Towards Robust and Versatile Causal Discovery for Business Applications

Giorgos Borboudakis
Computer Science Department, Univ. of Crete
Gnosis Data Analysis IKE
borbudak@gmail.com

Ioannis Tsamardinos
Computer Science Department, Univ. of Crete
Gnosis Data Analysis IKE
tsamard.it@gmail.com

ABSTRACT

Causal discovery algorithms can induce some of the causal relations from the data, commonly in the form of a causal network such as a causal Bayesian network. Arguably however, all such algorithms lack far behind what is necessary for a true business application. We develop an initial version of a new, general causal discovery algorithm called ETIO with many features suitable for business applications. These include (a) ability to accept prior causal knowledge (e.g., taking senior driving courses improves driving skills), (b) admitting the presence of latent confounding factors, (c) admitting the possibility of (a certain type of) selection bias in the data (e.g., clients sampled mostly from a given region), (d) ability to analyze data with missing-by-design (i.e., not planned to measure) values (e.g., if two companies merge and their databases measure different attributes), and (e) ability to analyze data from different interventions (e.g., prior and posterior to an advertisement campaign). ETIO is an instance of the logical approach to integrative causal discovery that has been relatively recently introduced and enables the solution of complex reverse-engineering problems in causal discovery. ETIO is compared against the state-of-the-art and is shown to be more effective in terms of speed, with only a slight degradation in terms of learning accuracy, while incorporating all the features above.The code is available on the mensxmachina.org website.

Keywords

Causal Discovery; Semi-Markov Causal Models; Latent Variables; Selection Bias; Multiple Datasets; Bayes-Ball Algorithm; Answer Set Programming; Interventional Data

1. INTRODUCTION

Knowledge of causal relations is necessary to plan effective interventions, such as launching a new advertising campaign or a promotion. Causal discovery algorithms attempt to induce some of these relations from data, often presenting them in the form of a network such as a causal Bayesian network. Unfortunately, we argue that most approaches leave much to be desired for a true business application. For example, Bayesian networks assume that there are no latent confounding factors or selection bias, which is unrealistic and leads to erroneous inductions [21, 19]. Other algorithms admit latent confounders but are nevertheless brittle to statistical errors and small sample sizes. Other desired features of the algorithms are also missing, e.g,. the ability to impose causal prior knowledge such as “no measured quantity causally affects the age of a client” or to co-analyze data that follow different distributions, e.g. before and after a promotion.

Recently, the problem of causal discovery has been formulated as a logic-based inverse engineering problem [25, 14, 13, 24]. The approach has paved the way for algorithms that are able to handle much more complex settings, imposing fewer, less restrictive assumptions. The price to pay is increased computational overhead. The main idea is the following: each conditional (in)dependence discovered in the data through a statistical hypothesis test imposes a constraint on the presence or absence of paths in the (unknown) causal graph, after accounting for the effects of interventions and the presence of selection. The graphs that satisfy all constraints are possible solutions to the problem. This approach leverages decades-long research in logic-based inference engines such as SAT solvers, logic-programming, and answer set programming. ETIO (from the Greek word for “cause”) is a proposed algorithm that follows this approach and demonstrates the usefulness and potential of these recent advances to causal discovery on business data. Specifically, ETIO has the following features important for a business application:

Admits latent variables: Latent confounding factors present inherent problems to causal Bayesian networks and lead to erroneous inductions. Confounding factors are quantities that causally affect two or more of the measured quantities. For example, if the true graph contains the subgraph $X \leftarrow L \rightarrow Y$ where L is not measured or included in the data, a dependence is induced between X and Y. *Any* Bayesian network algorithm will asymptotically return either the network $X \rightarrow Y$ or $X \leftarrow Y$ trying to explain the dependence. The induced network is equivalent to the true one for predictive purposes but erroneous for causal purposes: neither X nor Y causally affect each other. The presence of latent confounders is a possibility that cannot be excluded a priori in most realistic scenarios. ETIO employs more advanced formalisms and theory to account for the

KDD '16 August 13-17, 2016, San Francisco, CA, USA

ACM ISBN 978-1-4503-4232-2/16/08.

DOI: http://dx.doi.org/10.1145/2939672.2939872

presence of latent confounders, namely semi-Markov causal models (**SMCM**) [22].

Admits selection bias in the data: Selection bias is another source of errors if ignored. Imagine that X and Y are two factors with no causal relation between them that both affect whether a person chooses to stay in New York State denoted as N. A business that operates in New York and deals only with local clients gathers data conditional on $N = 1$. In their data X and Y will be found dependent. A Bayesian network algorithm will induce that either $X \rightarrow Y$ or $X \leftarrow Y$ trying to explain the dependence which is wrong from a causal perspective. Again, for predictive modeling selection bias presents no problems as long as one applies the model to the same population, in this case New York State clients. However, if ignored, the presence of correlations due to selection (conditioning) leads to *causal* inductions that are wrong even for the same population. ETIO can handle selection that depends on the observed variables, but not on latent confounders and observed variables at the same time.

Handling data with missing-by-design values: Suppose a business merges two of their internal databases containing some common variables stored and some distinct, e.g., in case of an acquisition of a new company. Thus, the pooled data matrix contains large blocks of missing values. These values are called missing-by-design in the statistical terminology [8]. Using the logic-based approach, ETIO is able to make use of all of the available data and make non-trivial inferences such as inducing relations between variables never jointly measured (similarly to [27]).

Handling data from different interventions: In another scenario, a business initiates an intervention, e.g., a promotion, an ad campaign, or a change in their portal. The data distribution before and after the intervention may be different; however, the internal causal mechanisms that determine customer behavior have not changed, only the exogenous conditions and stimuli to the customers. ETIO looks for causal models that fit to the data after accounting for the effect of manipulations similarly to [14, 13, 24] and can handle data under different interventions and distributions.

Incorporating Causal or Associative Prior Knowledge: Typically, the semantics of the variables carry important causal information. For example, no client attribute causally affects their age; dropping one's subscription occurs after all other recorded variables and thus cannot be causally affecting any other measured quantity, such as the cost of the insurance plan. More complicated scenarios are also possible to encode such as asserting the belief that taking senior driving courses improves driving skills (i.e., asserting the presence of a causal path in the graph). ETIO accepts this type of causal knowledge facts and employs them to constrain the set of admissible solutions.

In the next sections, we demonstrate and explain the capabilities of ETIO on a use case from the insurance domain. We also perform empirical evaluation experiments against the only other alternative in the literature [13]. The latter does not implement all the features of ETIO but follows a similar approach and could potentially be extended to this direction. The main differences are the encoding scheme and conflict resolution strategy used. In simulated experiments on observational data ETIO shows better scaling up, while performing slightly worse in terms of learning accuracy. We also compare ETIO against FCI [21, 28], a prototypical algorithm for causal discovery admitting latent confounders and selection bias in the data. ETIO and FCI perform similarly in terms of learning accuracy. However, the main goal and contribution of this paper is to introduce and illustrate the potential of the logic-based approach to causal discovery to address the needs of business applications.

