Learning Integrated Structure from Distributed Databases with Overlapping Variables

David Danks

October 28, 2003

Technical Report No. CMU-PHIL-149

Philosophy

Methodology

Logic

Carnegie Mellon

Pittsburgh, Pennsylvania 15213

Learning Integrated Structure from Distributed Databases with Overlapping Variables

David Danks
Department of Philosophy, Carnegie Mellon University and
Institute for Human & Machine Cognition, University of West Florida
ddanks@cmu.edu

Abstract

The rapid growth of distributed databases, as well as growing concerns over privacy in those databases, has led to the existence of multiple datasets for overlapping sets of variables. Moreover, we often are interested in learning more than just the structure underlying each of the databases; we want to know the structure underlying the full set of variables. In this paper, we show that we can partially learn the structure underlying the full set of variables given only the learning output for each of the datasets. In fact, we can – under certain conditions – remove edges (from the final output) between variables that do not appear in the same databases. These results point towards parallelizable Bayes net learning algorithms for very large (in the number of variables) datasets, as well as possible psychological theories of human causal learning.

1 INTRODUCTION

There are a variety of practical machine learning problems in which we have multiple datasets over overlapping (in a sense to be made precise later) variables from a range of sources. For example, we might have a range of medical data (e.g., from hospitals, insurance claims, and doctors' offices), or remote sensing data (e.g., meteorological measurements). Moreover, these multiple datasets typically cannot be integrated into a single complete dataset, whether because of privacy concerns (in the former example) or asynchronous monitoring (in the latter example). Despite this restriction, we still want to recover as much information as possible about the structure (either causal or correlational) underlying the full set of variables.

Similar problems concerning structure learning from datasets with overlapping variables arise when we have very large (in the number of variables) datasets. In those cases, we might want to pursue a "divide-and-conquer" strategy for learning the full structure (though see Friedman, *et al.*, 1999, for a different strategy). Unfortunately, that strategy requires some way of integrating the learning results for each of the subsets.

This problem also arises in the context of human causal learning. People appear to have relatively large-scale causal knowledge, but clearly only obtain data on relatively few variables at a time. We can thus ask whether there is a normative theory for the integration of people's patchwork learning.

In this paper, we explore the learnability of underlying structure given multiple datasets with overlapping variables. These issues were previously addressed in Danks (2002), but that work assumed (i) only two overlapping datasets, and (ii) Causal Sufficiency: Every common cause C of two variables in a dataset D is also in D. In this paper, we remove both of those assumptions, thereby yielding algorithms that are potentially useful for realistic problems.

We will denote random variables by X, Y, Z, and assume that they are all discrete, though this assumption is not necessary; we only must be able to compute (conditional) independence for any pair of variables. We further assume that we have n distinct datasets (denoted by D_1, …, D_n), and that the variables in a particular dataset D are given by $v(D)$. Let $\mathbf{V}$ be the set of all variables that appear in at least one data set; that is, $\mathbf{V} = \bigcup_{i=1}^{n} v(D_i)$.

A natural framework for this domain is that of Bayesian networks (or Bayes nets). A Bayes net is composed of

two related elements: (i) a directed acyclic graph over (nodes corresponding to) the random variables; and (ii) a joint probability distribution over the random variables. These two elements are connected *via* two assumptions. The *Markov assumption* is that every variable is independent of its non-descendants conditional on its parents. The *Faithfulness assumption* (Stability in Pearl, 2000) is that the only independencies in the joint probability distribution are those implied by the Markov assumption. Due to space considerations, we do not provide a more detailed overview of Bayes nets here; many detailed introductions are available elsewhere (including Pearl, 1988, 2000; Spirtes, *et al.*, 1993/2001).

If the variables in the datasets are non-overlapping, then nothing can be learned beyond the learning results for each individual dataset. The interesting case arises when the variables in the datasets form a "connected" system, in the sense that we can move from one dataset to another using some "chain" of overlapping datasets. Formally expressed, we assume that:

$$\forall i, j \exists a_1, \ldots, a_m \left[\begin{array}{l} v(D_i) \cap v(D_{a_1}) \neq \varnothing \wedge \\ \forall k < m \left[v(D_{a_k}) \cap v(D_{a_{k+1}}) \neq \varnothing \right] \wedge \\ v(D_{a_m}) \cap v(D_j) \neq \varnothing \end{array} \right]$$

Note that this assumption does *not* imply that any two arbitrary datasets overlap; it is possible that each dataset overlaps with only two other datasets, regardless of n.

Given this way of structuring the problem, the central questions we address in this paper are:

1. Given some learning results for the multiple datasets, D_1, ..., D_n, what can be learned about the structure for **V**, the union of the variables in the datasets?
2. Given prior learning and data for all of the variables, how can we most efficiently learn the true underlying structure/equivalence class?

We might naturally expect that the answer to question 1 would just be "Nothing." That question asks what can be learned about the structure for **V**, even though we have no datapoints with values for every variable in **V**. Perhaps surprisingly, we will find (in section 3) that we can sometimes learn quite a bit about the structure underlying **V**, including removing edges between variables that never appear in the same dataset. Given a partial answer to question 1, there turns out to be a simple algorithm that is more efficient than ignoring the prior learning. Section 4 thus focuses on the question of finding the *most* efficient algorithm. We first (in section 2) explore different strategies for learning Bayes net structure from data in order to determine which will be most effective for answering these questions.

