Markov Blanket Algorithms for Feature Selection

C.F. Aliferis M.D., Ph.D.

Discovery Systems Laboratory
Department of Biomedical Informatics
Vanderbilt University

Joint work with:

Ioannis Tsamardinos Ph.D.

Overview

- Background, Motivation, Prior Work
- A Framework for Feature Selection
- Initial results
- Conclusions & Future Work

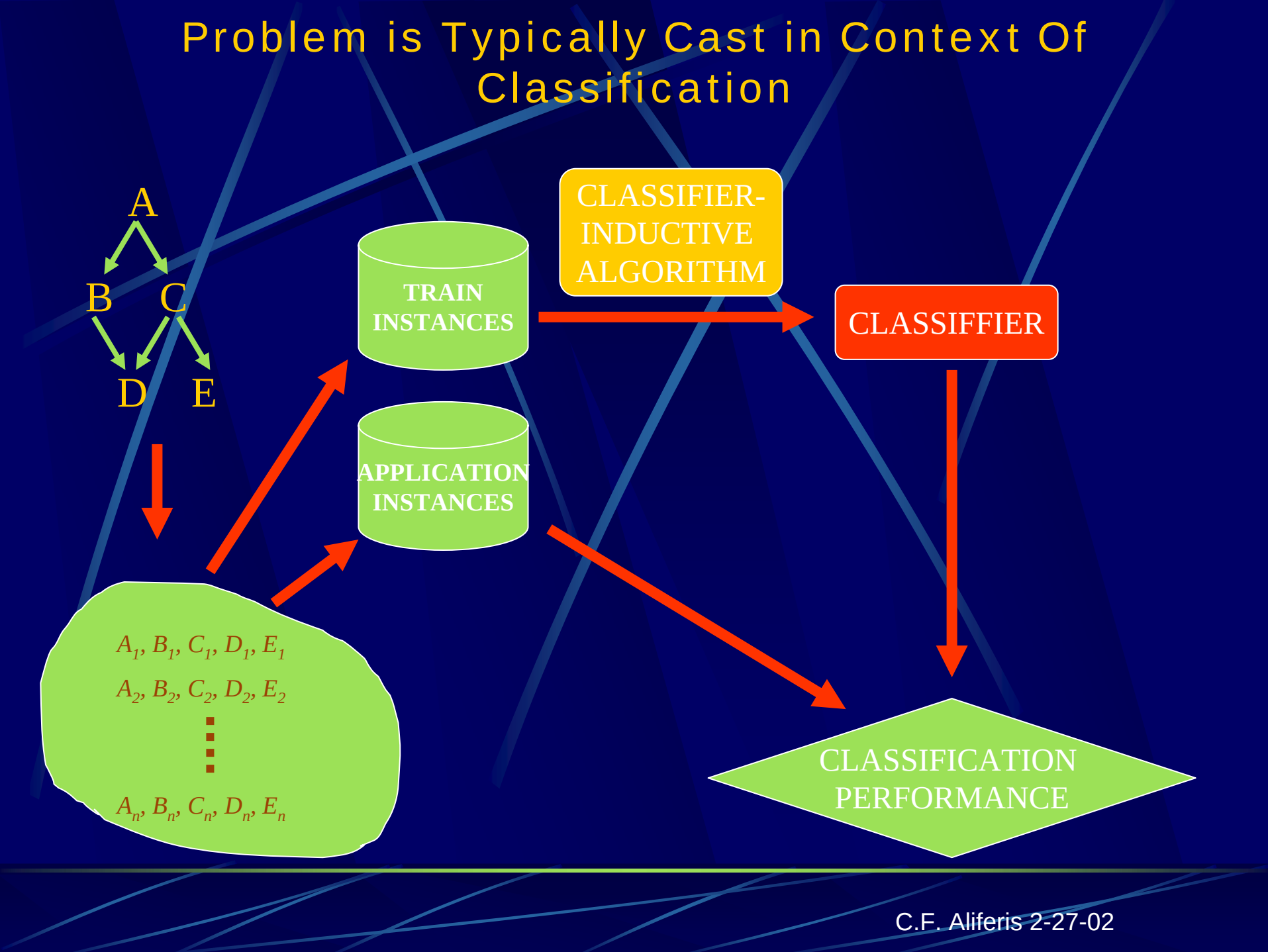
Background

- Datasets with very large numbers of variables are emerging very rapidly in several scientific and commercial settings
  - Genomics
  - Proteomics
  - Clinical Databases
  - Web-derived data: customer transactions, network traffic information, consumer activity profiles,…

Background

- Quantitative modeling is important in many fields:
  - computer science, machine learning, data mining, statistics, engineering…
- At the core of modeling using data is the need to separate the *relevant* from the *irrelevant* features
- During the last three decades, a vast literature on methods for *feature selection* has emerged
- This is still a formidable problem!

Problem is Typically Cast in Context Of Classification



A Terminology Note:

- I will use *Feature* interchangeably with *Variable*
- *I will use Classification* interchangeably with
  - *Diagnosis,*
  - *Prediction,* and
  - *Prognosis*

An Application Note:

- Although feature selection research has traditionally been geared toward classification, in many situations we need to select features to perform induction of causal models in the hope of understanding a data-generating process (causal hypothesis generation, causal modeling) and/or manipulating it.

What Happens If Many Features Are Irrelevant?

- In the sample limit, and assuming infinite time to construct a classifier, the more features we have the better
- In practice, overabundance of features is undesirable for most classifier inducing algorithms because the sample size gets fragmented into exponential many segments (to the number of features).
- This in turn makes estimation of distributions difficult and increases sensitivity to noise within each sample segment.
- As a result, irrelevant features can cause poor generalization for many learning algorithms
- In learning experiments involving most practical algorithms a good selection of features often yields models with better generalization performance than when using the full feature set

What Happens If Many Features Are Irrelevant?

- Algorithms that are robust to a low feature-to-sample ratio do exist (SVMs, to a lesser extent NNs)
- For those and from an applications perspective, inclusion of irrelevant features is still undesired because the produced models :
  - are unnecessarily expensive to use in terms of cost of observing features
  - are more difficult to understand
  - are more computationally expensive to execute in practical settings
  - make understanding of the characteristics and structure of the problem domain problematic

Prior Work

- Statistical and engineering methods:

  - projection-based:
    - PCA/SVD
  - selection-based:
    - Forward
    - Backward
    - or Combined steepest-ascent greedy search

Prior Work

- Machine Learning Approaches:
  - *Wrapper* feature selection algorithm = a search procedure in the space of all possible feature subsets, that uses the classifier algorithm that will be used to induce the final classifier as evaluation function for assessing the quality of feature subsets.
    - Examples: run Genetic Algorithms embedding Neural Networks, or KNN; greedy forward selection embedding Simple Bayes or Decision Tree Induction

Prior Work

- Machine Learning Approaches:
  - *Filter* feature selection algorithm = an algorithm that selects features *independently* of the classifier algorithm that will be used to induce the final classifier.
    - *Examples:* Koller-Sahari, K2MB, Information-Theoretic algorithms, FOCUS, RELIEF, etc.

Prior Work

- Shortcomings of statistical and engineering methods:

  - projection-based:

    PCA/SVD → they create artificial features that make induction of a good classifier easier without telling us which features are important and why

  - selection-based:

    they typically fail to include the best feature subset (not sound); do not take into account feature non-linear interactions or do so in very restrictive ways; several non-conclusive and contradictory studies

Prior Work

- Pros & Cons of Machine Learning Approaches:
  - *Wrapper* feature selection algorithms
    - Pros: take into account interaction of classifier inducing algorithm and the feature selection process
    - Cons: Slow (run classifier induction for each feature subset examined); no guaranteed soundness (because of poorly-characterized heuristic search space); need separate feature selection procedure for each new classifier induction algorithm used

Prior Work

- Pros & Cons of Machine Learning Approaches:
  - *Filter* feature selection algorithm
    - Cons: do not take into account interaction of classifier inducing algorithm and the feature selection process, hence they are not sound
    - Pros: Faster (run classifier induction only after best feature subset is found); do not need separate feature selection procedure for each new classifier induction algorithm used

A Framework for Feature Selection: Strategy

- Define application goals
- Define *Relevance*
- Specify general conditions that allow algorithms that combine the advantages of filter methods with those of wrapper methods while avoiding their corresponding shortcomings
- Develop a theory that guides development of algorithms
- Use the theory to develop algorithms
- Characterize properties of resulting algorithms as precisely as possible in terms of:
  - Assumptions
  - Soundness
  - Completeness
  - Complexity
- Test algorithms with synthetic data
- Test with real data

A Framework for Feature Selection: Goals

- Select optimal features subset for classification
- Select optimal features subset for causal manipulation
- Select optimal features subset for local causal discovery

A Framework for Feature Selection: Goals

- Definition 1. Feature Selection Problem for Classification (FSPC).
  - Given: a set of features F and a target variable T instantiated by some process P; a classification algorithm $A(D_{tr}, F_i) \rightarrow a_i$ (where D_{tr} is any set of training instances, a_i is the model or class of models returned by $A()$ for some subset of F, F_i); and a classification performance metric M, defined over the space of the outputs of $A()$ and the space of test sets D_{te}.
  - Find: a subset F_s of F s.t. no subset F_s' of F yields higher or equal performance than F_s and has smaller cardinality than F_s.

A Framework for Feature Selection: Goals

- Definition 2. Feature Selection Problem for Causal Manipulation (FSPCM).
  - Given: a set of features *F*, a subset of *F* , *MF* of manipulatable features, a target variable *T*, a target value *t* of *T.*
  - Find: the minimal subset of *MF* called *MMB* (manipulatable markov blanket) for which there is an instantiation *mmb* such that p (*T* = *t* | *MMB*=*mmb*) is maximum.

