Large-Scale Variable Selection with Causal Interpretation

School of Health Information Sciences
Research Seminar
University of Texas at Houston
3-24-2004

Constantin F. Aliferis M.D., Ph.D.
Discovery Systems Laboratory,
Department of Biomedical Informatics,
Vanderbilt University

Acknowledgments

- Collaborators
  - Ioannis Tsamardinos Ph.D.
  - Douglas Hardin Ph.D.
  - Pierre Massion M.D.
- Students
  - Alexander Statnikov M.S.
  - Laura Brown M.S.
- Support
  - NIH
  - Vanderbilt University

Main Points

- Feature selection is significant for decision support and data modeling
- Computational causal induction is very important for biomedical discovery
- There exists a formal connection between the two: the Markov Blanket of a variable, solves the feature selection problem *and* the local causal induction problem
- This talk presents new algorithms to induce the Markov Blanket, presents experimental evaluations and discusses ongoing and future work

What is Feature Selection for classification?

- Given: a set of predictors ("features") V and a target variable T
- Find: minimum set F that achieves maximum classification performance of T (for a given set of classifiers and classification performance metrics)

Filters vs Wrappers: Wrappers

Let’s say we have predictors A, B, C and classifier *M.* We want to predict T given the smallest possible subset of {A,B,C}, while achieving maximal performance (accuracy)

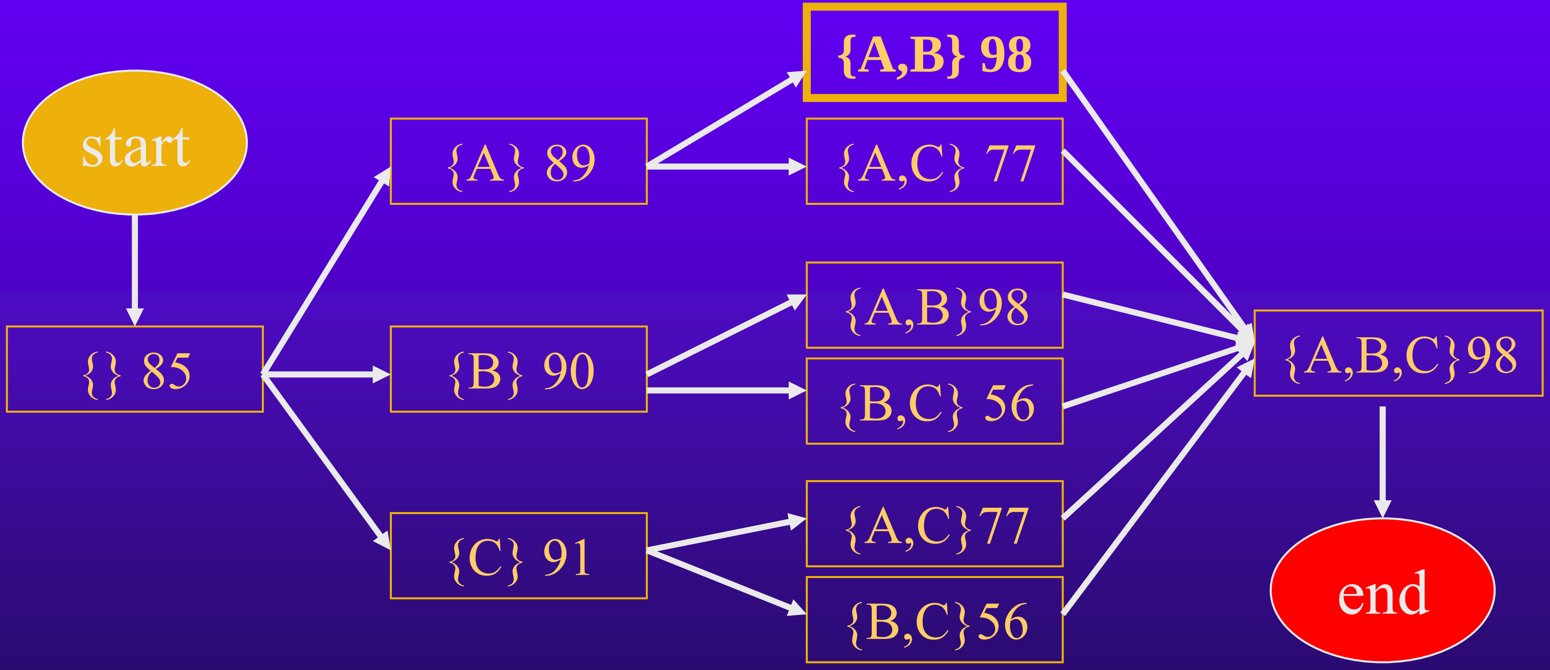
FEATURE SET	CLASSIFIER	PERFORMANCE
{A,B,C}	*M*	**98%**
{A,B}	*M*	**98%**
{A,C}	*M*	77%
{B,C}	*M*	56%
{A}	*M*	89%
{B}	*M*	90%
{C}	*M*	91%
{.}	*M*	85%

Why feature selection is important?

- May Improve performance of classification algorithm
- Classification algorithm may not scale up to the size of the full feature set either in sample or time
- Allows us to better understand the domain
- Cheaper to collect a reduced set of predictors
- Safer to collect a reduced set of predictors

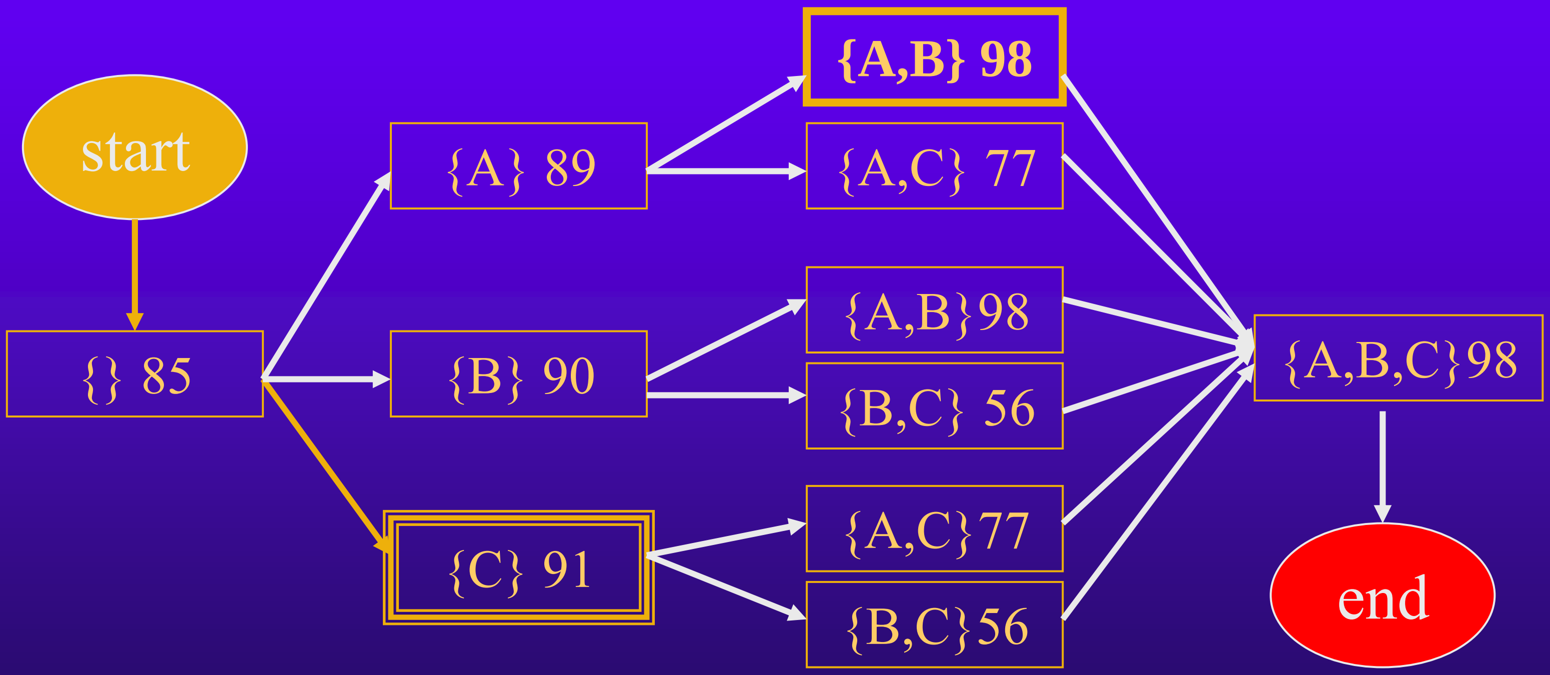
Filters vs Wrappers: Wrappers

The set of all subsets is the power set and its size is $2^{|V|}$. Hence for large *V* we cannot do this procedure exhaustively; instead we rely on *heuristic search* of the space of all possible feature subsets.



Filters vs Wrappers: Wrappers

A common example of heuristic search is hill climbing: keep adding features one at a time until no further improvement can be achieved.



Filters vs Wrappers: Filters

In the filter approach we do not rely on running a particular classifier and searching in the space of feature subsets; instead we select features on the basis of statistical properties. A classic example is univariate associations:

FEATURE	ASSOCIATION WITH TARGET	
{A}	89%	Threshold gives suboptimal solution
{B}	90%	Threshold gives optimal solution
{C}	91%	Threshold gives suboptimal solution

Example Feature Selection Methods in Biomedicine: Univariate Association Filtering