2. BACKGROUND

We assume the reader's familiarity with Bayesian networks (**BN**); see [21, 19, 17] for an introduction to causality and Bayesian networks. **Semi-Markov causal models** (**SMCM**) [22] (also known as acyclic directed mixed graphs (ADMG)) are generalizations of causal BNs that represent the presence of latent confounders. An SMCM represents both (in)dependence relations as well as causal relations among variables. The structure of SMCMs is a graph $\mathcal{G} = \langle \mathbf{V}, \mathbf{E} \rangle$, where $\mathbf{V}$ is a set of nodes and $\mathbf{E}$ a set of edges. Nodes represent variables, whereas edges represent relations between variables. SMCMs may contain both, directed ($\rightarrow$) and bi-directed ($\leftrightarrow$) edges. A directed edge $X \rightarrow Y$ denotes that X is a direct cause of Y in the context of the measured variables, while a bi-directed edge $X \leftrightarrow Y$ denotes that X and Y share a latent common cause. SMCMs represent the causal and probabilistic properties of marginals of BNs. There may be at most one edge of each type (directed or bi-directed) between two nodes. The graph $\mathcal{G}$ of a SMCM is **acyclic**, that is, there is no directed cycle in it so that causal feedback is excluded. A node X is a **parent** (**ancestor**) of Y, and Y is a **child** (**descendant**) of X, if $X \rightarrow Y$ (a directed path from X to Y, denoted as $X \dashrightarrow Y$) is in $\mathcal{G}$. An edge between X and Y is **into** Y, if $X \rightarrow Y$ or $X \leftrightarrow Y$, and is **out of** Y otherwise ($X \leftarrow Y$). Similarly, a (not necessarily directed) path between X and Y is into Y if the previous node on the path is into Y, and out of Y otherwise. A triplet $\langle X, Y, W \rangle$ of nodes is called a **collider** in $\mathcal{G}$ if both X and W are into Y.

Selection bias [6, 2] arises if the samples are not uniformly randomly selected from a population but their inclusion in the dataset depends on one or more variables, e.g. in case a business mostly has clients from a specific region. Selecting a sample for inclusion in the data can be modeled as a dummy binary variable N. All samples are thus conditioned on the fact that $N = 1$: *selection bias is equivalent to conditioning.* If N is causally affected by observed variables $X \rightarrow N \leftarrow Y$ then X and Y will be found dependent in the sample. Thus, selection bias introduces spurious dependencies among variables not present in the full population. Selection bias cannot be ignored in a practical business application; one needs to model it during causal discovery. ETIO admits the possibility that some observed variables S may be causing selection N. We assume that selection in the sample does not causally affect any other observed variable, e.g., the fact that a client has been included in our database does not affect any other of their attributes. In addition, we assume that selection is not caused by latent confounders (e.g. $X \leftarrow L \rightarrow N \leftarrow Y$ is not permitted); this complicates modeling selection and is left for future work. ETIO does not directly need to model the dummy variable N; what is important to reason with are the variables that causally affect N. The following concept definition of **m-separation**, here generalized to include selection, is fundamental in causal modeling:

Algorithm 1 Bayes-Ball for SMCMs with selection

```
Input: SMCM $\mathcal{G}$, Node $X$, Conditioning Nodes $\mathbf{Z}$
Output: Nodes $\mathbf{Y}$ m-connected to $X$ given $\mathbf{Z}$
Visit $X$ from node $X$ {Case 0}

When visiting a node $Y$ from node $W$:
if $W = Y$ or ($W \leftarrow Y$ and $Y \notin \mathbf{Z}$) then {Case 1}
    Visit $U$ if $Y \rightarrow U$ or $Y \leftarrow U$ or $Y \leftrightarrow U$ or $Y, U \in \mathbf{S}$
end if
if ($W \rightarrow Y$ or $W \leftrightarrow Y$) and $Y \in \mathbf{Z}$ then{Case 2}
    Visit $U$ if $Y \leftarrow U$ or $Y \leftrightarrow U$
end if
if ($W \rightarrow Y$ or $W \leftrightarrow Y$) and $Y \notin \mathbf{Z}$ then {Case 3}
    Visit $U$ if $Y \rightarrow U$ or $Y, U \in \mathbf{S}$
end if
```

Definition 1. In a SMCM G with causes of selection in the set S (possibly empty), a path p in G between nodes X and Y is **m-connecting** relative to (condition to) a set of nodes $\mathbf{Z}_S = \mathbf{Z} \cup S$, $(X, Y \notin \mathbf{Z}_S)$, if: (a) every non-collider on p is not a member of $\mathbf{Z}_S$, and (b) every collider on p is a member of $\mathbf{Z}_S$, or an ancestor of some member of $\mathbf{Z}_S$. X and Y are said to be **m-separated** by $\mathbf{Z}_S$ if there is no m-connecting path between them relative to $\mathbf{Z}_S$. Otherwise, X and Y are **m-connected** by $\mathbf{Z}_S$.

*The major assumption of the prototypical algorithms for causal discovery [21] is that X and Y are independent conditioned on $\mathbf{Z}$ and selection S if and only if X and Y are **m-separated** by $\mathbf{Z}_S$.* The "if" part is called the **Causal Markov Condition** and the "only if" part the **Faithfulness Condition** [21]. Intuitively, both conditions together imply that (in)dependencies appear only due to the causal structure and the selection process, not due to the exact parameterization of the distribution. *In(dependencies) are determined by performing an appropriate conditional independence test on the data.* ETIO employs Bayesian tests that return the probability that a given conditional dependence holds in the data. The main principle in the logic-based approach to causal discovery is that results of the tests of independence correspond to m-connection or m-separation constraints that should hold in the unknown causal graph, after accounting for selection and interventions (discussed below). For example, if a given independence is more probable than the corresponding dependence, an m-separation constraint is imposed. ETIO imposes the constraints that correspond to test results, in order of probability, while removing conflicting test results.

Interventions (a.k.a. manipulations, perturbations) modify the structure of the causal graph and change the data distribution, thus requiring special treatment and modeling. Reviews of various types of interventions, as well as discussions on their implications in causal discovery can be found in [15, 9]. The type of interventions we focus on are described following [9]. Such interventions may be modeled by including an additional indicator variable I for each different intervention with directed edges into the manipulated variables, taking values 1 in the samples where the specific intervention was present and 0 otherwise. **Structural interventions** (also known as **hard interventions**) are interventions that completely cut-off the influence of other causes on the manipulated variable (target), that is, they remove all edges into the target. For example, forcing all drivers to install an anti-lock breaking system (ABS) removes all its influences completely. **Parametric interventions** (also known as soft interventions) on the other hand affect the distribution of the target variable, but do not remove the influence of other variables and the corresponding edges. For example, a promotion to install an ABS increases the probability of installation, but does not eliminate all other causes. An intervention is **confounding** if it affects multiple target variables simultaneously. Naturally, an experimental dataset may stem from different combinations of interventions. Hereafter, we assume that probability distributions resulting from interventions remain faithful to the underlying, possibly manipulated, causal structure [10]. In addition, we assume that interventions are **exogenous**, that is, they are not causally affected by and are not confounded with the modeled variables.

For structural interventions incoming and bidirectional edges of the targets are removed in the graph that models the data where $I = 1$; for parametric interventions these edges are not removed. In each case, an additional edge is included from I to each target variable (in a structural intervention this is actually not necessary). An intervention is **uncertain** if its targets are not exactly known. For uncertain interventions ETIO will have to figure out from the data the possible targets; for certain interventions, the corresponding manipulation edge removals and additions can be imposed as facts. Finally, we say that an intervention is **possibly ineffective** if it is not known whether it is parametric or structural (see [9] for a discussion). ETIO can handle possibly ineffective interventions too.

