
Integrating locally learned causal structures with overlapping variables

Anonymous Author(s)

Affiliation

Address

email

Abstract

In many domains, data are distributed among datasets that share only some variables; other recorded variables may occur in only one dataset. There are several asymptotically correct, informative algorithms that search for causal information given a *single* dataset, even with missing values and hidden variables. There are, however, no such reliable procedures for distributed data with overlapping variables, and only a single heuristic procedure (Structural EM). This paper describes an asymptotically correct procedure, ION, that provides all the information about structure obtainable from the marginal independence relations. Using simulated and real data, the accuracy of ION is compared with that of Structural EM, and with inference on complete, unified data.

1 Introduction

In many domains, researchers are interested in predicting the effects of variable interventions, such as policy changes. Such predictions require knowledge of causal relations, rather than simply a compact representation of joint probabilities, such as in classification. There is a long history of research on algorithms for learning causal structure from data, possibly with missing values and hidden variables [1][2][3], but these algorithms all assume that every variable of interest is jointly measured in a *single* dataset. In practice, such datasets are not always readily available and often cannot be constructed due to privacy, ethical or financial concerns. For instance, models of the United States and United Kingdom economies share some but not all variables, due to different financial recording conventions; fMRI studies with similar stimuli may record different variables, since fMRI images vary according to magnet strength, data reduction procedures, etc.; and U.S. states report some of the same educational testing variables, but also report state-specific variables. The standard causal structure learning algorithms cannot be directly applied to cases such as these, since there are variables, which may be causally relevant, that are never jointly measured in any dataset.

More formally, suppose one has n datasets D_1, \dots, D_n , each measuring a corresponding variable set V_1, \dots, V_n , where all variables are either continuous or discrete. We assume that the variable sets are connected: for any pair V_i, V_j , there exists a (possibly empty) sequence of variable sets such that any two consecutive sets have non-empty intersection. (Connectedness is strictly weaker than the assumption that every pair of variable sets has non-empty intersection.) The central question is: what (if anything) can be inferred about the causal structure for $\mathbf{V} = V_1 \cup \dots \cup V_n$ given only D_1, \dots, D_n ? We assume that there are no cross-dataset IDs, and so no complete dataset can be constructed, even in principle. If there are cross-dataset IDs, then the central problem is computationally efficient transfer of information (as in [4][5]), not in-principle learnability.

To our knowledge, there are no useful algorithms that explicitly address this problem in its most general form (though see [6][7]), since essentially all structure learning algorithms assume a single input dataset. The standard response is to convert this into a “single dataset, missing values” prob-

lem: concatenate the multiple datasets into a single, large dataset, where the variables unmeasured in each dataset have missing values for datapoints from that dataset. Statistical matching [8] and multiple imputation [9] procedures estimate the missing values by assuming an underlying model (or small class of models), estimating model parameters using available data, and then using the model to interpolate the missing values. While the assumption of some underlying model is unproblematic when one is doing classification, it implies that these procedures only correctly learn the *causal* structure when the assumed underlying model happens to be correct. They are thus unreliable methods for causal inference.

In contrast, Structural EM is a reliable heuristic procedure for learning structure from datasets with missing data [10]. In Structural EM, one assumes some initial model M_0 , and then iterates until convergence between (a) using the current model M_i to estimate the missing values for a new dataset D_{i+1} (the E step); and (b) using D_{i+1} to learn a new best-fitting model M_{i+1} (the M step). The present problem is a non-standard application for Structural EM, as Structural EM assumes that missing data are missing at random (or that indicator variables can be used to make them so), but the pattern of missing values in this concatenated dataset is highly structured.

We present a novel algorithm—the *Integration of Overlapping Networks (ION) algorithm*—that uses independence constraints to learn a causal structure (or more properly, a set of possible causal structures) over a set of variables given multiple datasets, each of which measures only a proper subset of the variables. The ION algorithm is provably sound and complete, and its accuracy both is superior to the Structural EM algorithm, and compares favorably to baseline causal structure learning algorithms that have access to a complete dataset, on both synthetic and real-world data.