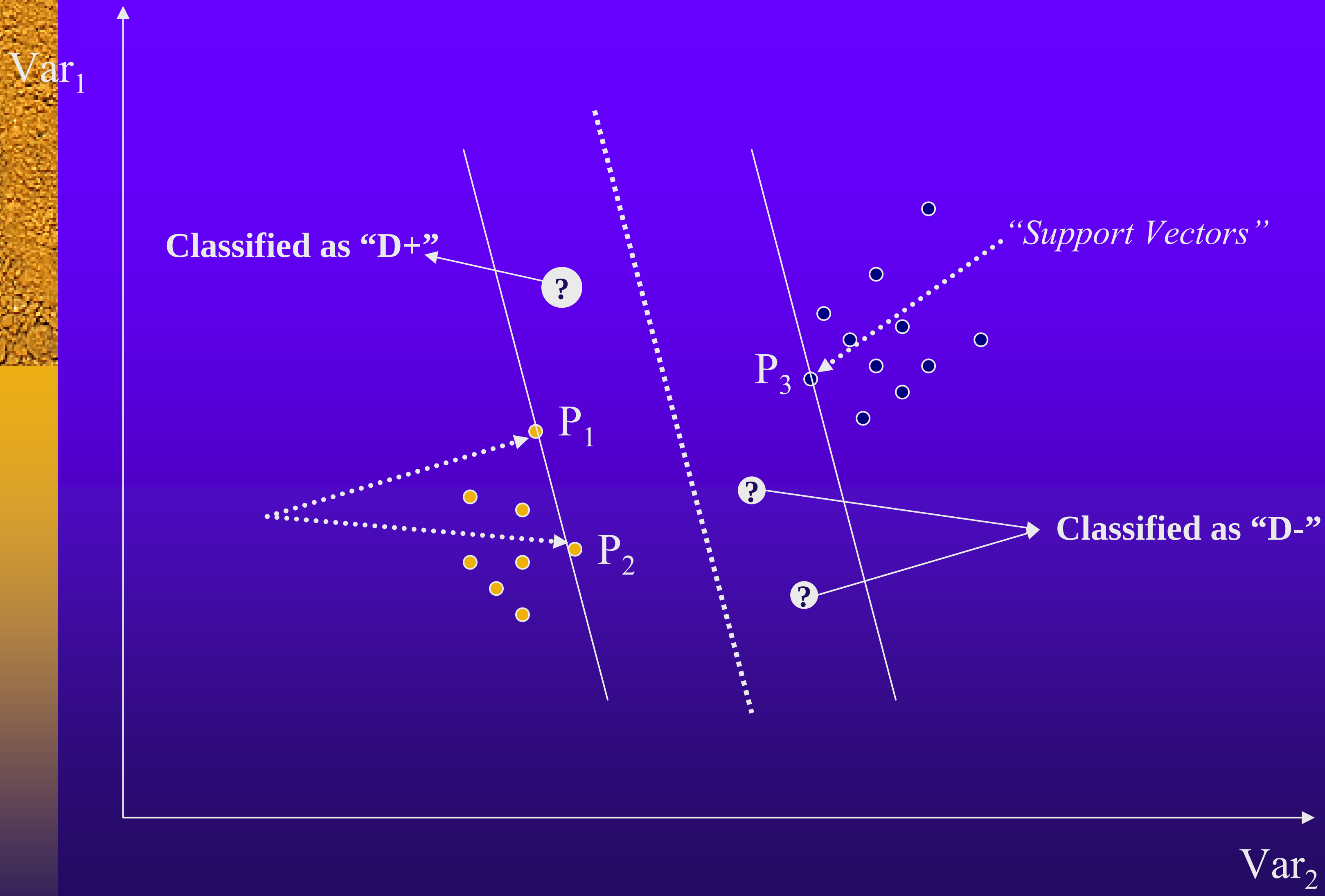
- Order all predictors according to strength of association with target
- Choose the first *k* predictors and feed them to the classifier
- Various measures of association may be used: X^2, G^2, Pearson r, Fisher Criterion Scoring, etc.
- How to choose *k*?
- What if we have too many variables?

Example Feature Selection Methods in Biomedicine: Recursive Feature Elimination

– Filter algorithm where feature selection is done as follows:

1. build linear Support Vector Machine classifiers using *V* features
2. compute weights of all features and choose the best *V/2*
3. repeat until 1 feature is left
4. choose the feature subset that gives the best performance (using cross-validation)

Support Vector machines



Example Feature Selection Methods in Bioinformatics: GA/KNN

Wrapper approach whereby:

- heuristic search=Genetic Algorithm, and
- classifier=KNN

Discovery of Causal Knowledge

- ***Diagnosis***
  - Knowing that "people with cancer often have yellow-stained fingers and feel fatigue", diagnose lung cancer
- ***Prevention***
  - Need to know that "Smoking causes lung cancer" to reduce the risk of cancer
- ***Treatment***
  - Knowing that "the presence of protein *X* causes cancer, inactivate protein *X*, using medicine *Y* that causes *X* to be inactive"

Causal Knowledge NOT required

Causal Knowledge required

Examples of Importance of Causal Discovery Today

- What SNP combination causes what disease?
- How genes and proteins are organized in complex causal regulatory networks?
- How behaviour causes disease?
- How genotype causes differences in response to treatment?
- How the environment modifies or even supersedes the normal causal function of genes?

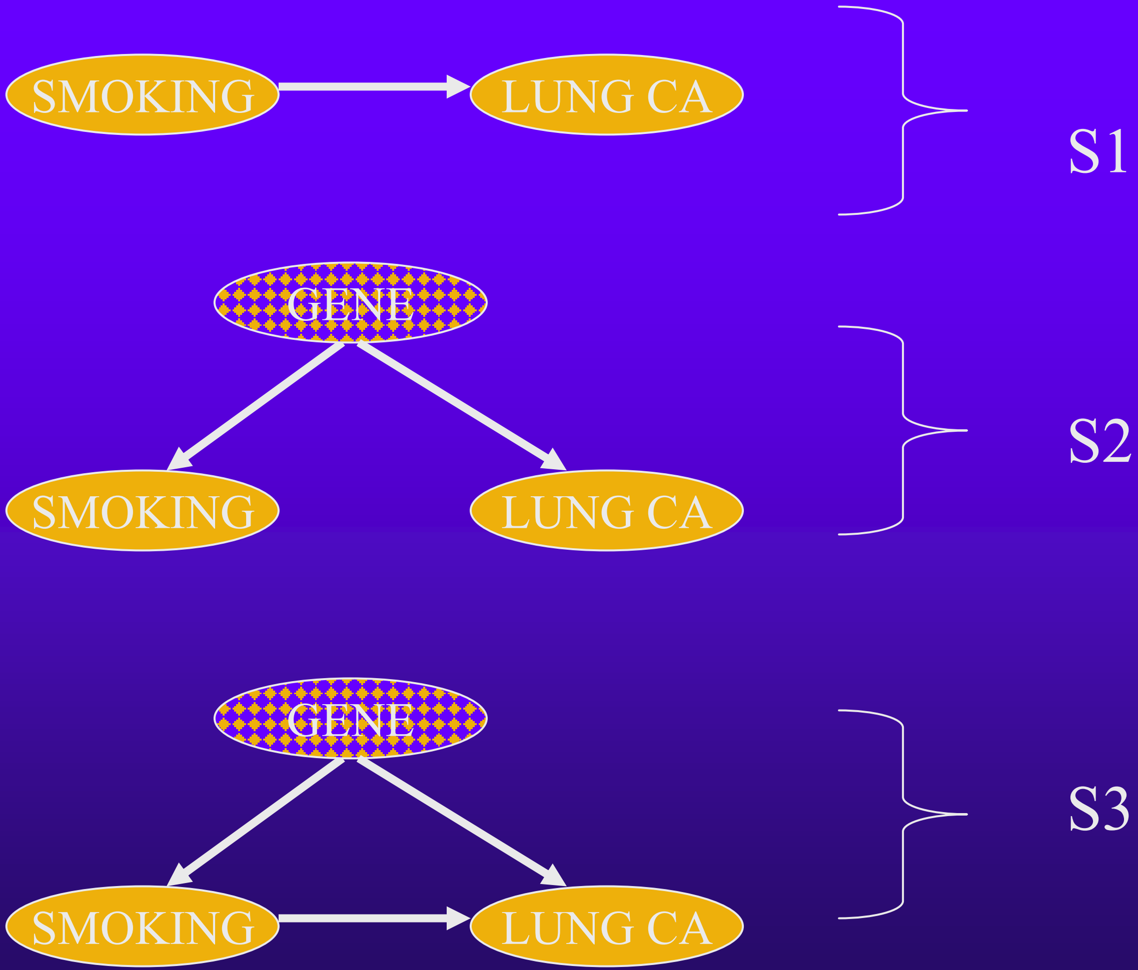
What is Causality?

- Thousands of years old problem, still debated
- A common operational informal definition is based on randomized experiments:
  - Assume the existence of a mechanism M capable of setting values for a variable A. We say that *A can be manipulated* by M to take the desired values.
  - Variable A causes variable B, *if*: in a hypothetical randomized controlled experiment in which A is randomly manipulated via M (i.e., all possible values a_i of A are randomly assigned to A via M) we would observe in the sample limit that $P(B= b \mid A= a_i) \neq P(B= b \mid A=a_j)$ for some $i \neq j$.

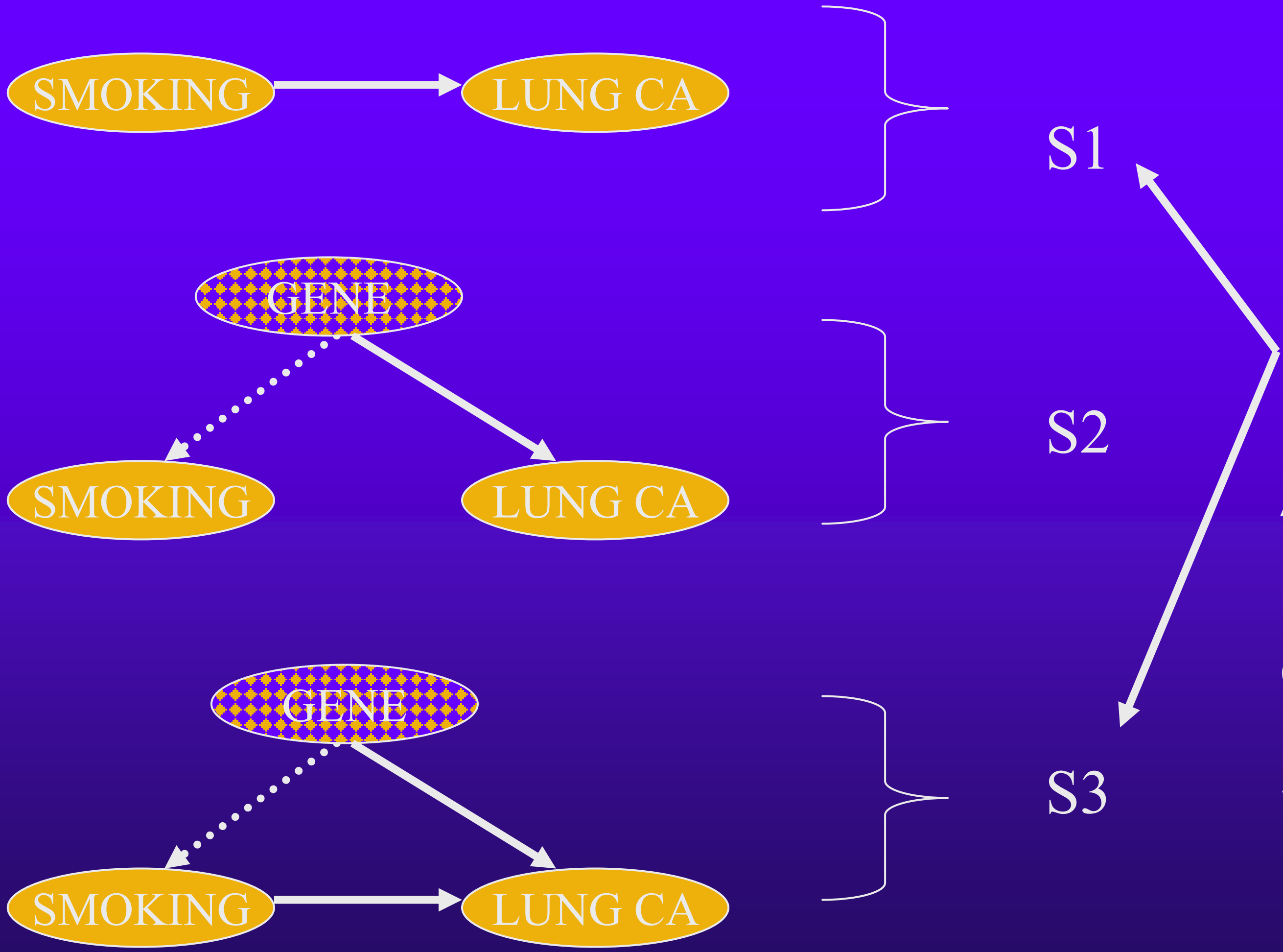
Causation and Association

- What is the relationship between the two?
- 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?)

Statistical Indistinguishability



RANDOMIZED CONTROLLED TRIALS



Association is still retained even after manipulating Smoking

RCTs Are *not* always feasible!

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

Large-Scale Causal Discovery without RCTs?

