Voting-based integration algorithm improves causal network learning from interventional and observational data: an application to cell signaling network inference

Meghamala Sinha[1*], Prasad Tadepalli[1], Stephen A. Ramsey[1,2]

1 School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, Oregon, USA

2 Department of Biomedical Sciences, Oregon State University, Corvallis, Oregon, USA

* sinham@oregonstate.edu

Abstract

In order to increase statistical power for learning a causal network, data are often pooled from multiple observational and interventional experiments. However, if the direct effects of interventions are uncertain, multi-experiment data pooling can result in false causal discoveries. We present a new method, "Learn and Vote," for inferring causal interactions from multi-experiment datasets. In our method, experiment-specific networks are learned from the data and then combined by weighted averaging to construct a consensus network. Through empirical studies on synthetic and real-world datasets, we found that for most of the larger-sized network datasets that we analyzed, our method is more accurate than state-of-the-art network inference approaches.

Introduction

Causal modeling is an important analytical paradigm in action planning, predictive applications, research, and medical diagnosis [1, 2]. A primary goal of causal modeling is to discover causal interactions of the form $V_i \rightarrow V_j$, where V_i and V_j are observable entities and the arrow indicates that the state of V_i influences the state of V_j. Causal models can be fit to passive observational measurements (*"seeing"*) as well as measurements that are made after performing external interventions (*"doing"*).

In many settings, observational measurements [3] are more straightforward to obtain than interventional measurements, and thus observational datasets are frequently used for causal inference. However, given only observational data, it is difficult to distinguish between compatible Markov equivalent models [4, 5]. For example, the three causal models $V_i \rightarrow V_j \rightarrow V_k$, $V_i \leftarrow V_j \leftarrow V_k$, and $V_i \leftarrow V_j \rightarrow V_k$ are Markov equivalent—each encodes the conditional independence statement $V_i \perp\!\!\!\perp V_k \mid V_j$. This ambiguity can in principle be resolved by incorporating measurements obtained from interventional experiments in which specific entities are targeted with perturbations. With the benefit of interventional measurements, Markov equivalent causal models can have different likelihoods, enabling selection of a maximum-likelihood model. These considerations have motivated the development of network learning approaches that are specifically designed to leverage mixed observational and interventional datasets [6].

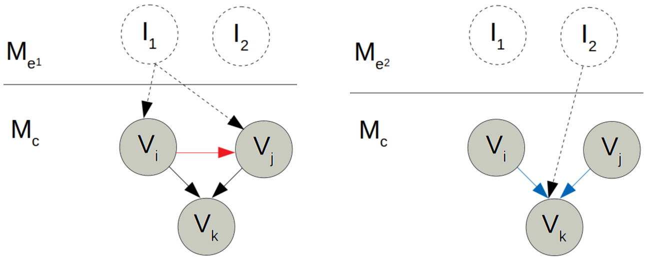
Learning a causal network from a mixed observational-interventional dataset poses methodological challenges, particularly in integrating datasets from different experiments and accounting for interventions whose effects are uncertain [7]. Due to batch effects, data collected from two different experiments might not be identically distributed and thus the two experiments may be incoherent from the standpoint of causal network model. As a result, directly combining data from different experiments can lead to errors in network learning. Interventions, too pose a challenge due to the fact that in real-world settings many interventions are (i) imperfect, meaning interventions are unreliable and have soft-targets (A "soft" target intervention, or "mechanism change," is an intervention that changes a target node's distribution's parameters, but does not render that it's independent of its parent nodes [7]), and (ii) uncertain, meaning that the "off-target" nodes are unknown. Classical causal learning algorithms are based on the assumption that interventions are *perfect* [1]; applying such algorithms to a dataset derived from imperfect interventions would likely yield spurious interactions. Eberhardt [8] classifies such errors into two types: a) *independence to dependence* errors, where two variables V_i and V_j that are independent are detected as dependent when data from the observational and interventional experiments are pooled (i.e., false positive detection of a causal interaction) and b) *dependence to independence* errors, where two variables V_i and V_j, that are dependent in an observational study are independent when the data from the observational study are pooled with data from an interventional study (i.e., a false negative for the interaction). Consensus has yet to emerge on the question of how—given two or more datasets generated from different interventions—the datasets should be combined to minimize such errors in the learned network model.

We have developed a new method, "Learn and Vote", for inferring causal networks from multi-experiment datasets. "Learn and Vote" can be used to analyze datasets from mixed observational and interventional studies and it is compatible with uncertain interventions. As it is fundamentally a data integration method, "Learn and Vote" is compatible with a variety of underlying network inference algorithms; our reference implementation combines "Learn and Vote" data integration with the Tabu search algorithm [9] and the Bayesian Dirichlet uniform (BDeu) [6, 10, 11] network score, as described below. To characterize the performance of "Learn and Vote", we empirically analyzed the network learning accuracies of "Learn and Vote" and six previously published causal network learning methods (including methods that are designed for learning from heterogeneous datasets) applied to six different network datasets. Of the six network datasets, the largest real-world dataset is a cell biology-based, mixed dataset (the Sachs et al. dataset [12]) with a known ground-truth network structure. On larger networks, we report superior (or in worst case, comparable) performance of "Learn and Vote" to the six previously published network inference methods.

Motivation and Background

Spurious dependencies and independencies

In this section, we introduce notation and describe how perturbations affecting two or more variables in a causal model can lead to spurious dependencies or independencies. Mathematically, a causal model M_c is described by a directed acyclic graph (DAG) containing a pair (V, E), where V is a set observable nodes (corresponding to random variables), E is a set of directed edges between pairs of nodes, $\mathrm{Pa}(V_i)$ represents the set of parent nodes of variable V_i, and $P(V)$ represents the joint probability distribution. In the context of network learning from interventional data, it is helpful to picture an intervention (say, I_1) as a separate type of node (denoted by a dashed circle in Fig. 1) that can be connected to its targets (say, V_i and V_j) by causal edges of a separate type (dashed arrow in Fig. 1). Applying classical network inference algorithms to



(a) False dependence

(b) False independence

Fig 1. Cross-experiment data pooling leads to network inference errors. Illustration of a simple hypothetical causal model M_c with three observable entities (V_i, V_j, and V_k. Two different interventional experiments are depicted: experiment M_{e^1} involves intervention I_1, and experiment M_{e^2} involves intervention I_2. Pooling measurements from the two experiments can cause two types of network inference errors: false positive edge (shown in (a) as a red arrow between V_i and V_j), and false negative edges (shown in (b) as blue arrows between V_i and V_k and between V_j and V_k).

measurements pooled from multiple interventional experiments can lead to two different types of learning errors, as we explain below.

