Sparse Approximations of Directed Information Graphs

Christopher J. Quinn
School of Industrial Engineering
Purdue University
West Lafayette, IN, USA
Email: cjquinn@purdue.edu

Ali Pinar
Data Sciences & Cyber Anal. Dept.
Sandia National Laboratories
Livermore, CA, USA
Email: apinar@sandia.gov

Jing Gao and Lu Su
Dept. of Computer Science and Engineering
University at Buffalo
Buffalo, NY, USA
Email: jing@buffalo.edu and lusu@buffalo.edu

***Abstract*—Given a network of agents interacting over time, which few interactions best characterize the dynamics of the whole network? We propose an algorithm that finds the optimal sparse approximation of a network. The user controls the level of sparsity by specifying the total number of edges. The networks are represented using directed information graphs, a graphical model that depicts causal influences between agents in a network. Goodness of approximation is measured with Kullback-Leibler divergence. The algorithm finds the best approximation with no assumptions on the topology or the class of the joint distribution.**

I. Introduction

Networks have become an increasingly integral part of life. Humans participate in networks, including social, communication, and financial networks. Humans are comprised of networks, including neuronal, gene regulatory, and metabolic. A major challenge to understand how these networks function and how to control them is to identify the connections, specifically causal influences, in the networks.

Two major approaches to determine and measure the strength of agent interactions are experimentation and using observed time-series. Experimentation directly determines influences through manipulating the states of certain nodes. However, experimentation can be costly or impractical, such as with the human brain or financial networks. An alternative is to use observed time-series to infer the connections.

Under certain conditions, such as no feedback, or for certain classes of distributions, causal influences can be learned from i.i.d. data [1], [2], [3]. Some methods for time series use prior assumptions on the interactions [4], hypothesize a particular topology and test that hypothesis with data [5], or assume a specific class of parametric models [6].

In some cases, approximations can be better than the full network. Approximations that are sparse, yet capture many of the network dynamics, could be more tractable to learn and use. Sparse approximations could be more tractable to learn since the search space is reduced.

Sparse approximations could be more useful by only depicting the few, strongest edges. For biological networks, approximations could generate hypotheses for connections, reducing the number of experiments. For adversarial networks, having an approximation with only a few, strong edges can greatly facilitate analysis of which edges to target (such as which road to blockade or communication link to break). Different tasks might require different levels of sparsity; it is important that the user can control the sparsity.

In this work, we propose an algorithm to identify optimal sparse approximations of networks of interacting agents. The method makes no assumptions about the distribution or the topology. Despite the generality, the algorithm is none-the-less significantly more tractable than finding the exact graph. Goodness of approximation is measured with KL divergence. The user controls the sparsity level (and computation time) by specifying the total number of edges. We use graphical models to assess how the network topology relates to the dynamics.

Probabilistic graphical models provide a powerful framework for characterizing statistical relationships. Markov and Bayesian networks, for example, are two widely used classes for depicting correlations in i.i.d. data [7]. For time-series, dynamic Bayesian networks can be used. However, they represent multiple variables from each time-series as separate nodes. Structure learning can be prohibitive for large dynamic Bayesian networks due to the number of potential edges. Instead, we use directed information graphs to model interacting processes [8], [9]. These graphical models represent each time-series as a node, resulting in a more compact depiction. They can be applied to any network, whether the time-series are linear or non-linear, parametric or non-parametric, discrete or continuous valued.

A. Our Contributions

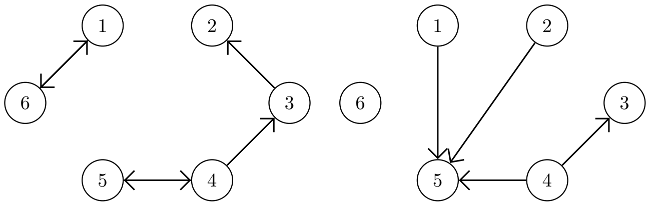
We propose an algorithm to find the optimal approximation with a user-specified number k of edges, with no assumptions on the underlying topology or distribution. The algorithm only requires computing marginals with $k+1$ processes, the theoretical minimum needed for optimality. We also demonstrate the performance of the approximations through simulations.