- Heuristics to the rescue…
- What is a heuristic?

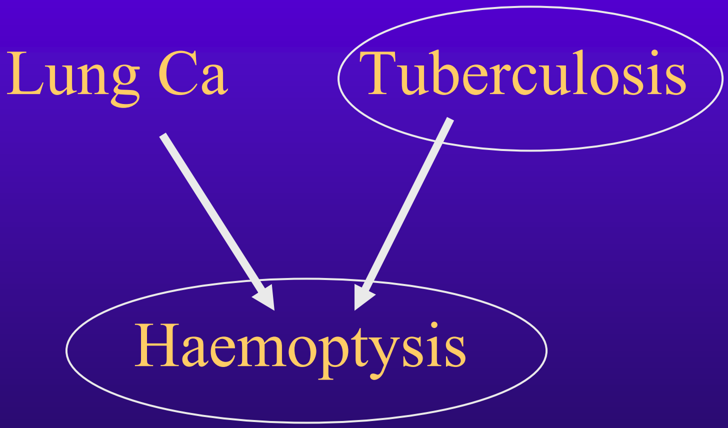
Causal Heuristic #1

- Surgeon's General's "Epidemiological Criteria for Causality" [Surgeon General of the United States 1964]: *A* is causing *B* with high likelihood if:
  1. *A* precedes *B*;
  2. *A* is strongly associated with *B*;
  3. *A* is consistently associated with *B* in a variety of research studies, populations, and settings;
  4. *A* is the only available explanation for *B* ("coherence");
  5. *A* is specifically associated with *B* (but with few other factors).

Causal Heuristic #2

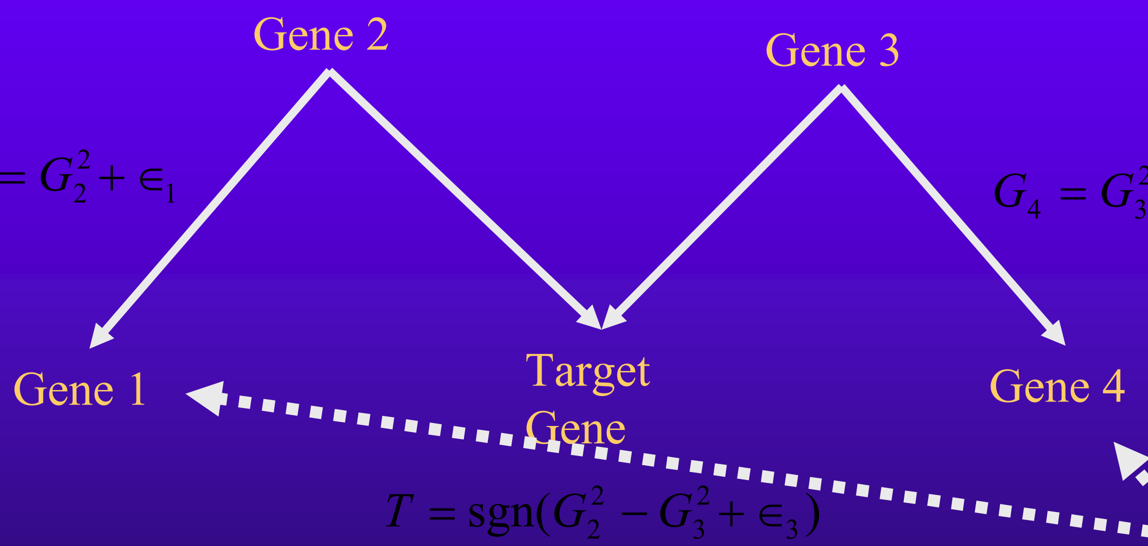
‘If *A* is a robust and strong predictor of *T* then *A* is likely a cause of *T*’

- Example: Feature selection
- Example: Predictive Rules



Causal Heuristic #2

‘If A is a robust and strong predictor of T then A is likely a cause of T’



$$G_1 = G_2^2 + \in_1$$

$$G_4 = G_3^2 + \in_2$$

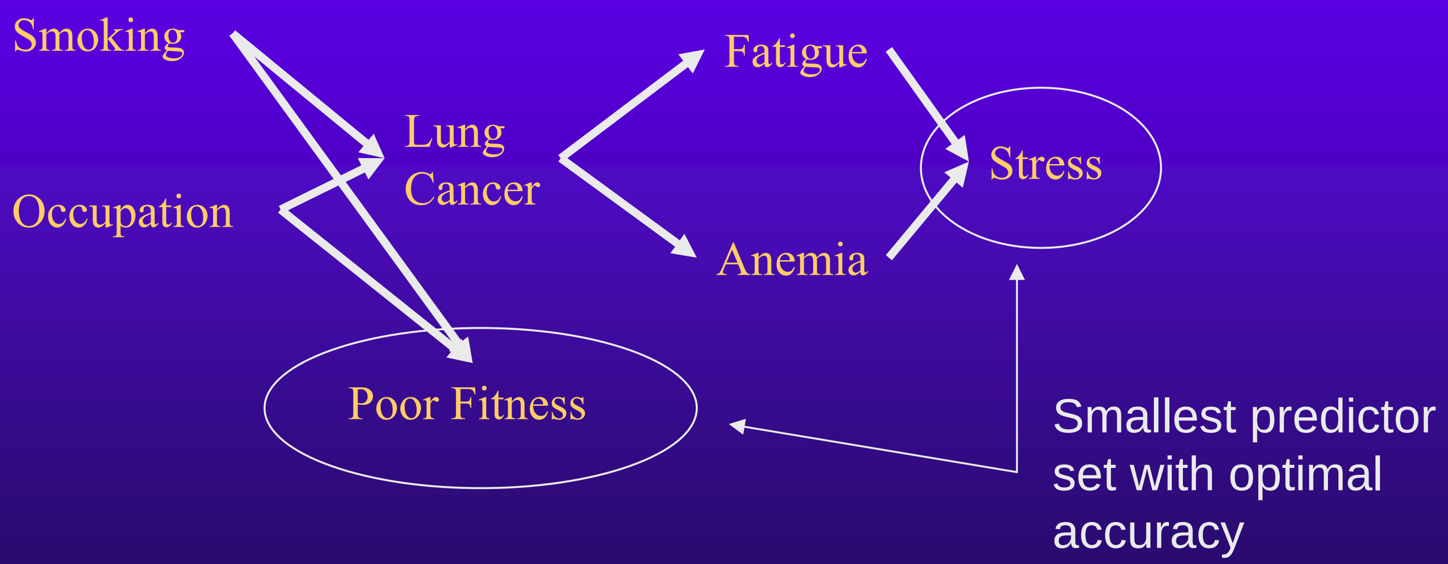
$$T = \mathrm{sgn}(G_2^2 - G_3^2 + \in_3)$$

$$T = \mathrm{sgn}(G_1 - G_4 + \in_1 + \in_2)$$

Linear SVM may select Gene 1 and Gene 2 as the minimal predictor set

Causal Heuristic #3

- ‘The closer *A* and *T* are in a causal sense, the stronger their correlation’ (localizes causality as well)



Causal Heuristic #4

‘If they cluster together they have similar or related function’.

The Problem with Causal Discovery

- Causal heuristics are unreliable
- Causation is difficult to define
- RCTs are not always doable
- Major "causal knowledge" does not have RCT backing!

Formal Computational Causal Discovery from Observational Data

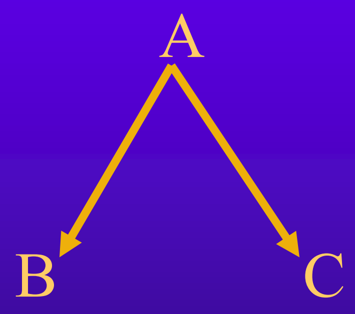
- Formal algorithms exist! Two Nobel prizes in economics in the last 5 years!
- Most are based on a graphical-probabilistic language called "Causal Probabilistic Networks (a.k.a. "Causal Bayesian Networks")
- Well-characterized properties of
  - What types of causal relations they can learn
  - Under 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: A tool for causal discovery

- BN=Graph (Variables (nodes), dependencies (arcs)) + Joint Probability Distribution + Markov Property
- Graph has to be DAG (directed acyclic) in the standard BN model



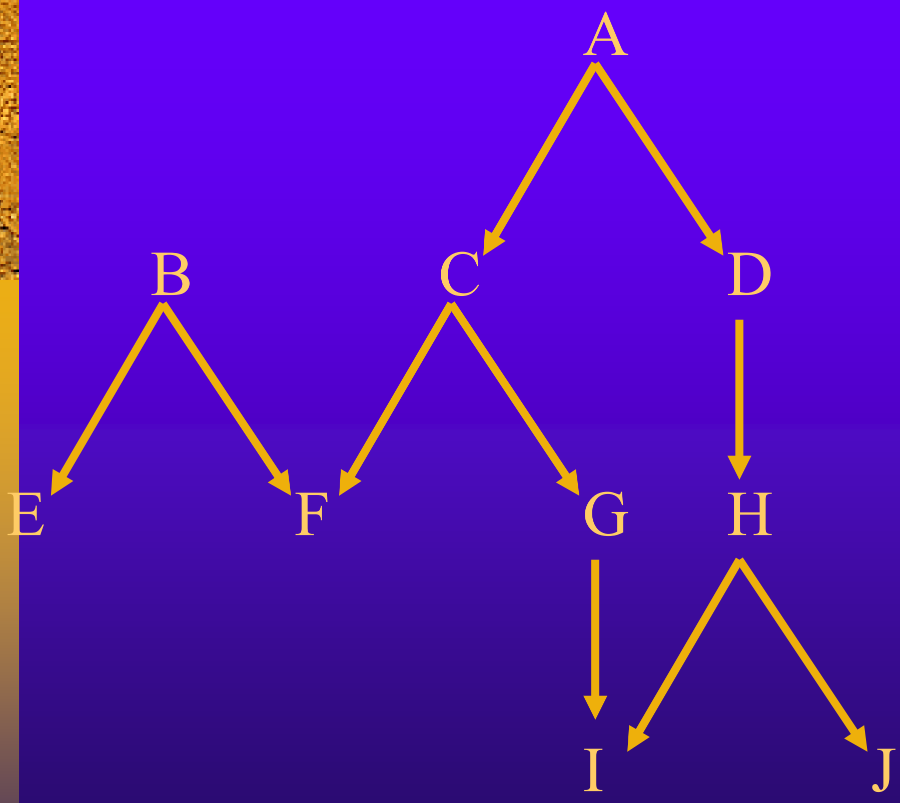
JPD

```
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
```

- Any JPD can be represented in BN form

Bayesian Networks: The Bayesian Network Model and Its Uses

- **<u>Markov Property</u>: the probability distribution of any node *N* given its parent *P* is independent of any subset of the non-descendent nodes *W* of *N***



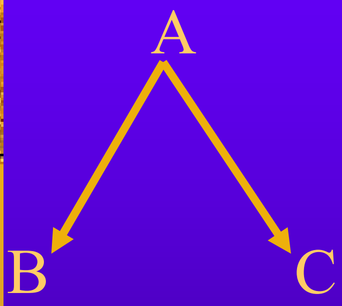
e.g., :

$D \perp \{B,C,E,F,G \mid A\}$

$F \perp \{A,D,E,F,G,H,I,J \mid B, C\}$

Bayesian Networks: The Bayesian Network Model and Its Uses

- The Markov property enables us to decompose (factor) the joint probability distribution into a product of prior and conditional probability distributions