1. **False causal dependence:** In the experiment depicted in Fig. 1a, V_i and V_j, which are not causally related in M_c ($V_i \not\to V_j$), are affected by intervention I_1. Due to the intervention's confounding effect, we have $V_i \not\perp\!\!\!\perp V_j$ in the combined model $M_{T_1} = M_c + M_{e^1}$ (we denote the joint distribution in the combined model by $P_1(\mathrm{V} \subset M_{T_1})$. Thus, pooling data from such different distributions may lead to spurious correlations between independent variables.

2. **False causal independence:** In the experiment depicted in Fig. 1b, the intervention I_2 on V_k removes all the incident arrows for V_k and cuts off the causal influences of V_i and V_j on V_k, causing $V_i \perp\!\!\!\perp \mathrm{Pa}(V_i)$. Pooling data from such models can cause the causal dependencies $V_i \to V_k$ and $V_j \to V_k$ in M_c to be missed (i.e., a "false negative" in the inferred network).

Review of prior literature

Classical causal learning methods fall into two classes: *constraint-based* methods (e.g., PC [2], FCI [13]), in which the entire dataset is analyzed using conditional independence tests; and *score based* methods (e.g., GES, GIES [14]), in which a score is computed from the dataset for each candidate network model. Both classes of methods were designed to analyze a single observational dataset, with the attendant limitations (in the context of multi-experiment datasets) that we described above. Several multi-dataset network inference approaches have been proposed that circumvent the above-described problems associated with cross-experiment measurement pooling. Cooper and Yoo [6] proposed a score-based algorithm that combines data from multiple experiments, each having perfect interventions with known targets. The approach was later refined by Eaton and Murphy [7] for uncertain and soft interventions [15]. The method of Claassen and Heskes [16] is based on imposing the causal invariance property across environment changes. Sachs et al. [12] analyzed a molecular biology dataset (which has since become a benchmark dataset for molecular network inference, a primary application focus of our work) using a variant of the Cooper-Yoo method. Chen et al. [17] proposed a subgraph-constrained approach, called Trigger, to learn a yeast gene regulatory network model from transcriptome and genotype data. In the Joint Causal Inference (JCI) [18] method, additional experimental context variables are introduced before data pooling. Notably, the aforementioned methods assume some prior knowledge about the network model. In contrast, our "Learn and Vote" method (see Methods and Datasets) requires no prior knowledge about the network model.

Network Combination Methods:

Another class of multi-dataset network inference approaches, which we call "network combination" methods, involve learning causal interaction statistics from each experiment followed by integration of the statistics to obtain a single consensus network. For example, in the ION [19] method, locally learned causal networks having overlapping variables are integrated. The constraint-based COmbINE [20] method is based on the estimation of variable-variable dependencies and independencies across separate experiments. The MCI [21] algorithm is a constraint-based method that exploits the 'local' aspect of causal V-structures [22]. However, none of these methods produce experiment-specific weighted graphs, instead enumerating experiment-specific partial ancestral graphs that are consistent with the data. In real-world datasets, due to a variety of factors (finite sampling, experiment-specific biases and confounding effects, measurement error, missing data, and uncertain/imperfect interventions), the confidence with which a given causal interaction $V_i \to V_j$ can be predicted within a given experiment will in many cases vary significantly from experiment to experiment (and in the case of incomplete measurements, may not be quantifiable at all in a given experiment). Thus, a network combination method compatible with experiment-specific edge weights would seem to offer a distinct advantage in the context of multi-experiment network inference. Furthermore, all of these methods assume that a single underlying causal model accounts for all observed causal dependencies. In real-world settings where experimental conditions change across experiments, this assumption seems unlikely to hold, motivating the need for network inference methods that can (1) score candidate interactions within individual experiment-specific datasets and (2) combine weighted edges from experiment-specific datasets into a consensus network.

Biological Signaling Networks

A cell signaling network is a type of causal network in which the state of a protein or other biomolecule influences the state of another protein or biomolecule downstream of it (denoted by a directed arc). Such networks are amenable to interventional experiments using molecular agents that target (i.e., activate or inhibit) specific molecules. Sachs et al. [12] used a Bayesian network approach to infer causal interactions among eleven signaling molecules in human CD4+ T-cells. In a series of nine experiments—two observational and seven with specific molecular interventions—they measured the activation levels of eleven phosphorylated proteins and phospholipids by flow cytometry (Figure 2). They found 17 true positive interactions, with 15 that were well-established in the biology literature and two that were supported by at least one study; their inferred network missed three arcs (false negatives) and it had no false positive arcs.

