A Constraint Optimization Approach to Causal Discovery from Subsampled Time Series Data

Antti Hyttinen[1]

HIIT, Department of Computer Science, University of Helsinki

Sergey Plis

Mind Research Network and University of New Mexico

Matti Järvisalo

HIIT, Department of Computer Science, University of Helsinki

Frederick Eberhardt

Humanities and Social Sciences, California Institute of Technology

David Danks

Department of Philosophy, Carnegie Mellon University

Abstract

We consider causal structure estimation from time series data in which measurements are obtained at a coarser timescale than the causal timescale of the underlying system. Previous work has shown that such subsampling can lead to significant errors in search for the system's causal structure if not properly taken into account. In this paper, we first consider the search for system timescale causal structures that correspond to a given measurement timescale structure. We provide a constraint satisfaction procedure whose computational performance is several orders of magnitude better than previous approaches. We then consider finite-sample data as input, and propose the first constraint optimization approach for recovering system timescale causal structure. This algorithm optimally recovers from possible conflicts due to statistical errors. We then apply the method to real-world data, investigate the robustness and scalability of our method, consider further approaches to reduce underdetermination in the output, and perform an extensive comparison between different solvers on this inference problem. Overall, these advances build towards a full un-

Email addresses: antti.hyttinen@helsinki.fi (Antti Hyttinen), s.m.plis@gmail.com (Sergey Plis), matti.jarvisalo@helsinki.fi (Matti Järvisalo), fde@caltech.edu (Frederick Eberhardt), ddanks@cmu.edu (David Danks)

[1] Corresponding author

derstanding of non-parametric estimation of system timescale causal structures from subsampled time series data.

Keywords: causality, causal discovery, graphical models, time series, constraint satisfaction, constraint optimization.

1. Introduction

Time-series data has long constituted the basis for causal modeling in many fields of science [12, 15, 22]. These data often provide very precise measurements at regular time points, but the underlying causal interactions that give rise to those measurements can occur at a much faster timescale than the measurement frequency. Time order information can simplify causal analysis since it can provide directionality, but time series data that undersamples the generating process can be misleading about the true causal connections [7, 19]. For example, Figure 1a shows the causal structure of a process unrolled over discrete time steps, and Figure 1c shows the corresponding structure of the same process, obtained by marginalizing every second time step. If we do not take into account the possibility of subsampling, then we might conclude that optimal control of V_2 requires interventions on both V_1 and V_3, when the influence of V_3 on V_2 is, in fact, completely mediated by V_1 (and so intervening only on V_1 suffices).

Standard methods for estimating causal structure from time series either focus exclusively on estimating a transition model at the measurement timescale (e.g., Granger causality [12, 13]) or combine a model of measurement timescale transitions with so-called "instantaneous" or "contemporaneous" causal relations that aim to capture interactions that are faster than the measurement process (e.g., SVAR [22, 15, 18]), though only very specific types of interactions can be captured with these latter models. In contrast, we follow Plis et al. [30, 31] and Gong et al. [11], and explore the possibility of identifying (features of) the causal process at the true timescale from data that subsample this process.

In this paper, we provide an exact inference algorithm based on using a general-purpose Boolean constraint solver [4, 10], and demonstrate that it is orders of magnitudes faster than the current state-of-the-art method by Plis et al. [31]. At the same time, our approach is much simpler and, as we show, it allows inference in more general settings. We then develop the approach to integrate possibly conflicting constraints obtained from the data. In addition to an application of the method to the real-world data, we investigate the robustness and scalability of our method, consider further approaches to reduce underdetermination in the output and perform an extensive comparison between different solvers on this inference problem. Moreover, unlike the method by Gong et al. [11], our approach does not depend on a particular parameterization of the underlying model and scales to a more reasonable number of variables.

This article considerably extends a preliminary version presented at International Conference on Probabilistic Graphical Models 2016 (PGM 2016) [17].

Most noticeably, Sections 6–9 of this article provide entirely new contents, including a real-world case study (Section 6), an evaluation of the impact of the choice of constraint satisfaction and optimization solvers on the efficiency of the approach (Section 7), and a discussion on learning from mixed frequency data (Section 8). Furthermore, new simulations on accuracy and robustness (Section 5, Figures 5-7) are now included.

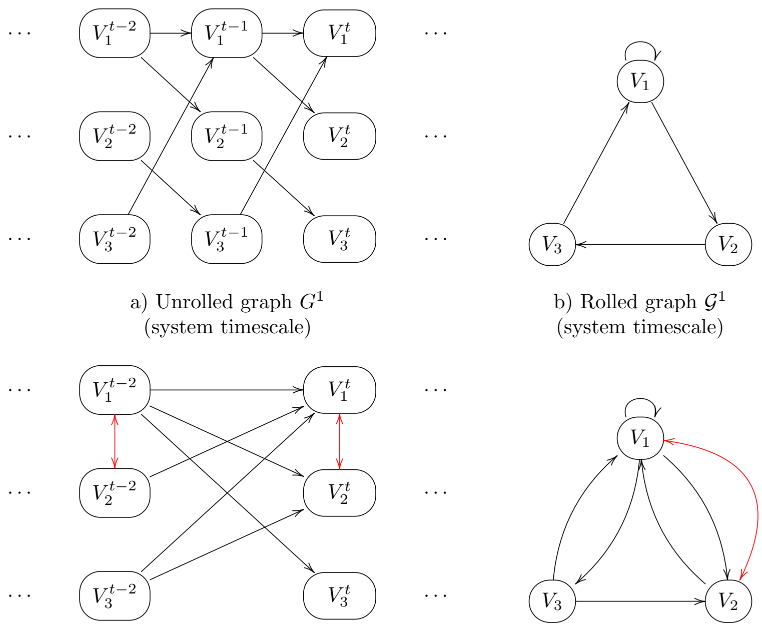
2. Representation

We assume that the system of interest relates a set of variables $\mathbf{V}^t = \{V_1^t, \ldots, V_n^t\}$ defined at discrete time points $t \in \mathbb{Z}$ with continuous ($\in \mathbb{R}^n$) or discrete ($\in \mathbb{Z}^n$) values [9]. We distinguish the representation of the true causal process at the *system timescale* from the time series data that are obtained at the *measurement timescale*. Following Plis et al. [31], we assume that the true between-variable causal interactions at the system timescale constitute a first-order Markov process; that is, that the independence $\mathbf{V}^t \perp\!\!\!\perp \mathbf{V}^{t-k} | \mathbf{V}^{t-1}$ holds for all $k > 1$. The parametric models for these causal structures are structural vector autoregressive (SVAR) processes or dynamic (discrete/continuous variable) Bayes nets. Since the system timescale can be arbitrarily fast (and causal influences take time), we assume that there is no "contemporaneous" causation of the form $V_i^t \rightarrow V_j^t$ [14]. We also assume that $\mathbf{V}^{t-1}$ contains all common causes of variables in $\mathbf{V}^t$. These assumptions jointly express the widely used causal sufficiency assumption (see [35]) in the time series setting.

The system timescale causal structure can thus be represented by a causal graph G^1 (as in a dynamic Bayes net) with only $V_i^{t-1} \rightarrow V_j^t$ edges, where $i = j$ is permitted (see Figure 1a for an example). Since the causal process is time-invariant, the edges repeat through t. In accordance with Plis et al. [31], for any G^1 we use a simpler, rolled graph representation, denoted by $\mathcal{G}^1$, where $V_i \rightarrow V_j \in \mathcal{G}^1$ iff $V_i^{t-1} \rightarrow V_j^t \in G^1$. That is, the rolled graph represents time only implicitly in the edges, rather than through variable duplication. Figure 1b shows the rolled graph representation $\mathcal{G}^1$ of G^1 in Figure 1a.

Time series data are obtained from the above process at the *measurement timescale*, defined by some (possibly unknown) integral sampling rate u. The measured time series sample $\mathbf{V}^t$ is at times $t, t-u, t-2u, \ldots$; we are interested in the case of $u > 1$, i.e., the case of subsampled data. A different route to subsampling would use continuous-time models as the underlying system timescale structure. However, some series (e.g., transactions such as salary payments) are inherently discrete-time processes [11], and many continuous-time systems can be approximated arbitrarily closely as discrete-time processes. Thus, we focus here on discrete-time causal structures as a justifiable, and yet simple, basis for our non-parametric inference procedure.

The structure of this subsampled time series can be obtained (leaving aside sampling variation) from G^1 by marginalizing the intermediate time steps. Figure 1c shows the measurement timescale structure G^2 corresponding to subsampling rate $u = 2$ for the system timescale causal structure in Figure 1a.



a) Unrolled graph G^1
(system timescale)

b) Rolled graph $\mathcal{G}^1$
(system timescale)

c) Unrolled graph G^2
(measurement timescale)

d) Rolled graph $\mathcal{G}^2$
(measurement timescale)

Figure 1: Example graphs with subsampling rate $u = 2$.