2 Integration of Overlapping Networks (ION) Algorithm

2.1 Formal preliminaries

We represent a causal structure for \mathbf{V} using a *directed acyclic graph* (DAG): each variable in \mathbf{V} corresponds to a node in the DAG, and $X \rightarrow Y$ denotes that X is a direct cause of Y relative to \mathbf{V} . An *undirected path* between X and Y is any sequence of nodes X, Z_1, \dots, Z_n, Y such that there is an edge between consecutive nodes in the sequence; that is, consecutive nodes are *adjacent*. A *directed path from X to Y* is an undirected path in which all edges are directed away from X (i.e., $X \rightarrow \dots \rightarrow Y$). X is an *ancestor* of Y if there is a directed path from X to Y . A DAG is acyclic because there cannot be a directed path from a node to itself. A triple of nodes $\langle X, C, Y \rangle$ is a *collider* if X and C , and Y and C , are adjacent, and there are arrowheads at C along both edges (e.g., $X \rightarrow C \leftarrow Y$). A collider is *unshielded* if there is no edge between X and Y . A *trek* between X and Y is an undirected path with no colliders.

There is a *d-connecting path between X and Y given set \mathbf{Z}* if there is an undirected path π between X and Y such that (i) no non-collider on π is in \mathbf{Z} ; and (ii) every collider on π is either in \mathbf{Z} , or an ancestor of a node in \mathbf{Z} . X and Y are *d-connected given \mathbf{Z}* if there is at least one such path, and are otherwise *d-separated*. A Bayesian network is a DAG G with a joint probability distribution P over the observed variables. The widely-assumed Markov and Faithfulness (Stability) conditions for the $\langle G, P \rangle$ pair imply d-separation relations in G are identical with conditional independence relations in P [11]. That is, G is a compact graphical representation of all conditional independencies in P .

A *partial ancestral graph* (PAG) represents a set of DAGs that share causal relationships over the observed variables. All DAGs represented by a PAG have the same d-separation/d-connection relations (over the observed variables). PAGs efficiently encode the causal information learnable from conditional independence relations. Nodes in a PAG correspond to observed variables, and the graph may contain four types of edges: \rightarrow , $\circ \rightarrow$, $\circ - \circ$ and \leftrightarrow . A \circ at a node indicates that an edge may be oriented as either \rightarrow or $-$ at that node. Fully directed edges indicate direct causation, though relative to the observed variables (e.g., $X \rightarrow Y$ in the PAG means that every DAG it represents has X as a causal ancestor of Y , though possibly via different, unobserved intermediate nodes); bidirected edges denote a latent common cause of the two nodes.

In a PAG, we mark a triple $\langle X, C, Y \rangle$ as a *definite non-collider* to indicate that \circ 's at C may only be oriented as \rightarrow or $-$ if such orientations do not result in making C a collider. We also say that X and Y are *possibly d-connected given \mathbf{Z}* if there is a path π between X and Y such that we

can (consistently) orient the \circ 's on π as \blacktriangleright or $-$ to produce an (actually) d-connecting path, and similarly for *possibly d-separated*. The same path π can be both possibly d-separating and possibly d-connecting (e.g., $X \circ - \circ Z \circ - \circ Y$, if Z is not a definite non-collider).

The ION algorithm uses conditional independence information to determine the set of PAGs that could possibly have produced the “marginal” datasets D_1, \dots, D_n over connected variable sets V_1, \dots, V_n . These conditional independence constraints are provided to ION as a set of PAGs, where the PAG for each dataset can be learned by any standard causal structure search algorithm for complete data, such as FCI [1] or GES [3].¹ Expert domain knowledge can also be encoded in input PAGs, if available.

We now restate our original problem: Given an input set G_1, \dots, G_n of PAGs learned for D_1, \dots, D_n , what is the set **Output** of PAGs over $\mathbf{V} = V_1 \cup \dots \cup V_n$, such that the marginalization of each PAG $O \in \mathbf{Output}$, with respect to V_i , is G_i ?