A Framework for Feature Selection: Goals

- Definition 3. Feature Selection Problem for Local Causal Discovery (FSPLCD).
  - Given: a set of features *F*, a subset of *F* ,a target variable *T*.
  - Find: the set of direct causes and direct effects of *T*.

A Framework for Feature Selection: Relevance

- After Kohavi and John:
  - Definition 4: Strongly relevant feature. A feature X_i is strongly relevant iff there exists some x_i, t, and s_i for which $p(X_i = x_i, S_i = s_i)>0$ s.t.: $p(T=t \mid X_i = x_i, S_i = s_i) <> p(T=t \mid S_i= s_i)$
  - Definition 5: Weakly relevant feature. A feature X_i is weakly relevant iff it is not strongly relevant, and there exists a subset of features S_i' of S_i for which there exists some x_i, t, and s_i' with $p(X_i = x_i, S_i' = s_i') >0$ s.t.: $p(T=t \mid X_i = x_i, S_i' = s_i') <> p(T=t \mid S_i'= s_i')$
  - Definition 6: Relevant feature. A feature is relevant if it is weakly or strongly relevant.
  - Definition 7: Irrelevant feature. A feature is irrelevant if it not relevant.

A Framework for Feature Selection: Relevance

- The intuition behind these definitions is straightforward: :
  - In the sample limit the best possible classifier is the optimal Bayes Classifier.
  - If by excluding a feature from the predictor set we get sub-optimal performance, this feature is considered thus to be necessary for optimal performance and labeled as strongly relevant.
  - If a feature is not strongly relevant but still conveys information about the target, we label it as weakly relevant.
  - If a feature is not weakly or strongly relevant (i.e., does not convey information about the target) we label it as irrelevant.

A Framework for Feature Selection: General Application Conditions

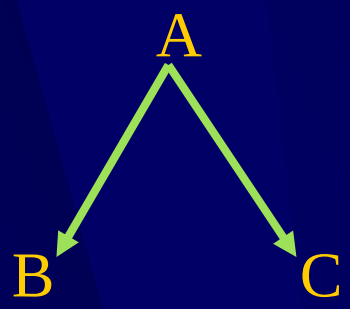
- Assumption I: All data is generated by processes that can be faithfully represented by Causal Probabilistic Networks.
- Assumption II: Classification is supervised.
- Assumption III: Classifiers may need to be calibrated.
- Assumption IV: i.i.d. data

A Framework for Feature Selection: Develop A Theory Guiding Algorithm Development

- Identify relevant Bayesian Network Properties & Algorithms
- Tie Bayesian Networks with the three feature selection problems

A Framework for Feature Selection: Bayesian Network Properties & Algorithms

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



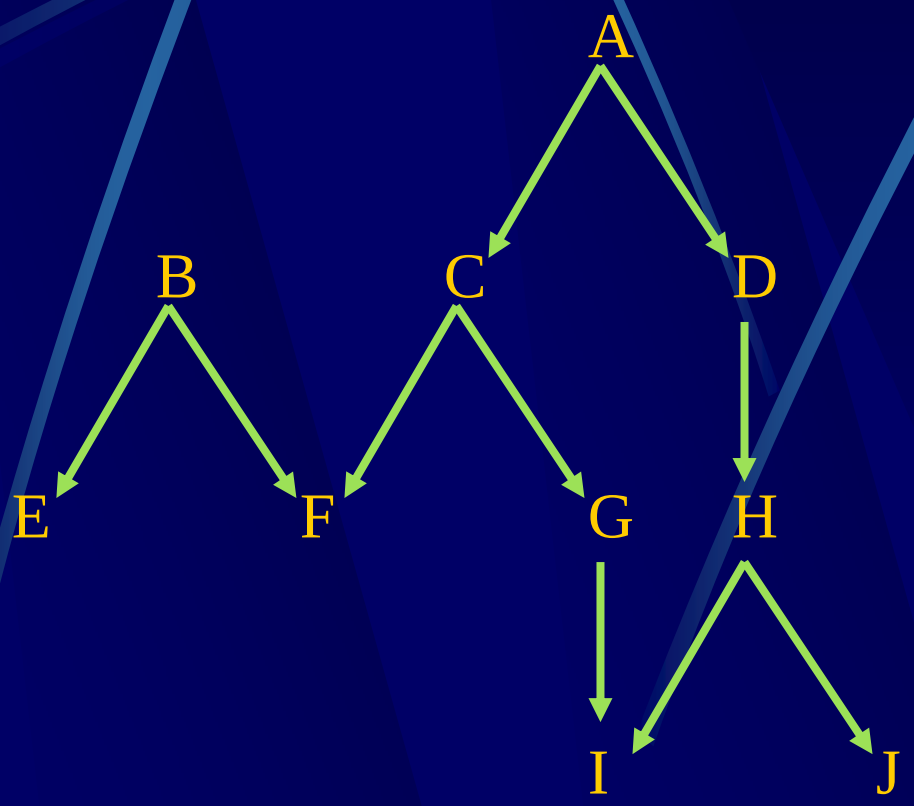
JPD

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

- Theorem 1 (Neapolitan): any JPD can be represented in BN form

A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- 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***



e.g., :

D ⊥ {B, C, E, F, G | A}

F ⊥ {A, D, E, F, G, H, I, J | B, C }

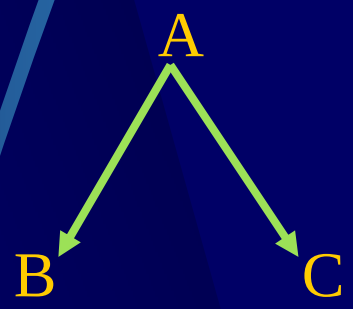
A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- Theorem 2 (Pearl): the Markov property enables us to decompose (factor) the joint probability distribution into a product of prior and conditional probability distributions

$$P(V_1, V_2, \ldots, V_n) = \prod_i p\,(V_i \mid \text{Parents}(V_i))$$

A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- Theorem 3 (Pearl): A DAG and set of conditional probabilities of each node given its parents defines a BN with a unique and valid jpd.



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

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
```

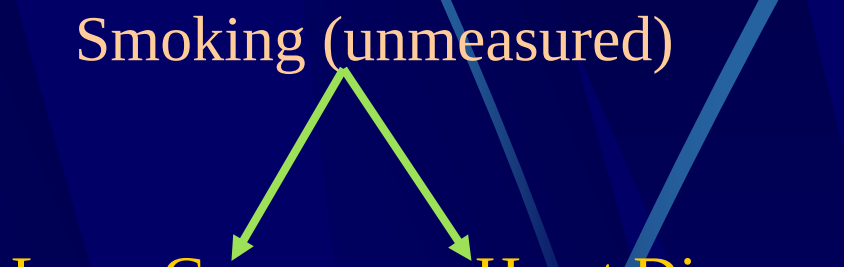
Becomes:

```
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!

A Framework for Feature Selection: Bayesian Network Properties & Algorithms

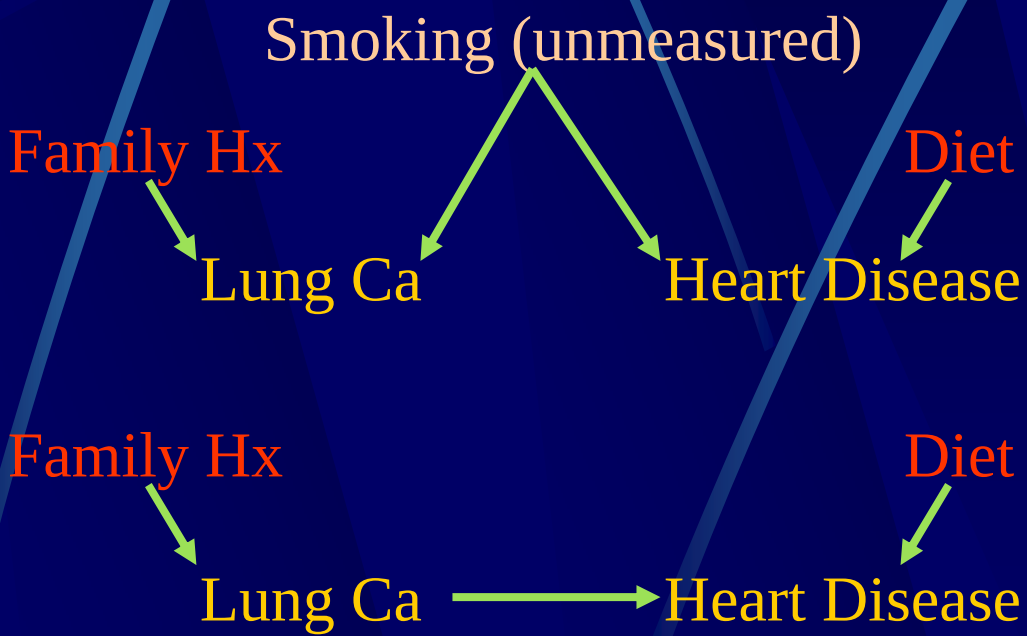
- BNs can help us learn causal relationships without doing experiments!



But Fisher says these two causal graphs are not distinguishable without doing an experiment (!?)

A Framework for Feature Selection: Bayesian Network Properties & Algorithms

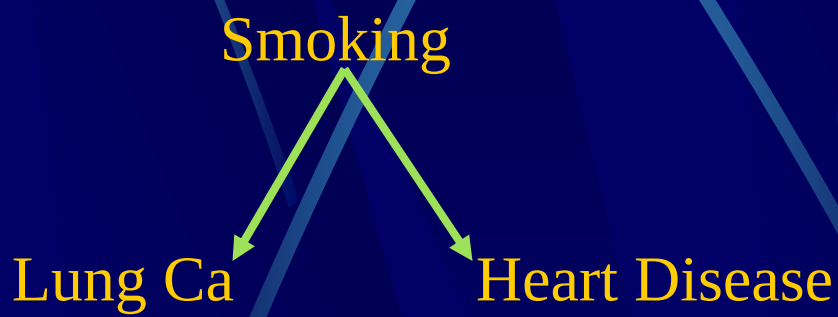
- BNs can help us learn causal relationships without doing experiments!