Uncertain interventions

Like most causal network learning approaches, the method used in the Sachs et al. study and in our re-analysis assumes perfect interventions, i.e., that each of the interventional agents targets exactly one of the signaling molecules. Such a perfect intervention assumption is likely not consistent with typical interventions in biological systems, due

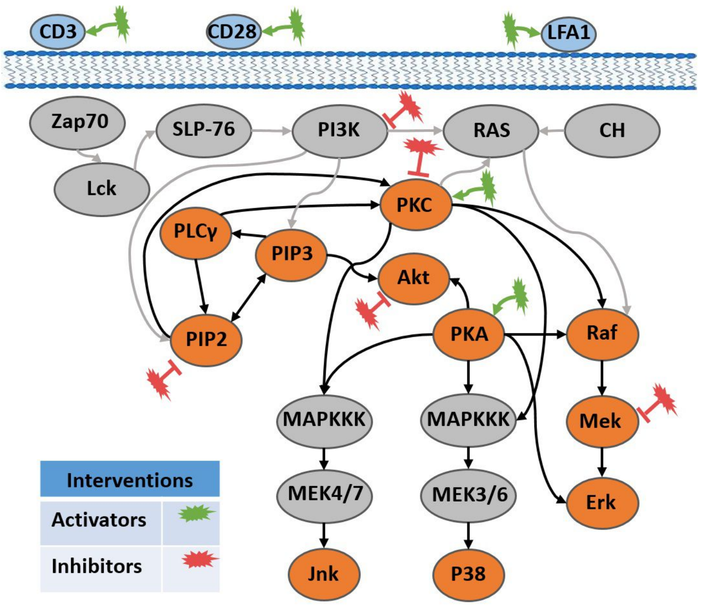


Fig 2. Biological network for the Sachs et al. study, showing interactions (arcs) and interventions (starred ellipses). The pathways represented by bold black lines are the Ground Truth known causal interactions, established through literature study.

to potential off-target effects of pharmaceutical agents. Moreover, in a biological system, the effects of certain types of interventions (for example, a gene knockout) may not be describable by forcing of a target node's state to a specific value in the observational network. In the Sachs et al. experiments, although the interventions are assumed to be perfect, they are known to have off-target effects, as shown by Eaton & Murphy (2007) [7]. Eaton & Murphy modeled chemical interventions as context variables in the network (assuming they had some known background knowledge about the underlying network) to learn the intervention's effects and found them to have multiple children. To summarize, in the context of current learning algorithms, there are three primary issues with pooling experimental data that were acquired with imperfect interventions:

1. Current algorithms might make mistakes since the arcs pointing towards the unknown targets are not removed or handled properly.

2. Although pooling data adds more confidence into learning the true causal arcs, it can also introduce spurious arcs with incorrect direction (see Fig. 4).

3. Each intervention might alter a mechanism or influence the local distribution in an unknown way [23].

Methods and Datasets

To avoid the problems arising from pooling data from different experiments in causal network learning, we propose the "Learn and Vote" method (shown in Fig. 3 and Algorithm 1). The method's key idea is to (1) learn a separate weighted causal network from the data generated in each experiment (which may be interventional or observational) by ignoring the directed arcs into the intervened variables and then (2) combine the experiment-specific networks by weighted averaging. The algorithm's inputs are, for each experiment, the values of the observed variables (V) in the experiments (we denote the number of variables by v and the number of experiments by k) and the identities of the known target nodes (stored as a list *intv*) for any interventions.

Algorithm 1 Learn and Vote

```
    Input: set of k experiments with dataset $D_1$,$D_2$...$D_k$
    Output: DAG $G^f$ = (E, V), final causal network
procedure OUR APPROACH
    for j = 1 to k do
        V = nodes In $D_j$
        intv = Intervened nodes in $D_j$
        randomNet = createRandNet(V, 100)
        for l = 1 to 100 do
            Net[l] = Tabu(randomNet[l], intv)
        arcProb[j] = arcStrength(Net)
    avgArcs = avgNetwork(arcProb)
    $G^f$ = learnDAG(avgArcs,Threshold)
```