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
```

Up to Exponential Saving in Number of Parameters!

$$P(\boldsymbol{V}) = \prod_{i} p(V_i|Pa(V_i))$$

A Formal Language for Representing Causality

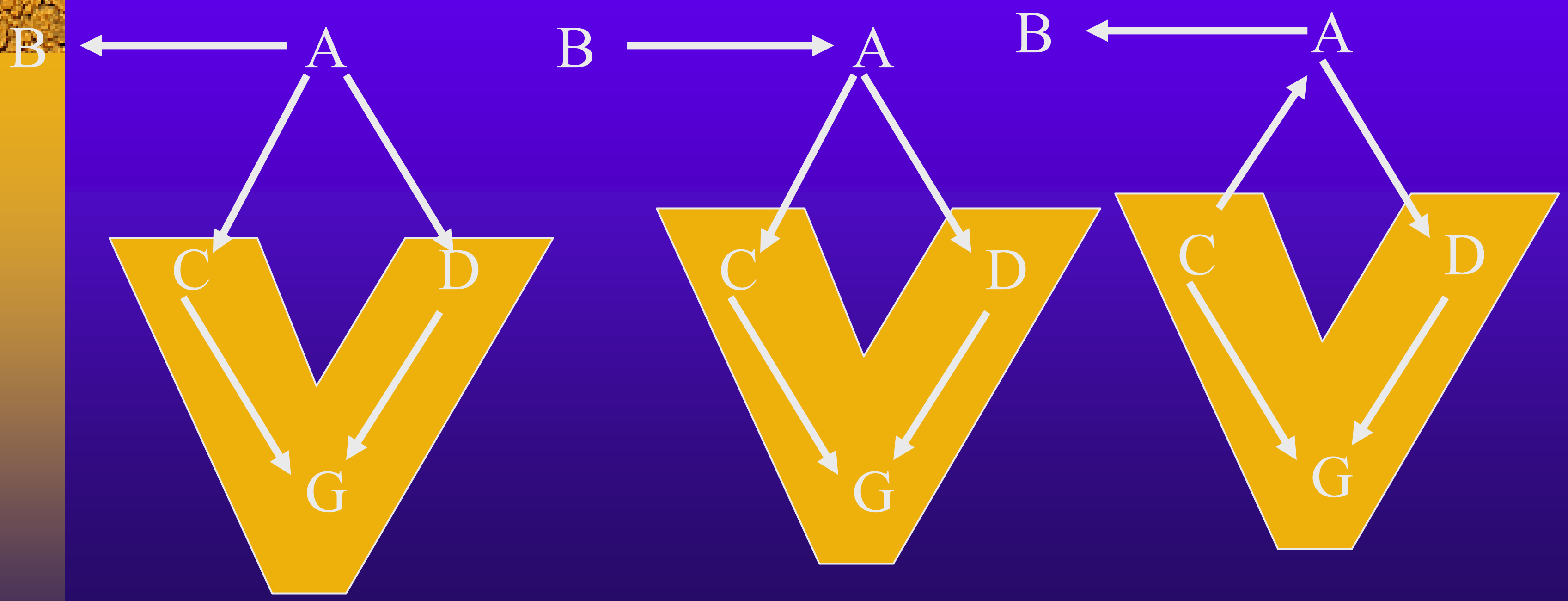
- Bayesian Networks
- Edges: probabilistic dependence
- Markov Condition: A node *N* is independent from non-descendants given its parents
- Probabilistic reasoning

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.
- All such BNs have the same edges (ignoring direction) same v-structures
- Statistically equivalent



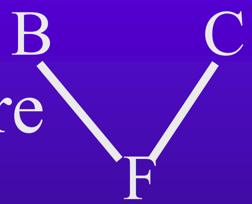
Causal Bayesian Networks

- If there is a (faithful) Causal Bayesian Network that captures the data generation process, it has to have the <u>same edges</u> and same <u>v-structures</u> as any (faithful) Bayesian Network that is induced by the data.
  - We can infer what the direct causal relations are
  - We can infer some of the directions of the edges

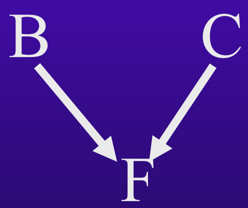
Faithfulness

- Fundamental condition for causal discovery
- Informally a distribution J is faithful to some BN X iff for arbitrary non-overlapping node subsets A,B,C in X, A is dependent of B given C whenever the Markov Condition does not imply that they are independent (otherwise they are independent). Hence X is a perfect dependence/independence map of the distribution
- In other words: when *d*-separation $\Leftrightarrow$ independence
- Faithful distributions constitute the vast majority of theoretical distributions

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)

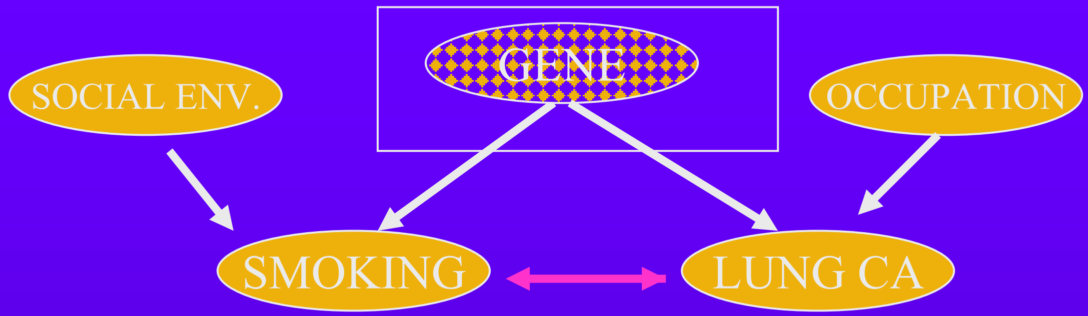
Learning Bayesian Networks: Search-and-Score

- Score each possible structure
- Bayesian score: 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.

Local Causal Discovery- Different Flavor

- Mani&Cooper 2000, 2001, Silverstein, Brin, Motwani, Ullman
- Rule 1: A, B, C pairwise dependent, Ind(A,C|B), A has no causes within the observed variables (e.g. temperature in a gene expression experiment), then
  - $A \rightarrow \ldots \rightarrow B \rightarrow \ldots \rightarrow C$
- Rule 2: $Dep(A,B|\varnothing)$, $Dep(A,C|\varnothing)$, $Ind(B,C|\varnothing)$, Dep(B,C|A), then
  - $B \rightarrow \ldots \rightarrow A \leftarrow \ldots \leftarrow C$
- Discovers a coarser causal model (ancestor relations and indirect causality)

FCI – Causal Discovery with Hidden Confounders



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

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

Other Causal Discovery Algorithms

- Large body of work in Bayesian (or other) search and score methods; still similar set of assumptions (Neapolitan 2004)
- Learning with linear Structural Equation Models in systems in static equilibria (allows feedback loops) (Richardson, Spirtes 1999)
- Learning in the presence of selection bias (Cooper 1995)
- Learning from mixtures of experimental and observational data (Cooper, Yoo, 1999)

In Summary:

- It is possible to perform causal discovery from observational data without Randomized Controlled Trials!
- Heuristic methods are typically used instead of formal causal discovery methods; their properties and their relative efficacy are unknown
- Causal discovery algorithms also make assumptions but have well-characterized properties
- There is a plethora of different algorithms with different properties and assumptions for causal discovery
- There is still plenty of work to be done

Linking Feature Selection and Causal Discovery

Linking Feature Selection and Causal Discovery

<u>Result1</u>

- Given:
  - Faithful distributions
  - Powerful enough learners (with “universal approximator” behavior)
  - Calibrated classification
  - Enough training sample
- The Markov Blanket of a target variable T is the solution to the feature selection problem. Furthermore the Markov Blanket always exists and is unique.

Linking Feature Selection and Causal Discovery

Corollary 1

- Under the conditions of result 1, the solution to the feature selection problem for some variable T contains the direct causes and direct effects of T (as well as the direct causes of the direct effects).

Linking Feature Selection and Causal Discovery

- The most widely used and accepted currently formal definitions of relevancy (Kohavi and John) are *(very* informally*):*
  - Strongly relevant features = always needed to obtain optimal classification
  - Weakly relevant features = contain information about the classification but are not necessarily required to achieve optimal classification
  - Irrelevant features = do not carry any information about the classification

Linking Feature Selection and Causal Discovery

Result2