2 LEARNING STRATEGIES

There are essentially two different strategies for learning Bayes nets from data: Bayesian (score-based) and constraint-based approaches (though see Spirtes & Meek, 1995, and Dash & Druzdzel, 1999, for examples of hybrid approaches). In Bayesian learning, we attempt to find the network(s) that best fits the observed data. Typically, we do so by providing a scoring function for a network given the data (e.g., the Bayes Information Criterion – BIC), and we then search through the space of possible networks to try to find the highest-scoring network (e.g., Cooper & Herskovits, 1992; Heckerman, *et al.*, 1995; Heckerman, 1998). Finding the optimal graph is NP-hard (Chickering, 1996), and so a score-based search might (i) start with some initial "seed graph," (ii) consider all possible one-edge changes in the graph (adding, removing, or reversing an edge), (iii) select the highest-scoring graph from this set, and then (iv) iterate until the current graph remains the highest-scoring. Alternately, the search procedure might use a technique such as simulated annealing (with the same local changes as above). These heuristic searches are computationally feasible, but not asymptotically correct (though see Chickering, 2002, for an expanded score-based algorithm that is asymptotically correct).

The central questions of this paper can not, however, be answered using the output of a score-based learning algorithm. Score-based algorithms all rely on the fact that, given a particular set of parents, the score for a particular variable is uniquely determined. If we have no joint data for a particular variable and a potential parent, however, the score for that variable given a set of parents will not be fully determined. Therefore, there will be significant numbers of possible structures that cannot, even in theory, be evaluated. This argument is not based on the computational complexity of the learning problem, but rather on the fact that the data needed to evaluate a particular score are simply not available.

Perhaps these questions could be answered from within a Bayesian framework if we used "true" Bayesian learning, rather than score-based search. That is, suppose we have a probability distribution over all possible networks (and for each network, a distribution over the possible parameters). We then update the probability distributions given the data using (hopefully) straightforward applications of Bayes' rule. These updates are based on scores similar to those used in standard score-based searches. At the end, the output of our algorithm is a probability distribution over the possible graphs.

Unfortunately, this strategy also faces serious difficulties. The primary problem is that there does not appear to be any principled way to reallocate the probability distributions for all of the subsets into a single probability distribution for the possible graphs over all of the

variables. The most natural strategy would be, for all subset probability distributions, to redistribute the probability for each graph G in that distribution over the possible **V**-graphs whose marginal distributions are Markov and faithful to G. Unfortunately, directed acyclic graphs are not closed under marginalization: there are **V**-graphs whose marginals over D_i are not representable as a (Markov and faithful) DAG. For example, the marginal distribution of $\{W, X, Y, Z\}$ in the graph in Figure 1 is not representable by any Markov and faithful DAG. Hence, the "natural strategy" described above will lead to possible **V**-graphs being assigned zero probability.

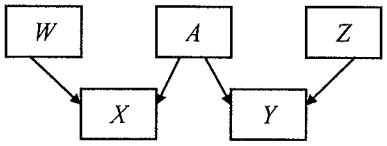

Figure 1: DAG not closed under marginalization

We could also try Bayesian search over a representation that is closed under marginalization, such as partial ancestral graphs (PAGs). PAGs are graphical structures that represent Markov equivalence classes over observed variables in the possible presence of latent variables and selection bias. Edges in a PAG convey ancestor relations. Specifically, the Y end of an $X - Y$ edge in a PAG can have one of three endpoints: (i) an arrowhead, indicating that X is not a descendant of Y; (ii) a straight edge, indicating that Y is an ancestor of X; or (iii) a circle, indicating that either of the above two cases is possible. Note that all permutations of endpoints on an edge are permissible, where a straight edge at both endpoints indicates there is no DAG represented by the PAG. So, for example, an X o→ Y edge means only that X cannot be a descendant of Y; in causal language, either X causes Y, there is an unobserved common cause of X and Y, or both.

The problem is, however, that we cannot score PAGs directly, and so we cannot update the probability distributions as required in Bayesian search. Instead, we can instantiate the PAG to a particular mixed ancestral graph (MAG), and since all MAGs for a particular PAG have the same BIC score, we can indirectly search over PAGs (see Spirtes, *et al.*, 1996, for an example of indirect PAG search). However, there is no known, computationally feasible algorithm for instantiating a PAG to a particular MAG, nor is there a computationally simple algorithm for determining which D_i-PAGs are marginally equivalent to a particular **V**-PAG.

Given the many difficulties facing the Bayesian/score-based approaches, we now consider whether a constraint-based learning algorithm might fare better. In the abstract, a constraint-based search procedure first calculates the set of independencies and associations in the data (using some statistical test), and then determines the equivalence class of graphs that could possibly have produced that data (i.e., the set of graphs whose Markov and faithful distributions all have the same independencies/ associations as the data). In practice, these algorithms do not compute all independencies/associations, but rather dynamically select which tests to perform. Exponentially many tests are required for the worst-case graphs.