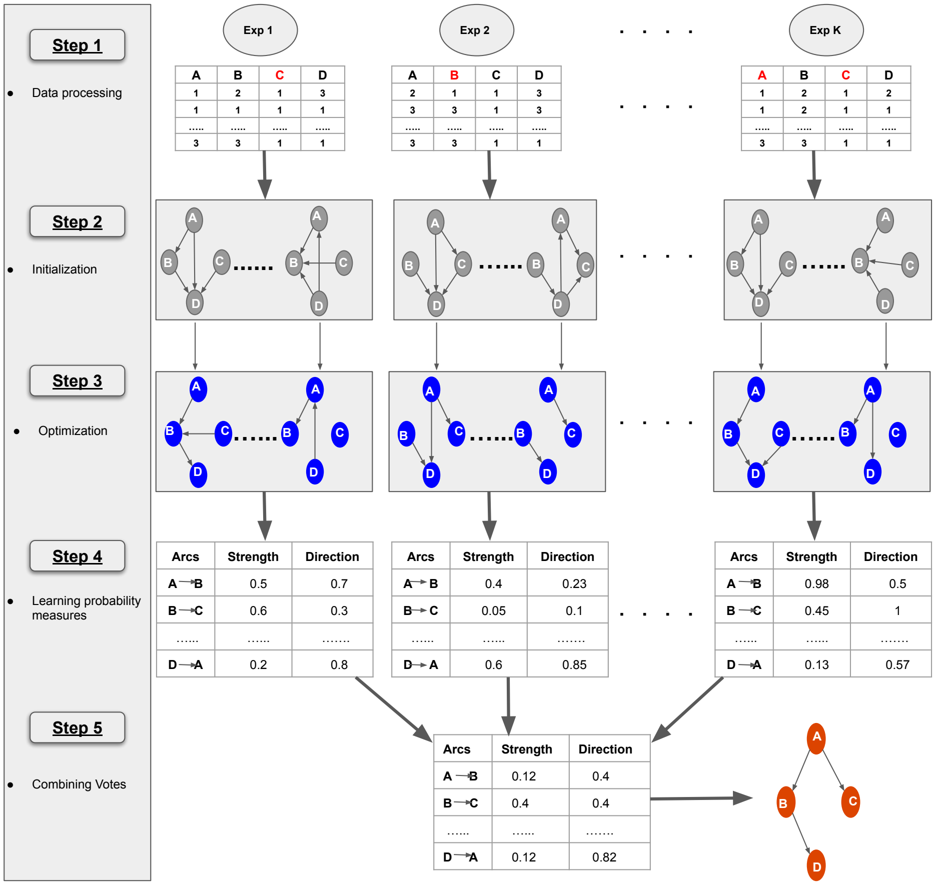
Scoring Function

We incorporate the effect of intervention in the score component associated with each node by modifying the standard Bayesian Dirichlet equivalent uniform score (BDeu) [6,10,11]. Given measurements D_j of variables V in experiment j, let G^j represent a DAG learned from it (with conditional distributions $P(V_i|\mathrm{Pa}(V_i)^{G^j})$, where $\mathrm{Pa}(V_i)^{G^j}$ is the set of parent nodes of V_i in DAG G^j). In a perfect interventional experiment, for the set $\mathrm{Int}(m)$ of intervened nodes in sample m, we fix the values of $V_i[m] \in \mathrm{Int}(m)$, meaning that we exclude $P(V_i[m] \mid \mathrm{Pa}(V_i)[m])$ from the scoring function for $V_i \in \mathrm{Int}(m)$. All the other unaffected variables are sampled from their original distributions. The distribution of D_j is per experiment and not a pooled dataset of all experiments as in the Sachs et al. method. We define an experiment-specific network score $S(G^j : D_j)$ as sum (over all variables V_i) of per-variable local scores $S_{\mathrm{local}}(V_i, U : D_j)$ of variables V_i. The left part of the equation is the prior probability assigned to the choice of set U as parents of V_i, and the right part is the probability of the data integrated over every possible parameterizations (θ) of the distribution.

$$S_{\mathrm{local}}(V_i, U : D_j) = \log P(Pa_i = U) + \log \int \prod_{m, V_i \notin \mathrm{Int}(m)} P(V_i[m]|U[m], \theta) dP(\theta).$$

Structure learning

Because our method uses local stochastic search (Tabu), we create an ensemble of n random starting DAGs (stored as `randomNet`, see Algorithm 1) using the procedure `createRandNet`. Empirically, we have found that $n = 100$ is adequate for the network multivariate datasets that we analyzed in this work to demonstrate empirical performance of our method (see Results). From each DAG in `randomNet`, we then search for an optimal network model using the Tabu search algorithm [9] and store the n networks in a list `Net`. The list `intv` of known targets is passed as an argument which incorporates interventions in the search algorithm by preventing the arcs to be incident on the targets. Next, we measure the probabilistic arc strength and direction (using the procedure `arcStrength`) for each arc as its empirical frequency given the list of networks in `Net`. We average the arc strengths for every directed arc over the networks in which corresponding target node was not intervened and store them as a list `arcProb`.



A	B	C	D
1	2	1	3
1	1	1	1
.....	.....	.....	.....
3	3	1	1

A	B	C	D
2	1	1	3
3	3	1	3
.....	.....	.....	.....
3	3	1	1

A	B	C	D
1	2	1	2
1	2	1	1
.....	.....	.....	.....
3	3	1	1

Arcs	Strength	Direction
A → B	0.5	0.7
B → C	0.6	0.3
......	......	....
D → A	0.2	0.8

Arcs	Strength	Direction
A → B	0.4	0.23
B → C	0.05	0.1
......	......	....
D → A	0.6	0.85

Arcs	Strength	Direction
A → B	0.98	0.5
B → C	0.45	1
......	......	....
D → A	0.13	0.57

Arcs	Strength	Direction
A → B	0.12	0.4
B → C	0.4	0.4
......	......	....
D → A	0.12	0.82

Fig 3. Workflow of "Learn and Vote": **Step 1** - Collecting data from k experiments (combination of observational and interventional studies). For interventional studies, the known targets (marked in Red) are incorporated as external perturbation during the search process. **Step 2** - Creating 100 random DAGs using the observed nodes, as a starting point. **Step 3** - Optimizing each of the 100 DAGs with data using Tabu search. **Step 4** - Calculating probability (in terms of strength and direction) of occurrence for every possible arc from the 100 optimized DAGs and storing them in tables. **Step 5** - Combining votes from all the tables by weighted averaging and constructing the final causal network, with arc strengths above a threshold (in this case 50%)

.

Combining results from experiments

Given arc information (in `arcProb`, see Algorithm 1) from each experiment, we average their strengths and directions over the number of experiments where the given arc is valid (using procedure `avgNetwork`). Finally, we compute the averaged arc strengths as `avgArcs` and threshold on arc strength (using a predefined `Threshold`) in order to produce the final DAG (using procedure `learnDAG`). We found that our method performs best at a 50% threshold. We implemented "Learn and Vote" in the R programming language, making use of the `bnLearn` package [24].

Datasets that we used for empirical performance analysis

From six published networks, we obtained nine datasets (with associated ground-truth networks) that we analyzed in this work. To avoid bias, from each network we used both observational and interventional datasets. For synthetic networks, as observations, we drew random samples. As interventions, we set some target nodes to fixed values. Next, in order to model uncertainty, we also set one or more of the target's children to different values (like "fat-hands" [7]) which are assumed to be unknown. Finally, we sampled data from each of the mutilated networks [25] :

- **Lizards:** a real-world dataset having three variables representing the perching behaviour of two species of lizards in the South Bimini island [26]. We generated one observational dataset and two interventional datasets from the lizards network.
- **Asia:** a synthetic network of eight variables [27] about occurrence of lung diseases and their relation with visits to Asia. For our empirical study, we created two mutilated networks: `Asia_mut1` has one observation and one interventional dataset, and `Asia_mut2` has one observational and two interventional datasets.
- **Alarm:** a synthetic network of thirty seven variables representing an alarm messaging system for patient monitoring [28]. For our study, we created two mutilated networks: `Alarm_mut1` has three observational and six interventional studies, and `Alarm_mut2` has five observational and ten interventional datasets.
- **Insurance:** a synthetic network of twenty seven variables for evaluating car insurance risks [29]. We created two mutilated networks: `Insurance_mut1`, from which we obtained one observational and five interventional datasets; and `Insurance_mut2`, from which we obtained three observational and eight interventional datasets.
- **gmInt:** a synthetic dataset containing a matrix of observational and interventional data from eight Gaussian variables, provided in the `pcalg-R` package.
- **Sachs et al.:** a cell signaling network and associated mixed observational-interventional dataset published by Sachs et al. [12], described above).