B. Related Work

There have been several works finding sparse approximations when the processes are multivariate auto-regressive using tree-approximations [10], [11] or Lasso [12], [13].

There have also been algorithms that find sparse approximations of directed information graphs. One finds directed tree approximations [14], connected graphs where each node

978-1-5090-1806-2/16/$31.00 ©2016 IEEE



(a) An in-degree one approximation.

(b) A k-sparse approximation.

Fig. 1. Diagrams of two approximations, one an in-degree one approximation with 6 edges and the other a k-sparse approximation with $k = 4$ edges.

has a single parent. That was generalized in [15], with a method to find connected graphs with more parents. Another finds approximations with user-specified in-degrees [16]. See Figure 1a for an example of an in-degree one approximation.

In contrast, the approximations this work considers are of the form in Figure 1b. The user simply specifies the total number of edges in the graph, letting the algorithm determine the in-degrees. The result is a sparser approximation that better approximates the full network. For networks of m nodes, (uniform) in-degree approximations like Figure 1a can only have m, $2m$, $3m$, etc. edges. In contrast, approximations like Figure 1b can have any number of edges.

The algorithm we propose uses dynamic programming to identify the best set of k edges. The algorithm searches over in-degree counts, choosing how many parents to assign to each node. Several works on Bayesian networks use dynamic programming to learn the exact graph, though they search over partial orders of the variables [17], [18].

II. Background

We first review notation and directed information graphs.

A. Notation and Information-Theoretic Definitions

We now define notation. We use ":=" for denoting.

- For a sequence $a_1, a_2, \ldots$, denote $(a_i, \ldots, a_j)$ as a_i^j and $a^k := a_1^k$. Let $[m] := \{1, \ldots, m\}$.
- We consider m discrete-time random processes over a horizon n with finite-alphabet X. Denote the ith random process at time t by $X_{i,t}$, the ith random process as $\mathbf{X}_i := (X_{i,1}, \ldots, X_{i,n})^\top$, the collection of all m processes as $\underline{\mathbf{X}} := (\mathbf{X}_1, \ldots, \mathbf{X}_m)^\top$, and a subset of L processes indexed by $A \subseteq [m]$ as $\underline{\mathbf{X}}_A := (\mathbf{X}_{A(1)}, \ldots, \mathbf{X}_{A(L)})^\top$.

  Remark 1. *We use finite-alphabet and finite time horizon to simplify the presentation. The results generalize.*
- Conditional and *causally conditioned* distributions [19] of $\mathbf{X}_i$ given $\mathbf{X}_j$ are

$$P_{\mathbf{X}_i|\mathbf{X}_j}(\mathbf{x}_i|\mathbf{x}_j) := \prod_{t=1}^{n} P_{X_{i,t}|X_i^{t-1},X_j^n}(x_{i,t}|x_i^{t-1}, x_j^n) \quad (1)$$

$$P_{\mathbf{X}_i\|\mathbf{X}_j}(\mathbf{x}_i\|\mathbf{x}_j) := \prod_{t=1}^{n} P_{X_{i,t}|X_i^{t-1},X_j^{t-1}}(x_{i,t}|x_i^{t-1}, x_j^{t-1}). \quad (2)$$

Note the similarity between (1) and (2), though in (2) the present and future, $x_{j,t}^n$, are not conditioned on. In [19], the present $x_{j,t}$ was conditioned on. The reason we remove it will be made clear in Remark 3.