Fisher is right of course; however if we know a cause of each variable of interest then, in many cases, we can derive causal associations without an experiment

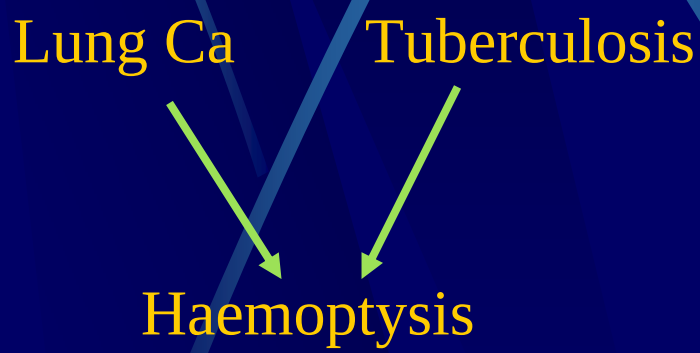
A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- The Markov property captures causality:
  - Confounders



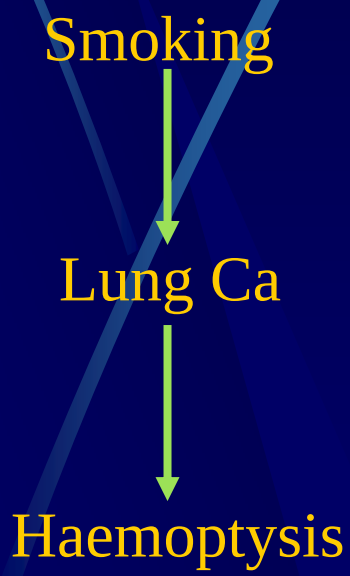
A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- The Markov property captures causality:
  - Modeling “explaining away”
  - Modeling/understanding selection bias



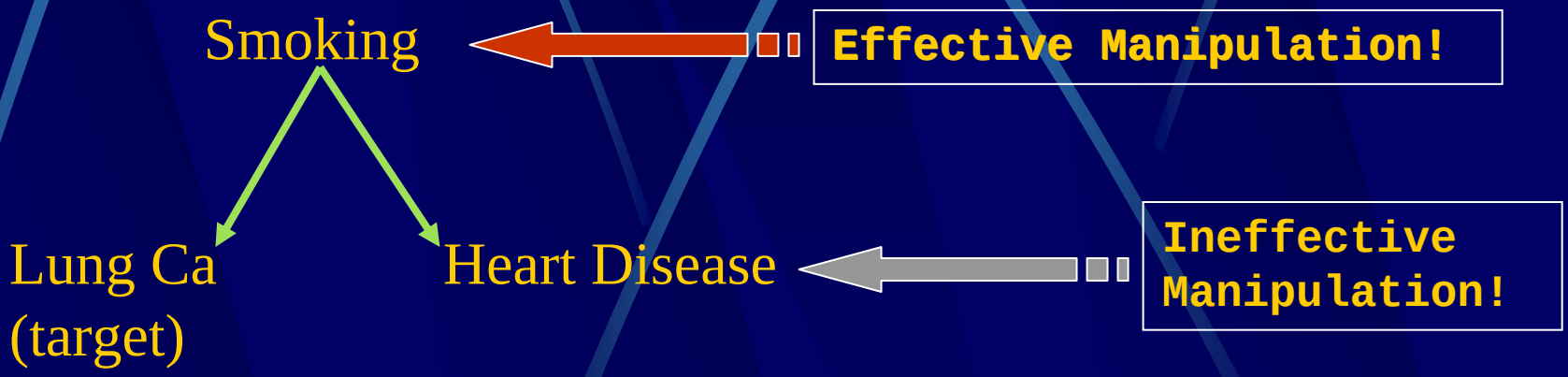
A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- The Markov property captures causality:
  - Modeling causal pathways



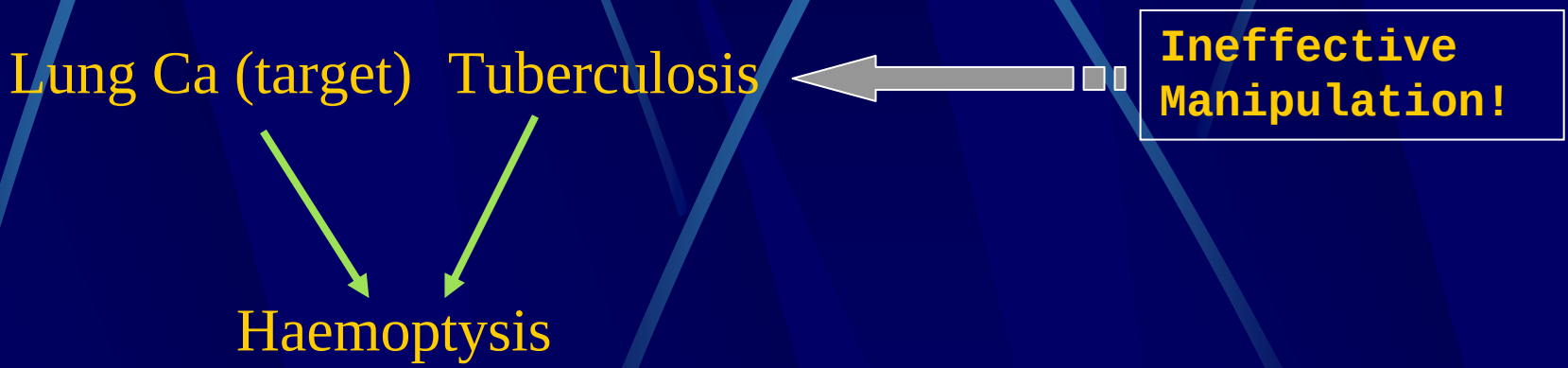
A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- The Markov property captures causality:
  - Manipulation in the presence of confounders



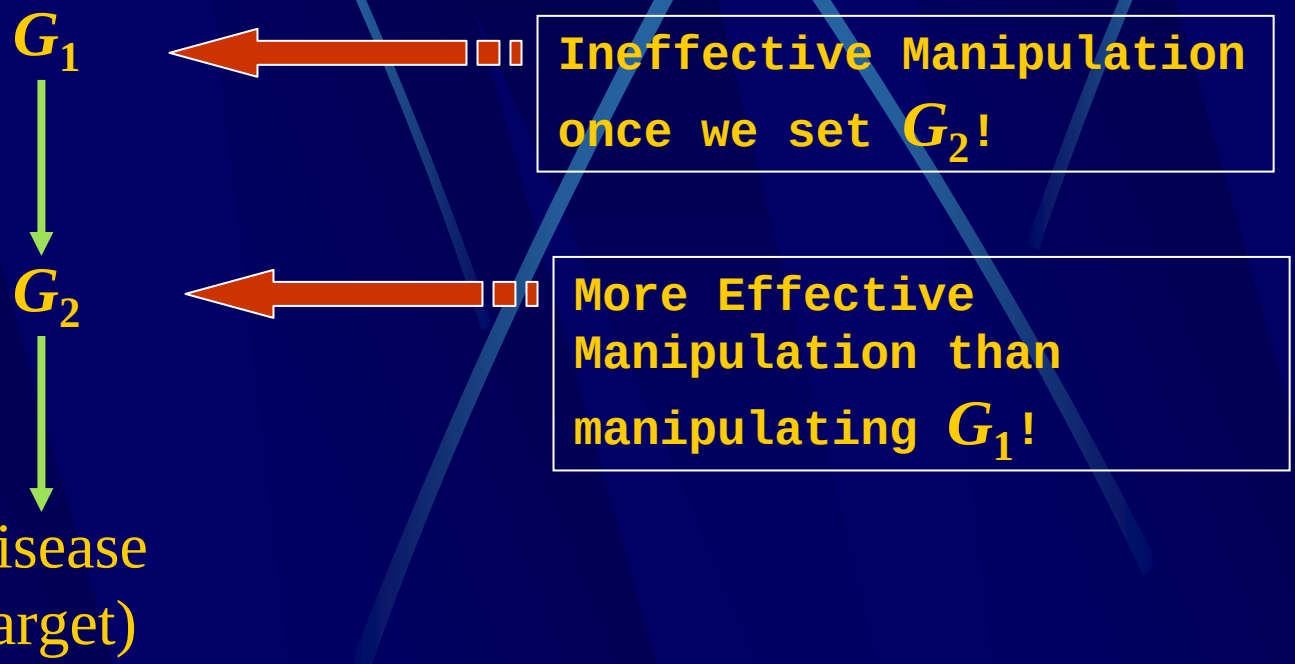
A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- The Markov property captures causality:
  - Manipulation in the presence of selection bias