Causal network learning methods that we compared to "Learn and Vote"

Using the aforementioned networks and datasets, we compared the accuracy of "Learn and Vote" for network inference to the following six algorithms (implemented in `R`):

- **PC:** We used the observational datasets to evaluate DAG-equivalent structures [2], and we used Fisher's z-transformation conditional independence test (varying α from 0 to 1).

- **GDS:** This is a greedy search method [14] to estimate Markov equivalence class of DAG from observational and interventional data. It works by maximizing a scoring function (L_0-penalized Gaussian maximum likelihood estimator) in three phases, i.e., addition, removal and reversal of an arrow, as long as the score improves.

- **GIES:** This algorithm [14] generalizes the greedy equivalence search (GES) algorithm (Chickering 2002) to include interventional data into observational data.

- **Globally optimal Bayesian Network (simy):** This is a score-based dynamic programming approach [30] to find the optimum of any decomposable scoring criterion (like BDe, BIC, AIC). This function (simy) estimates the best Bayesian network structure given interventional and observational data but is only feasible up to about 20 variables.

- **Invariant Causal Prediction (ICP):** This method by Peters et al., [31] calculates the confidence intervals for causal effects by exploiting the invariance property of a causal (vs. non-causal) relationship under different experimental settings. We implemented it in R, making use of the `InvariantCausalPrediction` package.

- **Sachs et al. method** The Bayesian network approach used by Sachs et al. was described in Methods and Datasets above.

For each of these methods except PC, the method implementations that we used were adapted for heterogeneous datasets (see citations above).

Performance measurement

For the purpose of quantifying the accuracies of the nine networks learned by each of the seven network algorithms, we treated the presence of an arc in the ground-truth dataset as a “positive” and its absence as a “negative”. For each inferred network and each algorithm, from the confusion matrix we computed precision, recall, and the F1 harmonic mean of precision and recall (we did not compute accuracy due to the inherent class imbalance of sparse networks), as shown in Table 1.

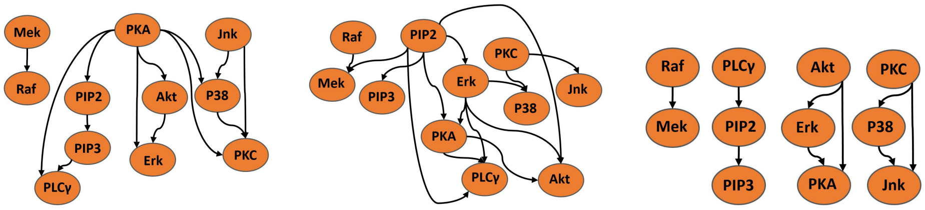
Results

Effect of interventions on network inference

Based on prior studies suggesting that incorporating data from interventional studies improves network inference (see Introduction), we re-analyzed the Sachs et al. [12] biological cell signaling dataset (for which a ground truth network was published [12]) using their published inference approach twice, first using observational samples only (Figure 4a) and then using an equal number of samples comprising 50% observational and 50% interventional data (Figure 4b). We found that sensitivity for detecting cell signaling interactions increases when data from observational and interventional experiments are co-analyzed (Fig. 4b), versus when only data from observational experiments are used (Fig. 4a). These results illustrate the benefit of using data from interventional experiments for causal network reconstruction.

Effect of pooling on network inference

Based on prior studies suggesting that pooling data from multiple experiments can lead to errors in network learning (see Introduction), we analyzed the same cell signaling



(a) TP: 3, FP: 10, FN: 17 **(b)** TP: 5, FP: 10, FN: 15 **(c)** TP: 4, FP: 5, FN: 16

Fig 4. Networks inferred by (a) pooling data from two observational experiments; (b) pooling data from an observational (anti-CD3/CD28) and an interventional experiment (AKT inhibitor); and (c) "Learn and Vote" using the same experiments as in the middle panel. The structure learning statistics used are True Positive (TP), False Positive (FP) and False Negative (FN). False positives are reduced by avoiding pooling.

dataset as in Fig. 4b, using the "Learn and Vote" method, in which data are not pooled. Compared to the the Sachs et al. inference method which was based on data pooling (Fig. 4b), use of "Learn and Vote" significantly reduced false positives, while increasing the overall robustness of the network learning (Fig. 4c).

Systematic comparative studies

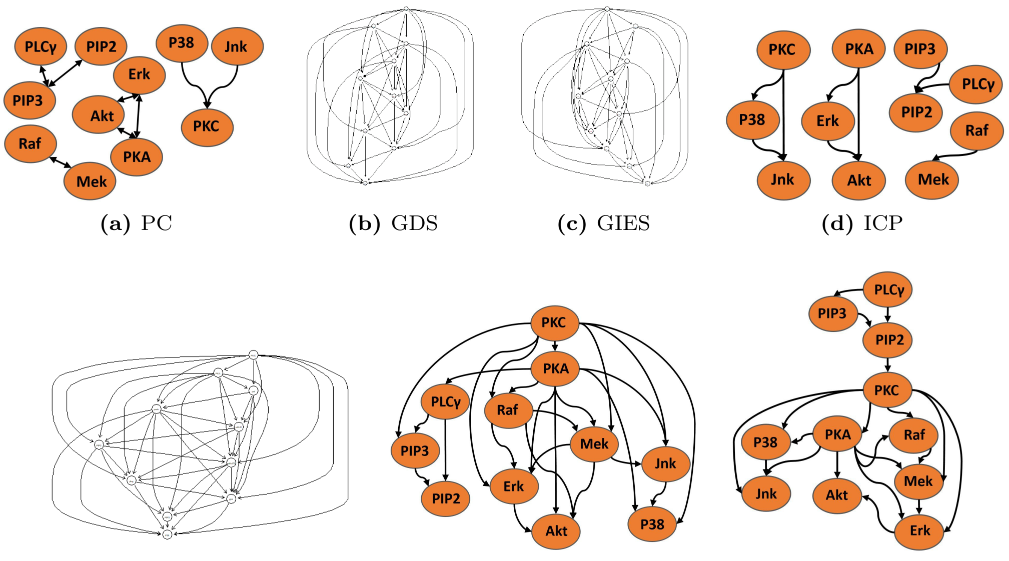
To study the performance characteristics of "Learn and Vote" for a broader class of network inference applications, we carried out a systematic, empirical comparison our method's performance with six previously published causal network learning methods using nine datasets (from six underlying networks of small to medium size, as described above in Methods and Datasets), spanning a variety of application domains.