Each directed edge in G^2 corresponds to a directed path of length 2 in G_1. For arbitrary u, the formal relationship between G^u and G^1 edges is

$$V_i^{t-u} \to V_j^t \in G^u \;\Leftrightarrow\; V_i^{t-u} \rightsquigarrow V_j^t \in G^1, \text{ where } \rightsquigarrow \text{ denotes a directed path.}^2$$

Subsampling a time series additionally induces "direct" dependencies between variables in the same time step [37]. The bi-directed arrow $V_1^t \leftrightarrow V_2^t$ in Figure 1c is an example: V_1^{t-1} is an unobserved (in the data) common cause of V_1^t and V_2^t in G^1 (see Figure 1a). Formally, the system timescale structure G^1 induces bi-directed edges in the measurement timescale G^u for $i \neq j$ as follows:

$$V_i^t \leftrightarrow V_j^t \in G^u \quad \Leftrightarrow \quad \exists(V_i^t \leftsquigarrow V_c^{t-k} \rightsquigarrow V_j^t) \in G^1, k < u.$$

Just as $\mathcal{G}^1$ represents the rolled version of G^1, $\mathcal{G}^u$ represents the rolled version of G^u: $V_i \to V_j \in \mathcal{G}^u$ iff $V_i^{t-u} \to V_j^t \in G^u$ and $V_i \leftrightarrow V_j \in \mathcal{G}^u$ iff $V_i^t \leftrightarrow V_j^t \in G^u$.

The relationship between $\mathcal{G}^1$ and $\mathcal{G}^u$—that is, the impact of subsampling—

[2] We assume a type of faithfulness assumption (see [35]), such that influences along (multiple) paths between nodes do not exactly cancel in G^u.

can be concisely represented using only the rolled graphs:

$$V_i \to V_j \in \mathcal{G}^u \quad \Leftrightarrow \quad V_i \overset{u}{\rightsquigarrow} V_j \in \mathcal{G}^1 \tag{1}$$

$$V_i \leftrightarrow V_j \in \mathcal{G}^u \quad \Leftrightarrow \quad \exists (V_i \overset{<u}{\leftsquigarrow} V_c \overset{<u}{\rightsquigarrow} V_j) \in \mathcal{G}^1, i \neq j \tag{2}$$

where $\overset{u}{\rightsquigarrow}$ denotes a path of length u and $\overset{<u}{\rightsquigarrow}$ denotes a path shorter than u (of the same length on each arm of a common cause). Using the rolled graph notation, the logical encodings in Section 3 are considerably simpler.

Danks and Plis [6] demonstrated that, in the infinite sample limit, the causal structure $\mathcal{G}^1$ at the system timescale is in general underdetermined, even when the subsampling rate u is known and small. Consequently, even when ignoring estimation errors, the most we can learn is an equivalence class of causal structures at the system timescale. We define $\mathcal{H}$ to be the estimated version of $\mathcal{G}^u$, a graph over $\mathbf{V}$ obtained or estimated at the measurement timescale (with possibly unknown u). Due to underdetermination, multiple $\langle \mathcal{G}^1, u \rangle$ pairs can imply $\mathcal{H}$, and so search is particularly challenging when u is unknown. At the same time, if $\mathcal{H}$ is estimated from data, it is possible, due to statistical errors, that no $\mathcal{G}^u$ has the same structure as $\mathcal{H}$. With these observations, we are ready to define the computational problems focused on in this work.

Task 1 *Given a measurement timescale structure $\mathcal{H}$ (with possibly unknown u), infer the (equivalence class of) causal structures $\mathcal{G}^1$ consistent with $\mathcal{H}$ (i.e. $\mathcal{G}^u = \mathcal{H}$ by Eqs. 1 and 2).*

We also consider the corresponding problem when the subsampled time series is directly provided as input, rather than $\mathcal{G}^u$.

Task 2 *Given a dataset of measurements of* $\mathbf{V}$ *obtained at the measurement timescale (with possibly unknown u), infer the (equivalence class of) causal structures $\mathcal{G}^1$ (at the system timescale) that are (optimally) consistent with the data.*

Section 3 provides a solution to Task 1, and Section 4 provides a solution to Task 2. Later sections further consider generalizations of these two basic tasks.

3. Finding Consistent System Timescale Structures

We first focus on Task 1. We discuss the computational complexity of the underlying decision problem, and present a practical Boolean constraint satisfaction approach that empirically scales up to significantly larger graphs than previous state-of-the-art algorithms.

3.1. On Computational Complexity

Consider the task of finding even a single $\mathcal{G}^1$ consistent with a given $\mathcal{H}$. A variant of the associated decision problem is related to the NP-complete problem of finding a matrix root.

Theorem 1. *Deciding whether there is a $\mathcal{G}^1$ that is consistent with the directed edges of a given $\mathcal{H}$ is NP-complete for any fixed $u \geq 2$.*

Proof. Membership in NP follows from a guess and check: guess a candidate $\mathcal{G}^1$, and deterministically check whether the length-u paths of $\mathcal{G}^1$ correspond to the edges of $\mathcal{H}$ [31]. For NP-hardness, for any fixed $u \geq 2$, there is a straightforward reduction from the NP-complete problem of determining whether a Boolean B matrix[3] has a uth root [21]: for a given $n \times n$ Boolean matrix B, interpret B as the directed edge relation of $\mathcal{H}$, i.e., $\mathcal{H}$ has the edge (i, j) iff $A^u(i, j) = 1$. It is then easy to see that there is a $\mathcal{G}^1$ that is consistent with the obtained $\mathcal{H}$ iff $B = A^u$ for some binary matrix A (i.e., a uth root of B). □

If u is unknown, then membership in NP can be established in the same way by guessing both a candidate $\mathcal{G}^1$ and a value for u. Theorem 1 ignores the possible bi-directed edges in $\mathcal{H}$ (whose presence/absence is also harder to determine reliably from practical sample sizes; see Section 5). Knowledge of the presences and absences of such edges in $\mathcal{H}$ can restrict the set of candidate $\mathcal{G}^1$s. For example, in the special case where $\mathcal{H}$ is known to not contain *any* bi-directed edges, the possible $\mathcal{G}^1$s have a fairly simple structure: in any $\mathcal{G}^1$ that is consistent with $\mathcal{H}$, every node has at most one successor.[4] Whether this knowledge can be used to prove a more fine-grained complexity result for special cases is an open question.

3.2. A SAT-Based Approach

Recently, the first exact search algorithm for finding the $\mathcal{G}^1$s that are consistent with a given $\mathcal{H}$ for a known u was presented by Plis et al. [31]; it represents the current state-of-the-art. Their approach implements a specialized depth-first search procedure for the problem, with domain-specific polynomial time search-space pruning techniques. As an alternative, we present here a Boolean satisfiability based approach. First, we represent the problem exactly using a rule-based constraint satisfaction formalism. Then, for a given input $\mathcal{H}$, we employ an off-the-shelf Boolean constraint satisfaction solver for finding a $\mathcal{G}^1$ that is guaranteed to be consistent with $\mathcal{H}$ (if such $\mathcal{G}^1$ exists). Our approach is not only simpler than the approach of Plis et al. [31], but as we will show, it also significantly improves the current state-of-the-art in runtime efficiency and scalability.

We present our approach using answer set programming (ASP) as the constraint satisfaction formalism[5] [28, 33, 10]. It offers an expressive declarative modeling language, in terms of first-order logical rules, for various types of NP-hard search and optimization problems. To solve a problem via ASP, one first

[3] Multiplication of two values in $\{0, 1\}$ is defined as the logical-or, or equivalently, the maximum operator.

[4] To see this, assume X has two successors, Y and Z, s.t. $Y \neq Z$ in $\mathcal{G}^1$. Then $\mathcal{G}^u$ will contain a bi-directed edge $Y \leftrightarrow Z$ for all $u \geq 2$, which contradicts the assumption that $\mathcal{H}$ has no bi-directed edges.

[5] Note the comparison to other solvers using the propositional SAT formalism in Section 7.

needs to develop an ASP program (in terms of ASP rules/constraints) that models the problem at hand; that is, the declarative rules implicitly represent the set of solutions to the problem in a precise fashion. Then one or multiple (optimal, in case of optimization problems) solutions to the original problem can be obtained by invoking an off-the-shelf ASP solver, such as the state-of-the-art `Clingo` system [10] used in this work. The search algorithms implemented in the `Clingo` system are extensions of state-of-the-art Boolean satisfiability and optimization techniques which can today outperform even specialized domain-specific algorithms, as we show here.

We proceed by describing a simple ASP encoding of the problem of finding a $\mathcal{G}^1$ that is consistent with a given $\mathcal{H}$. The input—the measurement timescale structure $\mathcal{H}$—is represented as follows. The input predicate `node/1` represents the nodes of $\mathcal{H}$ (and all graphs), indexed by $1 \dots n$. The presence of a directed edge $X \to Y$ between nodes X and Y is represented using the predicate `edgeh/2` as `edgeh(X,Y)`. Similarly, the fact that an edge $X \to Y$ is not present is represented using the predicate `no_edgeh/2` as `no_edgeh(X,Y)`. The presence of a bidirected edge $X \leftrightarrow Y$ between nodes X and Y is represented using the predicate `confh/2` as `confh(X,Y)` $(X < Y)$, and the fact that an edge $X \leftrightarrow Y$ is not present is represented using the predicate `no_confh/2` as `no_confh(X,Y)`.