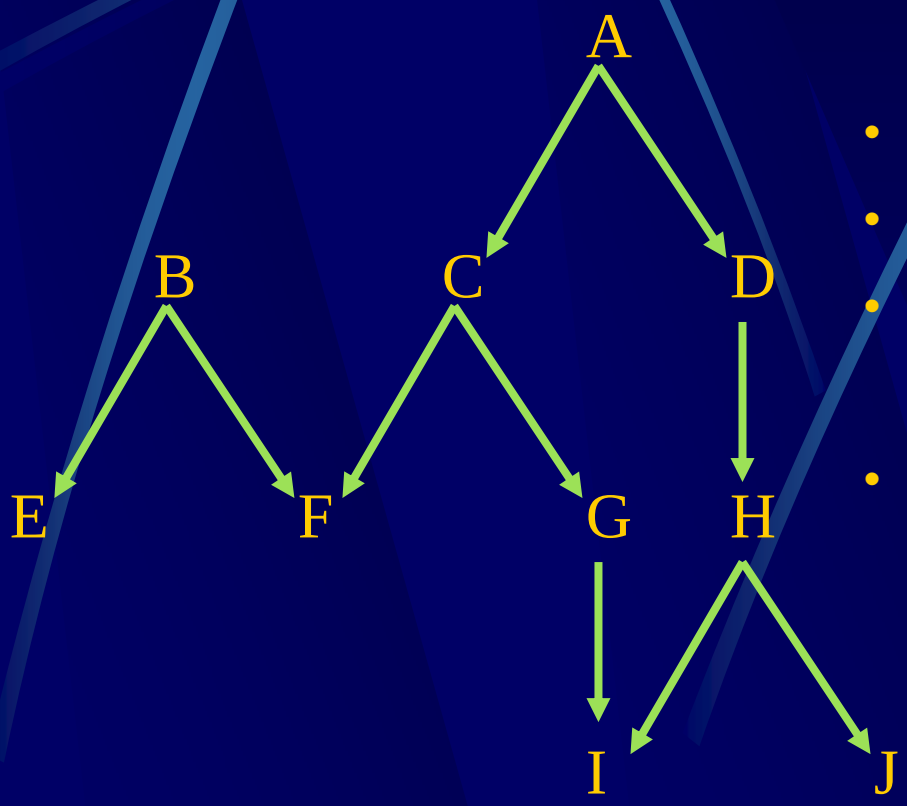
A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- The Markov property captures causality:
  - Identifying targets for manipulation in causal chains



A Framework for Feature Selection: Bayesian Network Properties & Algorithms

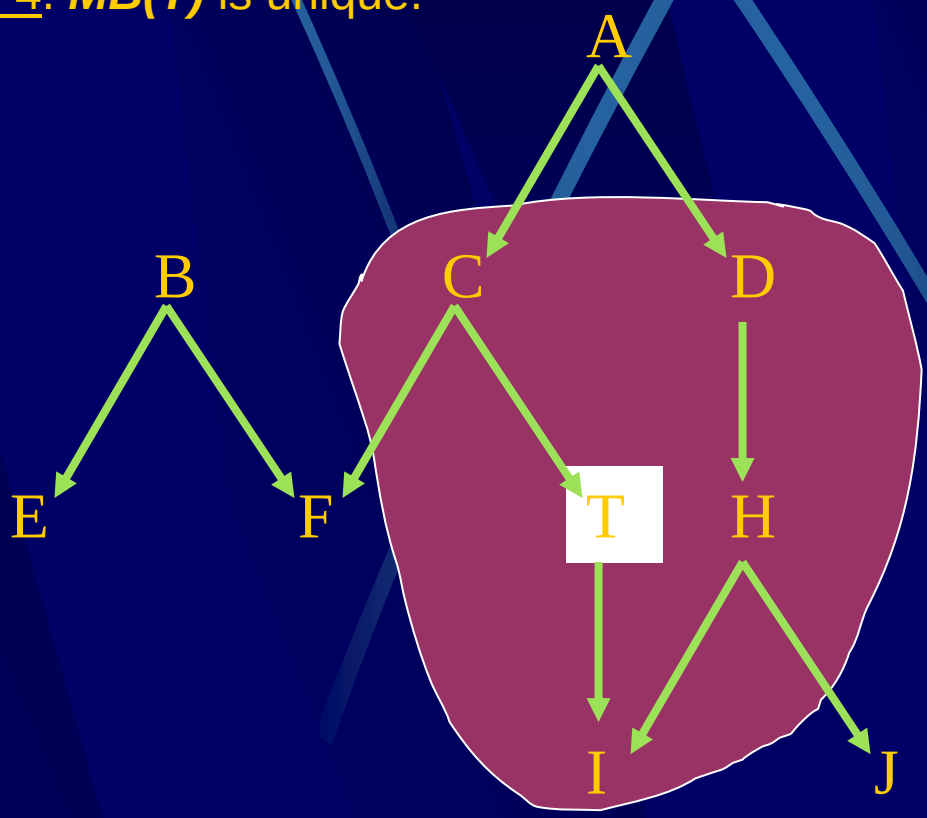
- Inference: Once we have a BN model of some domain we can ask questions:



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

A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- Markov Blanket of a feature *T*: The smallest feature subset conditioned on which all other features are independent of *T*.
- Theorem 4. ***MB(T)*** is unique.



A Framework for Feature Selection: Bayesian Network Properties & Algorithms

- Faithfulness. The graph *G* of some CPN *C* is faithful to a joint probability distribution *J* over feature set *V* if and only if every conditional dependence *P* entailed by *G* is also entailed by *J* and if every conditional dependence *P* entailed by *J* is also entailed by *G* .
- We say that a data-generating process *K* is faithfully represented by *C'*, if *K* in the sample limit produces data with joint probability distribution *D,* and *C'* is faithful to *D*.
- Causal Sufficiency. For every pair of measured variables in the training data, all their common parents are also measured.

A Framework for Feature Selection: Connecting Bayesian Network Properties With Feature Selection Problems

- Causal Discovery with BNs: Assume faithfulness of the data-generating process to the data; assume causal sufficiency; assume data is random sample from all instances produced by the process. Then an algorithm that generates the correct causal network given the data is:

PC Algorithm (Outline)

Phase I: find direct edges by using the criterion that A has a direct edge to B iff for all subsets of features there is no subset S, s.t. independent(A, B | S).

Phase II: orient edges in "collider" triplets (i.e., of the type: A->B<-C) using the criterion that if there are direct edges between A, B and between B,C, but not between A, C, and there is no subset containing B s.t. independent (A,C |B) then A->B<-B

Phase III: enter a constraint-propagation loop for orienting edges further by adding orientations until no further orientations can be produced using the two following criteria: (a) if A->B…->C and A-C then A->C also, and (b) if A->B-C then B->C

A Framework for Feature Selection: Connecting Bayesian Network Properties With Feature Selection Problems

- Proposition 1: In processes that can be faithfully represented by a CPN *C*, the Markov Blanket of a target feature *T*, MB(*T*), is the smallest set of optimal predictors of *T*.

- Proposition 2: For processes that can be faithfully represented by a CPN, and for classification algorithms that have universal approximator properties for the domain:
  - (a) all variables in *MB(T)* are strongly relevant to *T*,
  - (b) all variables that have paths to *MB(T)* but do not belong to *MB(T)* are weakly relevant to *T*, and
  - (c) all variables that have no paths to *MB(T)*, are strongly irrelevant to *T*.

A Framework for Feature Selection: Connecting Bayesian Network Properties With Feature Selection Problems

- Proposition 3: For processes that can be faithfully represented by a CPN, and for classification algorithms that have universal approximator properties for the domain:
  - the set *MB(T)* is the set of features that are strongly relevant to *T*,
  - the set of variables that have paths to *MB(T)* but do not belong to *MB(T)* is the set of features that are weakly relevant to *T*, and
  - the set of variables that have no paths to *MB(T)*, is the set of features that are strongly irrelevant to *T*.

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Backward Conditioning: An impractical (but still useful!) algorithm

BC algorithm Outline

Put all features in tentative MB

For each variable V in tentative MB
If V is independent of T given all features in tentative MB excluding V then
Remove V from tentative MB
EndFor

return Current-MB

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Backward Conditioning: properties

  - Assumptions:
    (a) The data-generating process is faithful to a CPN.
    (b) There exist reliable statistical tests of conditional independence for the data and features.
    (c) Causal sufficiency.
  - Correctness: The algorithm is sound and complete because if a feature belongs to *MB(T)*, then it cannot be removed because it will be dependent on *T* given any subset of the feature set. Thus Current_MB will always be a superset of *MB(T)*. Also if a feature is not a member of *MB(T)*, then conditioned on *MB(T)*, or any superset of *MB(T)* (Current_MB included) it will be independent of *T* and thus will be removed from Current_MB. Thus the algorithm returns *MB(T)*.

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Iterative Associative Markov Blanket : Heuristic improvement to reduce size of conditioning set.

IAMB algorithm outline

Phase I (forward)

Current-MB = empty
Enter a loop in which
Find the variable V that maximizes the | (association(V; T | Current-MB)) |
If (association(V; T | Current-MB)) is not zero (or higher than a threshold t1)
then add V to Current-MB
Until no feature is left with a non-zero conditional association

PhaseII (backward)

For each variable V in Current-MB
If association (V; T | Current-MB – {V, T}) = 0
Remove V from Current-MB
EndFor
Return Current-MB

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Iterative Associative Markov Blanket : properties

  - Assumptions:
    - (a) The data-generating process is faithful to a CPN.
    - (b) There exist reliable statistical tests of conditional independence for the given dataset and features.
    - (c) Causal sufficiency.
  - Correctness: In phase I the algorithm builds a conditioning set (tentative *MB(T)*) by selecting one-by-one the features that are most strongly associated with *T* given the current conditioning set. In phase II, a backward conditioning procedure similar to algorithm BC is employed. IAMB is sound because at the end of phase I the conditioning set (i.e., tentative *MB(T)*) is a superset of *MB(T)*. To see why this is the case consider that no member of *MB(T)* can fail to enter the conditioning set since its association with *T* will always be non-zero (otherwise it would be conditionally independent of *T* given some feature set and thus not a member of *MB(T)*).

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Bounded Iterative Associative Markov Blanket : Heuristic improvement to reduce size of conditioning set up to a point determined by examining the data and features.

Bounded IAMB (outline of modifications over IAMB):

Set a bound *b* for the conditioning set such that when the conditioning set grows to be size *b* during phase I the algorithm interrupts and starts backward conditioning on a smaller set than the full set of features.

If at least no feature gets removed the algorithm terminates.

If at least one feature gets removed, phase I is resumed and the same criteria apply.

Note: A reasonable *b* can be determined as the size that yields at least *n* samples per joint instantiation of the conditioning set (typical heuristic value: 5 cases). *b* can be constant or variable during the operation of the algorithm.