- Let $i, j \in [m]$ and $A \subseteq [m]\backslash\{i, j\}$. The causally conditioned directed information [20], [19] is

$$\mathrm{I}(\mathbf{X}_j \to \mathbf{X}_i \| \underline{\mathbf{X}}_A) := \sum_{t=1}^{n} \mathrm{I}(X_j^{t-1}; X_{i,t} | \underline{X}_{A\cup\{i\}}^{t-1}). \quad (3)$$

Remark 2. *While mutual information quantifies statistical correlation (in the colloquial sense of statistical interdependence), directed information quantifies statistical* causation *in the sense of Granger causality [16]. The main principle of Granger causality [6] is that when the past of $\mathbf{X}_j$ helps to predict the future of $\mathbf{X}_i$, even considering (conditioning on) the past of the whole network, then $\mathbf{X}_j$ is said to causally influence $\mathbf{X}_i$. Directed information* (3) *measures the correlation between the past of $\mathbf{X}_j$ and the future of $\mathbf{X}_i$, conditioned on the past of other processes. See [16] for more details as well as experiments on simulated and real-world data.*

Remark 3. *In* (2) *and* (3)*, there is no conditioning on the present $X_{j,t}$. This follows the original definition [20] and is consistent with Granger causality [6]. Later works such as [19] included conditioning on $X_{j,t}$ for the specific setting of communication channels.*

B. Directed Information Graphs

We now review directed information graphs.

Definition II.1. *[8], [9] A* directed information graph *is a probabilistic graphical model where each node represents a process $\mathbf{X}_i$ and an edge $\mathbf{X}_j \to \mathbf{X}_i$ is drawn if*

$$\mathrm{I}(\mathbf{X}_j \to \mathbf{X}_i \| \underline{\mathbf{X}}_{[m]\backslash\{i,j\}}) > 0.$$

Definition II.1 is in terms of individual edges. Under certain conditions, the graph corresponds to a particular factorization of the joint distribution. By the chain rule, the joint distribution $P_{\underline{\mathbf{X}}}$ factorizes over time as $P_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) = \prod_{t=1}^{n} P_{\underline{\mathbf{X}}_t|\underline{\mathbf{x}}^{t-1}}(\underline{\mathbf{x}}_t|\underline{\mathbf{x}}^{t-1})$. If given the full past, $\underline{\mathbf{X}}^{t-1}$, the processes $\{\mathbf{X}_1, \ldots, \mathbf{X}_m\}$ at time t are mutually independent, $P_{\underline{\mathbf{X}}}$ can be further factorized as

$$P_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) = \prod_{t=1}^{n}\prod_{i=1}^{m} P_{X_{i,t}|\underline{\mathbf{X}}^{t-1}}(x_{i,t}|\underline{\mathbf{x}}^{t-1}), \quad (4)$$

and $P_{\underline{\mathbf{X}}}$ is said to be *strictly causal*. Equation (4) can be written using causal conditioning notation (2) as $P_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) = \prod_{i=1}^{m} P_{\mathbf{X}_i\|\underline{\mathbf{X}}_{[m]\backslash\{i\}}}(\mathbf{x}_i \parallel \underline{\mathbf{x}}_{[m]\backslash\{i\}})$. A distribution $P_{\underline{\mathbf{X}}}$ is called *positive* if $P_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) > 0$ for all $\underline{\mathbf{x}} \in \mathsf{X}^{mn}$.

Theorem II.2. *[16] For a joint distribution $P_{\underline{\mathbf{X}}}$, if $P_{\underline{\mathbf{X}}}$ is positive and strictly causal, then the parent sets $\{A(i)\}_{i=1}^{m}$ in the directed information graph are the unique, minimal cardinality parent sets such that $\mathrm{D}(P_{\underline{\mathbf{X}}} \| \prod_{i=1}^{m} P_{\mathbf{X}_i\|\underline{\mathbf{X}}_{A(i)}}) = 0$.*

In searching for the optimal sparse approximation, $P_{\underline{\mathbf{X}}}$ need not be positive and strictly causal [16]. The proposed algorithm will search for parent sets, motivated by Theorem II.2.