If u is known, then it can be passed as input using `u(U)`; alternatively, it can be defined as a single value in a given range (here set to $1, \dots, 5$ as an example):

```
urange(1..5). % Define a range of u:s

1 { u(U): urange(U) } 1. % u(U) is true for only one U in the range
```

Solution $\mathcal{G}^1$s are represented via the predicate `edge1/2`, where `edge1(X,Y)` is *true* iff $\mathcal{G}^1$ contains the edge $X \to Y$. In ASP, the set of candidate solutions (i.e., the set of all directed graphs over n nodes) over which the search for solutions is performed, is declared via the so-called *choice construct* within the following rule, stating that candidate solutions may contain directed edges between any pair of nodes. If we have prior knowledge about edges that must (or must not) be present in $\mathcal{G}^1$, then that content can straightforwardly be encoded here.

```
{ edge1(X,Y) } :- node(X), node(Y).
```

The implied measurement timescale structure $\mathcal{G}^u$ for a candidate solution $\mathcal{G}^1$ is represented using the predicates `edgeu(X,Y)` and `confu(X,Y)`, which are derived in the following way. First, we declare the mapping from a given $\mathcal{G}^1$ to the corresponding $\mathcal{G}^u$ by declaring the exact length-L paths in a non-deterministically chosen candidate solution $\mathcal{G}^1$. For this, we declare rules that compute the length-L paths inductively for all $L \leq U$, using the predicate `path(X,Y,L)` to represent that there is a length-L path from X to Y.

```
% Derive all directed paths up to length U
path(X,Y,1) :- edge1(X,Y).
path(X,Y,L) :- path(X,Z,L-1), edge1(Z,Y), L <= U, u(U).
```

Second, to obtain $\mathcal{G}^u$, we encode Equations 1 and 2 with the following rules that form predicates `edgeu/2` and `confu/2` describing the edges $\mathcal{G}^1$ induces on the measurement timescale structure.

```
% Paths of length U, correspond to measurement timescale edges
edgeu(X,Y) :- path(X,Y,L), u(L).

% Paths of equal length (<U) from a single node result in bi-directed edges
confu(X,Y) :- path(Z,X,L), path(Z,Y,L), node(X;Y;Z), X < Y, L < U, u(U).
```

Finally, we declare constraints that require that the $\mathcal{G}^u$ represented by the `edgeu/2` and `confu/2` predicates is consistent with the input $\mathcal{H}$. This is achieved with the following rules, which enforce that the edge relations of $\mathcal{G}^u$ and $\mathcal{H}$ are exactly the same for any solution $\mathcal{G}^1$.

```
:- edgeh(X,Y), not edgeu(X,Y).
:- no_edgeh(X,Y), edgeu(X,Y).
:- confh(X,Y), not confu(X,Y).
:- no_confh(X,Y), confu(X,Y).
```

Our ASP encoding of Task 1 consists of the rules just described. The set of solutions of the encoding correspond exactly to the $\mathcal{G}^1$s consistent with the input $\mathcal{H}$.

3.3. Runtime Comparison

Both our proposed SAT-based approach and the recent specialized search algorithm MSL of Plis et al. [31] are correct and complete, so we focus on differences in efficiency, using the implementation of MSL by the original authors. Our approach allows for searching simultaneously over a range of values of u, but Plis et al. [31] focused on the case $u = 2$; hence, we restrict the comparison to $u = 2$.

We simulated system timescale graphs with varying density and number of nodes (see Section 5 for exact details), and then computed the implied measurement timescale structures for subsampling rate $u = 2$. This structure was given as input to the inference procedures. Note that the input consisted here of graphs for which there always is a $\mathcal{G}^1$, so all instances were satisfiable. The task of the algorithms was to output up to 1000 (system timescale) graphs in the equivalence class. The ASP encoding was solved by `Clingo` using the flag `-n 1000` for the solver to enumerate 1000 solution graphs (or all, in cases where there were fewer than 1000 solutions).

The running times of the MSL algorithm and our approach (SAT) on 10-node input graphs with different edge densities are shown in Figure 2 (left).

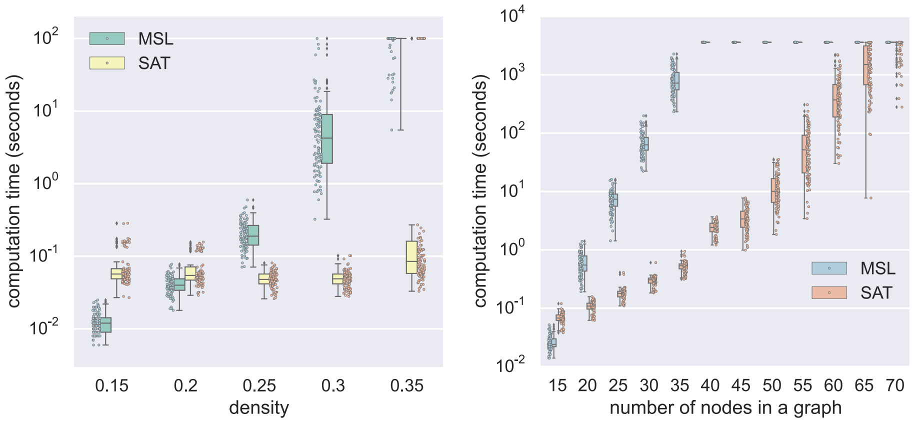


Figure 2: Running times. Left: for 10-node graphs as a function of graph density (100 graphs per density and a timeout of 100 seconds); Right: for 10%-dense graphs as a function of graph size (100 graphs per density and a timeout of 1 hour).

Figure 2 (right) shows the scalability of the two approaches in terms of increasing number of nodes in the input graphs and fixed 10% edge density. Our declarative approach clearly outperforms MSL. 10-node input graphs, regardless of edge density, are essentially trivial for our approach, while the performance of MSL deteriorates noticeably as the density increases. For varying numbers of nodes in 10% density input graphs, our approach scales up to 65 nodes with a one hour time limit; even for 70 nodes, 25 graphs finished in one hour. In contrast, MSL reaches only 35 nodes; our approach uses only a few seconds for those graphs. The scalability of our algorithm allows for investigating the influence of edge density for larger graphs. Figure 3 (left) plots the running times of our approach (when enumerating *all* solutions) for $u = 2$ on 20-node input graphs of varying densities. Note that here the instances are sorted by the running time for each individual density (curve). With a time limit of 1000 seconds we can solve 80% of the instances with 26% density, almost all of the instances with 25% density and all of the instances with 24% density. Thus, the running time is increased for denser graphs: in addition to more constraints, there are also more members in the equivalence classes. Finally, Figure 3 (right) shows the scalability of our approach in the more challenging task of enumerating *all* solutions over the *range* $u = 1, \ldots, 5$ simultaneously. This also demonstrates the generality of our approach: it is not restricted to solving for individual values of u separately.

4. Learning System Timescale Structures from Data

Due to statistical errors in estimating $\mathcal{H}$ and the sparse distribution of implied $\mathcal{G}^u$ in the space of possible undersampled graphs, the estimated $\mathcal{H}$ will often have *no* $\mathcal{G}^1$s with $\mathcal{G}^u = \mathcal{H}$. Given such an $\mathcal{H}$, neither the MSL algorithm nor our approach in the previous section can output a solution, and they simply

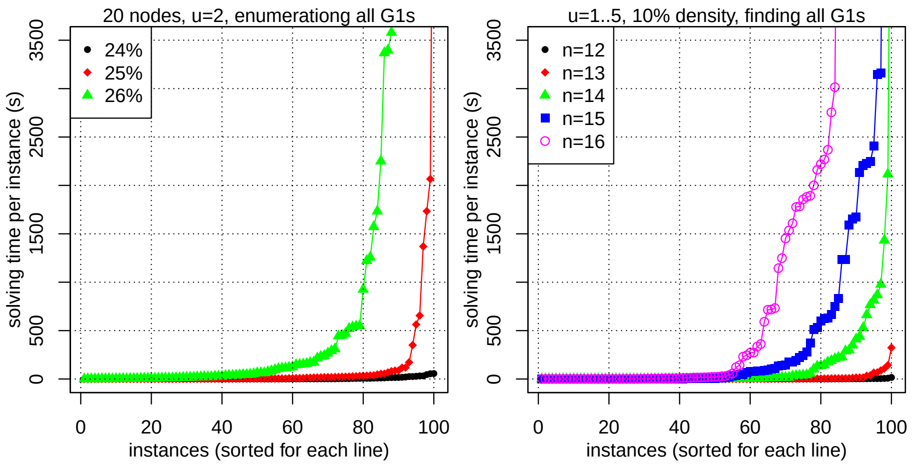


Figure 3: Top: Influence of input graph density on running times of our approach. Bottom: Scalability of our approach when enumerating all solutions over $u = 1, \ldots, 5$.

conclude that no solution $\mathcal{G}^1$ exists for the input $\mathcal{H}$. In terms of our constraint declarations, this is witnessed by conflicts among the constraints and the underlying model space for any possible solution candidate. Given the inevitability of statistical errors, we should not simply conclude that no consistent $\mathcal{G}^1$ exists for such an $\mathcal{H}$. Rather, we should aim to learn $\mathcal{G}^1$s that, in light of the underlying conflicts, are "optimally close" (in some well-defined sense of "optimal") to being consistent with $\mathcal{H}$. We now turn to this more general problem setting, and propose what (to the best of our knowledge) is the first approach to learning, by employing constraint optimization, from undersampled data under conflicts.[6] In fact, we can use the ASP formulation already discussed—with minor modifications—to address this problem.