2.2 Formal statement of the ION algorithm

1. **Initialize:** Set $\mathbf{CG}_1 = \{A_1\}$, where A_1 is a fully connected (with $\circ - \circ$ edges) graph over \mathbf{V} .
2. **Transfer information:** For each X, Y that are not adjacent in some G_i , remove the $X - Y$ edge. For each remaining edge in A_1 , if that edge appears in some G_i , then transfer its orientation(s) to A_1 (actual arrowheads/tails dominate definite non-colliders) and propagate that orientation throughout A_1 (e.g., if X is an ancestor of Y , Y is an ancestor of Z , and X and Z are adjacent, orient $X \circ - \circ Z$ as $X \circ \blacktriangleright Z$). If there are orientation conflicts between graphs (e.g., $X \blacktriangleright Y$ in G_i , and $X \blacktriangleleft Y$ in G_j), then ignore both orientations.²
3. **Block possibly d-connecting paths:** For $l = 2, \dots, |\mathbf{V}| - 1$, set $\mathbf{CG}_l = \emptyset$ and:
 - (a) For each $A_i \in \mathbf{CG}_{l-1}$, let $\mathbf{PDC}_i(\{X, Y\}, \mathbf{S}_j)$ be the length- l possibly d-connecting paths (in A_i) between non-adjacent $\{X, Y\}$ given \mathbf{S}_j , where \mathbf{S}_j d-separates $\{X, Y\}$ in some G_k . If all $\mathbf{PDC}_i = \emptyset$, then add A_i to \mathbf{CG}_l and go to A_{i+1} ; otherwise, continue.
 - (b) Construct **PossChanges**, the set of all sets of minimal combinations of graphical changes³ such that every path in every $\mathbf{PDC}_i(\{X, Y\}, \mathbf{S}_j)$ is no longer possibly d-connecting between $\{X, Y\}$ given \mathbf{S}_j .
 - (c) For each $\mathbf{PC}_k \in \mathbf{PossChanges}$, create the PAG $A_i(k)$ by applying the graphical changes in \mathbf{PC}_k to A_i . Propagate the orientation information in $A_i(k)$. If there exists $\{X, Y\}$ that are d-separated given \mathbf{Z} in $A_i(k)$, but are d-connected given \mathbf{Z} in some G_i , then reject $A_i(k)$; otherwise, add it to \mathbf{CG}_l .
4. **Generate full output set:** For each $A_i \in \mathbf{CG}_{|\mathbf{V}|-1}$, let \mathbf{PR} be the set of graphs with every combination of edge removal and retention in A_i (i.e., if A_i has k edges, then \mathbf{PR} has 2^k graphs). For each $A_i(j) \in \mathbf{PR}$ and each pair $\{X, Y\}$ of variables d-connected given \emptyset in some G_i , if $A_i(j)$ has no possible treks between $\{X, Y\}$, then reject it. If $A_i(j)$ has exactly one possible trek between $\{X, Y\}$, then orient every unoriented triple on the possible trek as a definite non-collider. For each acceptable $A_i(j)$, add to **Output** every PAG constructable by orienting all unshielded possible colliders as either colliders or marked definite noncolliders without blocking every trek between any $\{X, Y\}$ pair d-connected given \emptyset in some G_i .
5. Return **Output**.

2.3 Soundness and completeness of the algorithm⁴

The ION algorithm is sound if and only if nothing in the output contradicts known (from the datasets) conditional independence information. More formally, ION is sound iff no structure $O_k \in \mathbf{Output}$ entails a d-separation or d-connection that contradicts a d-separation or d-connection entailed by

¹The conditional Independence relations represented by the structures returned by these algorithms define *Markov equivalence classes*. Current research is focused on generalizing ION for scenarios when the input is not a Markov equivalence class.

²We are currently investigating more sophisticated ways of resolving conflicts.

³Edge removal, orienting \circ 's to create a collider, or marking a triple of nodes as a definite noncollider.

⁴We provide only proof sketches due to space constraints. Full proofs appear in the supplementary material.

some G_i . Soundness of ION thus follows directly from the following two theorems (and the fact that d-separation and d-connection are mutually exclusive, exhaustive relations).

Theorem 2.1 (d-separation preservation). If X and Y are d-separated given \mathbf{Z} in some G_i , then X and Y are d-separated given \mathbf{Z} in all $O_k \in \mathbf{Output}$.