III. Optimal k-Sparse Approximations

We now consider identifying the best, sparse approximation for a network $P_{\underline{\mathbf{X}}}$. Our goal is to specify a parent set $\widehat{A}(i)$ for each node i, such that the induced marginal distribution,

$$\widehat{P}_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) := \prod_{i=1}^{m} P_{\mathbf{X}_i \| \underline{\mathbf{X}}_{\widehat{A}(i)}}(\mathbf{x}_i \| \underline{\mathbf{x}}_{\widehat{A}(i)}) \tag{5}$$

is close the the full joint distribution $P_{\underline{\mathbf{X}}}$ and there are k edges total. We seek to minimize the KL divergence $\mathrm{D}(P_{\underline{\mathbf{X}}} \| \widehat{P}_{\underline{\mathbf{X}}})$. The user can select k according to the task and resources at hand. Note that the marginals in (5) are exact, but the parents are approximate (not necessarily a subset of true parents). Let $\mathcal{P}^k$ denote the set of all distributions of the form (5) with k edges total,

$$\mathcal{P}^k := \{\widehat{P}_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) : \sum_{i=1}^{m} |\widehat{A}(i)| = k\}.$$

Our objective is to identify the optimal approximation

$$\widehat{P}^*_{\underline{\mathbf{X}}} := \mathop{\arg\min}_{\widehat{P}_{\underline{\mathbf{X}}} \in \mathcal{P}^k} \mathrm{D}(P_{\underline{\mathbf{X}}} \| \widehat{P}_{\underline{\mathbf{X}}}).$$

Theorem III.1. *[14] For any distribution $P_{\underline{\mathbf{X}}}$ and set of approximations $\mathcal{P}$,*

$$\mathop{\arg\min}_{\widehat{P}_{\underline{\mathbf{X}}} \in \mathcal{P}} \mathrm{D}(P_{\underline{\mathbf{X}}} \parallel \widehat{P}_{\underline{\mathbf{X}}}) = \mathop{\arg\max}_{\widehat{P}_{\underline{\mathbf{X}}} \in \mathcal{P}} \sum_{i=1}^{m} \mathrm{I}(\mathbf{X}_{A(i)} \to \mathbf{X}_i).$$

Remark 4. *In [14], only the case $|A(i)| = 1$ for a positive and strictly causal $P_{\underline{\mathbf{X}}}$ was shown. The result generalizes [16].*

Theorem III.1 shows that to a certain extent, finding good approximations for directed information graphs (a global problem) is the same as finding good parent sets (a local problem). The choice of parent sets is not independent, however, because the total number of parents is restricted.

To find the best graph with only k edges, it is necessary to at least determine the best k parents for each node $\mathbf{X}_i$. This is because for each node $\mathbf{X}_i$, the graph with k edges into $\mathbf{X}_i$ is a feasible solution. To find the optimal set of k parents for each node $\mathbf{X}_i$, it is necessary and sufficient to compute statistics involving $k{+}1$ processes, namely $\{\mathrm{I}(\underline{\mathbf{X}}_S \to \mathbf{X}_i)\}_{\{i\}, S \subseteq [m], |S|=k}$ [16]. Since finding the k-sparse optimal approximation is strictly tougher, it is also necessary to compute statistics involving (at least) $k+1$ processes. The algorithm we develop will establish that is sufficient.

A. Algorithm Overview

The main algorithm, Algorithm 2: OPTkEDGES, uses dynamic programming to allocate the number of parents for each node. It first calls Algorithm 1: FILLINC to fill in a table C of directed information values between candidate parent sets of different sizes and their child nodes.

Algorithm 1 runs as follows. Initially, it creates a $(1 + \min(k, m-1)) \times m \times 2$ array C, with columns corresponding to processes $\mathbf{X}_i$ and rows corresponding to parent set sizes $l : \ 0 \le l \le \min(k, m-1)$. Recall that in a graph of m nodes, the maximum in-degree is $m-1$ though $k \ge m$ is permitted. For each $\mathbf{X}_i$ and $l : 0 \le l \le \min(k, m-1)$, compute

$$S^*_l \leftarrow \mathop{\arg\max}_{S \subseteq [m] \setminus \{i\}, |S|=l} \mathrm{I}(\underline{\mathbf{X}}_S \to \mathbf{X}_i).$$

Then $C[l+1, i, 1] \leftarrow \mathrm{I}(\underline{\mathbf{X}}_{S^*_l} \to \mathbf{X}_i)$, the largest directed information of l parents to $\mathbf{X}_i$, and $C[l+1, i, 2] \leftarrow S^*_l$, the parent set achieving the best value. Break ties arbitrarily. Indexing in C starts at 1, so row 1 corresponds to 0 parents.