In this more general setting, the input consists of both the estimated graph $\mathcal{H}$, and also (i) weights $w(e \in \mathcal{H})$ indicating the reliability of edges present in $\mathcal{H}$; and (ii) weights $w(e \notin \mathcal{H})$ indicating the reliability of edges absent in $\mathcal{H}$. Since $\mathcal{G}^u$ is $\mathcal{G}^1$ subsampled by u, the task is to find a $\mathcal{G}^1$ that minimizes the objective function:

$$f(\mathcal{G}^1, u) \quad = \quad \sum_{e \in \mathcal{H}} I[e \notin \mathcal{G}^u] \cdot w(e \in \mathcal{H}) + \sum_{e \notin \mathcal{H}} I[e \in \mathcal{G}^u] \cdot w(e \notin \mathcal{H}),$$

where the indicator function $I(c) = 1$ if the condition c holds, and $I(c) = 0$ otherwise. Thus, edges that differ between the estimated input $\mathcal{H}$ and the $\mathcal{G}^u$ corresponding to the solution $\mathcal{G}^1$ are penalized by the weights representing the reliability of the measurement timescale estimates. In the following, we first

[6] Given conflicts, Plis et al. [31] simply ran the MSL algorithm on graphs close to $\mathcal{H}$, which is not guaranteed to find an optimal solution, nor does it scale computationally.

outline how to generalize the ASP encoding from the preceding section to enable search for optimal $\mathcal{G}^1$ with respect to this objective function. We then describe two alternatives for determining the weights w. In the following section, we present simulation results on the relative performance of the different weighting schemes.

4.1. Learning by Constraint Optimization

To model the objective function for handling conflicts, only simple modifications are needed to our ASP encoding: instead of declaring *hard* constraints that require that the paths induced by $\mathcal{G}^1$ *exactly* correspond to the edges in $\mathcal{H}$, we *soften* these constraints by declaring that the violation of each individual constraint incurs the associated weight as penalty. In the ASP language, this can be expressed by augmenting the input predicates `edgeh(X,Y)` with weights: `edgeh(X,Y,W)` (and similarly for `no_edgeh`, `confh` and `no_confh`). Here the additional argument W represents the weight $w((x \to y) \in \mathcal{H})$ given as input. The following expresses that each conflicting presence of an edge in $\mathcal{H}$ and $\mathcal{G}^u$ is penalized with the associated weight W.

```
:~ edgeh(X,Y,W), not edgeu(X,Y). [W,X,Y,1]
:~ no_edgeh(X,Y,W), edgeu(X,Y). [W,X,Y,1]
:~ confh(X,Y,W), not confu(X,Y). [W,X,Y,2]
:~ no_confh(X,Y,W), confu(X,Y). [W,X,Y,2]
```

This modification provides an ASP encoding for Task 2; that is, the optimal solutions to this ASP encoding correspond exactly to the $\mathcal{G}^1$s that minimize the objective function $f(\mathcal{G}^1, u)$ for given u and input $\mathcal{H}$ with weighted edges.

4.2. Weighting Schemes

We use two different schemes for weighting the presences and absences of edges in $\mathcal{H}$ according to their reliability. To determine the presence/absence of an edge $X \to Y$ in $\mathcal{H}$, we simply test the corresponding independence $X^{t-1} \perp\!\!\!\perp Y^t \mid \mathbf{V}^{t-1} \setminus X^{t-1}$. To determine the presence/absence of an edge $X \leftrightarrow Y$ in $\mathcal{H}$, we run the independence test: $X^t \perp\!\!\!\perp Y^t \mid \mathbf{V}^{t-1}$.

The simplest approach is to use uniform weights for the estimated $\mathcal{H}$:

$$\begin{aligned} w(e \in \mathcal{H}) &= 1 \quad \forall e \in \mathcal{H}, \\ w(e \notin \mathcal{H}) &= 1 \quad \forall e \notin \mathcal{H}. \end{aligned}$$

Uniform edge weights resemble the search on the Hamming cube of $\mathcal{H}$ that Plis et al. [31] used to address the problem of finding $\mathcal{G}^1$s when $\mathcal{H}$ did not correspond to any $\mathcal{G}^u$, though our approach is much superior computationally.

A more intricate approach is to use pseudo-Bayesian weights following [16, 34, 24]. They used Bayesian model selection to obtain reliability weights for independence tests. Instead of a p-value and a binary decision, these types of tests give a measurement of reliability for an independence/dependence statement as a Bayesian probability. We can directly incorporate their approach of

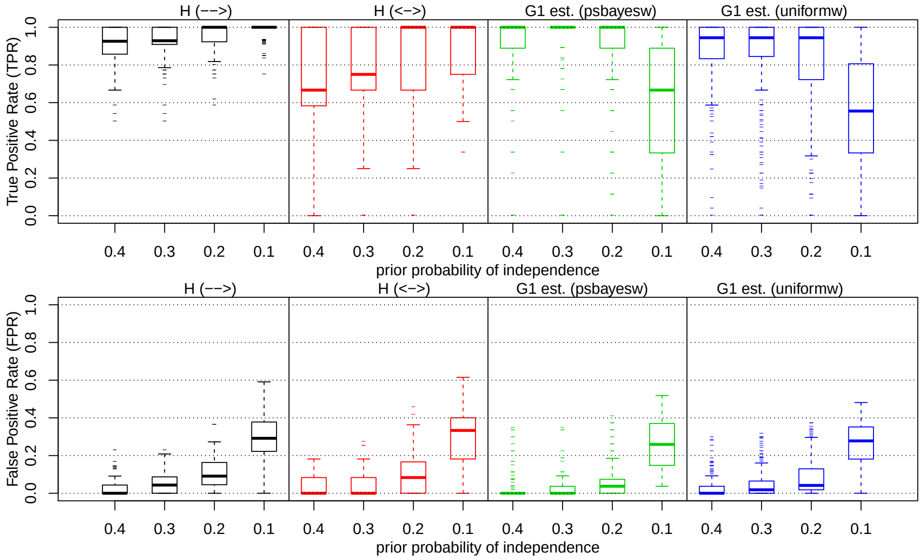


Figure 4: Accuracy of the optimal solutions when $u = 2$ with different weighting schemes and parameters (on x-axis). See text for further details.

using log-probabilities as the reliability weights for the edges. For details, see Section 4.3 of Hyttinen et al. [16]. Again, we only compute weights for the independence tests mentioned above in the estimation of $\mathcal{H}$.

5. Simulations

We use simulations to explore the accuracy and runtime efficiency of our approach in various different settings. For the simulations, system timescale structures $\mathcal{G}^1$ and the associated data generating models were constructed in the following way. To guarantee connectedness of the graphs, we first formed a cycle of all nodes in a random order (following Plis et al. [31]). We then randomly sampled additional directed edges until the required density was obtained. Recall that there are no bidirected edges in $\mathcal{G}^1$. We used Equations 1 and 2 to generate the measurement timescale structure $\mathcal{G}^u$ for a given u. When sample data were required, we used linear Gaussian structural autoregressive processes (order 1) with structure $\mathcal{G}^1$ to generate data at the system timescale, where coefficients were sampled from the two intervals $\pm[0.2, 0.8]$. We then discarded intermediate samples[7] to get the particular subsampling rate.[8]

[7] All sample counts refer to the number of samples after subsampling.

[8] `Clingo` only accepts integer weights; we multiplied weights by 1000 and rounded to the nearest integer.

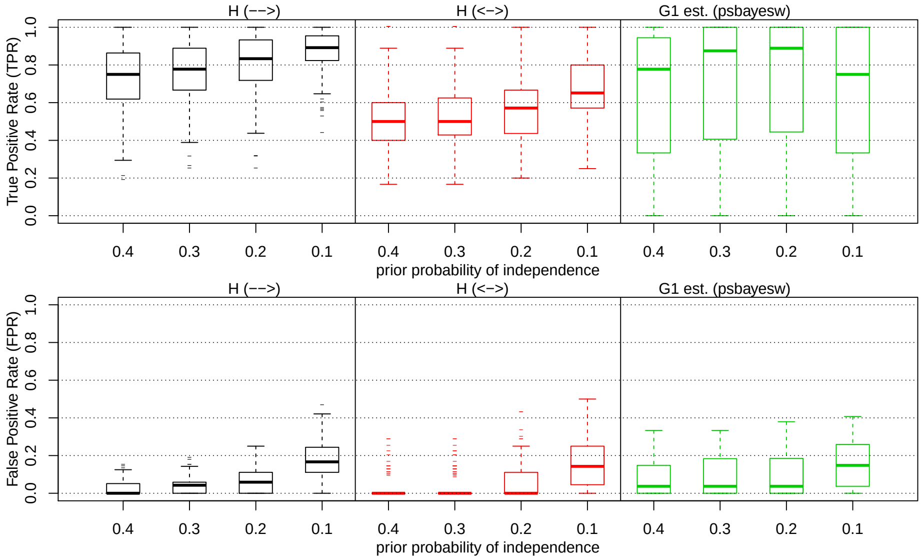


Figure 5: Accuracy of the optimal solutions when $u = 3$. See text for further details.