Proof Sketch. Step 3 provably ensures that every d-separation in some G_i is established in the graphs in some \mathbf{CG}_j . It thus suffices to show that those d-separations never convert back to d-connections. The graph change operations in Steps 3 and 4 are edge removal, collider creation and definite non-collider creation. Each provably never changes a d-separation into a d-connection. \square

Theorem 2.2 (d-connection preservation). If X and Y are d-connected given \mathbf{Z} in some G_i , then X and Y are d-connected given \mathbf{Z} in all $O_k \in \mathbf{Output}$.

Proof Sketch. If X and Y are d-connected given \mathbf{Z} in some G_i , but are d-separated in $A_k \in \mathbf{CG}_j$, then A_k is rejected at Step 3c. Step 4 ensures that every graph in **Output** has at least one trek between all X and Y d-connected given \emptyset in some G_i , and this constraint provably ensures that no other d-connection relations are changed. \square

Soundness implies that **Output** contain only structures consistent with the data. Conversely, completeness of the ION algorithm requires that *every* structure that is consistent with known conditional independence information be included in **Output**. Formally, ION is complete iff, if a structure S is consistent with every G_j , then the PAG H over \mathbf{V} representing S is in **Output**.

Theorem 2.3 (completeness). Let H be a PAG over \mathbf{V} such that, for any nodes $\{X, Y\}$ d-separated or d-connected by \mathbf{Z} in some G_i , X and Y are d-separated or d-connected, respectively, by \mathbf{Z} in H_i . Then, $H \in \mathbf{Output}$.

Proof Sketch. Any non-adjacencies or orientations transferred in Step 2 must be consistent with H , since H is consistent with each G_i . Provably, at least one sequence MHS of minimal hitting sets of graph changes in Step 3b will create a graph T with all the H -adjacencies (and perhaps extra edges), and no extra orientations (\blacktriangleright or $-$) on H -edges. Step 4 provably creates a structure U resulting from T that has *exactly* the same adjacencies as H , and no extra orientations. U therefore represents H , since H can be formed by orienting \circ -marked edges in U , and so H is represented in **Output**. \square

2.4 Complexity of the algorithm

The complexity of ION is dominated by the need to compute all *minimal hitting sets* of graph changes that block all possibly d-connecting paths between pairs of nodes that are d-separated in some G_i . This step is bounded by $O(2^{|\mathbf{V}|})$ in the worst case, which is clearly too costly in general. We reduce the computational burden by dividing the hitting set computation into a sequence of smaller hitting set problems, and then adopting a branch-and-prune strategy: try small sets of graph changes, and if they result in a d-separation inconsistent with the input, stop pursuing that line. This strategy helps significantly in practice, though cannot change the worst-case performance, as minimal hitting set computation and learning structure from complete data are both known to be NP-hard in general ([12] and [13], respectively).

Branching requires additional memory, but significantly helps to avoid a substantial computational burden. We considered three strategies for surveying the potentially problematic paths P_i between pairs of nodes $\{X, Y\}$ known to be d-separated in some G_i : (1) prune after blocking all P_i of a particular length, and iteratively increase the path length (described above); (2) prune after blocking all P_i between some $\{X, Y\}$, and iterate over all $\{X, Y\}$; and (3) prune after blocking all P_i of a particular length between some $\{X, Y\}$, and iterate over both dimensions. In all cases, we compute minimal hitting sets using Reiter’s algorithm [12]. We tested all three strategies on 2500 randomly generated DAGs with 4-7 nodes, and found that they all had good computational performance for the majority of cases. Strategy (1), however, struck the best balance of speed (i.e., pruning often enough) and memory (i.e., not branching too much) so we described it above, and use it in all subsequent studies. We report more about the computational performance of ION below.