Algorithm 1. FILLINC

```
Input: $k$, $m$, $\mathcal{DI}_{\mathrm{BndInd}}$
For $i$ in $m, \dots, 1$
    $C[1, i, 1] \leftarrow 0$
    $C[1, i, 2] \leftarrow \emptyset$
    For $l$ in $1, \dots, \min(k, m-1)$
        $S^*_l \leftarrow \arg\max_{S \subseteq [m] \setminus \{i\}, |S|=l} \mathrm{I}(\underline{\mathbf{X}}_S \to \mathbf{X}_i)$
        $C[l+1, i, 1] \leftarrow \mathrm{I}(\underline{\mathbf{X}}_{S^*_l} \to \mathbf{X}_i)$
        $C[l+1, i, 2] \leftarrow \{(S^*_l, i)\}$
Return $C$
```

To determine table C, the following set of directed information values $\mathcal{DI}_{\mathrm{BndInd}}$ is needed,

$$\begin{aligned} \mathcal{DI}_{\mathrm{BndInd}} := \{\mathrm{I}(\underline{\mathbf{X}}_S \to \mathbf{X}_i) : S \subseteq [m], i \in [m], \\ |S| \le \min(k, m-1)\}. \end{aligned}$$

Lemma III.2. *Each element $C[l{+}1, i, 2]$ filled by Algorithm 1 is the set of indices for an optimal set of l parents for $\mathbf{X}_i$, and $C[l+1, i, 1]$ is the corresponding directed information value.*

Proof. The proof follows immediately by construction. □

B. Finding the Optimal k-Sparse Approximation

Once table C is filled in with the best parent sets, we can use dynamic programming to identify the k-sparse optimal approximation. This procedure is formalized in Algorithm 2. Let $\mathrm{Bestk}(l, i)$ denote the largest total influence, measured as the sum of directed informations from parent sets to children, that can be achieved by picking l edges into any of the last $m - i + 1$ nodes $\{\mathbf{X}_i, \dots, \mathbf{X}_m\}$,

$$\mathrm{Bestk}(l, i) = \max_{\substack{\{\widetilde{A}(j)\}_{j=i}^m \\ s.t.\ \forall j:\ i \le j \le m\ \widetilde{A}(j) \subseteq [m] \setminus \{j\} \\ \sum_{j=i}^m |\widetilde{A}(j)| = l}} \sum_{j=i}^{m} \mathrm{I}(\underline{\mathbf{X}}_{\widetilde{A}(j)} \to \mathbf{X}_j)$$

Lemma III.3. *The function $\mathrm{Bestk}(l, i)$ satisfies the following recursion for all $l : 0 \le l \le k$ and all $i \in [m]$,*

$$\mathrm{Bestk}(l, i) = \max_{l' : 0 \le l' \le l} C[l'{+}1, i, 1] + \mathrm{Bestk}(l - l', i + 1). \tag{6}$$

Proof. The proof is omitted due to space limitations. It uses induction on i and Lemma III.2. □

To solve the recursion efficiently, store intermediate values in an array H. The dimensions are $(k{+}1) \times m \times 2$. $H[l{+}1, i, 1]$ stores $\mathrm{Bestk}(l, i)$, the value (sum of directed informations)