5.1. Accuracy

Figure 4 shows the accuracy of the different methods in one setting: subsampling rate $u = 2$, network size $n = 6$, average degree 3 (density 25%), sample size $N = 250$, and 200 datasets in total. The positive predictions correspond to presences of edges; when the method returned several solutions with equal cost, we used the mean solution accuracy to measure the output accuracy. The x-axis numbers correspond to the adjustment parameter for the statistical independence tests (prior probability of independence). The two left columns (black and red) show the true positive rate and false positive rate of the $\mathcal{H}$ estimation (compared to the true $\mathcal{G}^2$), for the different types of edges, using different statistical tests. Given 250 samples, we see that the structure of $\mathcal{G}^2$ can be estimated with a good tradeoff of TPR and FPR with the middle parameter values, but not perfectly. The presence of directed edges can be estimated more accurately. More importantly, the two rightmost columns in Figure 4 (green and blue) show the accuracy of the $\mathcal{G}^1$ estimation. Both weighting schemes produce good accuracy for the middle parameter values, although there are some outliers. The pseudo-Bayesian weighting scheme still outperforms the uniform weighting scheme, as it produces high TPR with low FPR for a range of threshold parameter values (especially for 0.3).

Figure 5 shows the accuracy when $u = 3$, $n = 6$, average degree 3 (density 25%), $N = 500$, and 200 datasets. The accuracy for edge presences in the measurement timescale graph $\mathcal{H}$ is lower than for $u = 2$, even though we have twice the number of samples (Figure 5, black, red). The problem is that measurement timescale edges here correspond to 3-edge paths, whose causal effects

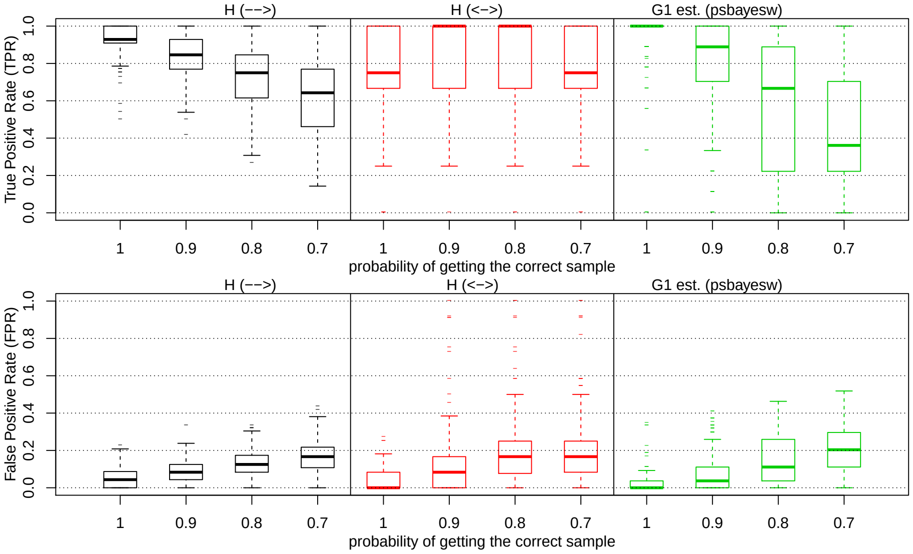


Figure 6: Accuracy of the optimal solutions when $u = 2$ and some samples are obtained at the adjacent timepoints.

will be smaller (on average) than 2-edge paths for a fixed interval of system timescale edge coefficients ($\pm[0.2, 0.8]$), and so are harder to detect. Nevertheless, the constraint optimization procedure achieves a good tradeoff between TPR and FPR for system timescale edges (Figure 5, green). Larger subsampling rates (u) require more samples for accurate $\mathcal{G}^1$ structure discovery, but not several orders of magnitude more data.

5.2. Robustness of the subsampling rate

Figure 6 shows the accuracy of this method when some of the samples are not obtained at the exact time assumed by the measurement timescale. Specifically, the x-axis specifies the probability with which we obtain the correct sample (for the given $u = 2$); otherwise, we take either the sample before or the sample after (synchronously for all variables), splitting the remaining probability. The results with probability 1 equal the result in Figure 4 with prior probability of independence 0.3 and a sample size of $N = 250$. These values were used in all runs in this plot. Unsurprisingly, as the "jitter" in the sampling process increases, the results deteriorate in terms of TPR and FPR. However, at least for the models and subsampling rate of $u = 2$ tested here, the inference is not overly sensitive. When the probability of a correct sample is 0.9, the results are still quite good, alleviating somewhat the dependence on the assumption of an exact subsampling rate. Naturally, there are many further permutations one could explore: jitter could affect variables independently of one another, jitter could be represented by a more complex distribution, we could explore the effect of jitter for different subsampling rates or when the subsampling rate

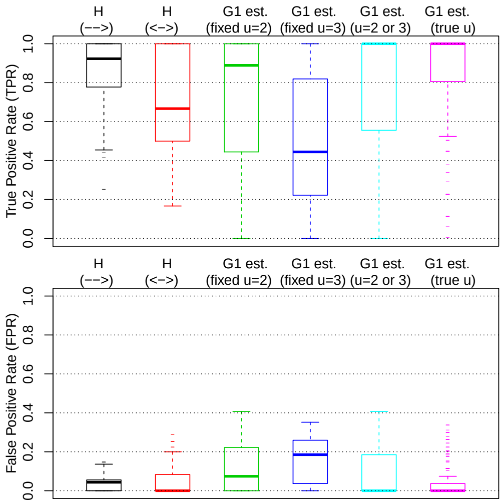


Figure 7: Accuracy when the true u is unknown. Two left boxplots show accuracy of the H estimate as before. The next three boxplots show the accuracy of our approach when, regardless of the true u, u is fixed to 2, or to 3, or left for the procedure decision, respectively. In the rightmost boxplot the true u was given as input.

is unknown. Moreover, jitter could have a persistent, rather than a local effect, in shifting subsequent measures as well. We have here only explored the very simple case mimicking the situation where the measurement device as a whole (i.e. simultaneously for all variables) comes out of synch with the system at random points without consequences for subsequent samples.

Figure 7 further examines the possibility to distinguish between different subsampling rates. We generated 500 samples of data from 200 models (average degree 3) with equal numbers of cases with $u = 2$ or $u = 3$. The two leftmost boxplots show the accuracy of the estimated H, which, given the mixture of $u = 2$ and $u = 3$, is between the accuracy of H obtained in previous simulations. The next two boxplots show the accuracy of the G_1 estimate, when the subsampling rate u for the search procedure is fixed to 2 or 3, respectively, regardless of the true u. As expected, the accuracy is mediocre in this case, since the method assumes the incorrect subsampling rate u in half of the runs. But when the method is left to determine the correct u itself, the accuracy improves again, as shown in the boxplots second form the right (the method was run with $u = 2...3$). In fact, the accuracy comes close to that of the rightmost boxplots, where the correct u was given as input to the procedure. Thus the procedure is often able to recognize the correct u. The longer tails indicate that at times the determination of u is not perfect.

5.3. *Scalability*

Finally, the running times of our approach are shown in Figure 8 with different weighting schemes, network sizes (n), and sample sizes (N). The subsam-

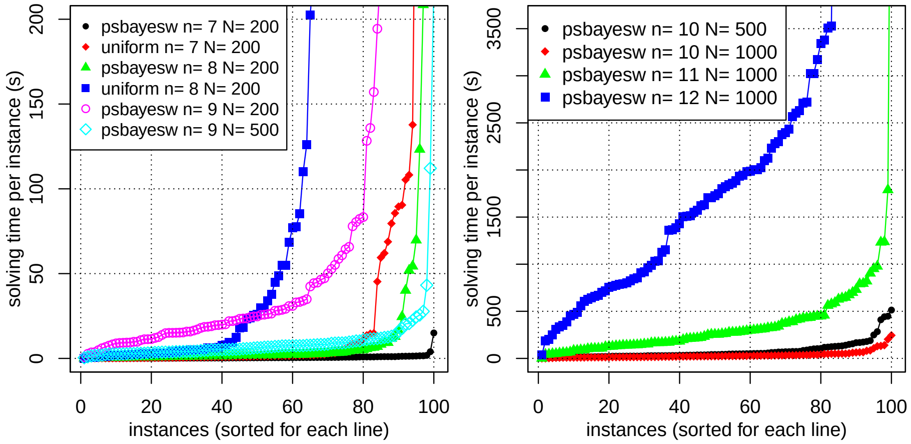


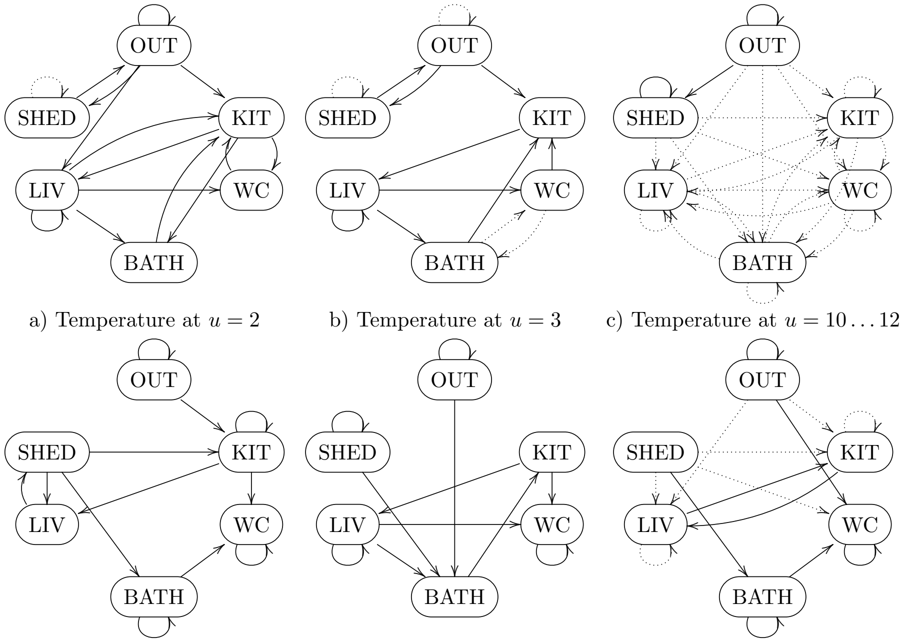
Figure 8: Scalability of our approach under different settings.

pling rate was again fixed to $u = 2$, and average node degree was 3. Figure 8 (left) shows that the pseudo-Bayesian weighting scheme allows for much faster solving: for $n = 7$, it finishes all runs in a few seconds (black line), while the uniform weighting scheme (red line) takes several minutes in the longest runs. Thus, the pseudo-Bayesian weighting scheme provides the best performance in terms of both computational efficiency and accuracy. The sample size has a significant effect on the running times: larger sample sizes take *less* time. Runs for $n = 9, N = 200$ (blue line) take longer than for $n = 9, N = 500$ (Figure 8 left, magenta vs. cyan lines). Intuitively, statistical tests should be more accurate with larger sample sizes, resulting in fewer conflicting constraints. For $N = 1000$, the global optimum is found here for up to 12-node graphs (Figure 8 right), though in a considerable amount of time.