- The Kohavi and John definitions of relevancy can be cast in terms of Bayes Networks in faithful distributions:
  - Strongly relevant features = members of the Markov Blanket
  - Weakly relevant features = variables with a path to the Markov Blanket but not in the Markov Blanket
  - Irrelevant features = variables with no path to the Markov Blanket

Linking Feature Selection and Causal Discovery

Result3

- Neither Wrappers nor Filters are inherently (i.e., over all possible data distributions and metrics) superior over the other approach. Why?
  – Wrappers are subject to the No Free Lunch Theorem for Optimization => some wrappers will be good for specific distributions and metrics and bad for others; there is no wrapper that dominates over any other over all possible distributions/metrics. A random walk in the space of feature subset is on the average as good as any wrapper.
  – There cannot be a definition of relevancy/filter algorithm that does not take into consideration the distribution and metric

Additional Theoretical Results

Linear (weight-based) SVM-based feature selection:

- Result 4: Identifies all Kohavi-John irrelevant features in faithful distributions
- Result 5: May remove strongly relevant features
- Result 6: May not remove weakly relevant features
- Result 7: Does not have a local causal interpretation

Novel Algorithms

- An algorithm (HITON) for inducing the Markov Blanket of a target variable T
- An algorithm (MMPC) for inducing the direct causes and direct effects of a target variable T
- An algorithm (MMBN) for inducing an undirected Bayesian Network by first inducing the local neighborhoods
- An algorithm (MMHC) for inducing the full Bayesian Network by first inducing the local neighborhoods and then performing search

Experiments

- Application of HITON to 5 different real-life datasets for feature selection
- Application of HITON to 4 different real-life stroke related diagnosis tasks for feature selection
- Application of MMPC, MMMB and MMBN/MMHC in simulated data from real-life BNs to assess causal discovery capabilities for discovery of direct causes/effects, Markov blankets, undirected BNs and full BNs

HITON: An algorithm for feature selection that combines MB induction with wrapping

ALGORITHM	SOUND	SCALABLE	SAMPLE EXPONENTIAL TO |MB|	COMMENTS
Cheng and Greiner	YES	NO	NO	Post-processing on learning BN
Cooper et al.	NO	NO	NO	Uses full BN learning
Margaritis and Thrun	YES	YES	YES	Intended to facilitate BN learning
Koller and Sahami	NO	NO	NO	Most widely-cited MB induction algorithm
Tsamardinos and Aiferis	YES	YES	YES	Some use BN learning as sub-routine
HITON	YES	YES	NO	

HITON: An algorithm for feature selection that combines MB induction with wrapping (rough outline)

- Step #1: Find candidate parents and children of *T;* use conditional independence testing to eliminate false positives; call the resulting set **PC(*T*)**
- Step #2: Find the **PC(.)** set of each member of **PC(*T*);** take the union of all these sets to be **PCunion**
- Step #3: Use conditional independence tests to filter out from **PCunion** the non-members of MB(*T)* that can be identified as such (not all can); call the resultant set TMB (tentative MB)
- Step #4: Apply heuristic search with a desired classifier/loss function and cross-validation to identify variables that can be dropped from TMB without loss of accuracy

HITON (Data *D*; Target *T*; Classifier *A*)
"returns a minimal set of variables required for optimal classification of *T* using algorithm *A*"
MB(T) = HITON-MB(D, T) // Identify Markov Blanket
Vars = Wrapper(*MB(T), T, A*) // Use wrapping to remove unnecessary variables
Return Vars

HITON-MB(Data D, Target T)
"returns the Markov Blanket of *T*"
PC = parents and children of T returned by HITON-PC(D, T)
PCPC = parents and children of the parents and children or T
$CurrentMB = PC \cup PCPC$
// Retain only parents of common children and remove false positives
$\forall$ potential spouse X in *CurrentMB* and $\forall$ *Y in* PC:
  if not$\exists$ S in $\{Y\} \cup \mathbf{V} - \{T, X\}$ so that $\perp (T ; X \mid S)$
  then retain X in *CurrentMB*
  else remove it
Return CurrentMB

HITON-PC(Data D, Target T)
"returns parents and children of *T*"

Wrapper(Vars, T, A)
"returns a minimal set among variables *Vars* for predicting *T* using algorithm *A* and a wrapping approach"
Select and remove a variable.
If internally cross-validated performance of *A* remains the same permanently remove the variable.
Continue until all variables are considered.

HITON-PC(Data D, Target T)

"returns parents and children of *T*"

CurrentPC = {}

Repeat

Find variable V_i *not* in *CurrentPC* that maximizes *association(Vi*, *T)*