Networks learned by the seven methods on the cell signaling dataset

On the Sachs et al. dataset, the consensus networks that each algorithm learned are shown in Fig. 5a-g; the networks varied significantly in terms of density, with GDS, GIES, and simy giving large numbers of edges, and PC and ICP giving relatively sparse networks (with the PC network having many ambiguous arc directions). For each of the methods, we tabulated the numbers of correct and incorrect (or missing) arcs in the consensus networks learned (Fig. 5h). The greedy algorithms (Fig. 5b,c) and simy (Fig. 5e) are able to find most of the true positive arcs at the cost of a large number of false positives. The consensus "Learn and Vote" network (Fig. 5g) improved over the consensus network obtained using the Sachs et al. inference method (Fig. 5f), by eliminating six false positive edges and gaining a true positive edge ($PIP2 \rightarrow PKC$) (Fig. 5h, rightmost two columns). We further note that two of the putatively false interactions that were detected by "Learn and Vote", ($P38 \rightarrow pjnk$) and ($PKC \rightarrow p44.42$), on further study are likely real interactions according to PCViz (www.pathwaycommons.org/pcviz) and PubMed (www.ncbi.nlm.nih.gov/pubmed). Moreover, our method had the lowest number of false positives among all seven methods and was tied for second-highest in terms of the number of true positives (Fig. 5h).

Quantifying performance of seven network learning algorithms

In Table 1, we summarize the performance, in terms of network learning precision, recall, and F1 score of the seven network inference methods applied to nine datasets (with associated ground-truth networks) that were described in Methods and Datasets. In terms of F1 accuracy, while the PC algorithm (which used *observational*



(a) PC **(b)** GDS **(c)** GIES **(d)** ICP

(e) simy **(f)** Sachs **(g)** Learn and Vote

Method	PC	GDS	GIES	ICP	simy	Sachs et al.	Learn & Vote
True Positive	8	18	17	9	19	17	18
False Positive	6	25	28	0	26	8	2
False Negative	12	2	3	11	1	3	2

(h) Performance

Fig 5. Consensus networks inferred from various algorithms (a-g) on the Sachs et al. cell signaling dataset. A bidirectional arrow between two nodes denotes that an interaction is predicted between the two nodes, but the direction of causality is ambiguous. In the table (h), each row corresponds to a component of the confusion matrix (true positives, false positives, and false negatives), and each column corresponds to a causal network inference method.

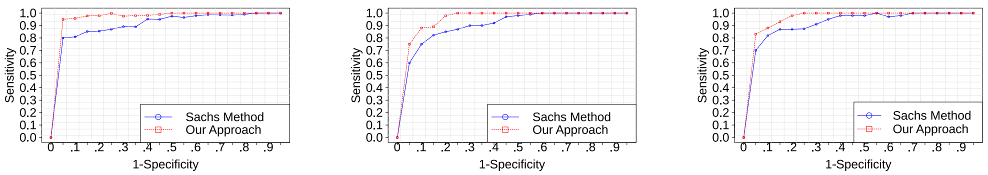
Table 1. Multi-dataset performance of "Learn & Vote" versus six other methods. Each row corresponds to a specific dataset derived from a specific underlying ground-truth network (as described in detail in Methods and Datasets). Each row is split into three performance measures (precision, recall, and the "F1" harmonic mean of precision and recall). For each sub-row, the highest performance measurement is boldfaced. Each column corresponds to a specific method for causal network inference (as described in detail in Methods and Datasets), with the performance measures of our method ("Learn and Vote") in the rightmost column. The symbol "n/a" denotes that no performance results were available for that method on that dataset. Here, the method "simy" is only feasible for networks containing up to 20 nodes, so it failed to produce results on the larger networks. The network **size** denotes the number of nodes in the indicated network. The network **type** is as follows: RW, real-world; S, synthetic.