6. Case Study: House data of Peters et al. [29]

In order to demonstrate the applicability to real-world data, we analyzed the house temperature and humidity data of Peters et al. [29]. The data includes 7265 samples of hourly temperature and humidity measurements of six sensors placed in a house (SHED=in the shed, OUT=outside, KIT=kitchen boiler, LIV=living room, WC=wc, BATH=bathroom) in the Black Forest. The house has heating, but the house is not in use for most of the year. This data was also partly analyzed by Gong et al. [11]. The measurements of this system were obtained at coarser intervals than the process of temperature and humidity changes are thought to take place. Since the data includes outside temperature and humidity measurements, the assumption of causal sufficiency at the system timescale seems a good approximation.

We analyzed the temperature and humidity components separately, and ex-



a) Temperature at $u = 2$ b) Temperature at $u = 3$ c) Temperature at $u = 10 \ldots 12$

d) Humidity at $u = 2$ e) Humidity at $u = 3$ f) Humidity at $u = 10 \ldots 12$

Figure 9: Results of the House data analysis. Edges with full lines are found to be present, absent edges are found to be absent, edges with dotted lines may be present or absent.

amined the differences of sequential measurements,[9] as this removed trends from each univariate time series. The temperature measurement timescale graph (obtained at 0.9 prior probability of independence) includes a total of 20 (out of 36) directed edges, and 8 (out of 15) bidirected edges, with varying pseudo-Bayesian weights. The humidity measurement timescale graph had the same total numbers of edges, although not the exact same edges.

As explained earlier, subsampling introduces underdetermination of the system timescale graph. Thus, we determined the presence of individual system timescale edges in the following way [23]. For each edge in $\mathcal{G}^1$, we ran the inference procedure first enforcing its presence and then enforcing its absence.[10] The difference in objective function values for the two outputs—the optimal $\mathcal{G}^1$s that do or do not contain the edge, respectively—indicates the support for the presence (absence) of the edge.

For the estimated $\mathcal{H}$, we computed $\mathcal{G}^1$s edgewise for subsampling rates of $u = 2, 3$. (Since the measurements were hourly, these correspond to time steps of 30 and 20 minutes, respectively.) The two temperature graphs for $u = 2$ and $u = 3$ (Figure 9a,b) differ substantially from one another, as do the two

[9] This may take out some of the influences selfloops would induce.

[10] This can be done by adding a simple clause to the input code "`edge(X,Y).`" to force the presence and "`:-edge1(X,Y).`" to enforce the absence of $X \to Y$.

humidity graphs (Figure 9d,e). These results provide empirical demonstrations of the impact of subsampling, as different choices of u imply different structures. At the same time, timesteps of 20 and 30 minutes arguably do not correspond to realistic time steps for the temperature and humidity changes measured by these data.

We thus considered larger subsampling rates $u = 10 \ldots 12$, which correspond to more realistic time steps of 5-6 minutes. As expected, there is more underdetermination for these u, but the results are also more plausible. Figure 9c suggests that the temperature outside is not directly influenced by the temperature in any of the rooms, but it directly influences the temperature in the shed. The data do not, however, uniquely determine how the outside temperature directly affects the temperatures in the rooms inside the house, nor the system timescale causal dependencies between temperatures in the rooms.

Similarly, the humidity structures for larger u are more plausible. Figure 9f suggests that the humidity level in the WC is driven by both bathroom and outside humidity, which is sensible since the WC is located next to the bathroom and has a window, according to Peters et al. [29]. In contrast, it seems unrealistic that the shed humidity would affect bathroom humidity as suggested by the graph. It is possible that the humidity in the shed provides information on the outside humidity, and so is mistaken for it. (We note that outside and shed measurements are also mixed in some results of Peters et al. [29].) The living room and kitchen boiler humidities seem to depend on each other directly, so the data suggest that the rooms may be adjacent, though that information was not provided by Peters et al. [29].

Overall, the processes controlling the temperature and humidity have differences and similarities. Determining the placement of sensors thus seems to require data from both measurements types.

7. Solver Performance Comparison

Thus far in this article we have considered `Clingo` as the only solver to find solutions to a declarative constraint encoding of the computational problems considered here. This raises the question to what extent the choice of the constraint solver affects the runtime performance of our approach. While the high-level ASP syntax is relatively easy to understand and modify, our approach can also be represented via propositional logic. The benefit of using propositional logic is that various SAT solvers, as well as MaxSAT solvers (as the Boolean optimization generalization of SAT), can be applied directly. In this section we evaluate the impact of the choice of SAT and MaxSAT solvers on the runtime efficiency of our approach.

7.1. Direct Propositional SAT Encoding

A direct propositional SAT encoding for finding a system timescale causal structure $\mathcal{G}^1$ consistent with a measurement timescale graph $\mathcal{H}$ for a known u is presented in Eqs. 3–10.

$$\begin{aligned}
&\vec{h}_{X,Y} & & & &\forall X,Y \in \mathbf{V} : X \rightarrow Y \in \mathcal{H} & (3)\\
\neg&\vec{h}_{X,Y} & & & &\forall X,Y \in \mathbf{V} : X \rightarrow Y \notin \mathcal{H} & (4)\\
&\overset{\leftrightarrow}{h}_{X,Y} & & & &\forall X,Y \in \mathbf{V} : X < Y, X \leftrightarrow Y \in \mathcal{H} & (5)\\
\neg&\overset{\leftrightarrow}{h}_{X,Y} & & & &\forall X,Y \in \mathbf{V} : X < Y, X \leftrightarrow Y \notin \mathcal{H} & (6)\\
&\vec{h}_{X,Y} &\Leftrightarrow& &\bigvee_{Z \in \mathbf{V}} (p^{u-1}_{X,Z} \wedge p^{1}_{Z,Y}) \quad &\forall X,Y \in \mathbf{V} & (7)\\
&p^{l+1}_{X,Y} &\Leftrightarrow& &\bigvee_{Z \in \mathbf{V}} (p^{l}_{X,Z} \wedge p^{1}_{Z,Y}) \quad &\forall X,Y \in \mathbf{V},\ l \in \{1..u-2\} & (8)\\
&\overset{\leftrightarrow}{h}_{X,Y} &\Leftrightarrow& &\bigvee_{l=1}^{u-1} \overset{\leftrightarrow}{h^{l}}_{X,Y} \quad &\forall X,Y \in \mathbf{V} :\ X < Y & (9)\\
&\overset{\leftrightarrow}{h^{l}}_{X,Y} &\Leftrightarrow& &\bigvee_{Z \in \mathbf{V}} (p^{l}_{Z,X} \wedge p^{l}_{Z,Y}) \quad &\forall X,Y \in \mathbf{V} :\ X < Y,\ l \in \{1..u-1\} & (10)
\end{aligned}$$

Essentially, Eqs. 3–6 enforce the input constraints imposed by $\mathcal{H}$. Following the ASP encoding presented earlier, Eqs. 7–10 encode the mapping from the $\mathcal{G}^1$'s—the edge relation of which is encoded as the length-1-path variables $p^1_{X,Y}$—that are consistent with $\mathcal{H}$.

7.2. Solver Comparison: Finding Consistent System Timescale Structures

The results of a runtime performance comparison between Clingo and two state-of-the-art SAT solvers, Glucose [2] and Lingeling [3], is presented in Figure 10 for $u = 3$, edge density of 10% and the numbers of nodes ranging from 27 (on left) to 30 (on right). Note that the plots give the running times of each of the three solvers sorted individually for each solver. In terms of runtime performance, the SAT solvers Glucose and Lingeling, both working directly on the propositional SAT encoding, exhibit noticeably improved performance over Clingo as the number of nodes is increased (right plot). Thus, in terms of runtime efficiency of our approach, it can be beneficial to apply current and future advances in state-of-the-art SAT solvers directly on the propositional level for improved performance. In these simulations the ASP paradigm does not show any particular computational advantage.