The FCI algorithm of Spirtes, *et al.* (1993/2001) is fairly efficient, and asymptotically correct with only the Markov and Faithfulness assumptions. The output of the FCI algorithm can be interpreted either as a PAG that is not necessarily fully oriented, or else as a partially oriented inducing path graph (POIPG – described below). The FCI algorithm also outputs a function **SepSet**(X, Y) that returns the set(s) **S** such that $X \perp\!\!\!\perp Y \mid \mathbf{S}$, or no return value at all if X and Y are associated conditional on every subset of the variables.

POIPGs do not focus on ancestral relations, but rather on inducing paths: an undirected path P is an inducing path between X and Y relative to **O** iff (i) every member of **O** $\cap P \setminus \{X, Y\}$ is a collider, and (ii) every collider on P is an ancestor of either X or Y. Perhaps more intuitively, there is an inducing path between X and Y relative to **O** iff X and Y are d-connected given any subset of **O** $\setminus \{X, Y\}$. A POIPG has the same endpoints as a PAG, but they have a slightly different interpretation. For the Y end of an $X - Y$ edge, (i) an arrowhead indicates that there is some inducing path between X and Y into Y; (ii) a straight edge indicates that every inducing path between X and Y is out of Y; and (iii) a circle indicates that either of the above cases is possible.

Regardless of the interpretation on the FCI output, the graphical structure is essentially a representation of the independencies and associations in the data. Thus, there is no theoretical barrier to integrating the outputs from overlapping sets of variables. The task at hand is thus one of determining how much of the **V**-output structure can be learned from partial independence information.

The primary drawback to constraint-based approaches is that their output can be quite sensitive to mistaken calculations of independence or association. Moreover, these algorithms do not explicitly incorporate (possibly relevant) information about the power and significance level of the statistical tests used. However, this drawback is not a significant problem whenever we have large amounts of data, which will typically be the case when we have distributed datasets.

Constraint-based learning algorithms thus seem to be well-suited to the questions posed in Section 1. They readily handle unobserved common causes (a likely difficulty when learning from distributed datasets), and their outputs can, in theory, be integrated. We now show that this possible integration can practically be performed.

3 INTEGRATING NETWORKS

3.1 The ION Algorithm

Throughout the remainder of this paper, we will assume that we have used the FCI algorithm of Spirtes, *et al.* (1993, 2001) on each of the datasets, and we will adopt the POIPG interpretation of the output.[1] We will denote the POIPG output for dataset D_i as G_i. We further denote the maximally informative (i.e., maximally oriented) POIPG for all of **V** by π. Note that π may actually contain more orientation information than G_V, the output of the FCI algorithm given complete data over **V**. The initial questions can now be re-expressed as:

1. Given $G_1, \ldots, G_n$, what can be learned about π?
2. Given $G_1, \ldots, G_n$ and complete data over **V**, how can we efficiently learn π?

In this section, we focus on question 1, and consider question 2 in the next section. Before providing a general algorithm for obtaining (partial) information about π, we provide a series of results, which will be used to construct the algorithm.

The absence of an $X - Y$ edge in any particular POIPG indicates that there is some subset **S** of the variables in that POIPG such that $X \perp\!\!\!\perp Y \mid \mathbf{S}$. Since **V** is a superset of $v(D_i)$ for all i, any edge absent in one of $G_1, \ldots, G_n$ must also be absent in π. Note that this result resolves conflicts in which G_i has an $X - Y$ edge but G_j does not: we should remove the edge.

We can actually do more than just reconcile differences between the subsets; we can also remove edges *across* subsets. Before showing exactly how to remove those edges, we first provide a theorem (all proofs provided in the Appendix) about the continuity of definite ancestry between some POIPG and the POIPG for any superset of variables.

Theorem 1: Let F be a POIPG for variables **T**, G be the true, underlying (unknown) DAG, $\mathbf{V} \supseteq \mathbf{T}$, and π be the maximally oriented POIPG for the inducing path graph of G over **V**. If X is a definite ancestor of Y in F, then X is a definite ancestor of Y in π.

We can then define the *reachable ancestors* of a variable X in POIPG G relative to a blocking set **S**:

$$RA_G(X, \mathbf{S}) = \{Y: \exists \mathbf{Z} = \{Z_1, \ldots, Z_n\} \text{ (possibly empty)}$$
$$\text{s.t. } Y \rightarrow Z_1 \rightarrow \ldots \rightarrow Z_n \rightarrow X \wedge \forall i\, Z_i \notin \mathbf{S}\}$$

Note that the edges in the definition must be fully directed; a path is not suitable if it contains a $Z_i \circ\!\!\rightarrow Z_{i+1}$ edge. Given this definition, we can then show that:

Theorem 2: Assume Markov and faithful data for POIPG G. If $X \perp\!\!\!\perp Y \mid \mathbf{S}$, then X and $Z \in RA_G(Y, \mathbf{S})$ are independent given **S**.