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Iterative Associative Markov Blanket : properties

  - Assumptions:
    - (a) The data-generating process is faithful to a CPN.
    - (b) There exist reliable statistical tests of conditional independence for the given dataset and features.
    - (c) Causal sufficiency.
  - Correctness:
    - (a) Bounded IAMB may not terminate if the size of the tentative MB becomes *b* and some features have not been examined.
    - (b) If the algorithm terminates it is sound.
    - (c) If *b* is fixed throughout the algorithm's operation no entry-removal loops may occur.
    - (d) If a variable *b is used* a finite number of entry-reentry loops may occur.

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Alternative way to cope with large size of conditioning set: Chunked Conditioning
- Similar conceptually to Bounded IAMB, however we use a full-network induction algorithm such as PC as conditioning test; the size of the maximum conditioning set should be larger than ***MB(T)*** and smaller than the smallest intractable set.

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

CC algorithm

%Phase I

Open = all features – T
Chunk = T
While Open<>{} and size(Chunk)<Ch
Selection = up to Ch-size(Selected) features from Open %choose according to univariate association or other heuristic, (or even randomly)
Open = Open - Selected
Chunk =Chunk U Selection

% constrained backward conditioning step
For each variable V in Chunk
If independent(V; T | all members of Chunk- {V,T}
Remove V from Chunk
EndIf
EndWhile

%Phase II

Open = all features – Chunk
Repeat loop as in phase I
Return Chunk-T

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Chunked Conditioning :

  - Assumptions:
    - (a) The data-generating process is faithful to a CPN.
    - (b) There exist reliable statistical tests of conditional independence for the given data and features that can operate on a set of variables up to size Ch.
    - (c) Causal sufficiency
  - Complexity: Given that there are *Ch* chunks, each one has size *V/Ch*. The complexity of running PC in each chunk is

$$\mathbf{O}\,(Ch^2 * (Ch - 1)^{k-1} / (k-1)!)$$

Thus the overall worst-case complexity is bounded by:

$$\mathbf{O}\,(V * Ch * (Ch - 1)^{k-1} / (k-1)!)$$

where *k* is the maximal degree (total number of causes and effects) of a feature in the domain. In comparison running PC on all features requires:

$$\mathbf{O}\,(V^2 * (V - 1)^{k-1} / (k-1)!)$$

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Chunked Conditioning : properties

  - Correctness: If the CPN induction algorithm asserts during phase I that a feature V is not in *MB(T)*, on the basis of the first Ch features, this means that either the feature is not a member of *MB(T)* or that it is a parent of a child of *T* and the child was not present in the first chunk. By induction, at the end of phase I the set Chunk will have all parents and children of *T* and some of the parents of its children. During phase II these parents of children cannot be missed since they will be in the Chunk.
  - Completeness: It is possible for Chunk to grow so large that the CPN induction algorithm "overflows" and CC stops without returning an output. Hence, when the algorithm terminates it is sound; It may not terminate however, depending on the size of ***MB(T)*** and the connectivity of the network.

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- 3-Phase IAMB : 3-Phase IAMB pursues a trade-off between larger number of conditional tests and smaller conditioning set.

3-Phase IAMB outline

% The algorithm first finds the parents and children of the target variable.

First step

1. **1. Cond-Set = empty**
2. **2. max-assoc = 1.0**
3. **3. while max-assoc not zero**
   a. **A. Find the variable V which maximizes assoc(V, T | Cond-Set), call this association max-assoc**
   b. **B. Cond-Set = Cond-Set union V**
   c. **C. Remove parents (ancestors) of children of Cond-Set from Cond-Set and mark them to not be re-entered later**
4. **4. End while**

% removing parents of children in Cond-Set

1. Look in all subsets of Cond-Set – A, where A any variable (so, for any variable and any subset of remaining variables).

2. If there is a subset S so that assoc(A, T | S) = 0, then A is not a parent or child of T

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

3-Phase IAMB (outline continued)

Second Step

After we find the parents and children of a variable T we run the algorithm recursively for the output nodes and find their parents and children. Then, we run PC on the union of all parents and children found to identify which nodes are indeed children, parents, or parents of children for T.

Third Step

Backward elimination as it is in IAMB

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- 3-Phase IAMB : properties
  - Assumptions:
    - (a) The data-generating process is faithful to a CPN.
    - (b) There exist reliable statistical tests of conditional independence for the given data and features that can operate on a set of variables up to size Ch.
    - (c) Causal sufficiency
  - Correctness/Completeness:
    - (a) 3-Phase IAMB is sound in the sample limit
    - (b) if we bound it, it exhibits same termination behavior as bounded IAMB
    - (c) If not bounded, it always terminates

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- 3-Phase IAMB : properties
  - Complexity: if *b* is the maximum size of the conditioning set, 3-Phase IAMB has complexity:

$$\mathbf{O}\,(b^3 * (b-1)^{k-1} / (k-1)!)$$

where *k* is the maximal degree (total number of causes and effects) of a feature in the domain.

Furthermore, 3-Phase IAMB is:

$$\mathbf{O}\,(b^{-1} * (V/b)^{k-1})$$

times faster than running PC on all features

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- Algorithms for local causal models & feature selection for manipulation: MMB1, MMB2.

MMB1 outline

Conduct a depth-first search in the set of all manipulatable features

Identify the next features to explore only if current parents are not manipulatable

To distinguish parents from children and parents of children run PC or similar

If PC cannot orient all edges, then keep expanding the network until necessary orientations are produced or until induction in the size of the partial network is intractable

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

MMB2 outline

Conduct a depth-first search in the set of all manipulatable features

Identify the next features to explore only if current parents are not manipulatable

To distinguish parents from children and parents of children run PC or similar

If PC cannot orient all edges, then consider all nodes with unoriented edges to *T* as parents of *T*

A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- MMB1, MMb2 : properties

  - Assumptions:
  - (a) The data-generating process is faithful to a CPN.
  - (b) There exist reliable statistical tests of conditional independence for the data and features.
  - (c) There exist reliable algorithms that can induce the CPN that generates the data given causal sufficiency.
  - (d) Causal sufficiency.

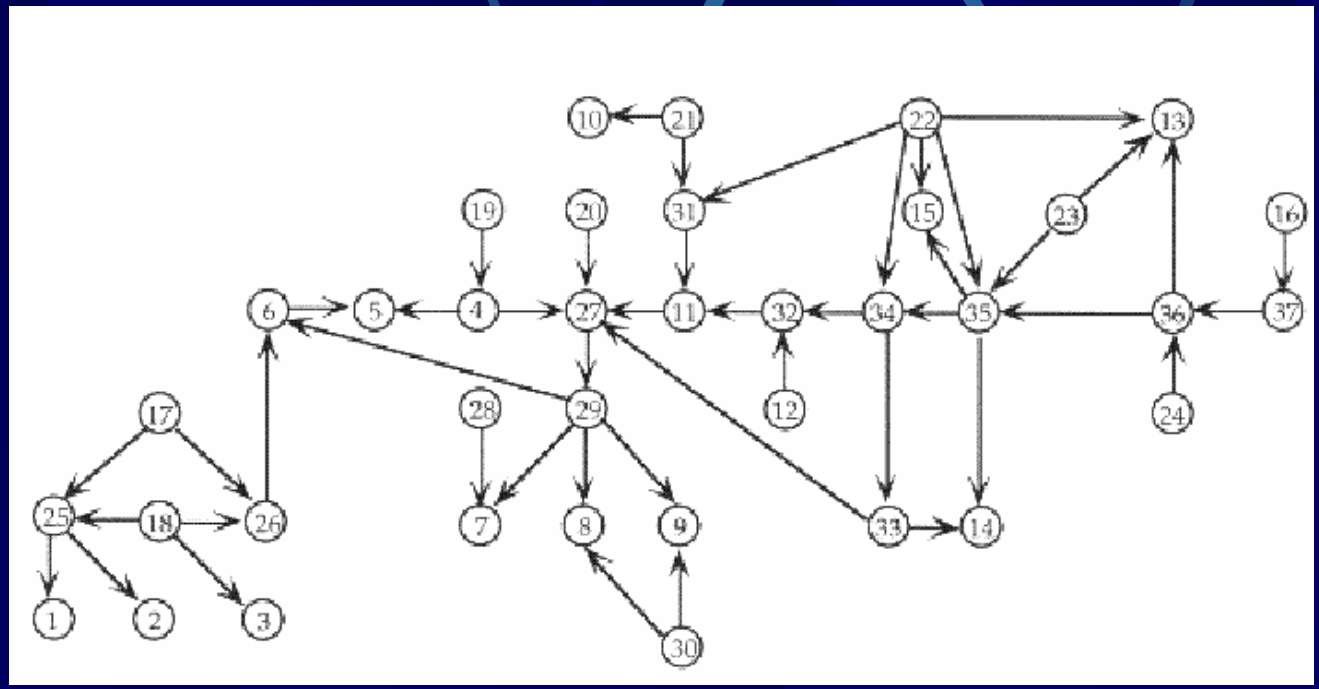
A Framework for Feature Selection: Use Theory To Develop & Characterize Algorithms

- MMB1, MMb2 : properties
  - Correctness/Completeness:
    - Trivially *MB(T)* may not be causally sufficient. However it can be shown that causal insufficiency when learning causal relationships within *MB(T)* does not entail false-positive or false-negative undirected edges. If *T* has at least two parents, or no parents and every child has parents other than *T,* then given the usual assumptions, all edge orientations derived by learning with the full set of features will be determined when learning only with *MB(T)*. However, when *T* has one or no parents, and every child has no parents other than *T,* then the edges between *T* and its parent and children may be unoriented.
    - MMB1: in the worst case the full set of features will be used for learning, hence the algorithm is sound *when* tractable.
    - MMB2: this strategy always results in a larger or equal set of manipulatable variables than the optimal one trading, optimality for tractability.