3. M-SEPARATIONS IN SMCMS

The Bayes-Ball algorithm [20] (shown in Algorithm 1 and adapted to fit our notation) identifies all nodes $\mathbf{Y}$ m-connected with one or multiple nodes X given a set $\mathbf{Z}$ in a Bayesian network (or DAG). Here it is extended to handle SMCMs with selection. In later sections, the algorithm is represented in logic so that m-connection constraints can be imposed to a logic solver. The Bayes-Ball algorithm leads to an efficient encoding and is the reason behind the improved computational performance of ETIO, as shown in Section 7.

The algorithm starts from nodes X and visits other nodes according to the definition of m-connection so that it only visits m-connected nodes. The set $\mathbf{S}$ contains all nodes which directly (in the context of modeled variables) affect selection. It is easy to see that the extended algorithm shown in Algorithm 1 is correct according to the definition of m-connection; a proof sketch follows. For latent variables, it suffices to see that each bi-directed edge between two nodes Y and U corresponds to a parent L_{YU} of them. Visiting U from Y through a bi-directed edge (cases 1 and 2) is equivalent to two steps in the search in the original algorithm: first, L_{YU} is visited as a parent of Y (cases 1 or 2) and then, U is visited as a child of L_{YU} (case 1). Cases 2 and 3 are trivially applicable when $W \leftrightarrow Y$, by substituting it with $W \leftarrow L_{WY} \rightarrow Y$, which matches with the preconditions of cases 2 and 3. The naive approach to handle selection is to include an additional node N in $\mathcal{G}$ and in the conditioning set $\mathbf{Z}$. An alternative way is to mark each node as to whether it affects selection or not, and to extend cases 1 and 3 of the algorithm. This can be done by allowing the algorithm in both cases to visit Y and U if both are selected; no

Table 1: Logic variables in the encoding and their semantics

$X \rightarrow Y$	X has an arrow into Y
$X \leftrightarrow Y$	X and Y are confounded
$X \dashrightarrow Y$	X is an ancestor of Y
$X \underset{Z,D}{\cdots} Y$	mconn$(X, Y, \mathbf{Z})$ in dataset D
$X \underset{Z,D}{\cdot\cdot >} Y$	mconn$(X, Y, \mathbf{Z})$ in dataset D (path into Y)
$X \underset{Z,D}{\cdot\cdot -} Y$	mconn$(X, Y, \mathbf{Z})$ in dataset D (path out of Y)
X^S_D	X is used for selection in dataset D
X^I_D	X is manipulated (hard) in dataset D

matter the incoming edge from W, Y and U are trivially m-connected through N, as N is conditioned on, as long as Y is not in $\mathbf{Z}$. Finally, since $U \rightarrow N$, the resulting m-connecting path is out of U. One important thing to note is that Y and U are not necessarily distinct nodes, otherwise cases such as $X \rightarrow Y \leftarrow W$ would not be handled correctly if Y is in $\mathbf{S}$.

4. THE BASIC ETIO ALGORITHM

A solution to a causal discovery problem is a causal graph that implies the same conditional (in)dependencies as found in the data, also called as a graph that fits the data. Typical causal discovery algorithms return either an arbitrary solution or another type of graph representing the equivalence class of solutions (called essential graph or PDAG for Bayesian networks). In the type of problems handled by ETIO, the set of equivalent solutions cannot be represented by a graph and their number can grow exponentially; complete enumeration of solutions is impractical. ETIO instead follows what is called the **query-based** approach, where the user queries the algorithm about some causal structural features of interest. For example, the user may query whether X causes Y in all solutions and thus necessarily entailed by the data (m-connections and m-separations) and the assumptions. ETIO is able to reason using a possibly incomplete set of dependence, independence and prior knowledge constraints, which is not possible using classical causal discovery algorithms. Thus, ETIO can handle missing-by-design data or ignore statistical test results that are deemed unreliable. We proceed by providing a high-level overview of the ETIO algorithm; implementation details, as well as the specific instantiation decisions we used are described in Section 4.2.

The input to the ETIO algorithm are: (a) a set of datasets $\mathbf{D}$, (b) meta-information about the datasets $\mathbf{MI}$, such as whether selection bias may be present, if they are observational or inverventional data, and if so, which are the known (if any) intervention targets and what type of interventions were performed, (c) a set of structural prior knowledge constraints $\mathbf{PK}$, and (d) query features $\mathbf{Q}$ of the causal structure that should be output (for example, directed edges and ancestral relations). The implementation of the algorithm also accepts parameters that dictate whether to admit latent variables and/or selection bias or not, not shown here for brevity; by default, we assume both are possible. ETIO first performs conditional independence tests on the data and represents the results in logic; it also represents known facts about selection and targets of interventions. It then selects a consistent subset of dependence and prior knowledge constraints (in case there are conflicts) and finally it identifies invariant features implied by the input constraints. The procedure is summarized in Algorithm 2.

Algorithm 2 Basic ETIO Algorithm

```
Input: Datasets $\mathbf{D}$, Meta-Information $\mathbf{MI}$, Prior Knowledge $\mathbf{PK}$, Queries $\mathbf{Q}$
Output: Query Results
constraints ← createConstraints($\mathbf{D}$, $\mathbf{MI}$, $\mathbf{PK}$)
constraints ← resolveConflicts(constraints)
results ← makeInferences(constraints, $\mathbf{Q}$)
```

4.1 Imposing and Encoding Constraints

In this section we show how to encode dependence, independence and various kinds of prior knowledge constraints in first-order logic. We proceed by defining the primitive logic variables used in our encoding. For clarity, we will call *variable* a binary variable in the logic problem and *node* a random variable of our data distribution that appears in the unknown causal graph. Instead of using a predicate notation such as DirectedEdge(X, Y) we use the notation $X \rightarrow Y$ as more convenient.

We are seeking the structure of a causal graph with nodes as many random variables appear in the union of the datasets as well as dummy variables I_D indicating the presence of a possible intervention in a dataset D. For each pair of nodes X and Y, we introduce a primitive variable for the presence or absence of a directed edge $X \rightarrow Y$, and one for the bidirected edge $X \leftrightarrow Y$. After imposing logical constraints, the truth assignment of these variables will correspond to an SMCM that fits all datasets after accounting for interventions and selection. We also define the auxiliary variable $X \dashrightarrow Y$ to denote the presence or absence of a directed path from X to Y.

A second set of primitive variables denoted as $X \underset{Z,D}{\cdots} Y$ denotes the presence or absence of the m-connection of X with Y in dataset D given subset Z. It is important to notice that different m-connections may hold after accounting for the interventions or selections in different datasets, thus the dataset parameter is necessary. There is one such variable for each pair X and Y of nodes, each input dataset, and each conditioning set Z that appears in any conditional independence test performed by the algorithm. *The algorithm determines and imposes the truth value of the $X \underset{Z,D}{\cdots} Y$ variables by performing conditional independence tests on the available datasets.* Not all possible tests need to be performed or considered, as is the case for example when blocks of data are missing. Section 4.2 explains the details of the strategy for performing tests.

