) in Decision Support Systems technology
- Bayesian Network Learning useful in automatically constructing Decision Support systems and inferring causal hypotheses
- Current BN learning algorithms can reliable reconstruct from data BNs with tens of thousands of variables (MMHC algorithm)

Summary and Conclusions

- Current Markov Blanket algorithms can reliable identify the MB variables among hundreds of thousands of variables
- Markov Blanket variable selection algorithms (HITON) are state-of-the-art in variable selection
- BN learning algorithms useful in representing and reasoning with causality and for inferring causal relations and hidden variables
- Incredible potential of formal causal discovery methods!

Future Directions

- Many ideas for improving BN learning in terms of computation time, accuracy of reconstruction, and relaxing the assumptions.
- Connecting SVMs and Markov Blanket techniques (Hardin, Tsamardinos, Aliferis ICML 2004).
- Validate experimentally the causal hypothesis generated by the BN-based algorithms in biological data.

The Discovery Systems Laboratory

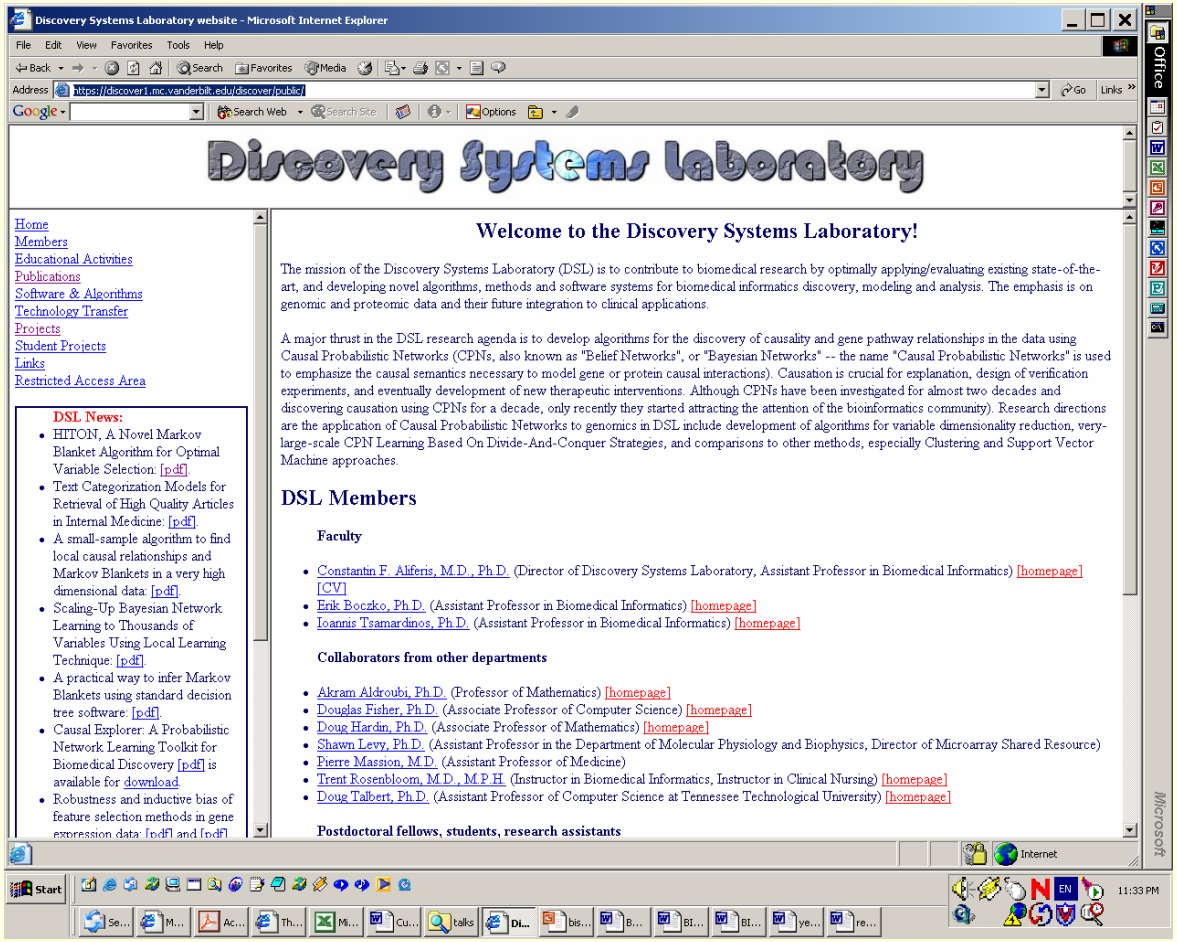
Papers, Software, Information

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

Discovery Systems Laboratory

For more Information (causal discovery tools, publications, contact information)

http://discover1.mc.vanderbilt.edu/discover/public/



The Discovery Systems Laboratory

- Machine Learning expertise and software
- Current projects:
  - Prediction of electrolyte lab results given previous lab results and clinical data
  - Causal analysis of lung cancer gene expression data
  - Causal analysis of epidemiological breast cancer data
  - Diagnosis of stroke given proteomic data
  - Analysis of effect of human hormones on the proteomic profile

Related References

- C. F. Aliferis, I. Tsamardinos, A. Statnikov. "HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection" *In Proceedings of the 2003 American Medical Informatics Association (AMIA) Annual Symposium*, November 8-12, 2003, Washington, DC, USA, pages 21-25
- C. Aliferis, I. Tsamardinos, A. Statnikov, L.E. Brown. "Causal Explorer: A Probabilistic Network Learning Toolkit for Biomedical Discovery" *In Proceedings of the 2003 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS)*, June 23-26, 2003, Las Vegas, Nevada, USA, CSREA Press
- I. Tsamardinos, C.F. Aliferis, A. Statnikov. "Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations" *In Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 24-27, 2003, Washington, DC, USA, ACM Press, pages 673-678
- I. Tsamardinos, C.F. Aliferis, A. Statnikov. "Algorithms for Large Scale Markov Blanket Discovery" *In Proceedings of the 16th International Florida Artificial Intelligence Research Society (FLAIRS) Conference*, May 12-14, 2003, St. Augustine, Florida, USA, AAAI Press, pages 376-380
- I. Tsamardinos and C.F. Aliferis. "Towards Principled Feature Selection: Relevance, Filters, and Wrappers" *In Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, January 3-6, 2003, Key West, Florida, USA
- L. Frey, D. Fisher, I. Tsamardinos, C.F. Aliferis, A. Statnikov. "Identifying Markov Blankets with Decision Tree Induction" *In Proceedings of the Third IEEE International Conference on Data Mining (ICDM)*, November 19-22, 2003, Melbourne, Florida, USA, IEEE Computer Society Press, pages 59-66
- D. Hardin, I. Tsamardinos, C. Aliferis. : A Theoretical Characterization of Linear SVM-Based Feature Selection". International Conference in Machine Learning 2004 (to appear).
- Y. Aphinynanaphongs, C. Aliferis. "Text Categorization Models For Retrieval Of High Quality Articles in Internal Medicine". *American Medical Informatics Association (AMIA) Annual Symposium*, November 8-12, 2003, Washington, DC, USA.
- Y. Aphinynanaphongs, C. Aliferis. "Learning Boolean Queries For Article Quality Filtering". MEDINFO 2004 (to appear).

Software Available and Other Resources

- Causal Explorer: Standard and early algorithms for causal discovery.
- MultiCat SVM library: a library with all major multicategory 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

- Tutorial on Machine Learning
- Slides from all our talks and presentations