7.3. Solver Comparison: Learning System Timescale Structures from Data

As with the ASP encoding given earlier, the SAT encoding given as Eqs. 3–10 is easily extended to solve the optimization problem underlying the task of learning system timescale structure from undersampled data. In the language of MaxSAT, the only change required is to make the constraints in Eqs. 3–6 soft, and to declare that the cost incurred from not satisfying these individual constraints equals that of $w(e \in \mathcal{H})$ (for Eqs. 3,5) or $w(e \notin \mathcal{H})$ (for Eqs. 4,6) for the

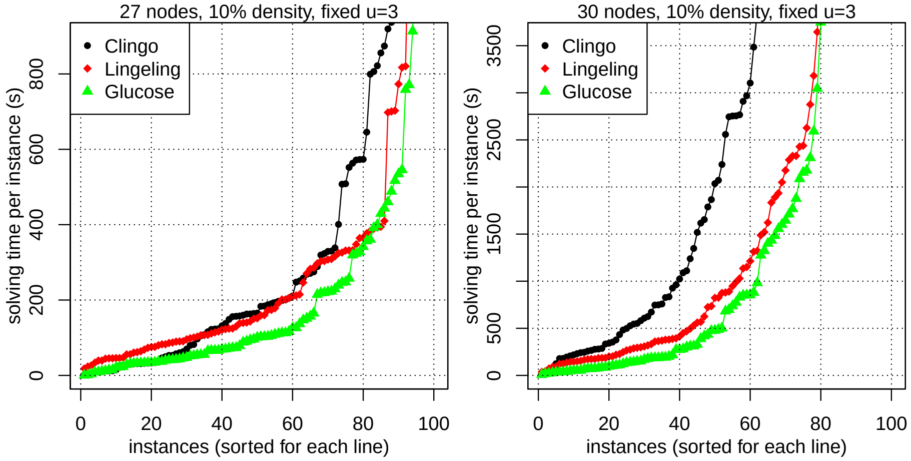


Figure 10: Comparison of running times for different solvers finding a single graph in the equivalence class.

corresponding edge e. This enables a comparison of the runtime performance of Clingo's default branch-and-bound based search for an optimal solution to those of other MaxSAT solvers implementing alternative algorithmic approaches on the direct propositional MaxSAT encoding. Results comparing the performance of Clingo to that of the modern MaxSAT solvers Eva500a [27], LMHS [32], MSCG [26], Open-WBO [25], PrimalDual [5], and QMaxSAT [20], as well as the commercial integer programming (IP) solver CPLEX run on a standard IP translation of MaxSAT [8, 1], are shown in Figure 11. Here we observe that Clingo's branch-and-bound approach is among the best performing solvers (with the considered problem parameters). However, the results also suggest that QMaxSAT, and so-called model-based approaches using a SAT solver to search for an optimal solution over the objective function range with a top-down strategy, can improve on the runtime efficiency of our approach. These results clearly show that the choice of the underlying Boolean optimization solver can indeed have a noticeable influence on the practical efficiency of the approach. There is at least some potential for further improving the runtime performance of our approach by making use of advances in MaxSAT solver technology.

8. Learning from Mixed Frequency Data

In some contexts we may have obtained data from the same system at different subsampling frequencies. Two cases can be distinguished here: First, the subsampled time series may be anchored to the same underlying process such that one may know about the offset between the two.[11] For approaches to this

[11] For example, in the special case where we have two simultaneously measured data sets with $u = 2$ and 1 time step offset could be combined to give a dataset that has no subsampling.

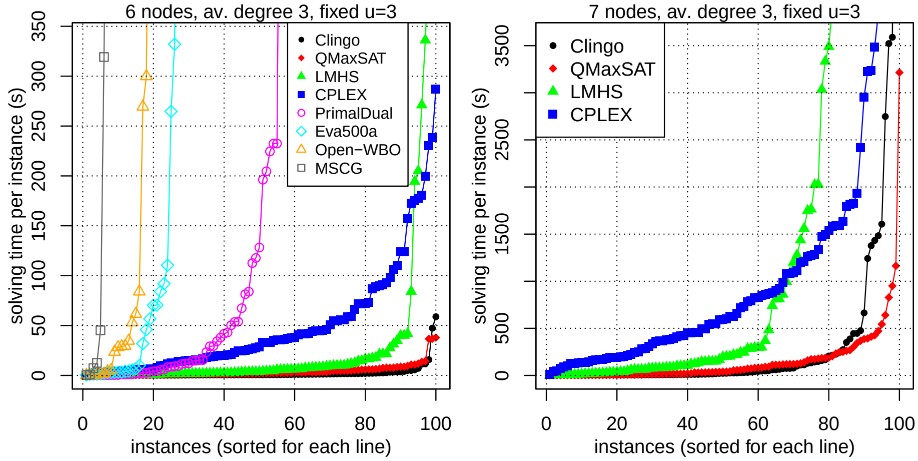


Figure 11: Comparison of running times for different solvers finding the optimal graph.

case see Tank et al. [36], who treat this issue as a missing data problem in a parametric setting. The second case we consider here is one where the subsampled time series are taken at different times and cannot be coordinated to the same instance of an underlying time series. A natural question is how much more can be learned by integrating information from multiple sampling rates. If one sampling rate is an integer multiple of the other, then (provably) nothing additional can be learned. A more interesting situation arises when neither sampling rate is a multiple of the other. For example, suppose the causal system operates at a 1-second timescale. If the system is measured every 2 seconds in one dataset, and every 3 seconds in another dataset, then we have $u_1 = 2/3 \cdot u_2$. More generally, if u_1/u_2 is non-integer, then when (if ever) is the equivalence class of $\mathcal{G}^1$ that satisfies both $\mathcal{H}_1$ & $\mathcal{H}_2$ smaller than the equivalence class for either $\mathcal{H}$ individually? We can start to answer this question using the constraint satisfaction approach of this paper with only minor modifications.

For example, suppose the true system timescale structure is given in Figure 12a. That is, the system includes four independent time series with self loops. Undersampling does not change this graph, so the measurement timescale structures for $u = 2$ and for $u = 3$ will also be the graph in Figure 12a. For this measurement timescale graph, the system timescale structure is not uniquely determined for either $u = 2$ or $u = 3$: for example, the system timescale structure in Figure 12b produces Figure 12a with $u = 2$, and Figure 12c produces Figure 12a with $u = 3$. In fact, *any* system timescale edge can be present or absent given either of the measurement timescale graphs alone.[12] However, if this measurement timescale graph is found at *both* $u = 2$ and $u = 3$, then the system timescale structure can be uniquely determined: Figure 12b produces

[12] The node labels in Figure 12b and c can be permuted.

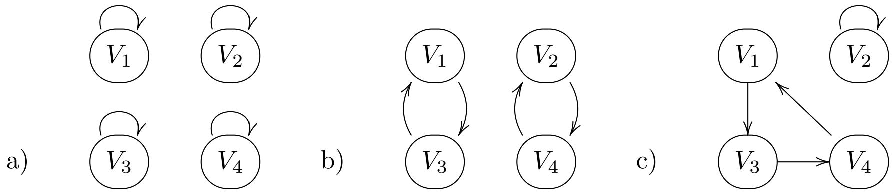


Figure 12: Example graphs. See text for details.

a different measurement timescale graph for $u = 3$ and Figure 12c produces a different measurement timescale graph for $u = 2$. And of course, the same observations hold if the us are multiplied by a constant (e.g., if $u = 4$ and $u = 6$).

To examine the prevalence of this phenomenon, we exhaustively considered all $65536 (= 2^{4\cdot 4})$ different 4-variable $\mathcal{G}^1$s, and compared the number of equivalence classes given input at a single subsampling rate, versus given inputs at two subsampling rates. A greater number of equivalence classes means a higher chance that a random graph will be uniquely identifiable, and so the number of equivalence classes is an approximate (inverse) measure of the extent of underdetermination.

For input at a single undersampling rate, we have $u = 2 \Rightarrow 24265$ equivalence classes; $u = 3 \Rightarrow 7544$ equivalence classes; and $u = 4 \Rightarrow 3964$ equivalence classes. These results with a single undersampled input graph thus replicate the known result that underdetermination is a significant problem, and it rapidly worsens as u increases [30, 31].

If we instead have measurement timescale graphs for both $u = 2, 3$, then we have 26720 equivalence classes, which is only slightly more than the number for $u = 2$ by itself. That is, underdetermination is not substantially reduced if we additionally measure at $u = 3$ when we already have measurements at $u = 2$. Similarly, for $u = 3, 4$ we have 7814 equivalence classes; again, there is a reduction in underdetermination compared to $u = 3$ by itself, but it is quite small. This analysis assumes that all $\mathcal{G}^1$ are equally likely, and it is an open question whether measurements at different undersampling rates would have more impact for certain classes of $\mathcal{G}^1$ (e.g., connected graphs).

9. Discussion

We have assumed that all common causes of measured variables are themselves measured, but this assumption is frequently violated in real-world data. Constraint satisfaction methods have elsewhere been used with success to identify causal relations in the presence of unobserved common causes or latent variables [16, 23]. For time series data, dropping the assumption of causal sufficiency (in the system timescale) generates complications. Even if the system timescale process including latent variables is assumed to be first order Markov, the Markov order of the measurement timescale (naturally without the latent variables) can be arbitrarily larger.[13] That is, variables arbitrarily far in the past can (directly in the measurement timescale) cause variables at the current timestep. We would thus need to both enrich the notation for $\mathcal{G}^u$ to encode the time lags of direct causal effects, and also modify the statistical tests used to estimate these connections.