We distinguish between two different types of m-connecting paths: two variables X and Y may be m-connected by a path that is either into Y or out of Y, denoted respectively by the auxiliary variables $X \underset{Z,D}{\cdot\cdot >} Y$ and $X \underset{Z,D}{\cdot\cdot -} Y$. The distinction is made to implicitly keep track of the edge that led to an m-connecting path from X to Y (that is, it is equivalent to recording the previous node W of the Bayes-Ball algorithm). As we will see below, those constraints are used to incrementally encode m-connecting paths.

A third set of primitive variables denoted by X^S_D and X^I_D represent the fact whether node X is causing selection in

Table 2: Inference rules

$$
\begin{aligned}
X \dashrightarrow Y &\Leftrightarrow X \rightarrow Y \vee (X \dashrightarrow U \wedge U \dashrightarrow Y) && (1)\\
\neg Y \rightarrow X &\Leftarrow X \dashrightarrow Y && (2)\\
X \underset{Z,D}{\cdots} Y &\Leftrightarrow X \underset{Z,D}{\cdot\cdot >} Y \vee X \underset{Z,D}{\cdot\cdot -} Y && (3)\\
X \underset{Z,D}{\cdot\cdot -} U &\Leftrightarrow (X = U \wedge X \notin Z) && (4a)\\
&\vee \ (X \underset{Z,D}{\cdot\cdot -} Y \wedge U \rightarrow Y \wedge Y \notin Z \wedge \neg Y_D^I) && (4b)\\
&\vee \ (X \underset{Z,D}{\cdot\cdot >} Y \wedge U \rightarrow Y \wedge Y \in Z \wedge \neg Y_D^I) && (4c)\\
&\vee \ X \underset{Z,D}{\cdot\cdot -} Y \wedge Y_D^S \wedge U_D^S \wedge Y \notin Z && (4d)\\
&\vee \ X \underset{Z,D}{\cdot\cdot >} Y \wedge Y_D^S \wedge U_D^S \wedge Y \notin Z && (4e)\\
X \underset{Z,D}{\cdot\cdot >} U &\Leftrightarrow (X \underset{Z,D}{\cdot\cdot -} Y \wedge Y \rightarrow U \wedge Y \notin Z \wedge \neg U_D^I) && (5a)\\
&\vee \ (X \underset{Z,D}{\cdot\cdot >} Y \wedge Y \rightarrow U \wedge Y \notin Z \wedge \neg Y_D^I \wedge \neg U_D^I) && (5b)\\
&\vee \ (X \underset{Z,D}{\cdot\cdot -} Y \wedge Y \leftrightarrow U \wedge Y \notin Z \wedge \neg U_D^I) && (5c)\\
&\vee \ (X \underset{Z,D}{\cdot\cdot >} Y \wedge Y \leftrightarrow U \wedge Y \in Z \wedge \neg Y_D^I \wedge \neg U_D^I) && (5d)
\end{aligned}
$$

D or is the target of a hard intervention in D respectively. There are X_D^S and X_D^I variables for each node X that appears in any input dataset, and each dataset D. If in a dataset D a hard intervention has occurred with a known target X, then X_D^I can be set to true. If it is set to false, then the algorithm assumes that X is not a hard target of I in D. If it is not set, the algorithm will try to infer the value of X_D^I from the rest of constraints, if possible. Similarly, for causes of selection, X_D^S can be set to true or false if it is a known fact, or be left unknown to be determined by the algorithm. Obviously, the more information is known regarding the targets of interventions and causes of selection, the more inferences can be made by the algorithm. All variables defined and their semantics are shown in Table 1.

Having defined the variables, we now present the inference rules that constrain their truth values and ensure their semantics are respected. The complete list of inference rules, including selection and interventions, is shown in Table 2. *For the sake of simplicity we omit the existential quantifier $\exists$; for example, the right part of rule 1 should normally we written as $X \rightarrow Y \vee (\exists U \ X \dashrightarrow U \wedge U \dashrightarrow Y)$.* For the moment, we ignore interventions and prior knowledge constrains that are dealt with in subsequent subsections.

Rule 1 is used to define ancestral relations, while rule 2 enforces acyclicity. The remaining rules directly encode the Bayes-Ball algorithm in first-order logic. Rule 3 encodes that an m-connecting path between two variables X and Y relative to a separating set and dataset must be either into or out of Y. Rule 4a corresponds to case 0 in the Bayes-Ball algorithm, that is, that each variable is m-connected to itself. Rules 4b, 4d, 5a and 5c correspond to case 1 in Bayes-Ball: the right-hand side of those rules requires that there is an m-connecting path between X and Y that is out of Y, and that Y is not in the separating set. Similarly, rules 4c and 5d match case 2, while rules 4e and 5b match case 3 of the Bayes-Ball algorithm. Note that all rules except 2 are bidirectional which is necessary to be able to make useful inferences; implicitly this stems from the Faithfulness Condition which translates to the fact that if m-connection holds a dependence is implied.

Encoding Interventions. We consider various different cases of both, structural and parametric interventions. Specifically we assume that the interventions are exogenous, non-confounding (also called independent) and not uncertain, but allow for possibly ineffective interventions. Under those assumptions we can handle: (a) single or multiple independent structural interventions, and (b) single or multiple independent parametric interventions. In principle the proposed encoding can also be extended to handle confounding, uncertain and non-exogenous interventions, but we did not further investigate those cases in this work.

In order to encode structural interventions it suffices to label the nodes as intervened or not intervened; we use variables X_D^I to encode whether or not X has been intervened in dataset D. For possibly ineffective interventions one can simply omit assigning a truth value to the respective variable. Recall that, in case of structural interventions all incoming edges at the target variable are eliminated and thus not observed in the data. Thus, it suffices to forbid certain m-connecting paths stemming from such constraints. Specifically, whenever an inference rule in groups 4 and 5 requires that there is an edge into a variable Y, we must ensure that Y is not manipulated; if Y is manipulated, the manipulated graph would not contain any edges into Y, and thus no such m-connection would be observed.

Parametric interventions, as already mentioned, can be encoded by including an additional indicator variable for each target variable. This variable can only have an edge into its respective target variable. In case it is possibly ineffective, the value of the edge variable can be set to unknown. Finally, note that when including data from parametric interventions, the data have to contain at least two values for the indicator variable, in order to be able to perform conditional independence tests. No further special treatment is needed, at least not for the cases we consider.

Incorporating Causal Prior Knowledge. Prior knowledge has several advantages: (i) it can reduce the number of errors in the output, as it helps in filtering out inconsistent input, (ii) it can increase the number of inferences made by the algorithm, and (iii) it may also decrease the total running time of the algorithm. ETIO accepts **structural prior knowledge** as in [4], i.e., knowledge about the causal structure in the form of hard constraints: structures not consistent with it are eliminated from the solution set. Interested readers on methods for structure learning with prior knowledge may refer to [1].

In principle, all kinds of structural prior knowledge can be incorporated, as long as it can be expressed in first-order logic. Examples of common types of structural prior knowledge are: (a) presence/absence of direct causal edges, (b) presence/absence of direct connections (causal edges and/or latent variables), (c) presence/absence of possibly indirect causal relations (that is, ancestral relations), (d) (conditional) dependence and independence constraints, (e) root or leaf nodes, that is, no incoming or outgoing edges respectively, (f) limits on the in/out-degree of nodes, (g) complete or partial order of variables. All of those can be easily encoded using the primitive propositions (logic variables) defined in Table 1. Note that, all of those constraints have already appeared in the literature [18, 7, 4]. However, none of the previous approaches is general enough to handle both SMCMs and such a variety of structural prior knowledge.