Initial Experiments

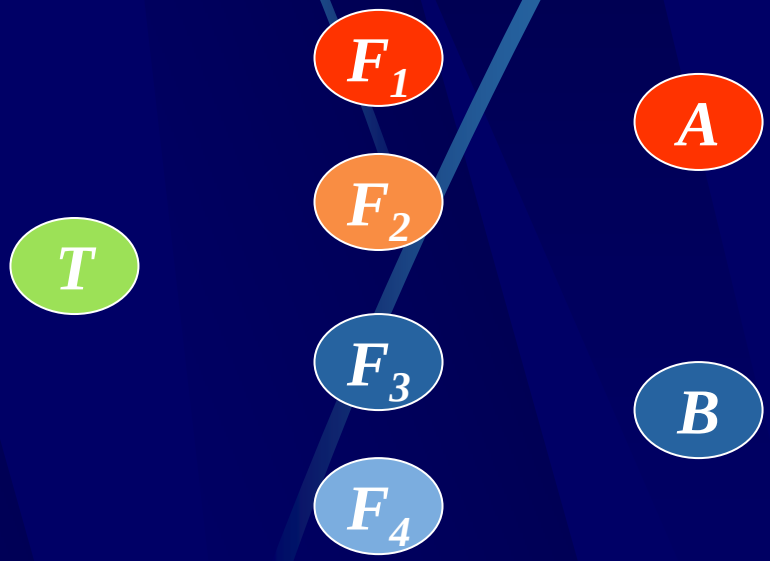
- Experiment 1: Simulated data from Alarm.

  The ALARM dataset model diagnosis in the medical domain of emergency care and it is often used as a benchmark for machine learning algorithms. In this experiment we run IAMB with training data of several different sample sizes ranging from 100 to 20,000 cases trying to recover the Markov Blankets of 10 randomly chosen features (among the 37 total features of the network).

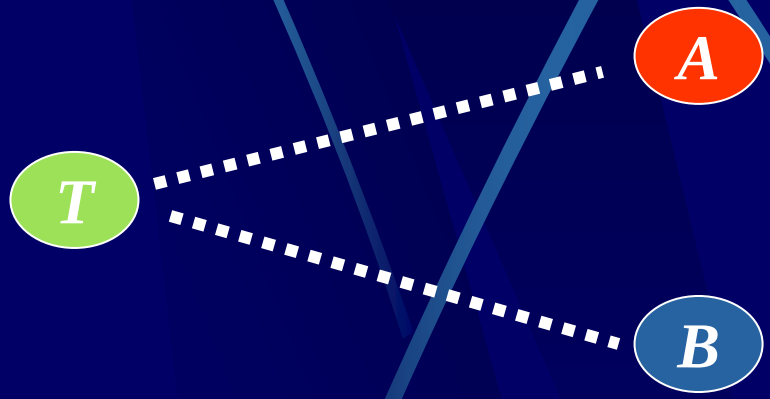


Initial Experiments

As a baseline comparison algorithm we use the Koller-Sahami (KS) feature selection algorithm that is designed to heuristically find the Markov Blanket of a target feature . Consider this example:

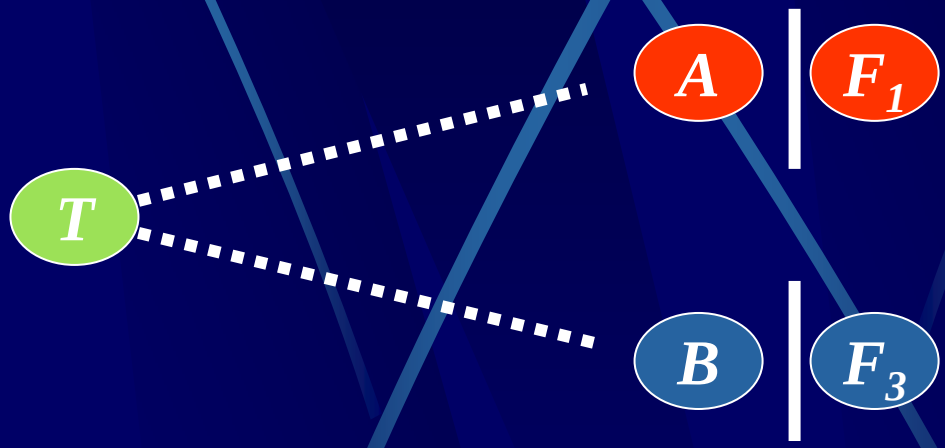


Initial Experiments



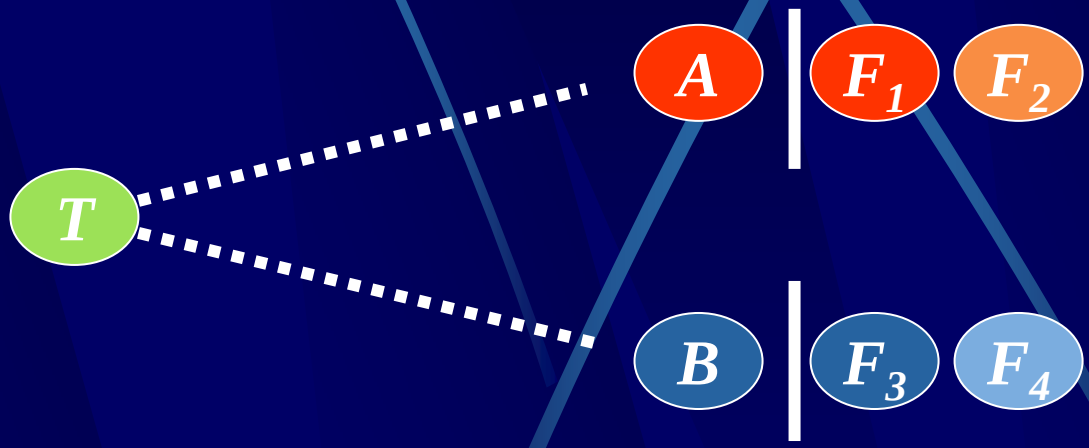
K=0 (univariate association)

Initial Experiments



K=1

Initial Experiments



K=2

Initial Experiments

Since KS outputs all features in order that corresponds to the algorithm's assessment of the likelihood that a feature belongs to *MB(T)*, while IAMB outputs a set number of features we use the following comparison method:

- for both algorithms we compute sensitivities and specificities and for the KS algorithm we also derive the ROC curve.
- we then examine whether the single point representing IAMB's output is to the outside (better), on (equal), or inside (worse) the KS ROC. We label these contingencies as "W", "D", "L" correspondingly.

Initial Experiments

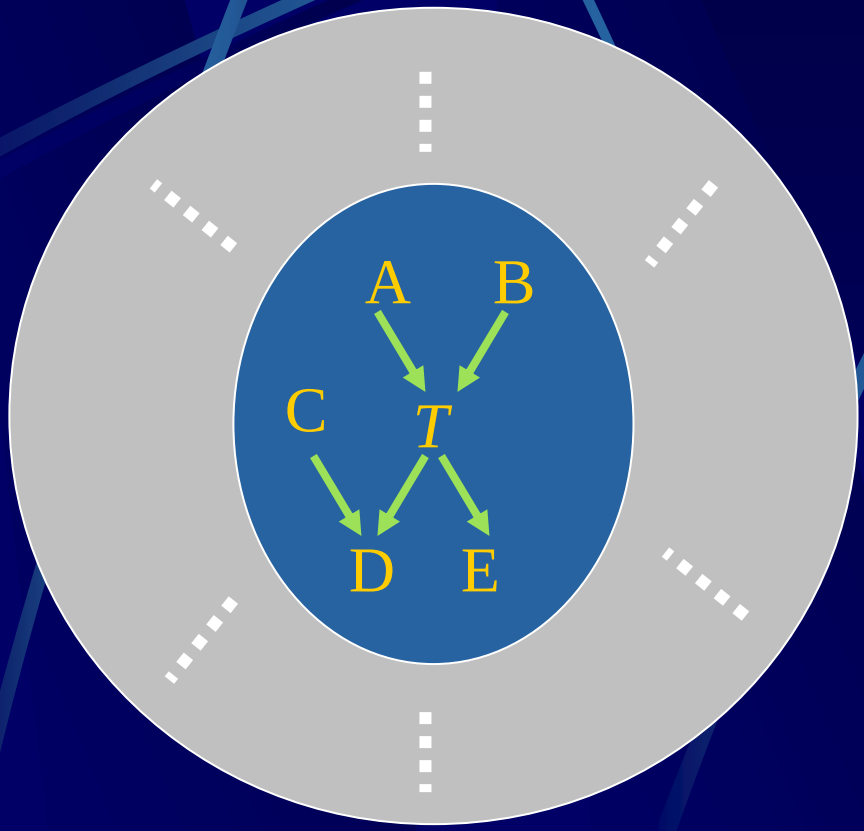
	Sample Size					*Total*
K	100	500	1000	10000	20000	
0	3/6/1	5/4/1	5/4/1	6/3/1	6/4/0	25/21/4
1	7/2/1	4/5/1	5/4/1	8/1/1	9/1/0	33/13/4
2	7/2/1	9/1/0	7/3/0	8/1/1	7/2/1	38/9/3
Total	17/10/3	18/10/2	17/11/2	22/5/3	22/7/1	**96/43/11**