The absence of an $X - Y$ edge in some G_i indicates there is some **S** such that $X \perp\!\!\!\perp Y \mid \mathbf{S}$. Therefore, X and $Z \in RA_G(Y, \mathbf{S})$ must be independent given **S**, and so cannot be adjacent in π, even though X and Z might never appear in the same dataset.

An example may help illustrate the use of this result. Suppose we have two overlapping variable sets: $\{W, X, Y\}$ and $\{X, A, B\}$. The learned graphs are: $W \rightarrow X \leftarrow Y$ and $X \rightarrow A \leftarrow B$, and we are trying to learn (as much as possible) about π.[2] The latter graph implies that X and B are unconditionally independent, and so B is independent (given the empty set) from all definite ancestors of X reachable relative to the empty set (i.e., all definite ancestors). Theorem 1 tells us that any definite ancestors of X in some G_i (W and Y in this particular example) are definite ancestors of X in π. Therefore, W and Y cannot be adjacent to B in π. Notice that we have removed the $W - B$ and $Y - B$ edges without seeing a single datapoint with values for both W and B, or Y and B.

If $\mathbf{S} = \varnothing$, then we can easily apply the above results, since Theorem 1 says that a definite ancestor in G_i is an ancestor in the inducing path graph for any superset of variables, and so we can conclude that any definite ancestor of X in G_i is reachable in π relative to the empty set. In fact, a wider range of sets are unproblematic. Define the potential ancestors of X in POIPG G as those Y such that (i) there is an undirected path between X and Y, and (ii) there are no arrowheads pointing towards Y on the path. Note that any definite descendant of X is necessarily not a potential ancestor of X. We can then prove the following theorem.

Theorem 3: If **S** contains no potential ancestors of X in G, then $RA_G(X, \mathbf{S}) = RA_G(X, \varnothing)$.

This theorem is not particularly helpful as it is currently written, since we do not yet have a good algorithm for determining the potential ancestors of X in G, the unknown true, underlying graph. We can get around this problem, though, using the following theorem.

[1] PAG versions of the theorems proven in this section are almost certainly provable. We do not, however, explore that possibility here.

[2] We should note that FCI learning on $\{W, X, Y\}$ in the absence of prior knowledge would return the POIPG: $W \circ\!\!\rightarrow X \leftarrow\!\!\circ Y$, and so W and Y would not be in $RA_G(X, \varnothing)$. This example requires further information (e.g., partial temporal ordering of the variables).

Theorem 4: Let **S** be some subset of the variables in G_i (not containing X), and let $X \in G_i$. For all $s \in \mathbf{S}$, if s is not a potential ancestor of X in G_i, then s is not a potential ancestor of X in G.

By chaining Theorem 3 and Theorem 4, we have the usable rule that: if no variable in **SepSet**(X, Y) is a potential ancestor of X in G_i, then any definite ancestor of X is a reachable ancestor of X. In addition to removing edges, we can also do some partial (tentative) orientation of the edges in π, using the following corollary of Theorem 1:

Corollary 1: Let F be a POIPG for variables **T**, G be the true, underlying (unknown) DAG, $\mathbf{V} \supseteq \mathbf{T}$, and π be the maximally oriented POIPG for the inducing path graph of G over **V**. If $X \rightarrow Y$ in F, and X and Y are adjacent in π, then $X - Y$ is oriented as $X \rightarrow Y$ in π.

This corollary results because other orientations would imply the existence of either an underlying cycle, or else an inducing path into X and into Y relative to **T**, contrary to the $X \rightarrow Y$ edge in F (full proof provided in the Appendix). Thus, if $X \rightarrow Y$ in some G_i, we can tentatively orient the $X - Y$ edge in π as $X \rightarrow Y$, recognizing that the edge might not exist at all.

We can also conditionally orient some edges. Define the "almost reachable" ancestors of X in G relative to **S** as those variables Y that would be reachable ancestors, except that the $Y - Z_1$ edge is $Y \circ\!\!\rightarrow Z_1$. If $X \perp\!\!\!\perp U \mid \mathbf{S}$ (the precondition of Theorem 2) and Y is an almost reachable ancestor of X, then although we cannot remove the edge between Y and U, we can determine that Y must be a collider on the $X - \ldots - Y - U$ path (if there is one), else X and U would be associated given **S**. Therefore, we can orient the tentative $Y - U$ edge into Y. We also know that the $Y \circ\!\!\rightarrow Z_1$ edge must be $Y \leftrightarrow Z_1$ if there is a $Y - U$ edge; unfortunately, we have no easy way to express this sort of conditional orientation, and so we exclude this step from the ION algorithm below.

Finally, consider the conditions under which we can assert that an $X - Y$ edge *must* be in π. Clearly, there must be some G_i such that $X - Y$ occurs in that POIPG, and no G_j without $X - Y$. More importantly, the presence of the edge in various of the subsets might be due to the exclusion of certain variables from those subsets. Hence, we can only assert the presence of an $X - Y$ edge when there is some D_i containing every variable in every *potential* trek between X and Y, as well as X and Y. This is the only way that we can determine whether X and Y are independent conditional on every potential trek.