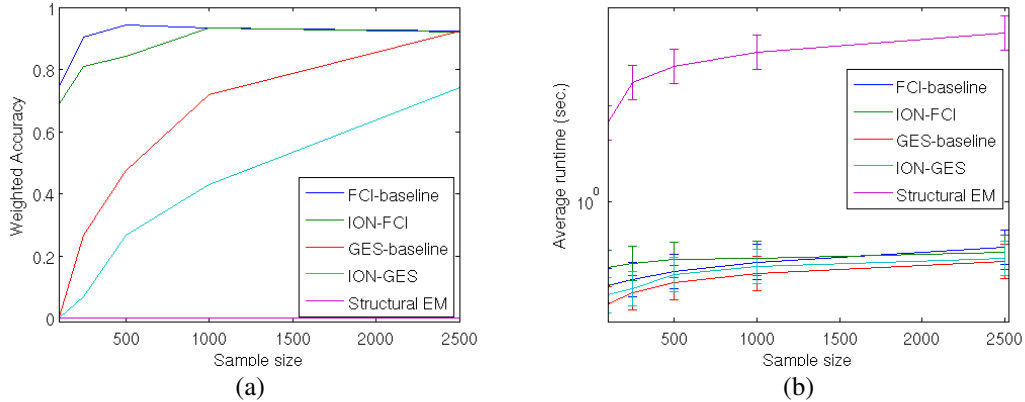


Figure 1: (a) Weighted accuracy and (b) Average runtime for 4-node DAGs

3 Experimental Results

3.1 Synthetic data

We evaluated the performance of ION with synthetic data so that its output could be compared to known ground truth. The parameters for the simulations were: $|\mathbf{V}|$ —the number of nodes in the underlying DAG; n —the number of “marginal” datasets; and N —the sample size. For each setting of these parameters, we generated 100 random DAGs of size $|\mathbf{V}|$. For each DAG, we then randomly divided the variables into n (connected) variable sets V_1, \dots, V_n , and generated a separate dataset of size N for each V_i . The ION algorithm takes PAGs as input, but is agnostic about the source of those PAGs. We thus generated input PAGs from both FCI⁵ [1] and GES [3]; we will refer to these as *ION-FCI* and *ION-GES*, respectively. We also concatenated the n datasets (with missing values) to produce an input dataset for Structural EM (i.e., the input dataset for Structural EM had $n \times N$ datapoints). We ran Structural EM with 5 random restarts (random “chains” were used as the initial model M_0), and always allowed the algorithm to converge. Finally, as a baseline of what performance could possibly be obtained, we applied FCI and GES to a single, complete dataset of size N from each DAG; we refer to these as *FCI-baseline* and *GES-baseline*, respectively.⁶

ION usually returns a larger set of possibly structures than FCI or GES, since it does not have access to the full dataset, and Structural EM returns only a single structure selected as most probable. We thus measured performance using weighted accuracy: an algorithm scored only $1 - \frac{|\mathbf{Output}|}{|\mathbf{Possible}|}$, where **Output** is the set of DAGs represented by the output of an algorithm and **Possible** is the set of all $|\mathbf{V}|$ -node DAGs, if it included the correct structure in its output set (and 0, otherwise). This measure penalizes algorithms that return very large output sets, and can be interpreted as the “useful” reduction in the search space (since excluding the correct structure is an unhelpful reduction). Figure 1 provides the weighted accuracies and average runtimes for all five algorithms using $|\mathbf{V}| = 4$, $n = 2$, and $N \in \{100, 250, 500, 1000, 2500\}$. Interestingly, the weighted accuracy of ION-FCI is remarkably similar to FCI-baseline for larger sample sizes, despite the fact that ION-FCI does not have access to the full dataset. Both GES-baseline and ION-GES improve as sample size increases, and ION-GES is only approximately 10–20% worse than GES-baseline. These four algorithms also ran quite quickly on average (under 10 seconds). In contrast, Structural EM performed quite poorly: it never learned the correct structure, and was also significantly slower than the other algorithms.

To better understand the performance of the algorithms, we examined the types of errors that each makes. Figure 2 shows, for each algorithm, the average errors of edge commission, edge omission, and orientation (e.g., \blacktriangleright instead of $-$). The poor performance of Structural EM is straightforwardly explained by Figures 2(a) and 2(b): it consistently makes large numbers of edge commission errors,

⁵Technically, we used a modified version of FCI that is more robust to small sample size errors, following the C-PC recommendations given in [14].

⁶We do not have a method to calculate the “effective sample size” for ION, but it presumably falls in $[N, N \times n]$. Use of size N datasets thus likely underestimates the “proper” baseline accuracy given full data.