Correctness. A full proof of correctness and completeness is omitted due to lack of space. A proof sketch would follow the one-to-one correspondence with the Bayes-Ball algorithm to prove that all conditional (in)dependencies found in the data are imposed as m-connections or m-separations in the logical representation, in a way that every truth assignment to the variables corresponds to a causal graph that entails the same m-connections. The algorithm is **query-complete** in the sense that it will output true for all queries (e.g., presence of a given edge) that are entailed by the data and the assumptions.

4.2 Implementation Details and Decisions

Performing Conditional Independence Tests. In step 1, the ETIO algorithm performs several conditional independence tests on each input dataset $\mathbf{D}$. Typical non-Bayesian tests are the G^2 for nominal nodes and the Fisher z-test for continuous nodes (a.k.a. partial correlation test). One could perform all possible conditional independence tests. This is feasible for small datasets of around 10 variables, as in most of our experiments. It may however be undesirable since conditioning on too many variables reduces the statistical power of the test, as well as increasing the number of constraints to consider. For larger datasets, one can perform as many tests as possible, e.g., considering all possible conditioning sets up to k variables. For sparse networks a value of $k = 5$ should suffice. For even larger datasets, one could focus on a set of nodes of interests and retrieve their Markov Blankets in a recursive fashion [26]; we intend to explore this direction in future work. Other strategies for selecting which tests to perform and focus on are possible and an interesting direction to explore to scale the algorithms. ETIO performs non-Bayesian tests that return p-values of the null hypothesis of conditional independence. It then employs an empirical-Bayesian method introduced in [24] (called MPR) to convert p-values of dependencies and independencies into probabilities. This allows ETIO to rank constraints to satisfy, in case there are conflicts. Alternative Bayesian methods have been proposed in [16, 5]; the former is implemented and explored in our experiments. However, the method in [24] has very low computational overhead and is suggested for large problems.

Conflict Resolution Strategy. In order to resolve conflicts, a consistent subset of all input constraints has to be selected. Hyttinen et al. [13] use the method described in [16] to compute the probability of dependence or independence. Their method then tries to identify a subset of constraints that maximizes the product of the weights of all satisfied constraints. Although this has the advantage that it maximizes a well-defined objective function, it comes at a high computational cost. Instead, we chose a greedy approach, following [24]. The constraints are ranked in order of confidence (i.e., the maximum of the probability of dependence or independence), and are considered in that order. If including a constraint leads to an inconsistency it is discarded, and is included in the reasoner otherwise.

Performing Inferences. In this paper we employ answer set programming (ASP) for conflict resolution, as well as for performing inferences. ASP is a declarative programming language, which is especially suited for computationally hard problems. Due to its declarative nature, it is well suited for encoding complicated problems. Answer set solvers usually consist of two phases: in the first phase, the input program is grounded[1], and subsequently solved, often using a SAT solver. As an answer set solver we chose Clingo [11] (version 4.5.4). Clingo supports multi-shot solving [12], which allows incremental grounding and solving. This is especially useful for performing the greedy conflict resolution we use. Without multi-shot solving capabilities, the whole grounding and solving process would have to be repeated for each constraint checked during conflict resolution, increasing the computational cost dramatically. Finally, in order to identify all invariant features based on the input queries, Clingo can be run in *enumeration mode*, which identifies the intersection of all possible solutions. Alternatively, for each desired output feature f, one can query the solver and ask whether f and $\neg f$ can be satisfied; if one is not satisfiable, then its negation holds. This leads to a linear number of additional solver queries for each input query. A better approach is to also keep track of all encountered solutions and avoid repeating solver queries if some literal has already been encountered in some solution. We plan to further investigate this direction in the future.

5. RELATED WORK

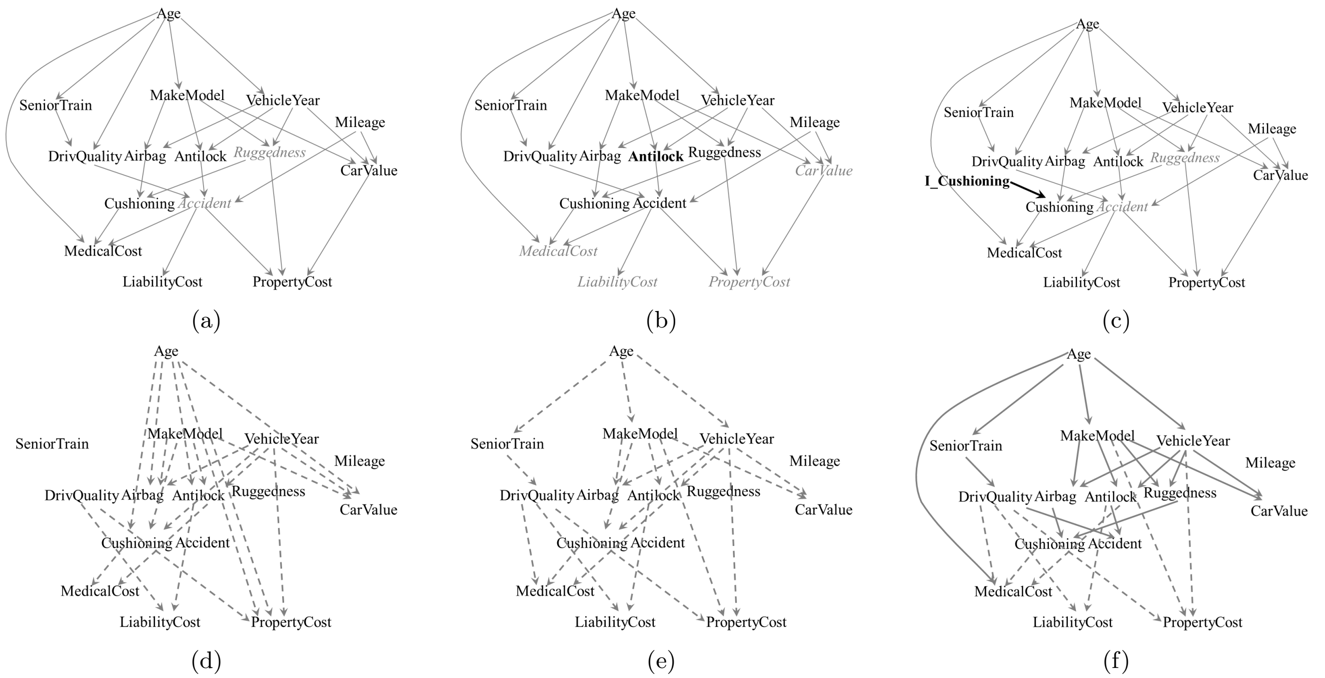
We review related constraint-based methods for causal discovery. Methods such as the PC and FCI algorithms [21, 28] learn an equivalence class of BNs and maximal ancestral graphs (an extension of BNs also admitting latent variables and selection bias; see [24] for its connection to SMCMs) respectively from a single dataset by performing a series of conditional independence tests. Recently, there has been some work on handling more general cases, such as multiple datasets with missing-by-design values, observational and interventional data. A recent overview of such methods can be found in [23]. Apart from methods that try to identify a complete causal model, there also exist some query-based methods that only identify features of the causal model [25, 14, 24]. However, there does not exist any method that is able to handle all possible cases we consider.