Algorithm 2. OPTkEDGES

```
Input: $k$, $m$, $\mathcal{DI}_{\text{BndInd}}$
 1. $H[\cdot,\cdot,1] \leftarrow 0$, $H[\cdot,\cdot,2] \leftarrow \emptyset$
 2. $C \leftarrow$ FILLIN$C$($k$, $m$, $\mathcal{DI}_{\text{BndInd}}$)
 3. For $l$ in $0,\dots,\min(k,m-1)$
 4.     $H[l+1,m,1] \leftarrow C[l+1,m,1]$
 5.     $H[l+1,m,2] \leftarrow C[l+1,m,2]$
 6. For $i$ in $m-1,\dots,1$
 7.     For $l$ in $0,\dots,\min(k,m-1)$
 8.         $l^* \leftarrow \arg\max_{l':0\le l'\le l} C[l'+1,i,1] + H[l-l'+1,i+1,1]$
 9.         $H[l+1,i,1] \leftarrow C[l^*+1,i,1] + H[l-l^*+1,i+1,1]$
10.         $H[l+1,i,2] \leftarrow C[l^*+1,i,2] \cup H[l-l^*+1,i+1,2]$
11. Return $H[k+1,1,2]$
```

of the parent sets picked. $H[l,i,2]$ records which parent sets achieve that value. The array H can be filled in from column $i=m$ back to $i=1$, and for each column i filling in row one ($l=0$) to $k+1$ ($l=k$).

We now briefly describe the pseudo-code. Since the recursion "ends" at column $i=m$, if l' edges are left over, then $\mathbf{X}_m$ will have l' parents. Lines 3-5 set the final column of H. Lines 6-10 compute values in reverse order of the recursion, with line 10 tracking the current sets of parents picked for $\mathbf{X}_i,\dots,\mathbf{X}_m$ and line 9 tracking the cumulative value. Line 11 returns the parent sets in the optimal k-sparse approximation.

Theorem III.4. *Algorithm 2 identifies the optimal k-sparse approximation $\widehat{P}^*_{\underline{\mathbf{X}}} \in \mathcal{P}^k$.*

Proof. By Theorem III.1, the set of k edges which maximizes the sum of directed informations from parent sets to children corresponds to the optimal approximation. By Lemma III.3, the recursion (6) finds the k edges that maximize the sum, according to C. By construction, H stores intermediate values of the recursion (6). By Lemma III.2, C is filled correctly. □

By filling the first column in H, for $\mathbf{X}_1$, Algorithm 2 finds not only the optimal k-sparse approximation, but also the optimal l-sparse approximation for all $0 \le l \le k$. In Algorithm 2, line 11, $H[l+1,1,2]$ could be called instead.

C. Asymptotic Efficiency

To analyze algorithmic complexity, we need to bound the computation time needed for directed information. Suppose the directed information $\mathrm{I}(\underline{\mathbf{X}}_B \to \mathbf{Y} \| \underline{\mathbf{X}}_{B'})$ will be computed following the definition (3). If the joint distribution is Markov order r, the alphabet size is $|\mathsf{X}|$, and the number of processes in the term is $|B|+|B'|+1=k+1$, the run-time is [15]

$$\mathcal{O}(n|\mathsf{X}|^{(k+1)r+1}). \tag{7}$$

Alternative methods could have different run-time bounds.

The complexity of finding the best k-sparse approximation is dominated by Algorithm 1, computing directed information values. Algorithm 1 computes all elements of $\mathcal{DI}_{\text{BndInd}}$, running in $\mathcal{O}(m(\sum_{l=1}^{k}\binom{m-1}{l}n|\mathsf{X}|^{(l+1)r+1}))$ using (7), when $k \le m-1$. When the sparsity level is much smaller than the number of nodes, $m \gg k$, this simplifies to $\mathcal{O}(m^{k+1}n|\mathsf{X}|^{kr}))$ and $\mathcal{O}(m^{k+1}n)$ for fixed sparsity k and alphabet size $|\mathsf{X}|$. If $k \ge m$, the run-time is fixed as $\mathcal{O}(m(\sum_{l=1}^{m-1}\binom{m-1}{l}n|\mathsf{X}|^{(l+1)r+1})) = \mathcal{O}(mn(1+|\mathsf{X}|^r)^{m-1})$.