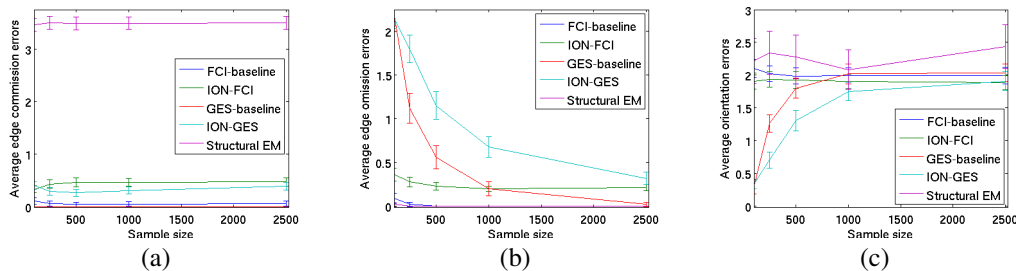


Figure 2: (a) Edge commissions, (b) Edge omissions, and (c) Orientation errors for 4-node DAGs

and almost no edge omission errors. Structural EM almost always returns the complete graph, and so is quite unreliable for structure search, at least when there is a considerable amount of data missing in a highly structured pattern.⁷ The ION algorithms consistently had slightly more errors (on average) than the baseline comparisons, and the baseline algorithms performed similarly to previous simulation studies of them.

Structural EM quickly becomes computationally intractable for larger DAGs. The runtime of Structural EM is closely tied to the number of missing values, which increases rapidly as the number of nodes in a structure increases. For example, Structural EM with 5 random restarts required 22-26 hours to converge for each of eight 5-node test DAGs ($n = 2$ and $N = 2500$). In [10], Structural EM was stopped after 30 minutes even if it had not yet converged. This strategy did not work for these datasets, however, since Structural EM only completed an average of three iterations (*not* restarts) after 60 minutes. In light of these considerations and the overall poor performance of Structural EM, we exclude it from the remaining simulations.

Figure 3 shows the weighted accuracies for three additional sets of simulations, all with $N \in \{100, 250, 500, 1000, 2500\}$. 3(a): $|\mathbf{V}| = 5$ & $n = 2$; 3(b): $|\mathbf{V}| = 6$ & $n = 2$; and 3(c): $|\mathbf{V}| = 6$ & $n = 3$. All are similar to the 4-node case, especially for FCI-baseline and ION-FCI. Moreover, the ION output set sizes remained relatively small. Figure 3(d) shows, for the $|\mathbf{V}| = 6$ & $n = 2$ parametrization, where there is the most uncertainty about the correct structure, the size of the output set returned for each of the 100 test DAGs (averaged over N) as the fraction of all possible 6-node DAGs. As expected, ION returns more structures than FCI and GES, but even the largest output set (an ION-FCI output) included less than 1% of the search space of possible 6-node DAGs.

For arbitrary connected datasets $\mathbf{D} = \{D_1, \dots, D_n\}$, as $|\mathbf{V}|$ increases, the number of structures over \mathbf{V} that predict the same d-separations and d-connections as \mathbf{D} also increases as a complex function of (i) the variable overlap in \mathbf{D} ; and (ii) features of the structures learned for each D_i (e.g., graph degree, number of colliders, etc.). For higher dimensional datasets with small overlap, simply enumerating all such possible structures can pose a significant computational and memory burden (for any approach). For some connected datasets, there may simply be too much uncertainty for ION or any complete algorithm to be computationally tractable given current hardware constraints, or even particularly effective in reducing the search space (just as there are no effective hypothesis tests for extremely small sample sizes). For higher dimensional datasets, the criteria that input structures must satisfy to be good candidate inputs for ION are largely unknown. We can confirm the existence of examples for the $|\mathbf{V}| = 15$, $n = 2$, and $N = 6000$ case where ION returns the correct structure in less than five minutes, but the extremely large space of candidate sets of higher dimensional datasets and corresponding structures make comprehensive simulation studies intractable in general, due to the number of cases with significant computational and memory burdens. Many real-world overlapping variable cases involve fewer than 15 variables, but we aim to investigate ION’s performance when $|\mathbf{V}| > 15$, and its scalability in high dimensions more generally, in future research.

3.2 Real-world data

We also tested the performance of ION-FCI on a real-world dataset measuring IQ and various neuroanatomical traits [15]. We used a single real-world dataset so that we could again apply FCI-

⁷We note in passing that previous evaluations of Structural EM (e.g., [10]) used KL-divergence as a metric.