The methods closest to our approach are the methods by Hyttinen et al. [14] (HHEJ2013 hereafter), Hyttinen et al. [13] (HEJ2014 hereafter), and COmbINE [24] following the logic-based approach to causal discovery. HHEJ2013 does not perform any conflict resolution and thus cannot be applied in practice since statistical tests almost always contain some conflicts. COmbINE does not handle selection, prior knowledge, or soft interventions, but has introduced several key ideas and is the precursor to ETIO. Unlike ETIO, HEJ2014 is able to also handle cyclic linear models. It uses a Bayesian method [16] to compute the probability of dependence or independence, and then tries to find a causal graph that maximizes the product of probabilities of all satisfied constraints which has a high computational cost. There are two major differences with ETIO: the conflict resolution strategy and the way constraints are encoded in logic.

6. A USE CASE EXAMPLE

In this section we will present a possible scenario for using ETIO for causal discovery for a car insurance company. The scenario introduces and demonstrates in sequence the features of the algorithm, the range of scenarios it can handle, and the type of inferences it makes. In all cases ETIO was executed with an oracle for the (in)dependence conditional tests.

[1] For example, in our case the first-order logic rules would be transformed to propositional logic rules, depending on the number of input variables, datasets and conditioning sets.



(a) (b) (c)

(d) (e) (f)

Figure 1: A part of the insurance network [3]. Grey nodes represent unobserved variables, dashed edges represent possibly indirect causal relations, while direct edges correspond to direct causal relations. Three different cases are considered: **(a)** observational dataset, **(b)** observational dataset selected based on *Antilock*, and **(c)** interventional dataset where a soft intervention has been performed on *Cushioning*. **(d)** Inferences made by only using data from network (a). **(e)** Inferences made by using data from network (a), as well as prior knowledge that *Age* is a root node, *MedicalCost, LiabilityCost* and *PropertyCost* are leaf nodes and are causally affected by *SeniorTrain*. **(f)** Inferences made by using all networks, as well the prior knowledge.

The insurance network: We assume a true causal model in the form of a Bayesian network that generates the data. The true network is of course unknown to the algorithm. We employ a simplified version of the insurance network [3] that was created by a human expert in the field. The ground-truth network is shown in Figure 1 (a). The nodes correspond to attributes of the customers, their cars, as well as well as how much each customer may cost the company, measured by the medical, liability and property cost in case of an accident. Companies that collect related data have the goal of identifying causal factors of the cost nodes, in order to reduce cost. An example of such a factor is whether or not a car has an airbag (Airbag node) which causally affects MedicalCost. Knowing this, the company may either increase the price for customers that do not have an airbag installed, or may try to persuade customers to install an airbag by promising a reduced price.

Latent variables: Let us assume that a company measures most of the nodes in Figure 1 (a) for each client in their data. Unfortunately, most likely some of the latent confounders are bound not to be measured. In this case, whether or not a driver will have an accident while being a customer (the Accident node) cannot be measured as it is not known in advance. Other nodes may be omitted from measuring because they did not seem as important at the time of designing the database. In Figure 1 (a) we will assume that *Ruggedness* and *Accident* are latent shown in a light font. Because these nodes are common causes of two or more measured nodes, they are latent confounders: Bayesian network algorithms will induce the wrong causal relations. In Figure 1, since *Ruggedness* has not been measured and affects both *Cushioning* and *PropertyCost*, a dependence is observed between *Cushioning* and *PropertyCost*. Such a dependence cannot be broken by conditioning on any set of the measured variables. Thus, any Bayesian network algorithm will (asymptotically) identify the edge *Cushioning* → *PropertyCost* or *Cushioning* ← *PropertyCost*. The direction of the edge will be arbitrary. However, as there is no causal relation between them, *the edge, if causally interpreted, would lead to wrong conclusions*: improving the cushioning of the car will not affect the property cost in case of an accident; the reverse edge direction makes even less sense. Therefore, if latent confounders are possible, one should use algorithms for learning models that admit them, such as the FCI [21, 28] algorithm for learning maximal ancestral graphs. *Figure 1 (d) shows the direct and indirect causal relations any sound and complete procedure such as ETIO would identify, denoted with solid and dashed edges respectively.* Indeed, ETIO does not output any causal relation between *Cushioning* and *PropertyCost*. However, ETIO is not able to identify any causal relation as direct (not possibly mediated by any other variable). *We remind the reader that the figures with ETIO results (i.e., subfigures (d) - (f)) only show the causal relations ETIO is able to prove: missing edges do not imply that they are not possible.*

Incorporating prior knowledge: For some variables it is safe to make certain assumptions based on their semantics. It is reasonable to assume that *Age* is a root node, and is not caused nor confounded with any of the measured variables. Furthermore, we can also assume that the cost variables are leaf nodes as they appear after an accident and recording of all other nodes. Finally, we assume that *SeniorTrain* causally influences all costs, possibly indirectly. There may be additional such cases, but for the sake of demonstration we only consider the ones above. Figure 1 (e) shows the inferences made by ETIO when these pieces of causal prior knowledge are included. We can see that, using prior knowledge increases the amount of inferences made, while also refining them. For instance, note that without using the prior knowledge (see Figure 1 (d)) it was inferred that *Age* is a causal factor of *PropertyCost*, but after including prior knowledge it is inferred that *Age* only indirectly affects it through *SeniorTrain*, *MakeModel* and *VehicleYear*.

Including data with selection and overlapping node sets: Assume that there has been a retrospective study on antilock systems and how they affect accident rates; the measured node subset is shown in Figure 1 (b). In this study, the samples were selected such that the proportion of different types of antilock systems is equal: for a number of cars with antilock system, an equal number was selected for inclusion in the study. Thus, sample selection is affected by the *Antilock* variable. Such data are called **case-control** data, a type of selected data. Because of selection bias the dataset has a different distribution than the one contained in the company's database. In addition, notice that this study measures a different (overlapping) set of nodes as the ones available to the insurance company; this is a case of data with missing-by-design values. Because of selection and missing values the pooled data cannot be analyzed using Bayesian network learning methods, even if there were no latent confounders. Due to selection bias in the data, spurious dependencies appear. For example, *MakeModel* would seem to be dependent with *VehicleYear*, yet there is no direct or indirect causal relation between them. FCI could handle selection bias in a single dataset but not in combination with missing-by-design values and prior knowledge. ETIO will analyze the original and the second available datasets together with the following interventional dataset.

Including interventional data: As a final example, consider the case were the company wants to reduce the medical cost of some of its clients. They decided to run a campaign that promotes the usage of cushioning. Since not all clients will respond to the campaign, this intervention is a soft intervention. On the network this is modeled as an additional indicator variable *I_Cushioning* that has an edge into *Cushioning* (see Figure 1 (c)). After the campaign, the company continues to gather additional data which can then be employed in combination with the previous two datasets and available prior knowledge. The results of the analysis on the three datasets by ETIO that include observational, selected, and interventional data measuring different node sets and including prior knowledge are shown in Figure 1 (f). Note that, including additional data further refines the inferences made compared to Figure 1 (e). For example, the previously inferred causal relations between *Age* and *SeniorTrain*, *MakeModel* and *VehicleYear* are now found to be direct (in the context of the measured variables), denoted with solid lines. Furthermore, the algorithm also makes the non-trivial inference that *Age* is a direct cause of *MedicalCost*.