We now determine the asymptotic efficiency of Algorithm 2, OPTkEDGES, for $k \le m-1$. It uses C to fill table H. H has $k+1$ rows and m columns. Each element needs to look up no more than k elements in the previous column, so Algorithm 2's run-time is $\mathcal{O}(mk^2)$ given C.

The total run-time is thus $\mathcal{O}(m^{k+1}n|\mathsf{X}|^{kr}))$ when $k \le m-1$ and $\mathcal{O}(mn(1+|\mathsf{X}|^r)^{m-1})$ otherwise. A brute-force search would, like Algorithm 2, require computing all elements of $\mathcal{DI}_{\text{BndInd}}$ first. It would then test all $\binom{m(m-1)}{k} = \mathcal{O}(m^{2k})$ graphs (for $k \le m-1$). Thus, once given C, Algorithm 2's run-time of $\mathcal{O}(mk^2)$ is a significant improvement.

Remark 5. *Algorithm 1 is impractical for large k and m. The main bottleneck is computing directed information values to fill in table C. This can be parallelized. Faster estimation methods would reduce the overall complexity.*

An alternative heuristic is to only compute directed information values up to an in-degree limit (such as three or four). Algorithm 2 could be slightly modified to find the best k-sparse approximation with limited in-degrees. Furthermore, once $k \ge m-1$ (or a limit on the in-degree), then the complexity of Algorithm 1, FILLINC*, does not grow with k. A wide range of sparsity levels could be examined in that regime without an increase in complexity.*

IV. SIMULATIONS

In this section, we demonstrate the performance of optimal k-sparse approximations through network simulations.

1) Setup: The simulations consisted of 150 consensus games. In each game, a network of $m=10$ nodes was generated. Each node was randomly assigned parents, with the number of parents picked uniformly between two and six. See Figure 2a for an example network. At each time-step t, every node i would select one of two states $0/1$ based on its current state $X_{i,t}$, the state of its parents $\underline{\mathbf{X}}_{A(i),t}$, and its bias $a_1 \in [0,1]$ for state 1. The selection followed the distribution

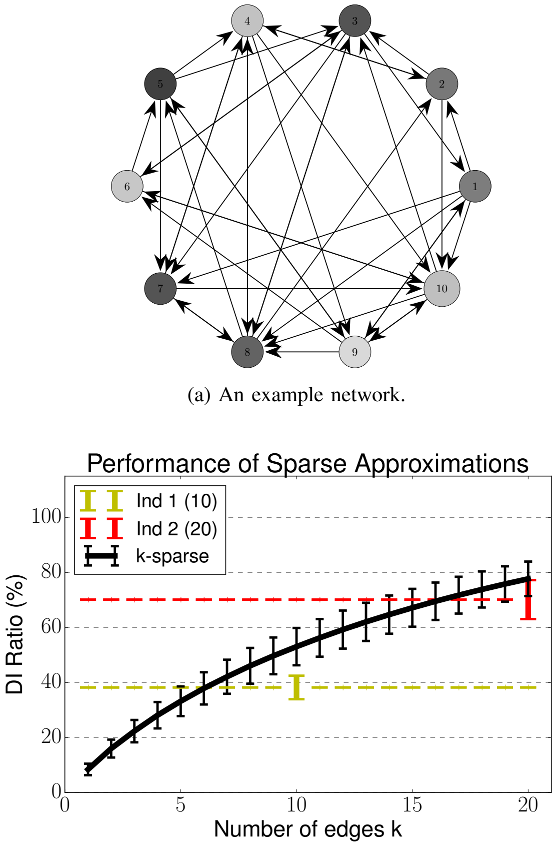
$$P_{X_{i,t+1}|\underline{\mathbf{X}}_{A(i)\cup\{i\},t}}(1|\underline{\mathbf{X}}_{A(i)\cup\{i\},t}) = \frac{a_1\gamma_1}{a_1\gamma_1 + a_0\gamma_0},$$

where $\gamma_1 := (X_{i,t} + \sum_{j\in A(i)} X_{j,t})$, the number of 1's that node i sees at time t, and γ_0 is defined similarly.