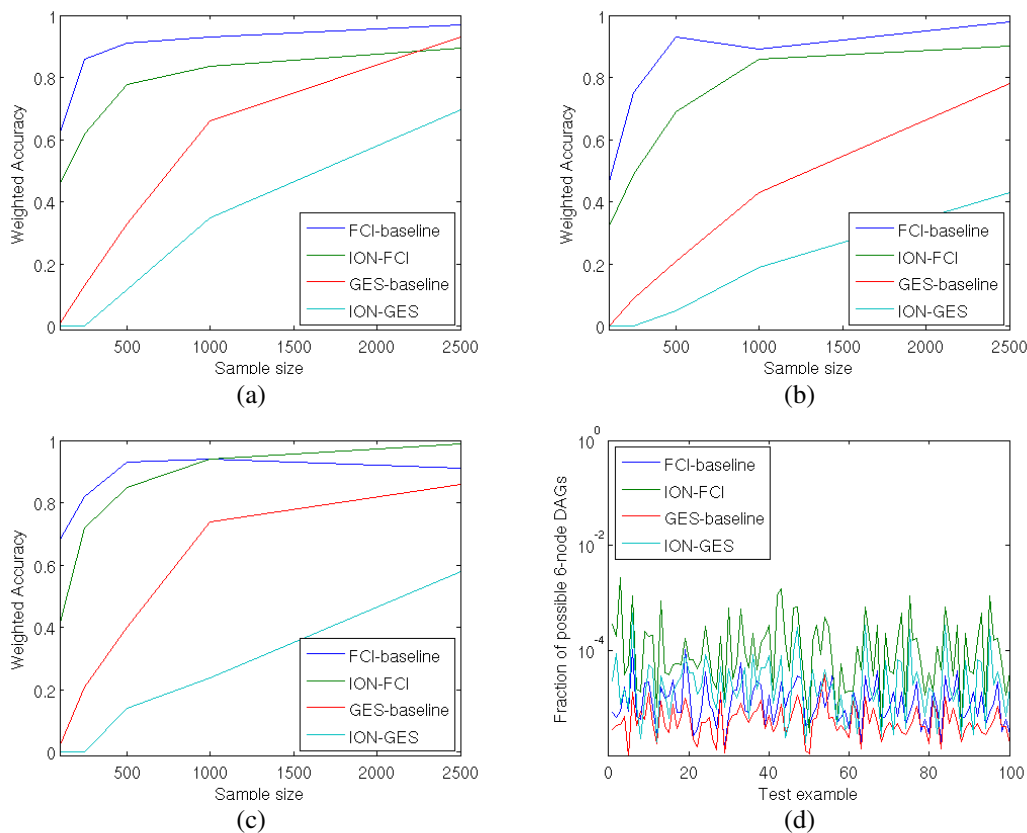


Figure 3: (a) Weighted accuracy for 5- and (b),(c) 6-node DAGs, and (d) Example output set sizes

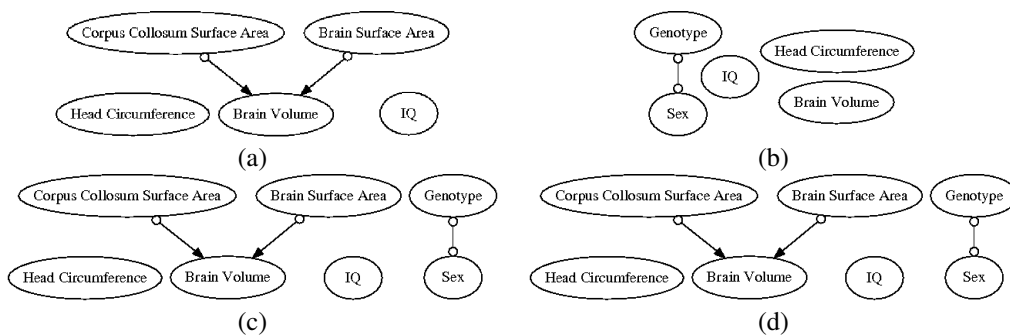


Figure 4: Output PAGs from (a),(b) FCI on “marginal” datasets; (c) FCI-ION; and (d) FCI-baseline