and admit V_i into *CurrentPC*

If there is a variable *X* and a subset S of CurrentPC s.t. $\perp(X : T \mid S)$

remove V_i *from CurrentPC;*

mark V_i *and do not consider it again*

Until *no more variables are left to consider*

Return *CurrentPC*

Dataset	Thrombin	Arrythmia	Ohsumed	Lung Cancer	Prostate Cancer
Problem Type	Drug Discovery	Clinical Diagnosis	Text Categorization	Gene Expression Diagnosis	Mass-Spec Diagnosis
Variable #	139,351	279	14,373	12,600	779
Variable Types	binary	nominal/ordinal /continuous	binary and continuous	continuous	continuous
Target	binary	nominal	binary	binary	binary
Sample	2,543	417	2000	160	326
Vars-to-Sample	54.8	0.67	7.2	60	2.4
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.

Figure 2: Dataset Characteristics

1. 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	*N/A*	92.04%	92.65%	*N/A*
Average	91.69%	91.68%	**92.7%**	90.95%
# of variables	34837	8709	**32**	139351

2. Clinical Diagnosis (Arrythmia)				
	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%	**65.85%**	65.15%
# of variables	279	96	**63**	279

3. Text Categorization (OHSUMED)				
	IG	X^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%	**84.79%**	83.04%	84.10%
# of variables	224	112	**34**	14373

4. 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%	**98.33%**
# of variables	330	19	**16**	12,600
5. 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%	**96.14%**	91.87%
# of variables	706	87	**16**	779

Averages Over All Tasks			
	Av. Over BaselineAlgorithms	HITON	ALL
Av. Perf. over classifiers	86.1%	**87.1%**	86.1%
Av. variable #	4540	**32.3**	33,476
Av. reduction	x 8	x **1124**	x 1

Figure 3: Task-specific and average model reduction performance (in bold, best performance per row; asterisks indicate that the corresponding algorithm yield the best model or a non-statistically significantly worse model than the best one).

Diagnosing Stroke with Proteomic Markers

Ischemic vs Hemorrhagic Stroke

	All	with binary discretization			Recursive Feature		Non-recursive SVM		UAF-KW (evaluation by PSVM)
		No wrapper	SVM wrappe	SVM wrappe	Linear	Polynomial	Linear	Polynomial	
PSVM	0.7456	0.6909	0.7781	0.7653	0.7709	0.7998	0.7379	0.7404	0.7722
RSVM	0.8054	0.8126	0.8276	**0.8290**	0.7833	0.8273	0.7657	0.8039	0.8043
Number of features in CV: mean (min-max)	29 (29-29)	2.5 (2-5)	2.4 (1-4)	2.9 (2-5)	7.6 (4-10)	11.9 (9-20)	6.1 (3-9)	12.9 (3-29)	6.9 (3-19)
Number of features in final model	29	2	1	3	10	11	9	14	3

Stroke vs Non-Stroke

	All	with ternary discretization			Recursive Feature Elimination		Non-recursive SVM **FS** (like UAF)		UAF-KW (evaluation by PSVM)
		No wrapper	PSVM wrappe	RSVM wrappe	Linear	Polynomial	Linear	Polynomial	
PSVM	0.8085	0.8020	0.7887	0.7899	0.7915	**0.8316**	0.7779	0.7958	0.7852
RSVM	0.8305	0.8021	0.8122	0.8009	0.8039	0.8264	0.8058	0.8107	0.7956
Number of features in CV: mean (min-max)	29 (29-29)	15.7 (12-19	11.5 (9-14)	12.2 (8-15)	11.6 (9-15)	14.2 (9-19)	12.6 (6-19)	18.6 (12-25	20.6 (5-29)
Number of features in final model	29	18	17	13	13	19	11	21	29

Stroke vs Mimic

	All	HITON with ternary discretization			Recursive Feature Elimination		Non-recursive SVM **FS** (like UAF)		**UAF-KW** (evaluation by PSVM)
		No wrapper	PSVM wrappe	RSVM wrappe	Linear	Polynomial	Linear	Polynomial	
PSVM	0.8821	0.8729	0.8605	0.8664	0.8936	0.8634	**0.8987**	0.8953	0.8813
RSVM	0.8691	0.8725	0.8743	0.8664	0.8707	0.8643	0.8926	0.8918	0.8796
Number of features in CV: mean (min-max)	29 (29-29)	8.7 (4-11)	7.3 (4-10)	7.7 (5-11)	12.3 (8-22)	14.2 (10-25)	14 (12-16)	14.1 (11-19)	25.2 (19-29)
features in final model	29	10	6	7	9	10	13	15	27

Ischemic vs Mimic

	All	HITON with ternary discretization			Recursive Feature Elimination		Non-recursive SVM FS (like UAF)		UAF-KW (evaluation by PSVM)
		No wrapper	PSVM wrappe	RSVM wrappe	Linear	Polynomial	Linear	Polynomial	
PSVM	0.8463	**0.8558**	0.8268	0.8455	0.8445	0.8400	0.8415	0.8396	0.8328
RSVM	0.8379	0.8394	0.8323	0.7931	0.8445	0.8250	0.8498	0.8170	0.8448
Number of features in CV: mean (min-max)	29 (29-29)	11.9 (8-14)	9.4 (7-13)	9 (5-11)	17.4 (7-23)	14.8 (5-29)	19.7 (13-26	18.8 (11-29)	13.4 (7-29)
Number of features in final model	29	14	11	12	16	11	15	15	29

Experiments with learning partial and full Bayesian Networks when the true structure is known

- State-of-the-art algorithms in Bayesian Network learning from data do not scale up to more than a few hundred variables

MMBN: An algorithm to learn the skeleton of a BN (rough outline)

- <u>Step #1</u>: Find candidate parents and children of every variable *V*i in ***V****;* use conditional independence testing to eliminate false positives; call the resulting set **PC(*V*i*)***