Although not shown in the figures, the algorithm is also able to make several non-trivial inferences between variables that were never measured together in any of the datasets: (a) *Ruggedness* is not directly connected (no direct causal relation and not confounded) with neither of *CarValue*, *MedicalCost* and *LiabilityCost*, (b) *Ruggedness* does not causally influence *CarValue* and *LiabilityCost* and is not causally affected by *LiabilityCost*, and (c) *Accident* and *Carvalue* are not causally related.

7. EXPERIMENTAL EVALUATION

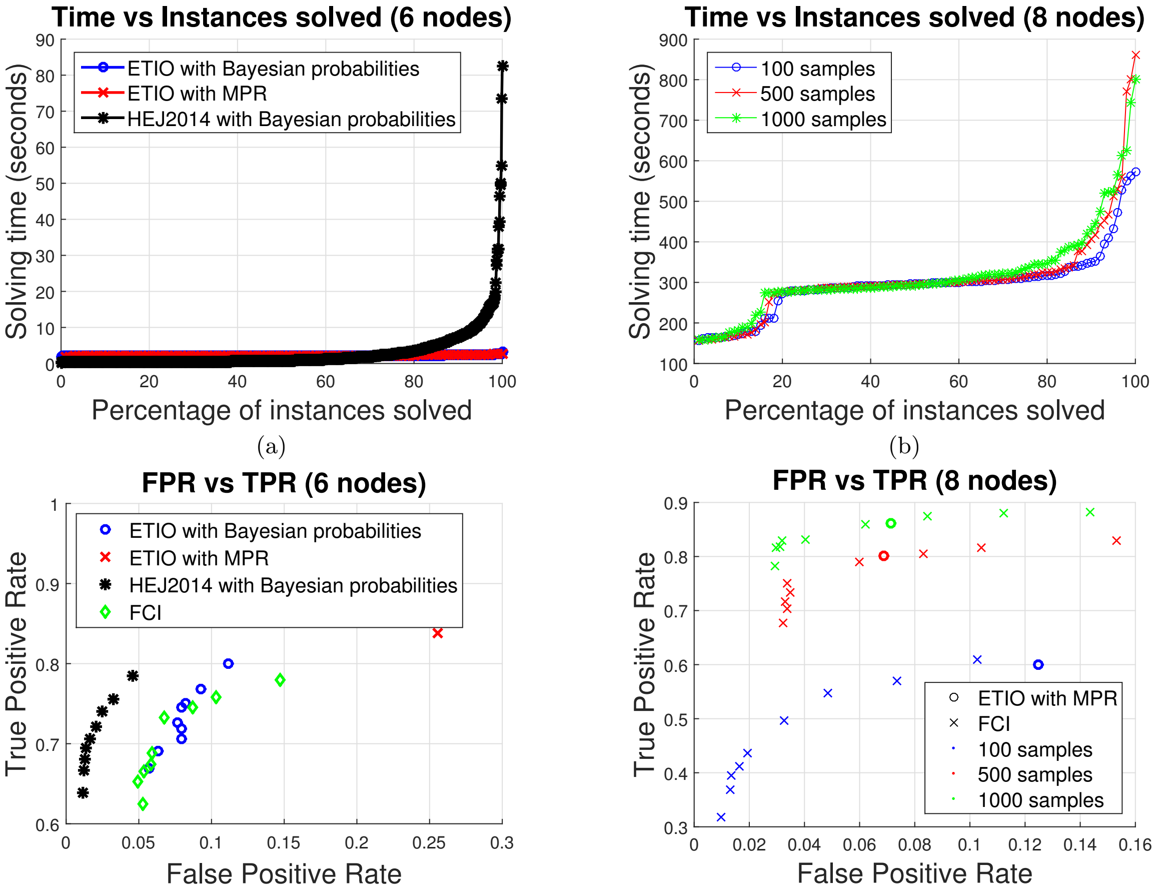
We evaluated ETIO and compared it with HEJ2014 [13] and FCI [28] on simulated data.

Data Generation: We generated acyclic networks with latent confounders. Direct edges and confounders were included in the network independently with probabilities P_d and P_c respectively. We used a linear parameterization and coefficients were sampled uniformly at random from the range $\pm[0.2, 0.8]$, following [13]; that is, each node is a linear function of its parents plus an error term, all having parameters in the aforementioned range. We performed two sets of simulations: in the first simulation we generated 100 networks with 6 nodes, $P_d = 0.2$ and $P_c = 0.1$, and generated one dataset with 500 samples from each network. Similarly, in the second simulation we generated 100 networks with 8 nodes, $P_d = 0.15$ and $P_c = 0.05$, and generated three datasets from each network, with sample sizes of 100, 500 and 1000. In order to compare all algorithms we used only a single observational dataset.

Algorithms: We used the implementation of HEJ2014 by Hyttinen et al. [13]. For ETIO we used both the MPR [24] and Bayesian method [16] used by HEJ2014 to rank constraints. The Bayesian method accepts a prior p, which measures the prior probability of independence. We used the values $\{0.1, 0.2, \ldots, 0.8, 0.9\}$ for p. For both, HEJ2014 and ETIO we ran all possible conditional independence tests. We implemented the complete version of FCI [28] without orientation rules for selection. FCI uses a threshold α on the p-value of an independence test to decide dependence or independence. For α we used the values {0.001, 0.003, 0.005, 0.007, 0.01, 0.03, 0.05, 0.07, 0.1}. As a conditional independence test we used the partial correlation test. As an answer set solver we used Clingo [11] (version 4.5.4) for both HEJ2014 and ETIO. Finally, we set a time limit of 20 minutes for each problem instance.

Comparison: To compare the learning accuracy of all algorithms, we followed Hyttinen et al. [13] and compared all dependencies and independencies represented by the output of HEJ2014, ETIO and FCI. The dependencies and independencies can be read-off the output of HEJ2014 (a SMCM for acyclic graphs with latent variables) and FCI using the Bayes-Ball algorithm. ETIO does not output a causal graph by default, but given the right set of queries a SMCM can be returned, at least in this case where all conditional independence tests are performed. As performance measures we used the true positive rate (TPR) and false positive rate (FPR), using dependencies as positives. Furthermore, we measured the running time of Clingo for HEJ2014 and ETIO only, in order to compare the efficiency of the different encodings and conflict resolution strategies. The running time of FCI was not measured, as FCI is much faster than both logic-based algorithms, especially for larger networks.