baseline as a comparison. We divided the variables into two overlapping sets based on domain grounds: (a) variables that might be included in a study on the relationship between neuroanatomical traits and IQ; and (b) variables for a study on the relationship between IQ, sex, and genotype, with brain volume and head circumference included as possible confounders. Figures 4(a) and 4(b) show the FCI output given only these restricted variable sets. Figure 4(c) provides the ION-FCI output, given these PAGs as input. In this particular case, the output of ION-FCI was a set with only a single PAG. Moreover, its output was exactly the same as FCI-baseline (Figure 4(d)), which had access to the full dataset. Thus, in this real-world case, ION-FCI recovers as much information about the structure as FCI-baseline, despite having access to less input information.

4 Conclusion

In scientific practice, we are often unable to construct a single, complete dataset containing every variable of interest (or doing so is very costly). We thus need some way of integrating the information about causal structure that is learnable from a collection of datasets with related variables [7]. Standard causal structure learning algorithms cannot be used, since they take only a single dataset as input. The lack of suitable algorithms constitutes a significant open problem.

To address this problem, we developed the ION algorithm: a novel, provably sound, and provably complete algorithm that learns the set of possible causal structures given connected datasets as input. While the complexity of ION is superexponential in $|\mathbf{V}|$ in the worst case, it performs well in practice when we use the branch-and-prune strategy described above. ION performed (a) significantly better than its nearest competitor (Structural EM) in both accuracy and computational tractability; and (b) comparably to baseline algorithms with access to all of the data. Ongoing and future research is focused on establishing the performance of ION for larger $|\mathbf{V}|$ and n ,⁸ as well as developing accurate and efficient methods for model averaging and model selection over the ION output set. Since ION is a complete algorithm, its output set can be quite large even with smaller values of $|\mathbf{V}|$ and n if the input does not reveal much about the overall structure for \mathbf{V} . We thus need methods for determining which causal relationships are present in all or most of the structures in the output (e.g., through model averaging). In other applications, we may be interested in finding a single most probable structure (e.g., through model selection).

References

- [1] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2nd edition, 2000.
- [2] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.
- [3] D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.
- [4] K. Sivakumar, R. Chen, and H. Kargupta. Learning Bayesian network structure from distributed data. *SIAM International Data Mining Conference*, 2003.
- [5] R. Chen, K. Sivakumar, and H. Kargupta. Collective mining of Bayesian networks from distributed heterogeneous data. *Knowledge and Information Systems*, 6:164–187, 2004.
- [6] D. Danks. Learning the causal structure of overlapping variable sets. In S. Lange, K. Satoh, and C. H. Smith, editors, *Discovery Science: Proceedings of the 5th International Conference*, pages 178–191. Springer-Verlag, 2002.
- [7] D. Danks. Scientific coherence and the fusion of experimental results. *The British Journal for the Philosophy of Science*, 56:791–807, 2005.
- [8] S. Rässler. *Statistical Matching*. Springer, 2002.
- [9] D. B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley & Sons, 1987.
- [10] N. Friedman. The Bayesian structural EM algorithm. In G. F. Cooper and S. Moral, editors, *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 129–138, 1998.
- [11] D. Geiger, T. Verma, and J. Pearl. Identifying independence in Bayesian networks. *Networks*, 20:507–534, 1990.
- [12] R. Greiner, B. A. Smith, and R. W. Wilkerson. A correction to the algorithm in Reiter’s theory of diagnosis. *Artificial Intelligence*, 41:79–88, 1989.
- [13] D. M. Chickering. Learning Bayesian networks is NP-complete. In *Proceedings of AI and Statistics*, 1995.
- [14] J. Zhang. *Causal inference and reasoning in causally insufficient system*. PhD thesis, Carnegie Mellon University, Pittsburgh, June 2006.
- [15] M. J. Tramo, W. C. Loftus, R. L. Green, T. A. Stukel, J. B. Weaver, and M. S. Gazzaniga. Brain size, head size, and IQ in monozygotic twins. *Neurology*, 50:1246–1252, 1998.

⁸The computational performance of ION is not affected by N .