Moreover, there can be more information contained in the pattern of time lags (i.e., *which* past variables directly cause the present) than is given by the Markov order of the system. As just one example, suppose $\{X^{t-2}, X^{t-4}, \ldots\} \rightarrow Y^t$. The simplest (in terms of number of latents) structure that explains these influences (i) has a latent L through which X influences Y (i.e., $X^{t-2} \rightarrow L^{t-1} \rightarrow Y^t$); and (ii) L is part of a 2-loop with another latent M (i.e., $L^{t-1} \rightarrow M^t$ and $L^t \leftarrow M^{t-1}$). In contrast, if we have $\{X^{t-2}, X^{t-3}, \ldots\} \rightarrow Y^t$, then the simplest structure has only a single latent L through which X influences Y, but where L has a self-loop (i.e., $L^{t-1} \rightarrow L^t$). The pattern of time lags for direct causes—in particular, the absence of certain time lags—thus contains information about the number and causal structure of the latent variables. Estimation of this pattern, however, can be quite complex statistically.

Subsampled time series data is particularly prone to violations of faithfulness. For example, the underlying process unrolled over time may include directed paths over many time steps that do not result in significant statistical dependence in the observed data. In addition, variables observed over subsequent time steps might be almost deterministically related. If $V_1^{t-1} \approx V_1^{t-2}$, then conditioning on V_1^{t-2} may render the statistical dependence through $V_2^t \leftarrow V_1^{t-1} \rightarrow V_3^t$ undetectable from any realistic sample sizes. In the current framework, both of these situations are treated as estimation errors in $\mathcal{H}$. Further modeling of these complications may help to achieve improved accuracy.

10. Conclusion

In this paper, we introduced a constraint optimization based solution for the problem of learning causal timescale structures from subsampled measurement timescale graphs and data. Our approach considerably improves the state-of-art; in the simplest case (subsampling rate $u = 2$), we extended the scalability by several orders of magnitude. Moreover, our method generalizes to handle different or unknown subsampling rates in a computationally efficient manner. Unlike previous methods, our method can operate directly on finite sample input, and we presented approaches that recover, in an optimal way, from conflicts arising from statistical errors. We demonstrated the accuracy, robustness and scalability of the approach through a series of simulations and applied it to real-world time series data. We expect that this considerably simpler approach will allow for the relaxation of additional model space assumptions in the future. In particular, we plan to use this framework to learn the system timescale

[13] This complication is independent of undersampling, and arises even if $u = 1$.

causal structure from subsampled data when latent time series confound our observations.

Acknowledgments

AH was supported by Academy of Finland Centre of Excellence in Computational Inference Research COIN (grant 251170) and Academy of Finland grant 295673. SP was supported by NSF IIS-1318759 & NIH R01EB005846. MJ was supported by COIN (grant 251170) and Academy of Finland grants 276412, 284591; and Research Funds of the University of Helsinki. FE was supported by NSF 1564330. DD was supported by NSF IIS-1318815 & NIH U54HG008540 (from the National Human Genome Research Institute through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative). 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] Ansótegui C, Gabàs J. Solving (weighted) partial MaxSAT with ILP. In: Gomes CP, Sellmann M, editors. Proc. CPAIOR. Springer; volume 7874 of *Lecture Notes in Computer Science*; 2013. p. 403–9.

[2] Audemard G, Simon L. Predicting learnt clauses quality in modern SAT solvers. In: Boutilier C, editor. Proc. IJCAI. 2009. p. 399–404.

[3] Biere A. Splatz, Lingeling, Plingeling, Treengeling, YalSAT entering the SAT Competition 2016. In: Balyo T, Heule M, Järvisalo M, editors. Proc. of SAT Competition 2016 – Solver and Benchmark Descriptions. University of Helsinki; volume B-2016-1 of *Department of Computer Science Series of Publications B*; 2016. p. 44–5.

[4] Biere A, Heule M, van Maaren H, Walsh T, editors. Handbook of Satisfiability; volume 185 of *FAIA*. IOS Press; 2009.

[5] Bjørner N, Narodytska N. Maximum satisfiability using cores and correction sets. In: Yang Q, Wooldridge M, editors. Proc. IJCAI. AAAI Press; 2015. p. 246–52.

[6] Danks D, Plis S. Learning causal structure from undersampled time series. In: NIPS 2013 Workshop on Causality. 2013. .

[7] Dash D, Druzdzel M. Caveats for causal reasoning with equilibrium models. In: Proc. ECSQARU. Springer; volume 2143 of *LNCS*; 2001. p. 192–203.

[8] Davies J, Bacchus F. Exploiting the power of MIP solvers in MAXSAT. In: Järvisalo M, Gelder AV, editors. Proc. SAT. Springer; volume 7962 of *Lecture Notes in Computer Science*; 2013. p. 166–81.

[9] Entner D, Hoyer P. On causal discovery from time series data using FCI. Proc PGM 2010;:121–8.

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

[11] Gong M, Zhang K, Schoelkopf B, Tao D, Geiger P. Discovering temporal causal relations from subsampled data. In: Proc. ICML. JMLR.org; volume 37 of *JMLR W&CP*; 2015. p. 1898–906.

[12] Granger C. Investigating causal relations by econometric models and cross-spectral methods. Econometrica 1969;37(3):424–38.

[13] Granger C. Testing for causality: a personal viewpoint. Journal of Economic Dynamics and Control 1980;2:329–52.

[14] Granger C. Some recent development in a concept of causality. Journal of Econometrics 1988;39(1):199–211.

[15] Hamilton J. Time series analysis. volume 2. Princeton University Press, 1994.

[16] Hyttinen A, Eberhardt F, Järvisalo M. Constraint-based causal discovery: Conflict resolution with answer set programming. In: Proc. UAI. AUAI Press; 2014. p. 340–9.

[17] Hyttinen A, Plis S, Järvisalo M, Eberhardt F, Danks D. Causal discovery from subsampled time series data by constraint optimization. In: Antonucci A, Corani G, de Campos CP, editors. Proc. PGM. JMLR.org; volume 52 of *JMLR Workshop and Conference Proceedings*; 2016. p. 216–27.

[18] Hyvärinen A, Zhang K, Shimizu S, Hoyer P. Estimation of a structural vector autoregression model using non-gaussianity. Journal of Machine Learning Research 2010;11:1709–31.

[19] Iwasaki Y, Simon H. Causality and model abstraction. Artificial Intelligence 1994;67(1):143–94.

[20] Koshimura M, Zhang T, Fujita H, Hasegawa R. Qmaxsat: A partial max-sat solver. Journal of Satisfiability, Boolean Modeling and Computation 2012;8(1/2):95–100.

[21] Kutz M. The complexity of Boolean matrix root computation. Theoretical Computer Science 2004;325(3):373–90.

[22] Lütkepohl H. New introduction to multiple time series analysis. Springer Science & Business Media, 2005.

[23] Magliacane S, Claassen T, Mooij JM. Ancestral causal inference. In: Proc. NIPS. 2016. .

[24] Margaritis D, Bromberg F. Efficient Markov network discovery using particle filters. Computational Intelligence 2009;25(4):367–94.

[25] Martins R, Manquinho VM, Lynce I. Open-WBO: A modular MaxSAT solver. In: Sinz C, Egly U, editors. Proc. SAT. Springer; volume 8561 of *Lecture Notes in Computer Science*; 2014. p. 438–45.

[26] Morgado A, Ignatiev A, Marques-Silva J. MSCG: Robust core-guided MaxSAT solving. Journal on Satisfiability, Boolean Modeling and Computation 2015;9:129–34.

[27] Narodytska N, Bacchus F. Maximum satisfiability using core-guided maxsat resolution. In: Brodley CE, Stone P, editors. Proc. AAAI. AAAI Press; 2014. p. 2717–23.

[28] Niemelä I. Logic programs with stable model semantics as a constraint programming paradigm. Annals of Mathematics and Artificial Intelligence 1999;25(3-4):241–73.

[29] Peters J, Janzing D, Schölkopf B. Causal inference on time series using restricted structural equation models. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, editors. Proc. NIPS 26. Curran Associates, Inc.; 2013. p. 154–62.

[30] Plis S, Danks D, Freeman C, Calhoun V. Rate-agnostic (causal) structure learning. In: Proc. NIPS. Curran Associates, Inc.; 2015. p. 3285–93.

[31] Plis S, Danks D, Yang J. Mesochronal structure learning. In: Proc. UAI. AUAI Press; 2015. p. 702–11.

[32] Saikko P, Berg J, Järvisalo M. LMHS: A SAT-IP hybrid MaxSAT solver. In: Creignou N, Berre DL, editors. Proc. SAT. Springer; volume 9710 of *Lecture Notes in Computer Science*; 2016. p. 539–46.

[33] Simons P, Niemelä I, Soininen T. Extending and implementing the stable model semantics. Artificial Intelligence 2002;138(1-2):181–234.

[34] Sonntag D, Järvisalo M, Peña J, Hyttinen A. Learning optimal chain graphs with answer set programming. In: Proc. UAI. AUAI Press; 2015. p. 822–31.

[35] Spirtes P, Glymour C, Scheines R. Causation, prediction, and search. Springer, 1993.

[36] Tank A, Fox E, Shojaie A. Identifiability of non-Gaussian structural VAR models for subsampled and mixed frequency time series. In: SIGKDD Workshop on Causal Discovery. 2016. .

[37] Wei W. Time series analysis. Addison-Wesley, 1994.