We now introduce a new representation: a U-POIPG (*U*ncertainty-*POIPG*) has the same edgetypes as a regular POIPG, and some edges additionally have a subscript to indicate that the presence or absence of that edge is unknown: '$*\!\!-\!\!*_?$'. More precisely, let **D** be some set of datasets and **G** be the set of POIPGs that could have produced the datasets in **D**. We say that a U-POIPG U *represents* **D** iff (i) if every POIPG in **G** has some edge (absence), then U has a definite edge (absence); (ii) otherwise, U has an uncertain edge; and (iii) a definite edge in U is partially oriented iff every POIPG in **G** has the same partial orientation for that edge.

Given all of these results, we can provide the following *Integration of Overlapping Networks (ION)* algorithm for partially determining the structure of π given all G_i's. Recall that **SepSet**(X, Y) returns the set **S** (if any) such that $X \perp\!\!\!\perp Y \mid \mathbf{S}$.

ION Algorithm:

Input: A set of POIPGs, $\{G_1, \ldots, G_n\}$ learned from the datasets $\mathbf{D} = \{D_1, \ldots, D_n\}$, and the **SepSet** outputs.

Output: A U-POIPG G for the datasets **D**

1. Construct G, the complete graph over **V** using $\circ\!\!-\!\!\circ_?$ for each ordered pair of edges.
2. For each G_i, and for all pairs, X, Y of non-adjacent variables in G_i,
   a. remove the corresponding edge from G,
   b. if no variables in **SepSet**(X, Y) are potential ancestors of X (alternately, Y) in G_i, then remove all edges in G between Y (X) and any definite ancestors of X (Y). We may have to consider multiple G_k's to determine the full set of definite ancestors (e.g., if $Z \rightarrow X$ in G_j, and $W \rightarrow Z$ in G_k).
3. For all $X \rightarrow Y$ edges in some G_i, if the $X \circ\!\!-\!\!\circ_? Y$ edge still exists in G, orient it as $X \rightarrow_? Y$.
4. If $X *\!\!-\!\!*_? Y$ exists in G, let $\mathbf{T}(X, Y)$ be the set of possible-treks between X and Y, and let $var(\mathbf{T})$ be the set of variables that appear on some $T \in \mathbf{T}$. If there is a D_i such that $var(\mathrm{T}) \cup \{X, Y\} \subseteq D_i$, then replace $X *\!\!-\!\!*_? Y$ with $X *\!\!-\!\!* Y$.

We do not provide here a proof that G represents the datasets **D**, since it is an open question whether the ION algorithm is complete (either for edge removal or edge orientation). Since the edge removal step requires full orientations, the ION algorithm will likely perform best on datasets of moderate size; if the datasets are too small, we likely will not be able to orient the edges sufficiently.

Note that the input to the algorithm is simply a set of POIPGs (and the associated **SepSets**). Thus, we could (at least in theory) use a Bayesian/score-based algorithm to determine the POIPGs and separating sets, and then feed that into the ION algorithm. This strategy, however,

would not exploit the advantages of Bayesian/score-based learning, so we do not explore it further here.

3.2 Sample Run of the ION Algorithm

Suppose the true underlying structure is given in Figure 2, and the overlapping variable sets are: $D_1 = \{X, Y, Z, B\}$ and $D_2 = \{A, B, C\}$. The FCI POIPG output for the two datasets is given in Figure 3.

We can now run the ION algorithm using these POIPGs as input. First, we construct the complete graph over all of the variables. We then remove edges that are absent in the POIPGs (e.g., $X - B$, $Y - B$). Furthermore, for one of these edges ($B - C$), **SepSet** $= \emptyset$. Therefore, we can remove edges between C and all definite ancestors of B: namely, Z. Even though C and Z never occur in the same dataset, we can remove the edge between them. Note that we cannot remove edges between C and X, Y because the latter two variables are not definitely ancestors, but we can orient the tentative edge between C and X, Y into X, Y.

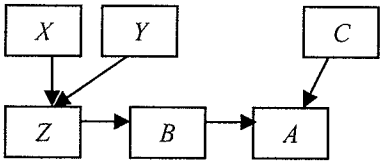

Figure 2: Example graph

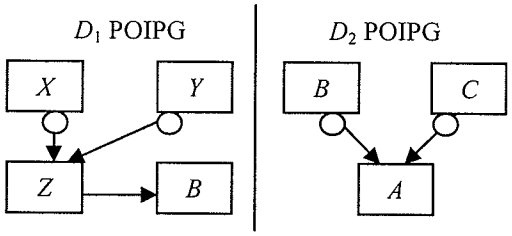

Figure 3: Output POIPGs

The intuitive justification for the cross-dataset edge removal and orientation can be seen quite easily in this example. If there were a $Z - C$ edge, then regardless of its orientation, there would be a trek connecting B and C, so they would be unconditionally associated. Similarly, if there were an $X - C$ edge that didn't form a collider with the $Z - X$ edge, then there would again be a trek connecting B and C (and similarly for Y). The D_2 data tells us that B and C are unconditionally independent, though, so we can remove/orient these tentative edges.