Solving Time: Figure 2 (a) shows the solving time for HEJ2014 and ETIO on 100 datasets with 500 samples from 100 networks with 6 nodes. The x-axis shows the percentage



(a)

(b)

(c)

(d)

Figure 2: (a,b) Solving time of HEJ2014 and ETIO. The x-axis shows the percentage of instances solved in time less than the respective value on the y-axis each. ETIO outperforms HEJ2014 on data with 6 variables. HEJ2014 did not run on data with 8 variables for the time limit of 20 minutes. (c,d) TPR and FPR of FCI, HEJ2014 and ETIO. FCI and ETIO outperform HEJ2014. FCI and ETIO perform similarly, but ETIO does not have any hyperparameter to tune, in contrast to FCI.

of solved instances, sorted by time in ascending order, while the y-axis shows the running time in seconds. This means that for a specific value X on the x-axis, X % of the instances have been solved each in time less than or equal to the corresponding value on the y-axis. For example, in Figure 2 (a) around 90 % of all instances have been solved by HEJ2014 in less than 10 seconds each. We see that ETIO takes about the same time (2-3 seconds) for all instances, regardless of the weighting scheme used. HEJ2014 on the other hand, although slightly faster for some instances, takes much longer for the harder instances. *This shows that the encoding and conflict resolution strategy of ETIO are more efficient than the one of HEJ2014.*

Figure 2 (b) only shows the running time of ETIO on 300 datasets, 100 for each sample size (100, 500 and 1000), generated from 100 networks with 8 nodes. HEJ2014 was not run on those data as it took too long to complete. We see that the running times are very similar across sample sizes, although there are a few cases with 500 and 1000 samples that ran slower. This can be attributed to the fact that more samples lead to lower p-values (that is, more dependencies), which may slow down the algorithm. In general, the more independencies are identified the faster ETIO runs.

Quality of Results: Figures 2 (c) and (d) show the TPR and FPR for different parameter values (prior and threshold). Although not shown in the figures, smaller priors for the Bayesian scoring method and larger thresholds for FCI correspond to points with higher TPRs. The results of the first simulation (Figure 2 (c)) show that HEJ2014 outperforms ETIO and FCI, at least in this experimental setting. Specifically, all algorithms give comparable results in terms of TPR, but HEJ2014 consistently has about 5% less FPR than both, ETIO and FCI. This difference in performance between HEJ2014 and ETIO is due to the different conflict resolution strategy used. Recall that ETIO uses a greedy strategy, whereas HEJ2014 identifies the optimal subset of constraints which maximizes the product of weights assigned to each constraint. ETIO with Bayesian probabilities performs similarly to FCI. Using the MPR method, ETIO achieves the highest TPR, but the FPR also increases in contrast to the Bayesian method. In the second simulation (Figure 2 (d)) ETIO perform similarly to FCI in most settings. Note that, the comparison favors FCI, as multiple thresholds were used, while ETIO uses the MPR method which does not have any hyperparameters. In practice it is not known in advance which is the best threshold. For example, common thresholds such as 0.01 and 0.05 (the 5th and 7th largest TPRS in the figures) perform similarly, if not worse, than ETIO with MPR. *Thus, ETIO with MPR performs at least as well as FCI, without using any hyperparameters.*

8. DISCUSSION AND CONCLUSIONS

We propose the ETIO algorithm for causal discovery from multiple heterogeneous datasets, where latent confounders, selection bias, or interventions may have occurred. ETIO can also handle missing-by-design data and incorporate causal prior knowledge. Compared to the state-of-the-art algorithm that can also handle multiple datasets it is computational more efficient. ETIO is an instance of the logic-based approach to integrative causal discovery demonstrating its potential for applications to business data. It also points to interesting new directions for future research to increase the scalability and learning performance of this type of methods.

9. ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their comments. This work was funded by the ERC Consolidator Grant No 617393 CAUSALPATH.

References

[1] N. Angelopoulos and J. Cussens. Bayesian learning of Bayesian networks with informative priors. *Ann. Math. Artif. Intell.*, 54(1-3):53–98, 2008.

[2] E. Bareinboim, J. Tian, and J. Pearl. Recovering from selection bias in causal and statistical inference. AAAI, 2014.

[3] J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Mach. Learn.*, 29(2-3):213–244, 1997.

[4] G. Borboudakis and I. Tsamardinos. Incorporating causal prior knowledge as path-constraints in Bayesian networks and maximal ancestral graphs. ICML, 2012.

[5] T. Claassen and T. Heskes. A Bayesian approach to constraint based causal inference. UAI, 2012.

[6] G. F. Cooper. A Bayesian method for causal modeling and discovery under selection. UAI, 2000.

[7] C. P. de Campos, Z. Zeng, and Q. Ji. Structure learning of Bayesian networks using constraints. ICML, 2009.

[8] M. D'Orazio, M. D. Zio, and M. Scanu. *Statistical Matching: Theory and Practice (Wiley Series in Survey Methodology).* John Wiley & Sons, 2006.

[9] F. Eberhardt. *Causation and Intervention.* PhD thesis, Carnegie Mellon University, USA, 2007.

[10] F. Eberhardt. Direct causes and the trouble with soft interventions. *Erkenntnis*, 79(4):755–777, 2013.

[11] M. Gebser, R. Kaminski, B. Kaufmann, M. Ostrowski, T. Schaub, and M. Schneider. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107–124, 2011.

[12] M. Gebser, R. Kaminski, P. Obermeier, and T. Schaub. Ricochet Robots Reloaded: A Case-study in Multi-shot ASP Solving. IULP, 2015.

[13] A. Hyttinen, F. Eberhardt, and M. Järvisalo. Constraint-based causal discovery: Conflict resolution with answer set programming. UAI, 2014.

[14] A. Hyttinen, P. Hoyer, F. Eberhardt, and M. Järvisalo. Discovering cyclic causal models with latent variables: A general SAT-based procedure. UAI, 2013.

[15] K. B. Korb, L. R. Hope, A. E. Nicholson, and K. Axnick. Varieties of causal intervention. PRICAI, 2004.

[16] D. Margaritis and S. Thrun. A Bayesian multiresolution independence test for continuous variables. UAI, 2001.

[17] R. E. Neapolitan. *Learning Bayesian Networks.* Prentice Hall, 2003.

[18] R. T. O'Donnell, A. E. Nicholson, B. Han, K. B. Korb, M. J. Alam, and L. R. Hope. Incorporating expert elicited structural information in the CaMML causal discovery program. AUJCAI, 2006.

[19] J. Pearl. *Causality, Models, Reasoning, and Inference.* Cambridge University Press, Cambridge, U.K., 2000.

[20] R. D. Shachter. Bayes-ball: Rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams). UAI, 1998.

[21] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search.* MIT Press, Cambridge, MA, 2nd edition, 2000.

[22] J. Tian and J. Pearl. On the identification of causal effects. Technical report, Technical Report R-290L, UCLA Cognitive Systems Laboratory, 2003.

[23] R. E. Tillman and F. Eberhardt. Learning causal structure from multiple datasets with similar variable sets. *Behaviormetrika*, 41(1):41–64, 2014.

[24] S. Triantafillou and I. Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *J. Mach. Learn. Res.*, 16:2329–2348, Oct. 2015.

[25] S. Triantafillou, I. Tsamardinos, and I. Tollis. Learning causal structure from overlapping variable sets. AISTATS, 2010.

[26] I. Tsamardinos, L. Brown, and C. Aliferis. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Mach. Learn.*, 65(1):31–78, 2006.

[27] I. Tsamardinos, S. Triantafillou, and V. Lagani. Towards integrative causal analysis of heterogeneous data sets and studies. *J. Mach. Learn. Res.*, 13, Apr. 2012.

[28] J. Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artif. Intell.*, 172(16-17):1873–1896, 2008.