Dataset	**size**	**type**	**Metric**	**PC**	**GDS**	**GIES**	**ICP**	**simy**	**Sachs et al.**	**Learn & Vote**
Lizards	3	RW	Precision	**1**	**1**	**1**	0	**1**	**1**	**1**
			Recall	**1**	**1**	**1**	0	**1**	0.5	0.5
			F1 score	**1**	**1**	**1**	0	**1**	0.667	0.667
Asia_mut1	8	S	Precision	**1**	0.625	0.625	**1**	0.316	0.77	**1**
			Recall	0.75	0.625	0.625	0.5	0.75	**0.875**	0.75
			F1 score	**0.857**	0.625	0.625	0.666	0.444	0.824	**0.857**
Asia_mut2	8	S	Precision	**1**	0.857	0.857	**1**	0.304	0.666	**1**
			Recall	0.75	0.75	0.75	0.5	**0.875**	0.75	0.75
			F1 score	**0.857**	0.8	0.8	0.666	0.493	0.706	**0.857**
gmInt	8	S	Precision	0.75	0.889	0.889	**1**	0.889	0.857	**1**
			Recall	0.75	**1**	**1**	0.375	**1**	0.75	0.75
			F1 score	0.75	**0.94**	**0.94**	0.545	**0.94**	0.8	0.857
Cell signaling	11	RW	Precision	0.571	0.419	0.377	**1**	0.422	0.68	0.89
			Recall	0.4	**0.9**	0.85	0.45	0.95	0.85	0.89
			F1 score	0.47	0.572	0.522	0.62	0.584	0.756	**0.89**
Insurance_mut1	27	S	Precision	0.714	0.36	0.362	0.7	n/a	**0.857**	0.8
			Recall	0.288	0.346	0.327	0.25	n/a	**0.577**	0.538
			F1 score	0.411	0.352	0.343	0.368	n/a	**0.689**	0.643
Insurance_mut2	27	S	Precision	**0.714**	0.355	0.366	0.64	n/a	0.676	0.686
			Recall	0.288	0.423	0.423	0.21	n/a	0.442	**0.461**
			F1 score	0.411	0.386	0.392	0.316	n/a	0.535	**0.552**
Alarm_mut1	37	S	Precision	0.666	0.25	0.26	**0.7**	n/a	0.625	0.564
			Recall	0.434	0.217	0.26	0.26	n/a	**0.446**	0.4
			F1 score	**0.526**	0.232	0.26	0.38	n/a	0.52	0.468
Alarm_mut2	37	S	Precision	0.666	0.411	0.513	0.6	n/a	0.725	**0.769**
			Recall	0.434	0.456	0.434	0.21	n/a	0.63	**0.642**
			F1 score	0.526	0.432	0.47	0.311	n/a	0.675	**0.7**

measurements) has strong performance on smaller networks, “Learn and Vote” has superior performance for learning the structure of larger networks. More broadly, “Learn and Vote” outperformed the other six algorithms in five out of nine studies in terms of precision, with the ICP method having second best performance. The positive predictive rate of our approach is higher for small or medium sized networks (i.e., fewer than 20 nodes) but decreases as the size of the network increases. In contrast, the greedy algorithms (GDS, GIES) perform well for smaller networks but suffer from lower precision on larger networks. In terms of F1, our approach outperformed the others in five out of nine studies and is more stable even when the network size increases. For very small networks (i.e., fewer than 10 nodes), the PC-based approach has good sensitivity, however, it leaves many of the arc directions ambiguous (Fig. 5a).

Sensitivity to threshold

To study the sensitivity of our results to the threshold parameter (which was set to 0.5) for predicting a causal arc, we compared the performance of “Learn and Vote” to that of the Sachs et al. method on three different network datasets (cell signaling, Asia_mut1, and Asia_mut2; see Methods and Datasets) by plotting the sensitivity versus false positive error rate (FPR) for various threshold values (Fig. 6a). On all three datasets, in terms of area under the sensitivity-vs-FPR curve, “Learn and Vote” has a higher score than the Sachs et al. method, with the most significant performance gap occurring at thresholds where the specificity is in the range of 0.7–0.9.



(a) Cell signaling

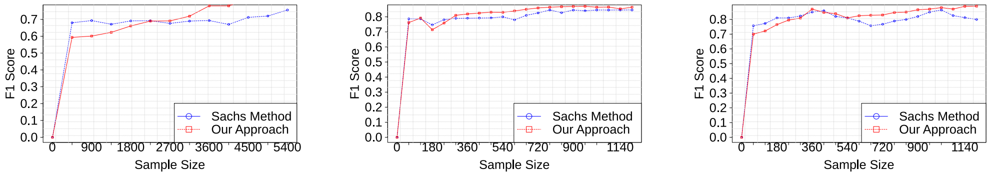
(b) Asia_mut1

(c) Asia_mut2

Fig 6. Sensitivity vs. FPR for “Learn and Vote” and the Sachs et al. method on three datasets: (a) Sachs et al. cell signaling; (b) Asia lung disease (mut1); and (c) Asia lung disease (mut2). The line plots are nonmonotonic due to the use of different random initial DAGs for different points on the line plot.

Effect of sample size

It seems intuitive that in cases where single-experiment sample sizes are very small, separately analyzing data from individual experiments would be expected to perform poorly relative to a pooling-based approach like the Sachs et al. method. To test this, we analyzed the how the relative performances of “Learn and Vote” and the Sachs et al. method vary with sample size on the Sachs et al. dataset (for which the Sachs et al. method was specifically developed). We sampled equal numbers of data points from each experiment to prevent bias towards a particular experiment. Fig. 7 shows the performance of our method versus the Sachs et al. method by varying the numbers of samples used from each experiment. When the number of samples per experiment is very small, learning from pooled data gives a better result. For the Asia network, which has eight nodes, when the number of samples per experiment is very small (e.g., 20 samples), the performance of “Learn and Vote” is no better than the pooling-based Sachs et al. method (Fig. 7b-c). Hence, when only a small amount of data are available it is a good idea to combine them irrespective of experimental conditions. However, for large enough sample size, we see that pooling degrades accuracy of network reconstruction.



(a) Cell signaling

(b) Asia_mut1

(c) Asia_mut2

Fig 7. F1 vs. sample size for Learn and Vote and the Sachs et al. method, for three datasets.

Discussion

Taken together, our results (Fig. 5 and Table 1) suggest that for analyzing datasets from studies that have imperfect interventions, greedy analysis methods (e.g., GDS, GIES) are not as accurate as "Learn and Vote". On the other hand, ICP is conservative due to its strict invariance property and helps reduce false causal arcs to a great extent, but at the cost of sensitivity (Fig. 5d). The relatively poor performance of the PC method on the Sachs et al. dataset likely reflects the fact that it does not utilize interventional data. In future work, we plan to study the case of handling uneven samples of data from different experiments. We also plan to extend the work by choosing which interventional target is more informative in an unknown network structure. Another improvement of our approach is to see how choosing the number of random DAGs (we have taken 100) scales with network size. For example, in case of larger graphs, 100 might not be sufficient while in smaller graphs it could be overkill. One possible improvement to "Learn and Vote" would be an adaptive method for selecting the number of random initial DAGs; this is an area of planned future work.