4 PARALLELIZABLE LEARNING

We can now consider question 2, in which we want to use the subset learning to learn π more efficiently when we are given complete data over all **V**. The above results point to the following simple strategy: (i) use the above algorithm to reduce the search space as much as possible; (ii) go through and check each $X \circ\!\!-\!\!\circ Y$ edge (with the usual steps of the FCI algorithm); then (iii) unorient all edges and reorient them (to ensure that all orientations are correct). Step (iii) could perhaps be optimized, but orientation is computationally a low-cost operation, and so it is unlikely that we would gain significantly from optimizing this step.

This straightforward algorithm provides one answer to the second question we originally asked, though we have not answered the question: Given prior learning on some subsets, how *much* faster is this algorithm than the FCI algorithm (without prior learning)? In the remainder of this section, we instead ask the reverse question:

3. Given complete data for **V** (of large cardinality), how should we divide the variables in **V** so that step (ii) of the ION algorithm maximally (or close to it) reduces uncertainty about π?

That is, given that we have some very large (in the number of variables) dataset, how should we divide the variables into overlapping subsets so that we minimize the amount of "clean-up" we have to do using the full data?

There are two different strategies we might pursue. One would be to use the fact that all edges removed during subset learning are also removed by the ION algorithm. Therefore, we might try to construct overlapping subsets of variables that are almost certainly *not* associated with each other, so that each subset graph will (hopefully) be quite sparse. For example, we might construct clusters of variables that are all (or almost all) unconditionally independent of each other, and then add variables to ensure that the subsets overlap. This strategy, while usable, does not seem to provide any significant advantages over simply learning on the whole dataset for **V**, since (by construction) these edges would have been removed early in the learning algorithm anyway.

A second strategy would be to focus on learning "families": a variable and its parents. Suppose we first learn the parents of X, and then we learn the parents of some (probably) child of X, call it Y. Then by an application of Theorem 2, no parents of X are adjacent to any parent of Y that is not also adjacent to X itself.[3] This result suggests trying to learn these local families, because they will (hopefully) often have the appropriate structure to enable us to remove edges without checking the data.

Of course, determining which variables to consider as parents is a highly non-trivial task. As a first try, we might consider some simple measure such as

[3] This claim is not quite right, since the FCI algorithm will only orient the edges in a family as $A \circ\!\!\rightarrow X \leftarrow\!\!\circ B$. We would actually need to include several grandparents of X in order to fully orient the *parent* $\circ\!\!\rightarrow X$ edges.

unconditional association or mutual information. Regardless of our choice, however, our learning algorithm is asymptotically correct (because we go back and check uncertain edges). Therefore, clustering the variables into suboptimal overlapping subsets will slow down the algorithm, but not lead to incorrect output.

By separating the variables into overlapping subsets and first learning on those subsets, we can easily parallelize this algorithm so that it can be run on multiple processors. We can thus potentially significantly reduce the effective run time of this algorithm on very large (in the number of variables) datasets.

This second strategy also points towards a normative theory of human causal learning, as there is growing evidence that people learn local causal families (e.g., Danks & McKenzie, submitted; Tenenbaum & Griffiths, 2000). The results here potentially provide a normative theory to explain how people should integrate this local causal knowledge into a large-scale causal structure. Whether they actually follow this theory is, of course, a separate empirical question.

This second strategy is related in end-goal to Friedman, *et al.*'s (1999) Sparse Candidate Algorithm. That algorithm iteratively cycles between step (i): selecting potential parents using a current "best guess network," and step (ii): finding the best-scoring Bayes net(s) with those potential parents. There are three major differences between the two approaches. First, the Sparse Candidate Algorithm is based on a score-based search, as opposed to the constraint-based search used here. Second, their algorithm does not necessarily provide useful information if the algorithm is stopped prior to finishing. If the above strategy is stopped midway, there is a clear interpretation to the output: every POIPG found to that point represents the correct Bayes net for the (marginal) joint distribution over the variables in that subset.

Third, and most importantly, the Sparse Candidate Algorithm (unlike this proposed algorithm) is not asymptotically correct for two different reasons. First, Friedman, *et al.* assume that every variable has at most k parents, and so their algorithm cannot learn the correct Bayes net for denser graphs. Also, their iterative procedure is not guaranteed to find all parents of a variable, since there is no constraint that ensures that every variable will (at some point) be considered as a potential parent of every other variable.

5 CONCLUSION

The ION algorithm presented in this paper provides an initial estimate of the amount of information regarding the global structure that can be extracted from local learning. Whether the ION algorithm is maximally informative or efficient remains an open problem. Nevertheless, it is an important first step, particularly since it demonstrates that the learnable global structure includes some information that, in a sense, goes beyond the information contained in any particular local learning.

The work in this paper points towards two interesting questions for future research. First, we have (by apparent necessity) focused on constraint-based algorithms. The most significant drawback of these algorithms, however, is their sensitivity to mistaken independence/association tests. With relatively small datasets, this sensitivity can pose a significant problem. There might thus be significant benefits on small datasets to exploring the strategy (mentioned in an earlier footnote) of instead using a Bayesian/score-based approach to determine the POIPGs (equivalence classes) provided as input for the ION algorithm. Unfortunately, that strategy faces the serious difficulty of scoring latent variable models.