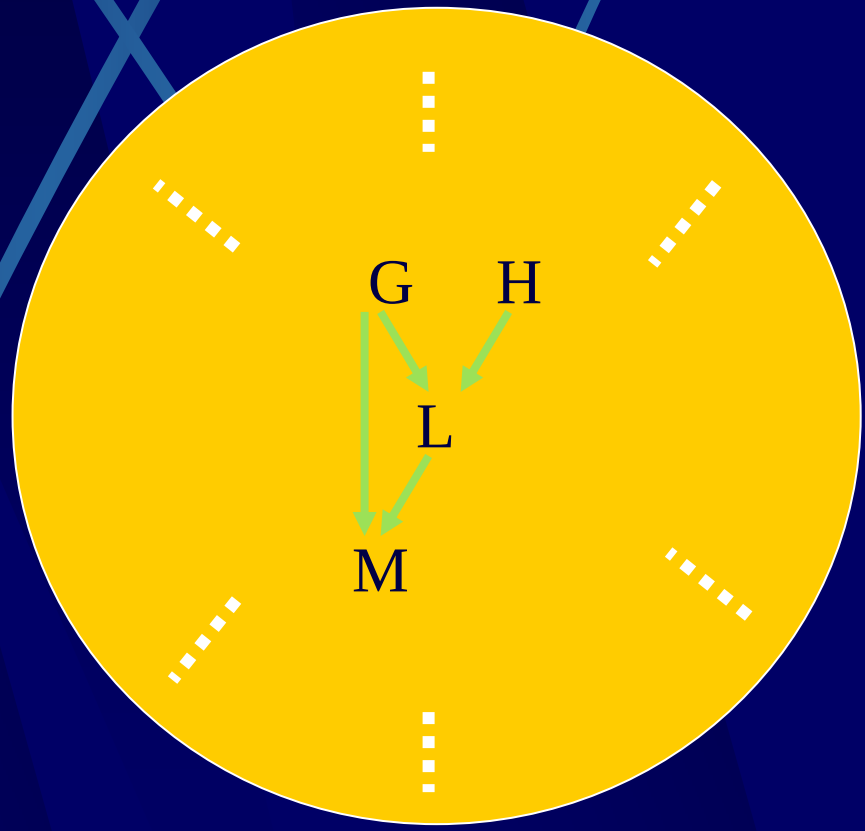
Figure 7: IAMB versus Koller-Sahami on Alarm

Initial Experiments

- Experiment 2: Simulated data from artificial CPNs with small Markov Blankets.
- We created a series of random CPNs subject to the following constraint: that they would have a small *MB(T)* with 6 members and a fixed structure (3 parents, 2 children, one parent of child).
- We started with a network with 25 features (*T*, *MB(T)*, 8 strongly irrelevant and 8 weakly irrelevant features). We parameterized the network randomly and used logic sampling to simulate cases from the network.
- We subsequently expanded the network to have 50, 100, 200, and 1000 features holding the *MB(T)* fixed and the proportion of strongly irrelevant to weakly irrelevant features constant (50%).

Initial Experiments





Initial Experiments

k=0	Size					
Features	*100*	*500*	*1000*	*10000*	*20000*	
25	L	W	D	W	W	
50	W	W	D	W	W	
100	D	W	D	W	W	
200	D	L	L	W	W	
1000	L	L	L	W	W	
						14/5/6

Initial Experiments

k=1	Size					
Features	*100*	*500*	*1000*	*10000*	*20000*	
25	L	W	W	W	W	
50	W	W	W	W	W	
100	W	W	W	W	W	
200	W	W	W	W	W	
						24/0/1

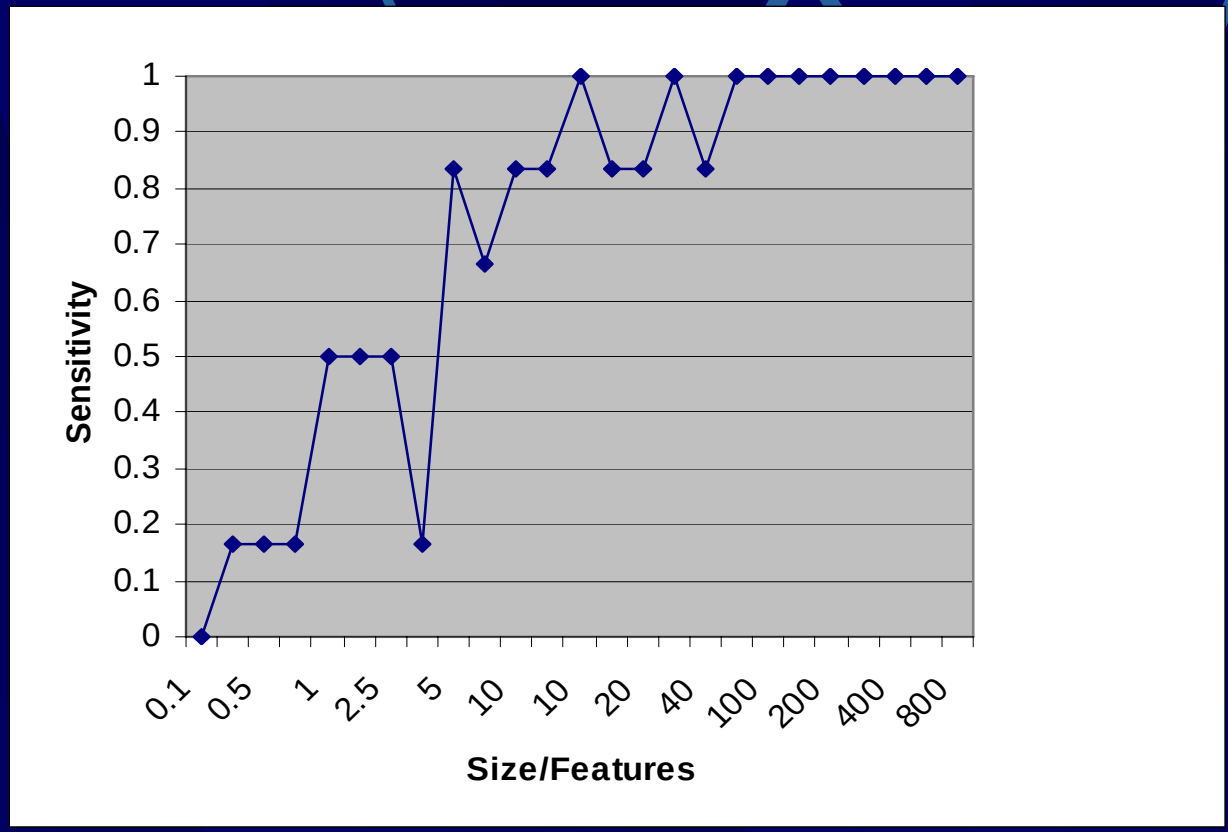
Initial Experiments

k=2	Size					
Features	*100*	*500*	*1000*	*10000*	*20000*	
25	W	W	W	W	W	
50	W	W	W	W	W	
100	W	W	W	W	W	
200	W	W	W	W	W	
						25/0/0

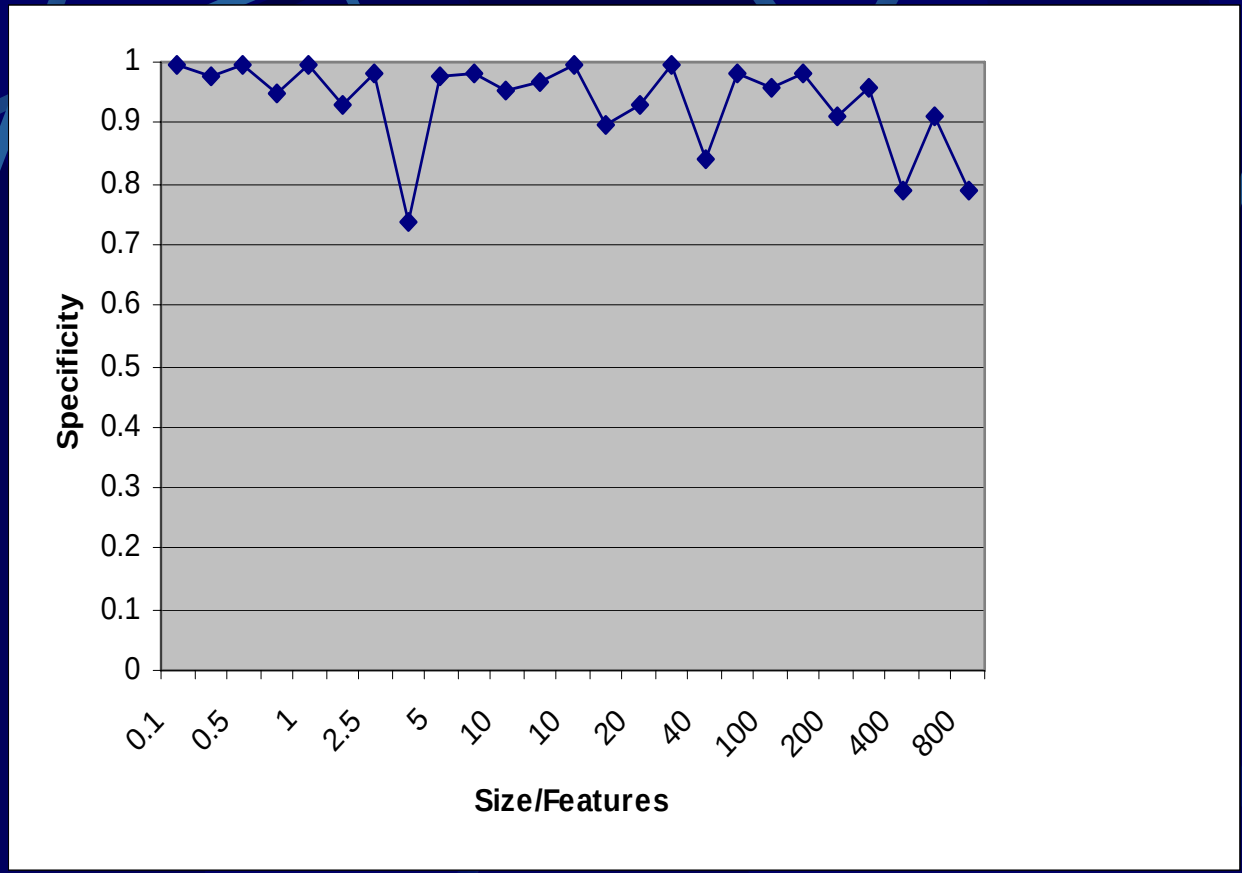
These results suggest that IAMB outperforms KS 56%, 96%, and 100% of the cases when *k* is 0, 1, and 2 respectively in this task. KS outperforms IAMB 24%, 4% and 0% when *k* is 0, 1, and 2 respectively.