Conclusion

We report a new approach, "Learn and Vote," for learning a causal network structure from multiple datasets generated from different experiments, including the case of hybrid observational-interventional datasets. Our approach assumes that each dataset is generated by an unknown causal network altered under different experimental conditions (and thus, that the datasets have different distributions). Manipulated distributions imply manipulated graphs over the variables, and therefore, combining them to learn a network might increase statistical power but only if it assumes a single network that is true for every dataset. Unfortunately, this is not always the case under uncertain interventions. Our results are consistent with the theory that simply pooling measurements from multiple experiments with uncertain interventions leads to spurious changes in correlations among variables and increases the rate of false positive arcs in the consensus network. In contrast, our "Learn and Vote" method avoids the problems of pooling by combining experiment-specific weighted graphs. We compared "Learn and Vote" with six other causal learning methods on observational and interventional datasets having uncertain interventions. We found that for most of the larger-network datasets that we analyzed, "Learn and Vote" significantly reduces the number of false positive arcs and achieves superior F1 scores. However, for cases where sample size per experiment is very small, we found that pooling works better. Our findings (i) motivate the need to focus on the uncertain and unknown effects of interventions in order improve causal network learning precision, and (ii) suggest caution in using causal learning algorithms that assume perfect interventions, in the context of real world domains that have uncertain intervention effects.

Acknowledgment

Research reported in this publication was supported in part by the National Center for Advancing Translational Sciences (NCATS), National Institutes of Health, through the Biomedical Data Translator program (award OT2TR002520 to SAR). The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

References

1. Pearl J. Causality: models, reasoning, and inference. Econometric Theory. 2003;19(675-685):46.
2. Spirtes P, Glymour C, Scheines R. Causation, prediction, and search. Adaptive computation and machine learning; 2000.
3. Hagmayer Y, Sloman SA, Lagnado DA, Waldmann MR. Causal reasoning through intervention. Causal learning: Psychology, philosophy, and computation. 2007; p. 86–100.
4. Koller D, Friedman N, Bach F. Probabilistic graphical models: principles and techniques. MIT press; 2009.
5. Pearl J, Mackenzie D. The Book of Why: The New Science of Cause and Effect. Basic Books; 2018.
6. Cooper GF, Yoo C. Causal discovery from a mixture of experimental and observational data. In: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 1999. p. 116–125.
7. Eaton D, Murphy K. Exact Bayesian structure learning from uncertain interventions. In: Artificial Intelligence and Statistics; 2007. p. 107–114.
8. Eberhardt F. A sufficient condition for pooling data. Synthese. 2008;163(3):433–442.
9. Glover F. Future paths for integer programming and links to artificial intelligence. Computers & operations research. 1986;13(5):533–549.
10. Heckerman D, Geiger D, Chickering DM. Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning. 1995;20(3):197–243.
11. Pe'er D, Regev A, Elidan G, Friedman N. Inferring subnetworks from perturbed expression profiles. Bioinformatics. 2001;17(suppl_1):S215–S224.
12. Sachs K, Perez O, Pe'er D, Lauffenburger DA, Nolan GP. Causal protein-signaling networks derived from multiparameter single-cell data. Science. 2005;308(5721):523–529.

13. Spirtes P, Meek C, Richardson T. Causal inference in the presence of latent variables and selection bias. In: Proceedings of the Eleventh conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 1995. p. 499–506.

14. Hauser A, Bühlmann P. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. Journal of Machine Learning Research. 2012;13(Aug):2409–2464.

15. Tian J, Pearl J. Causal discovery from changes. In: Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 2001. p. 512–521.

16. Claassen T, Heskes T. Causal discovery in multiple models from different experiments. In: Advances in Neural Information Processing Systems; 2010. p. 415–423.

17. Chen LS, Emmert-Streib F, Storey JD. Harnessing naturally randomized transcription to infer regulatory relationships among genes. Genome biology. 2007;8(10):R219.

18. Mooij JM, Magliacane S, Claassen T. Joint Causal Inference from Multiple Contexts. arXiv preprint arXiv:161110351. 2016;.

19. Danks D, Glymour C, Tillman RE. Integrating locally learned causal structures with overlapping variables. In: Advances in Neural Information Processing Systems; 2009. p. 1665–1672.

20. Triantafillou S, Tsamardinos I. Constraint-based causal discovery from multiple interventions over overlapping variable sets. Journal of Machine Learning Research. 2015;16:2147–2205.

21. Claassen T, Heskes T. Causal discovery in multiple models from different experiments. In: Advances in Neural Information Processing Systems; 2010. p. 415–423.

22. Mani S, Spirtes PL, Cooper GF. A theoretical study of Y structures for causal discovery. arXiv preprint arXiv:12066853. 2012;.

23. Tian J, Pearl J. Causal discovery from changes. In: Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 2001. p. 512–521.

24. Scutari M. Learning Bayesian networks with the bnlearn R package. arXiv preprint arXiv:09083817. 2009;.

25. Pearl J. Graphical models for probabilistic and causal reasoning. In: Quantified representation of uncertainty and imprecision. Springer; 1998. p. 367–389.

26. Schoener TW. The Anolis lizards of Bimini: resource partitioning in a complex fauna. Ecology. 1968;49(4):704–726.

27. Lauritzen SL, Spiegelhalter DJ. Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society Series B (Methodological). 1988; p. 157–224.

28. Beinlich IA, Suermondt HJ, Chavez RM, Cooper GF. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In: AIME 89. Springer; 1989. p. 247–256.

29. Binder J, Koller D, Russell S, Kanazawa K. Adaptive probabilistic networks with hidden variables. Machine Learning. 1997;29(2-3):213–244.

30. Silander T, Myllymaki P. A simple approach for finding the globally optimal Bayesian network structure. arXiv preprint arXiv:12066875. 2012;.

31. Peters J, Bühlmann P, Meinshausen N. Causal inference by using invariant prediction: identification and confidence intervals. Journal of the Royal Statistical Society B (Statistical Methods). 2016;78(5):947–1012.