Second, this work has all presupposed that there is some single underlying generative structure for the overlapping subsets. This assumption can obviously be false, particularly since the overlapping datasets will often have been gathered at different spatiotemporal locations (which explains the lack of a single integrated dataset). At the very least, we need a set of statistical tests to try to test this assumption. The most obvious test would simply be to compute the joint distributions for the variables in each overlap and check whether different subsets have different marginal joint distributions. It is, to our knowledge, an open question whether there are more sophisticated statistical tests.

References

Chickering, David M. 1996. "Learning Bayesian Networks is NP-complete." In *Proceedings of AI & Statistics V.*

Chickering, David M. 2002. "Optimal Structure Identification with Greedy Search." Microsoft Research Technical Report: MSR-TR-2002-10. Submitted to *Journal of Machine Learning Research.*

Cooper, G. F., and E. Herskovits. 1992. "A Bayesian Method for the Induction of Probabilistic Networks from Data." *Machine Learning*, 9: 309-347.

Danks, David. 2002. "Learning the Causal Structure of Overlapping Variable Sets." In S. Lange, K. Satoh, & C. H. Smith (eds.). *Discovery Science: Proceedings of the 5th International Conference.* Berlin: Springer-Verlag. pp. 178-191.

Danks, David, and Craig R. M. McKenzie. "Learning Complex Causal Structures." Submitted to *Cognitive Science.*

Dash, Denver, and Marek J. Druzdzel. 1999. "A Hybrid Algorithm for Constructing Causal Models." In K. B.

Laskey & H. Prade, eds. *Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference (UAI-1999)*. San Francisco: Morgan Kaufmann. pp. 142-149.

Friedman, Nir, Iftach Nachman, and Dana Peér. 1999. "Learning Bayesian Network Structure from Massive Datasets: The 'Sparse Candidate' Algorithm." In K. B. Laskey & H. Prade, eds. *Uncertainty in Artificial Intelligence: Proceedings of the 15th Conference (UAI-1999)*. San Francisco: Morgan Kaufmann. pp. 206-215.

Heckerman, David. 1998. "A Tutorial on Learning with Bayesian Networks." In M. I. Jordan, ed. *Learning in Graphical Models*. Boston: Kluwer. pp. 301-354.

Heckerman, David, Dan Geiger, and David Maxwell Chickering. 1995. "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data." *Machine Learning*, 20: 197-243.

Spirtes, Peter, and Christopher Meek. 1995. "Learning Bayesian Networks with Discrete Variables from Data." In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc. pp. 294-299.

Spirtes, Peter, Thomas Richardson, and Christopher Meek. 1996. "Heuristic Greedy Search Algorithms for Latent Variable Models." *Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics*, Fort Lauderdale, Fla. pp. 481-488.

Pearl, Judea. 1988. *Probabilistic Reasoning in Intelligent Systems*. San Francisco: Morgan Kaufmann Publishers, Inc.

Pearl, Judea. 2000. *Causality: Models, Reasoning, and Inference*. Cambridge: Cambridge University Press.

Spirtes, Peter, Clark Glymour, and Richard Scheines. 1993. *Causation, Prediction, and Search*. 2nd edition, 2001. Cambridge, Mass.: AAAI Press & The MIT Press.

Tenenbaum, Joshua B., and Griffiths, Thomas L. 2000. "Structure Learning in Human Causal Induction." In *Advances in Neural Information Processing 13*.

Appendix

Theorem 1: Let F be a POIPG for variables **T**, G be the true, underlying (unknown) DAG, $\mathbf{V} \supseteq \mathbf{T}$, and π be the maximally oriented POIPG for the inducing path graph of G over **V**. If X is a definite ancestor of Y in F, then X is a definite ancestor of Y in π.

Proof: We first prove the following lemma for a directed edge.

Lemma 1.1: If $U \rightarrow W$ in F, then there is a directed path from U to W in G^*.

Proof: $U \rightarrow W$ in F implies there is a set **P** of directed paths from U to W in G. We need to show that there is some directed path from U to W in G^*. For some $P \in \mathbf{P}$, let P^* be the ordered sequence of variables in $P \cap \mathbf{V}$. Choose the longest such P^*.

Consider any arbitrary pair A, B of neighboring variables in P^*, where A is closer to U on P. Since there is an inducing path between A and B relative to **V** that is out of A and into B (namely, the directed path in G), then by lemma 6.1.1 in Spirtes, *et al.* (1993), for any subset $\mathbf{S} \subseteq \mathbf{V}$, there is an undirected path in G^* that d-connects A and B given **S** that is out of A and into B. Let $\mathbf{S} = \varnothing$. The undirected path in G^* must contain no colliders, else it would not d-connect A and B given the empty set. The only undirected path out of A and into B with no colliders is a directed path from A to B in G^*.