- <u>Step #2</u>: return an undirected graph such that there is an edge between *X* and *Y* if and only if *X* is in **PC(*Y*)** *Y* is in **PC*(X*),** where *X, Y* are in ***V***

- ***Note:*** a special variant builds a region of radius *d* of the network around *T* using a breadth-first strategy

Simulated Bayesian Networks

- Needed networks that resemble real processes as much as possible
- Needed a way to vary the size of the networks to test scalability
- Identified a network used in a real decision support system (the ALARM network)
- Designed a method (tiling) to generate new larger networks from the original ALARM in a way that shares the structural and probabilistic properties

Experiment 1: MMBN vs State-of-the-Art Other Algorithms on ALARM

- 1000 randomly generated samples from the joint distribution of ALARM were fed to the algorithm
- Other algorithms Sparse Candidate using the Mutual Information and Scoring heuristic, *k*=10, TPDA, and PC
- All algorithms took less than a couple of minutes to complete on a Pentium Xeon with 2.4GHz

Algorithm	**Sensitivity**	**Specificity**	**Distance**
PC	98%	93%	7%
TPDA	91%	96%	9%
SC/MI	98%	94%	6%
SC/Sc	96%	94%	7%
MMBN	98%	95%	5%

Experiment 2:

- Tiled 270 copies of original ALARM, randomly interconnected them (tiling algorithm) (approximately 10,000 variables)
- Randomly sampled 1,000 training instances from the joint distribution of the network

Algorithm	Sens.	Spec.	Dist.	Time
MMBN	81%	99.9%	18%	62 hours

- Observation: specificity improves as variables increase. Why?
- The rate of increase of false positives (that reduce specificity) is lower than the rate of increase of true negatives

Experiment 3: Reconstructing the Local Neighborhood

- What if the number of variables is extremely large (e.g., millions)?
- Modified the algorithm to reconstructing the network in an inside-out fashion (breath-first-search) starting from a target node of interest
- Algorithm now returns the network in a radius *d* edges away from target *T*

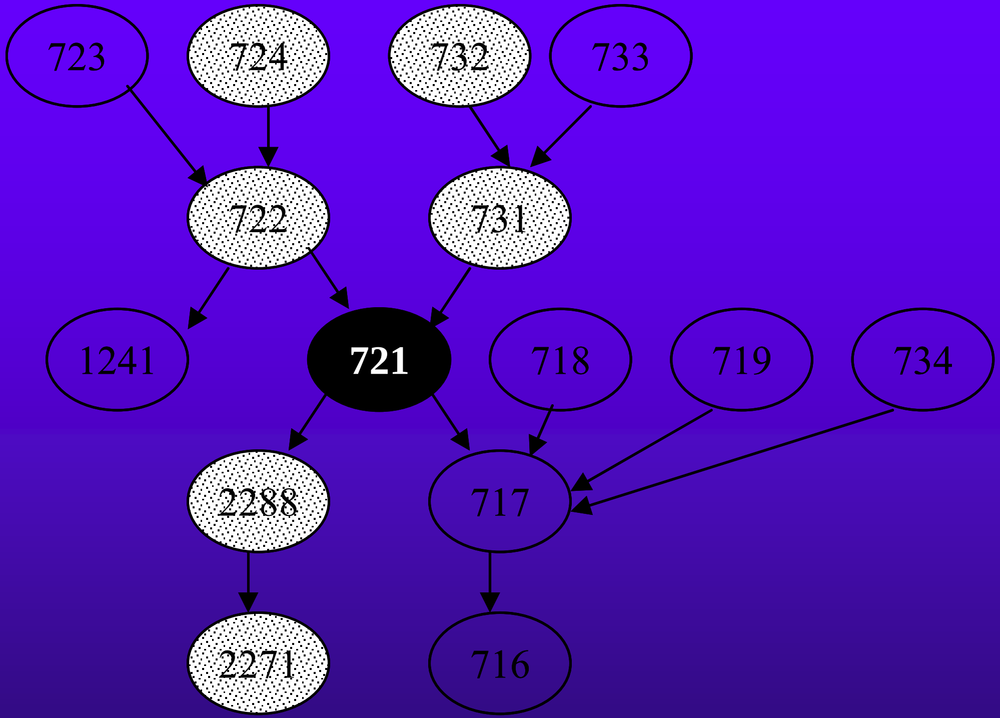
Experiment 3: Results

- Sampled 1,000 instances from 10,000 variable tiled ALARM
- Reconstructed each neighborhood of radius 1,2,3 and 4 with target each node
- Calculated the average performance metrics

Depth	Sens.	Spec.	Dist.
1	79.17%	100.00%	20%
2	67.48%	100.00%	32%
3	59.56%	100.00%	40%
4	52.84%	100.00%	53%

Experiment 3: Discussion

- Sensitivity quickly drops: missing one node at the previous level does not expand that node leading to miss more nodes
- Textured nodes are the true positives
- Example taken for depth $d=2$ and for a node with the average sensitivity and specificity



MMMB: An algorithm to induce the Markov Blanket (rough outline)

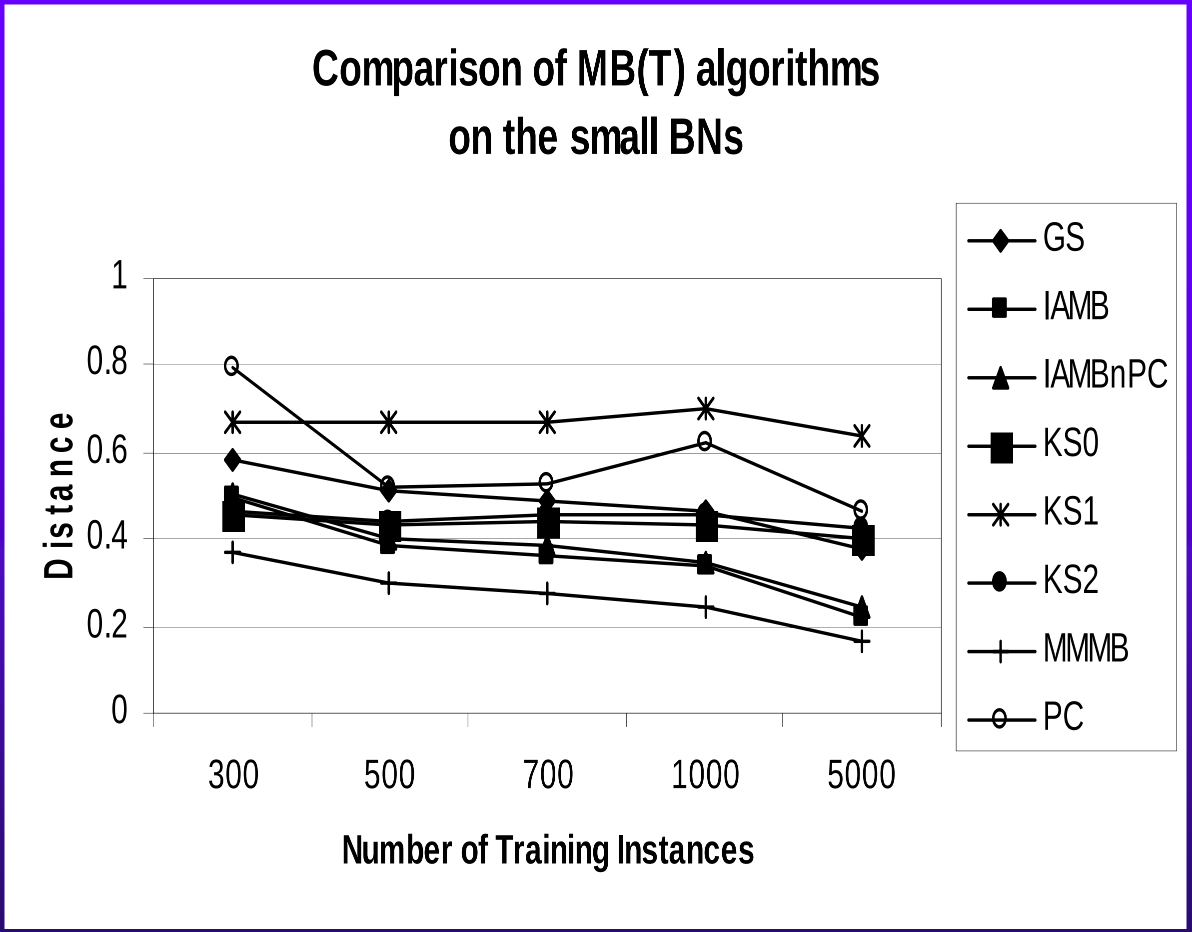
- Step #1: Find candidate parents and children of *T;* use conditional independence testing to eliminate false positives; call the resulting set **PC(*T*)**
- Step #2: Find the **PC(.)** set of each member of **PC(*T*);** take the union of all these sets to be **PCunion**
- Step #3: Use conditional independence tests to filter out from **PCunion** the non-members of MB(*T)* that can be identified as such (not all can); call the resultant set TMB (tentative MB)

- Note: very similar to HITON but omits the wrapping part and uses a more complicated heuristic for step #1

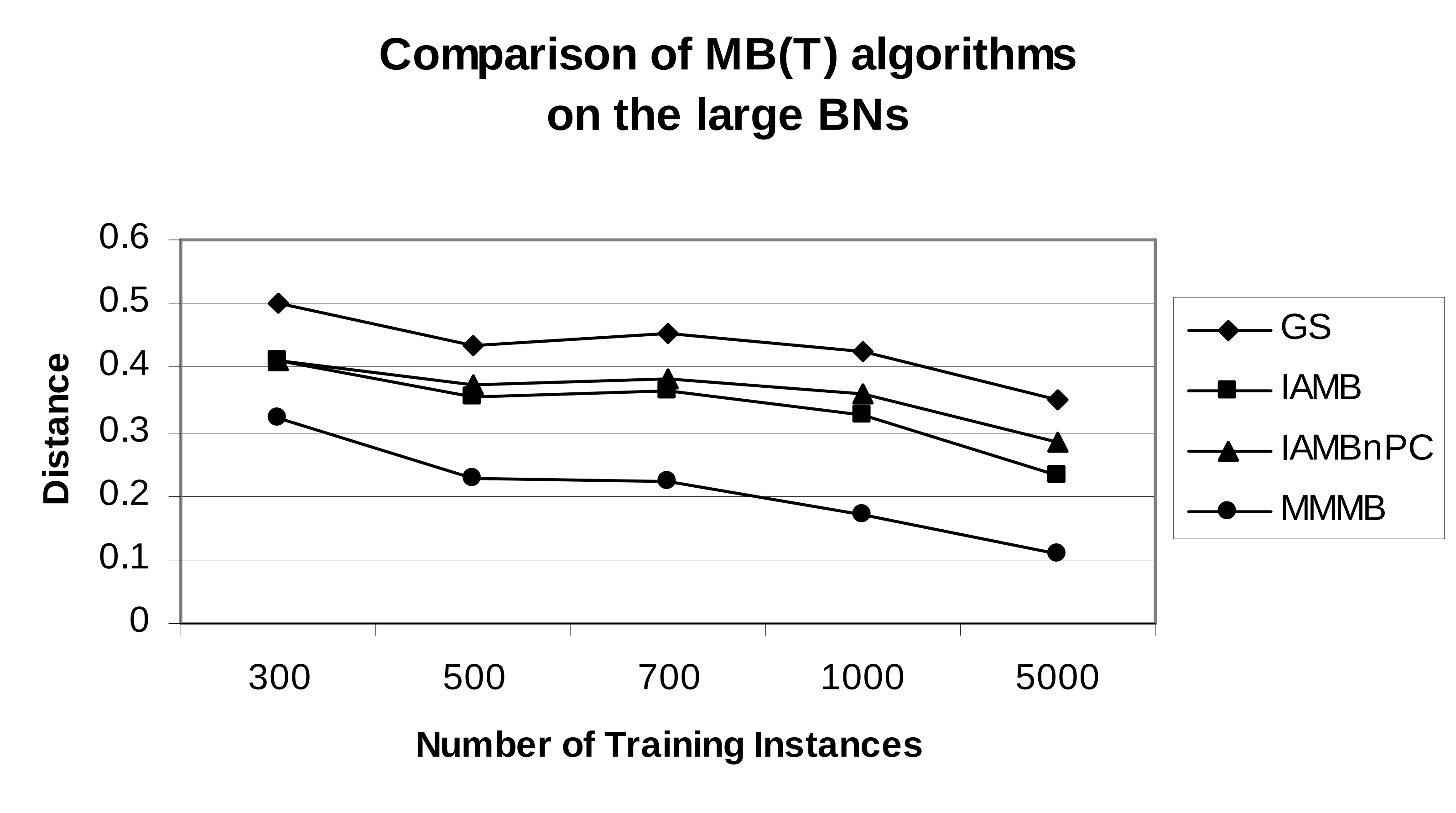
Datasets

- 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

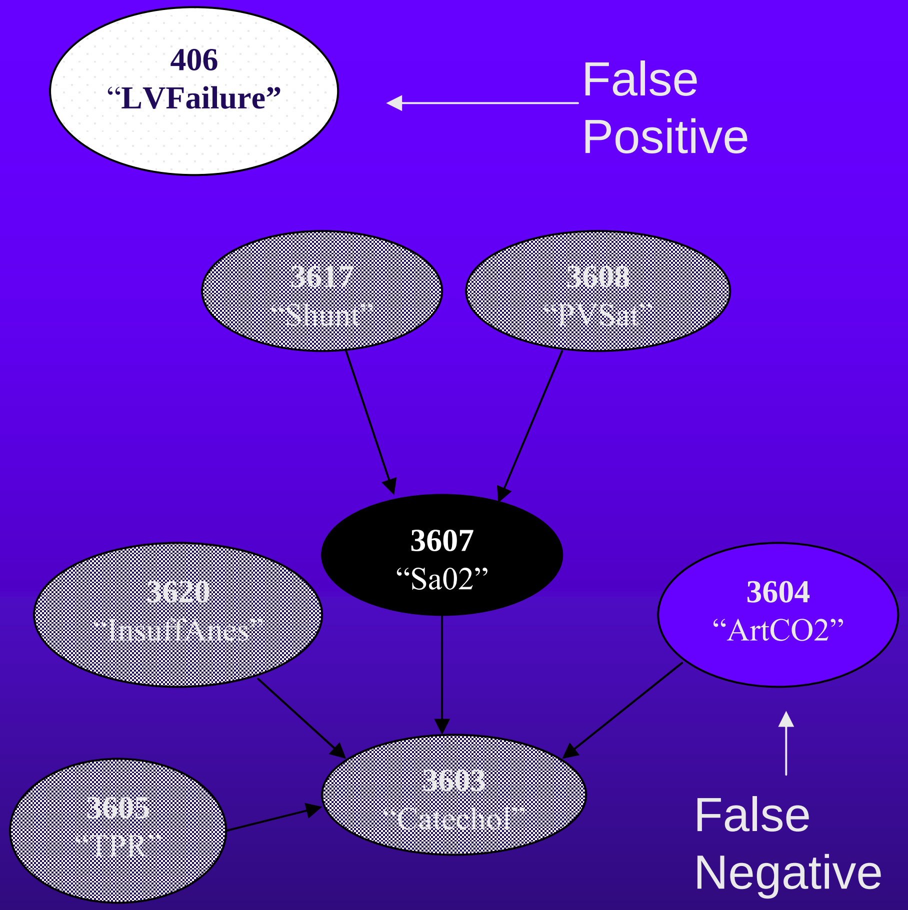


Discovering the Markov Blanket



Discovering the Markov Blanket

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



Reconstructing the Full Bayesian Network

- Algorithm <u>Max-Min Hill Climbing</u>:
  - Similar to MMBN but also orients the edges by applying Bayesian search-and-score (search=hill climbing with operators add edge, remove edge, orient edge, change orientation of edge; score is typically BDeu)
- 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 (log of the probability of the BN given the data)
  - Number of structural errors: wrong additions, deletions, or reversal of edges

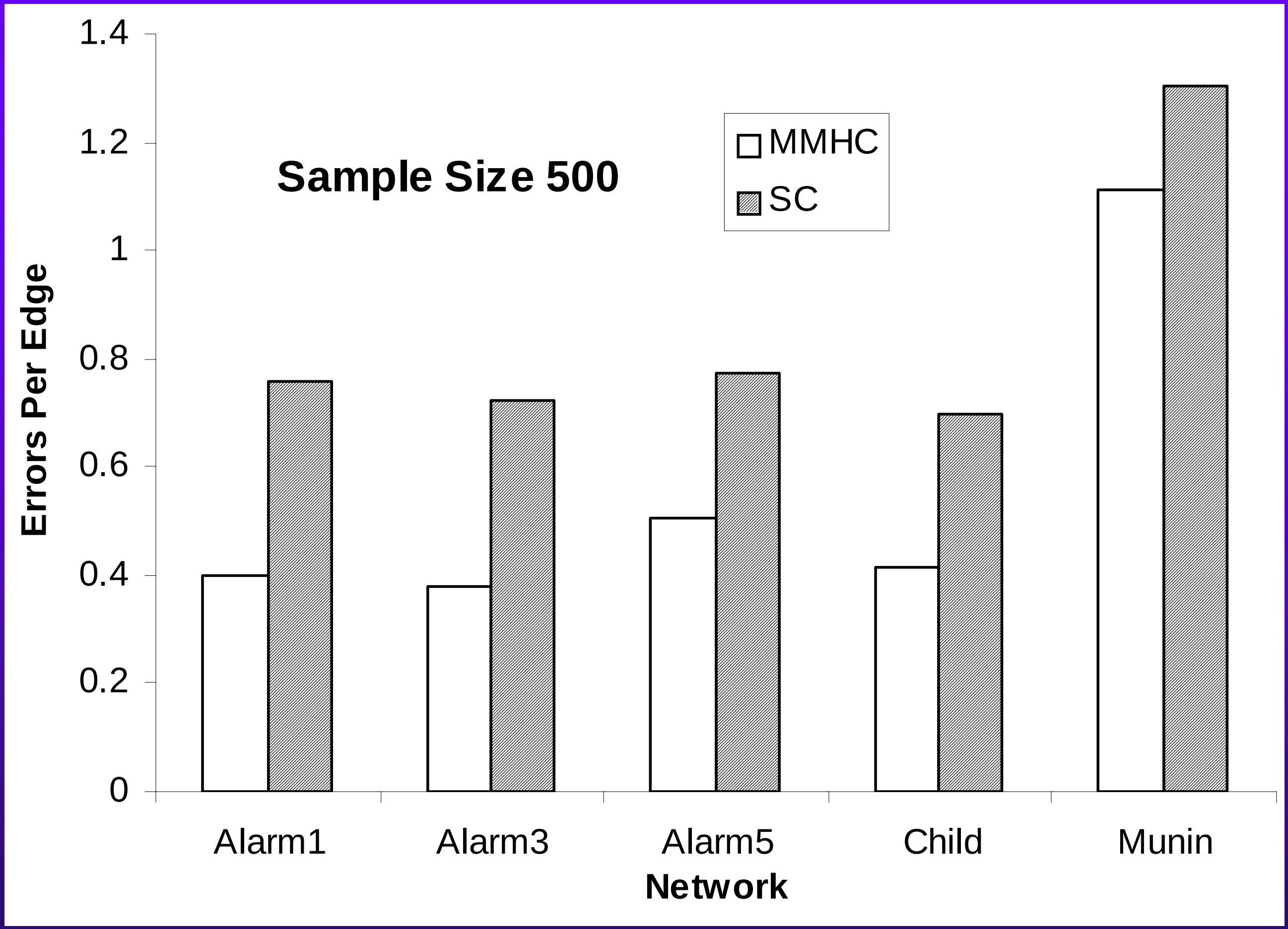
Bayesian Networks

Name	# Vars	# Edges	Description
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

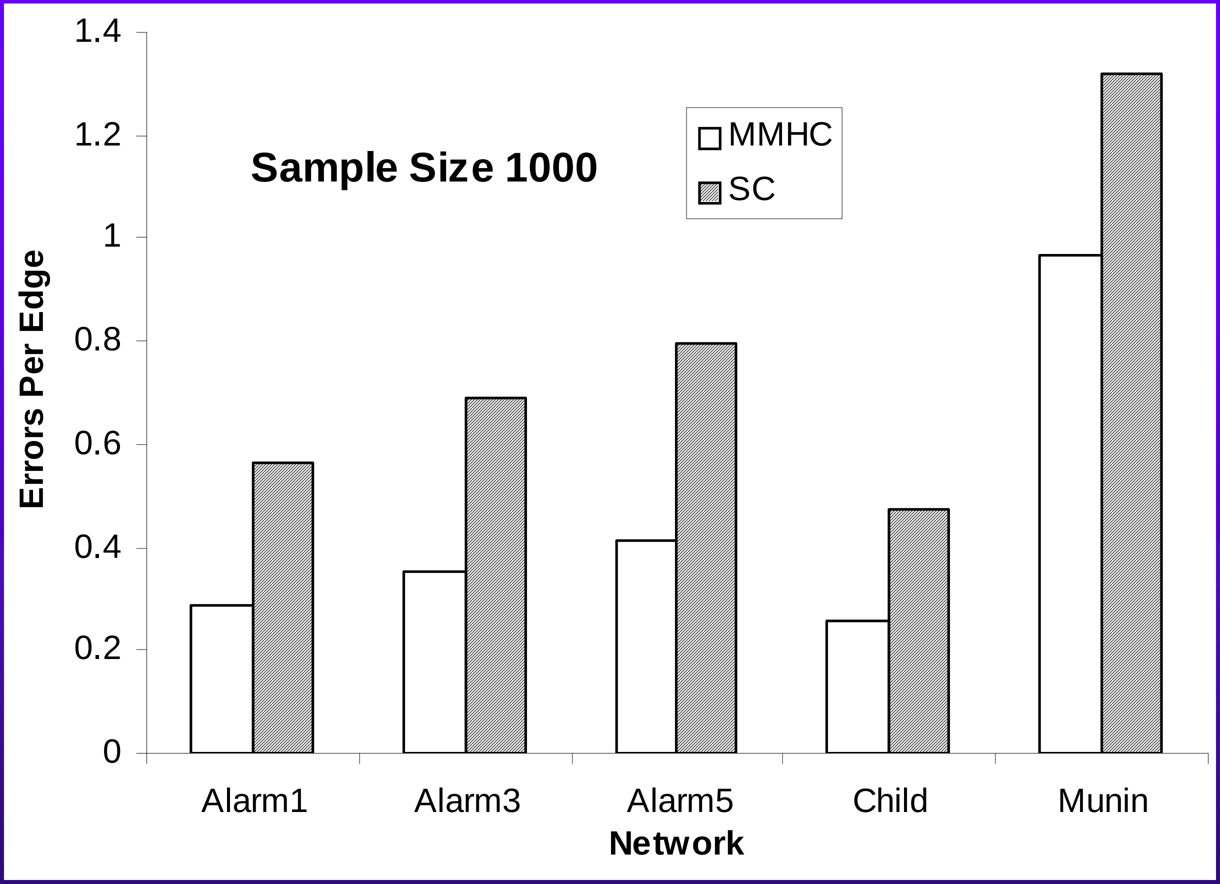
MMHC versus Sparse Candidate

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

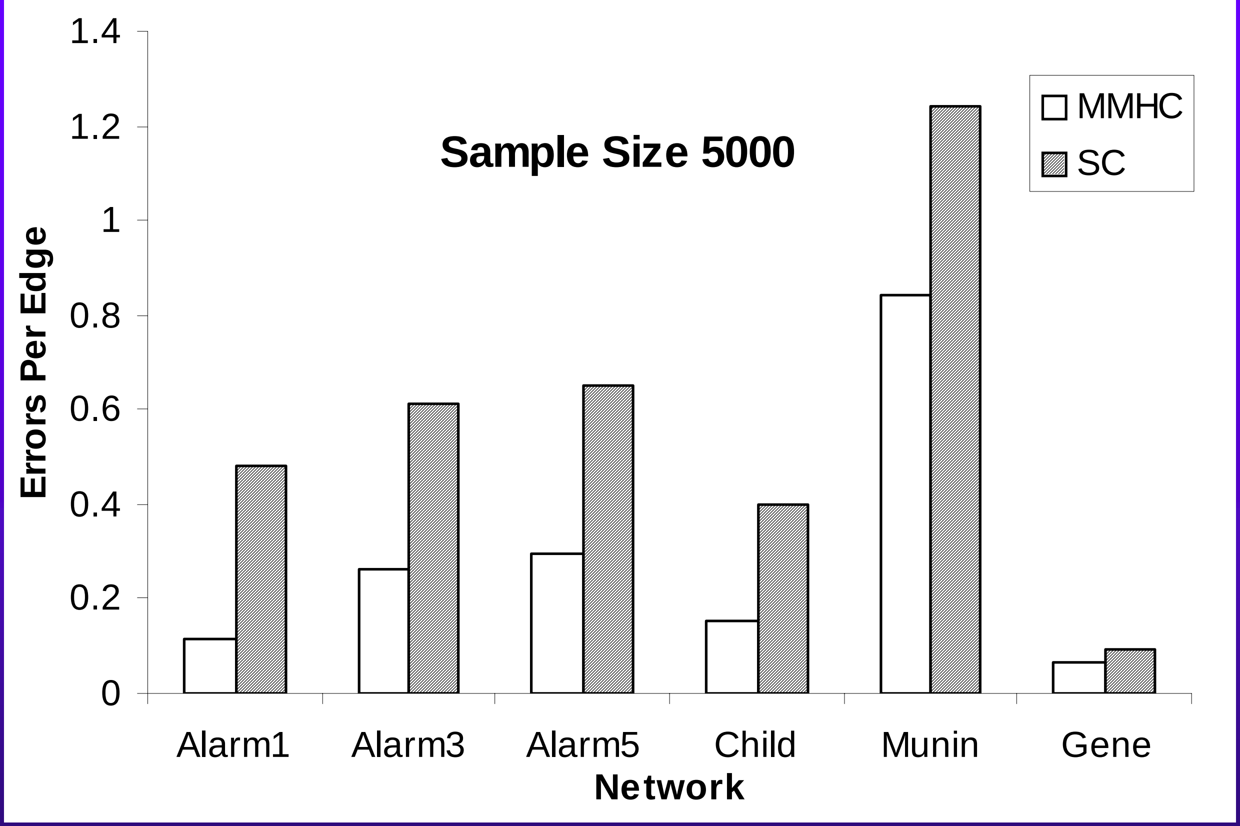
MMHC versus Sparse Candidate



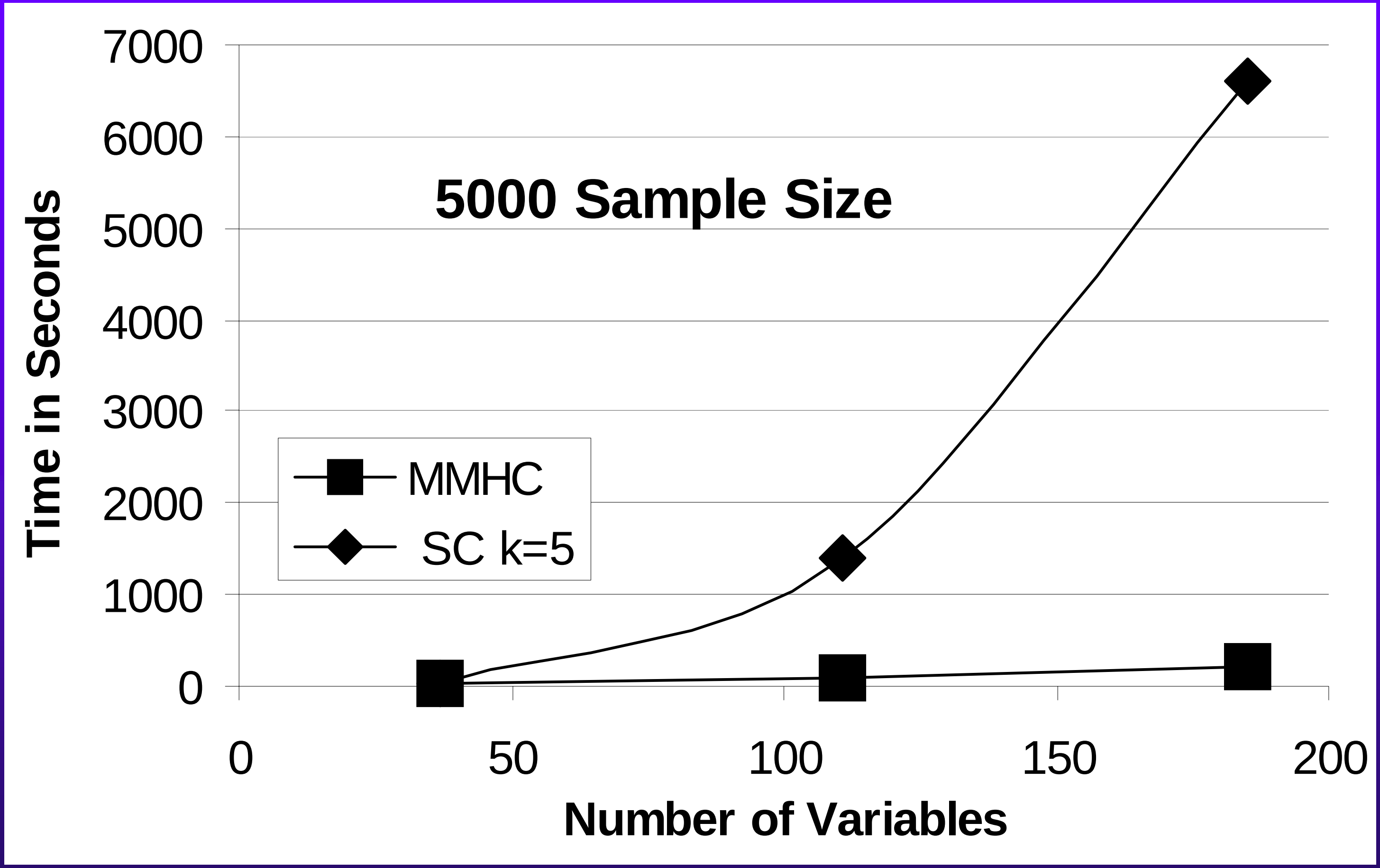
MMHC versus Sparse Candidate



MMHC versus Sparse Candidate



MMHC versus Sparse Candidate



Conclusions: Feature selection

- The new Markov Blanket algorithms find the smallest subset with maximum or near-maximum classification performance always outperforming in feature set parsimony and often in classification performance a variety of established methods in several datasets spanning diverse biomedical domains