Initial Experiments

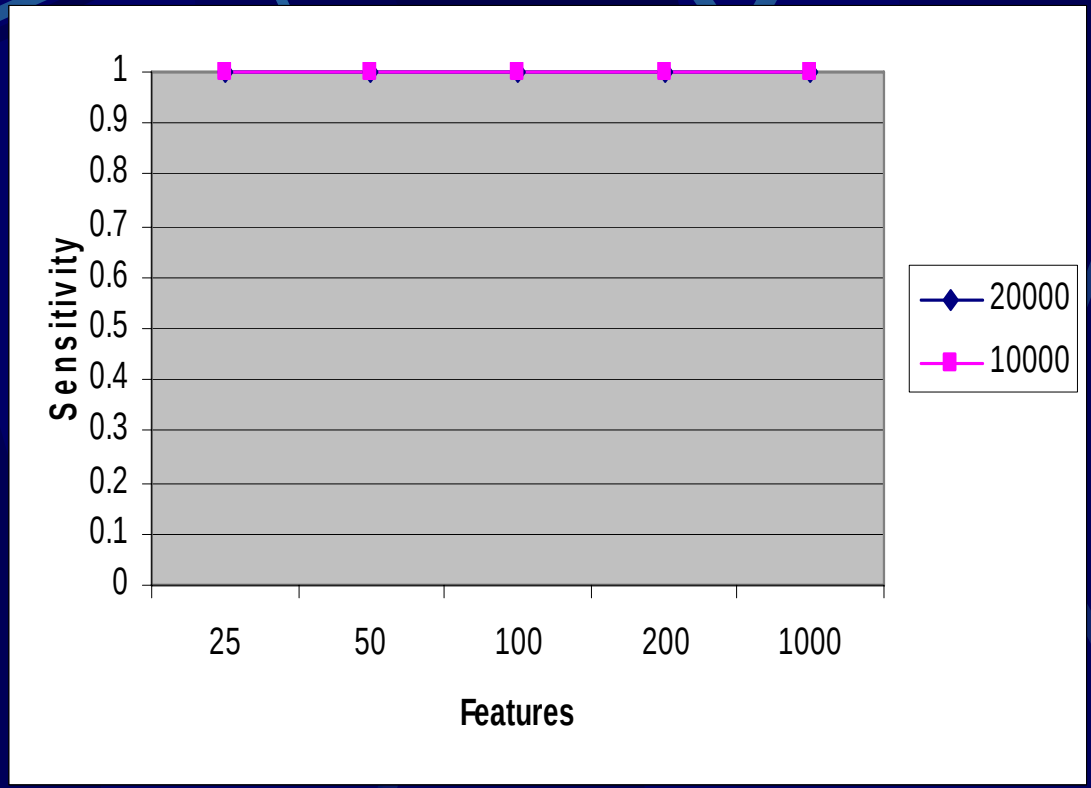
- We also examined how the sensitivity and specificity of IAMB varied as the number of irrelevant features increased.



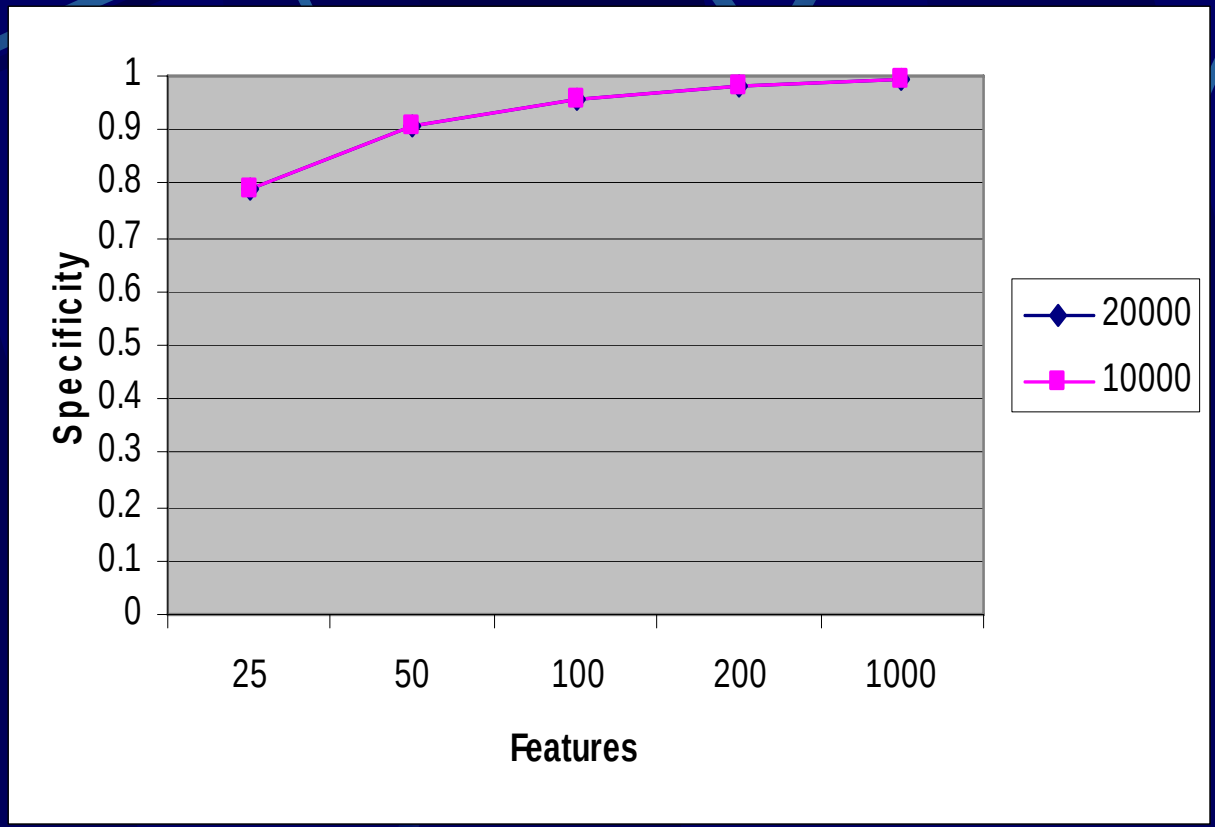
Initial Experiments



Initial Experiments

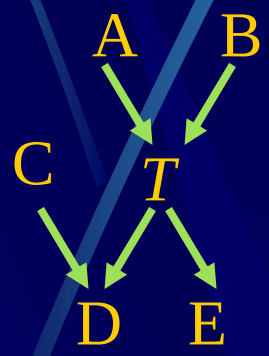


Initial Experiments



Initial Experiments

- Experiment 3: Local causal discovery using IAMB's output. Local structure was successfully recovered by running IAMB output through both PC and BN Power Constructor



Initial Experiments

- Experiment 4: Simulated data from artificial CPNs with larger Markov Blankets.
- We created a series of random CPNs subject to the following constraint: that they would have a *MB(T)* with 25 members and a fixed structure (10 parents, 5 children, 10 parents of children).
- We started with a network with 50 features (*T*, *MB(T),* 12 strongly irrelevant and 12 weakly irrelevant features). We parameterized the network randomly and used logic sampling to simulate cases from the network.
- We subsequently expanded the network to have 50, 100, 200, and 1000 features holding the *MB(T)* fixed and the proportion of strongly irrelevant to weakly irrelevant features constant (50%).

Initial Experiments

k=0	Size					
Features	100	500	1000	10000	20000	
51	W	W	L	L	W	
100	W	L	L	L	W	
200	L	L	L	L	W	
1000	D	L	L	L	NC	
						6 / 1 / 12
k=1	Size					
Features	100	500	1000	10000	20000	
51	W	W	W	D	W	
100	W	D	W	W	W	
200	L	D	D	W	W	
1000	D	D	D	W	NC	
						11/7/1
k=2	Size					
Features	100	500	1000	10000	20000	
51	W	W	W	L	W	
100	D	D	W	W	W	
200	D	W	W	W	W	
1000	D	W	W	NC	NC	
						13/4/1

- Although, IAMB is not equipped to handle the big conditioning set, it still outperformed the K-S algorithm 30 to 14 (with 12 draws)

Initial Experiments

Although we are doing better that the baseline algorithm, and specificity is very good, sensitivity can be improved. Can 3-Phase IAMB or CC deal with this size MB? The following results support this hypothesis:

- PC on 10,000 cases with 32 features (25 MB and 7 weakly relevant) retrieves 17/25 MB members
- BN Constructor on 20,000 cases and same feature set also retrieves 17/25 MB members
- It is certainly possible to increase the sensitivity of the algorithms by
- Setting higher p-value thresholds. We have not studied how this
- Will affect their specificity however.
- BN Constructor retains its performance when an additional 30 irrelevant features are included in the training data. This shows the feasibility of effective conditioning for this size of chunk in the CC algorithm.

Conclusions

- It is possible to effectively connect the tasks of selection of features for classification, manipulation , and causal discovery in a unified formal framework.
- Markov Blanket – based methods can solve the feature selection problem soundly when the Markov Blanket is small to the available data for a large class of classifier inducing algorithms, intended uses of selected features and data generating processes. These algorithms have the desired characteristics of filter approaches while at the same time do not sacrifice soundness.
- There is no ‘magical way” to solve this difficult class of problems when Markov Blankets are very big relative to the available data; sometimes we will need to trade off correctness for tractability, some other times completeness for tractability, or optimality for tractability.
- Out formal framework allows us to develop algorithms with well-specified properties and behaviors; different algorithms make different trade-offs helping us to find the best solution to a given application.

Future Research

- We are currently working to test our more advanced algorithms: CC, and 3-Phase IAMB and test them in large-scale simulated experiments, as well as with real data against wrapper and filter approaches.