Furthermore, by lemma 6.2.3 of Spirtes, *et al.* (1993), there must be an edge between A and B in G^*. Suppose the $A - B$ edge is not oriented in G^* as $A \rightarrow B$. Since there is a directed path from A to B in G^*, there must be some other sequence of variables $A \rightarrow C \rightarrow \ldots \rightarrow B$ that is a directed path in G^*. But this implies that there is some directed path from U to W in G that involves A, C, and B, *contra* the assumption that we had chosen the longest P^*.

Therefore, $A - B$ must be oriented as $A \rightarrow B$ in G^*. The concatenation of these $A \rightarrow B$ edges is a directed path out of U and into W, and so U is a definite ancestor of W in G^*. In fact, for *every* $P \in \mathbf{P}$ (the directed paths in G), the sequence of variables in P^* form a directed path from U to W in G^*. *End of proof of Lemma 1.1*

Since X is a definite ancestor of Y in F, there is a sequence of directed edges $X \rightarrow A_1 \rightarrow \ldots \rightarrow A_j \rightarrow Y$ in F. By the above result, there is a directed path in G^* for each of these directed edges, and the concatenation of these directed paths is a directed path from X to Y. ■

Theorem 2: Assume Markov and faithful data for POIPG G. If $X \perp\!\!\!\perp Y \mid \mathbf{S}$, then X and $Z \in RAG(Y, \mathbf{S})$ are independent given **S**.

Proof: Prove the contrapositive. Assume X and $Z \in RA_G(Y, \mathbf{S})$ are associated given **S**, and so d-connected given **S** in the underlying graph G. By definition of $RA_G(Y, \mathbf{S})$, there is a directed path from Z to Y in G^* with no nodes in **S**. If there is a directed path from A to B in some inducing path graph G^*, then there is a directed path from A to B in the underlying graph G. We can then append that directed path to the path (or paths) that d-connects X and Z in G. Since no nodes in **S** are on the directed path from Z to Y, X and Y must also be d-connected given **S**. Therefore, X and Y must be associated conditional on **S**. ■

Theorem 3: If **S** contains no potential ancestors of X in G, **then** $\mathrm{RAG}(X, S) = RA_G(X, \varnothing)$.

Proof: Consider $RA_G(X, \varnothing)$: the set of all definite ancestors of X. For any variable Y in this set, the variables on the directed path(s) from Y to X are also definite ancestors of X, and so are not members of **S** (since there is a suitable undirected path – the directed path itself). Therefore, every $Y \in RA_G(X, \varnothing)$ is also a member of $RA_G(X, \mathbf{S})$. The opposite direction, $RA_G(X, \mathbf{S}) \subseteq RA_G(X, \varnothing)$, follows immediately from the definition of RA_G. ■

Theorem 4: Let **S** be some subset of the variables in G_i (not containing X), and let $X \in G_i$. For all $s \in \mathbf{S}$, if s is not

a potential ancestor of X in G_i, then s is not a potential ancestor of X in G.

Proof: We prove the contrapositive: if s is a potential ancestor of X in G, then s is a potential ancestor of X in G_i. Suppose that s is a potential ancestor of X in G. Then there must be some undirected path $P = X - Z_1 - \ldots - s$ such that there is no Z_i *→ Z_{i+1} edge in the path. Note that this property implies that there are no colliders on P. Let P_i be the ordered sequence of variables on P that also appear in G_i. (Note that P_i has at least two elements, since both X and s are in G_i.) Consider any two sequential variables A, B on P_i. Since there are no colliders on P, and since no intervening variables on P appear in G_i, A and B must be adjacent in G_i (since they are associated regardless of G_i-conditioning set). Without loss of generality, let B be the variable closer to s on P_i. The sub-path of P between A and B is an inducing path over G_i, but it cannot be into B (else P must contain Q *→ B for some Q, contradicting the directionality property of P). If the $A - B$ edge in G_i is oriented as A *→ B, then every inducing path (over G_i) between A and B must be into B. Since there is at least one inducing path (over G_i) between A and B that is not into B (namely, the sub-path of P), the $A - B$ edge in G_i cannot be oriented as A *→ B. Therefore, P_i is a path between X and s such that no edges on the path have arrowheads on the s-end of the edge. Hence, s is a potential ancestor of X in G_i. ■

Proof of Corollary 1: Since X and Y are adjacent in G^* and all $H^* \in$ **Equiv**(G^*) have the same adjacencies, we need only show that $X \leftarrow Y$ and $X \leftrightarrow Y$ do not appear in any $H^* \in$ **Equiv**(G^*).

Suppose $X \leftarrow Y$ appears in some H^*, and so there must be a directed path from Y to X in the underlying graph. Since $X \rightarrow Y$ in F, there is also a directed path from X to Y in the underlying graph, and so we have a cycle.

Suppose instead that $X \leftrightarrow Y$ appears in some H^*. This edge implies that there is some inducing path over **V** between X and Y that is into X and into Y. An inducing path over **V** is an inducing path over $\mathbf{T} \subseteq \mathbf{V}$. Therefore, there must be an inducing path over **V** between X and Y that is into X and into Y. The $X \rightarrow Y$ edge in F, though, means that every inducing path over **T** is out of X and into Y. Therefore, $X \leftrightarrow Y$ cannot appear in any H^*. ■