Conclusions: Local Causal Discovery

- When the generating network is known it was shown that the new algorithms discover the true local neighborhood more reliably than state-of-the-art algorithms and more efficiently in sample and time

Conclusions with respect to full BN induction/Global Causal Discovery

- "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
- We showed that the new causally oriented feature selection algorithms can learn accurately the Markov Blanket, or the neighborhoods around a target variable, and using a divide-and-conquer approach the full network, or just the skeleton of the full network

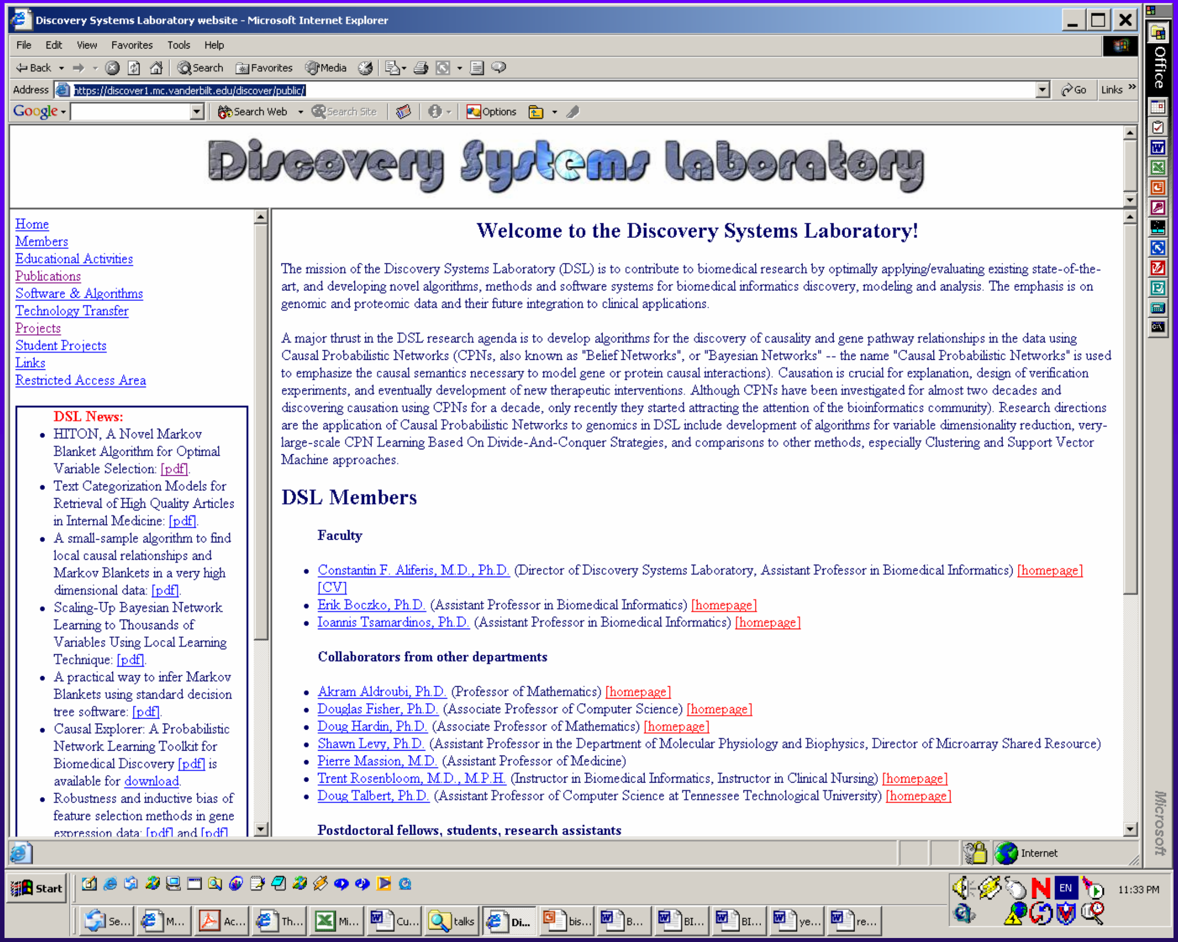
Open Problems & Future Work

- Relax the assumptions (faithfulness)
- Extend the investigation of how well current assumptions apply to practical distributions in biomedicine
- Methods for non-uniform cost feature selection
- Better theoretical understanding of behavior of non-MB methods
- Addressing other types of causal questions

Discovery Systems Laboratory

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

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



Discovery Systems laboratory

Home
Members
Educational Activities
Publications
Software & Algorithms
Technology Transfer
Projects
Student Projects
Links
Restricted Access Area

DSL News:

- HITON, A Novel Markov Blanket Algorithm for Optimal Variable Selection: [pdf].
- Text Categorization Models for Retrieval of High Quality Articles in Internal Medicine: [pdf].
- A small-sample algorithm to find local causal relationships and Markov Blankets in a very high dimensional data: [pdf].
- Scaling-Up Bayesian Network Learning to Thousands of Variables Using Local Learning Technique: [pdf].
- A practical way to infer Markov Blankets using standard decision tree software: [pdf].
- Causal Explorer: A Probabilistic Network Learning Toolkit for Biomedical Discovery [pdf] is available for download.
- Robustness and inductive bias of feature selection methods in gene expression data: [pdf] and [pdf]

Welcome to the Discovery Systems Laboratory!

The mission of the Discovery Systems Laboratory (DSL) is to contribute to biomedical research by optimally applying/evaluating existing state-of-the-art, and developing novel algorithms, methods and software systems for biomedical informatics discovery, modeling and analysis. The emphasis is on genomic and proteomic data and their future integration to clinical applications.

A major thrust in the DSL research agenda is to develop algorithms for the discovery of causality and gene pathway relationships in the data using Causal Probabilistic Networks (CPNs, also known as "Belief Networks", or "Bayesian Networks" -- the name "Causal Probabilistic Networks" is used to emphasize the causal semantics necessary to model gene or protein causal interactions). Causation is crucial for explanation, design of verification experiments, and eventually development of new therapeutic interventions. Although CPNs have been investigated for almost two decades and discovering causation using CPNs for a decade, only recently they started attracting the attention of the bioinformatics community). Research directions are the application of Causal Probabilistic Networks to genomics in DSL include development of algorithms for variable dimensionality reduction, very-large-scale CPN Learning Based On Divide-And-Conquer Strategies, and comparisons to other methods, especially Clustering and Support Vector Machine approaches.

DSL Members

Faculty

- Constantin F. Aliferis, M.D., Ph.D. (Director of Discovery Systems Laboratory, Assistant Professor in Biomedical Informatics) [homepage] [CV]
- Erik Boczko, Ph.D. (Assistant Professor in Biomedical Informatics) [homepage]
- Ioannis Tsamardinos, Ph.D. (Assistant Professor in Biomedical Informatics) [homepage]

Collaborators from other departments

- Akram Aldroubi, Ph.D. (Professor of Mathematics) [homepage]
- Douglas Fisher, Ph.D. (Associate Professor of Computer Science) [homepage]
- Doug Hardin, Ph.D. (Associate Professor of Mathematics) [homepage]
- Shawn Levy, Ph.D. (Assistant Professor in the Department of Molecular Physiology and Biophysics, Director of Microarray Shared Resource)
- Pierre Massion, M.D. (Assistant Professor of Medicine)
- Trent Rosenbloom, M.D., M.P.H. (Instructor in Biomedical Informatics, Instructor in Clinical Nursing) [homepage]
- Doug Talbert, Ph.D. (Assistant Professor of Computer Science at Tennessee Technological University) [homepage]

Postdoctoral fellows, students, research assistants