Each game consisted of 1000 rounds with the same network. Each round lasted for $t=10$ time steps or until consensus. The initial states were randomly generated. A high number of rounds was used to ensure accurate estimates.

The performance of the k-sparse approximations generated by Algorithm 2 were measured using the following ratio. The ratio is the sum of directed informations from chosen parent sets to children divided by that of the true parents,

$$\frac{\sum_{i=1}^{m}\mathrm{I}(\underline{\mathbf{X}}_{\widehat{A}(i)} \to \mathbf{X}_i)}{\sum_{i=1}^{m}\mathrm{I}(\underline{\mathbf{X}}_{A(i)} \to \mathbf{X}_i)}, \tag{8}$$



(a) An example network.

(b) Approximation performance, as 100% times ratio (8).

Fig. 2. Figure 2a shows an example network of $m = 10$ nodes. Darker nodes have a higher bias a_1. Figure 2b shows the simulation results for the best k-sparse approximations versus the best in-degree one and two approximations.

where $A(i)$ and $\widehat{A}(i)$ denote the true and inferred parent sets respectively. This ratio characterizes how much of the causal dynamics the estimated parent sets account for. Recall from Theorem III.1 that larger sums (eg. the numerator of (8)) correspond to better approximations. The directed information values were estimated using the plug-in empirical estimator (see, for instance, [16]).

For comparison, the ratio (8) was also calculated for the best in-degree one and in-degree two approximations [16] which used 10 and 20 edges respectively.

2) Results: Overall, the k-sparse approximation performed well. The mean and standard deviation the ratio (8) over 150 games is shown in Figure 2b. At $k = 10$ and 20 edges, the ratio had already reached 50% and 80% of the dynamics respectively. In contrast, the best in-degree one and two approximations had close to 38% and 70% using 10 and 20 edges respectively. On average, networks had forty edges.

V. Conclusion

We developed and tested an algorithm that finds the optimal, sparse approximation of a network where the user controls the sparsity level. The algorithm works regardless of the network topology or the class of the joint distribution. Important future directions include identifying faster, heuristic algorithms that are provably good for major classes of distributions, exploring the robustness of the approximations in the small-data regime, and developing automatic procedures for selecting k.

Acknowledgments

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[1] Z. Chen, K. Zhang, and L. Chan, "Causal discovery with scale-mixture model for spatiotemporal variance dependencies," in *Advances in Neural Information Processing Systems*, 2012, pp. 1727–1735.

[2] R. Chang, J. R. Karr, and E. E. Schadt, "Causal inference in biology networks with integrated belief propagation," in *Pacific Symposium on Biocomputing*, vol. 20. World Scientific, 2014, pp. 359–370.

[3] J. Pearl, *Causality: Models, Reasoning and Inference*, 2nd ed. Cambridge Univ Press, 2009.

[4] K. Friston, "Causal modelling and brain connectivity in functional magnetic resonance imaging," *PLoS biology*, vol. 7, no. 2, p. 220, 2009.

[5] J. B. Ullman and P. M. Bentler, *Structural Equation Modeling*. Wiley Online Library, 2003.

[6] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.

[7] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[8] C. J. Quinn, T. P. Coleman, N. Kiyavash, and N. G. Hatsopoulos, "Estimating the directed information to infer causal relationships in ensemble neural spike train recordings," *Journal of Computational Neuroscience*, vol. 30, no. 1, pp. 17–44, 2011.

[9] P. O. Amblard and O. J. J. Michel, "On directed information theory and Granger causality graphs," *Journal of Computational Neuroscience*, vol. 30, no. 1, pp. 7–16, 2011.

[10] D. Materassi and G. Innocenti, "Topological identification in networks of dynamical systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1860–1871, 2010.

[11] V. Y. F. Tan and A. S. Willsky, "Sample complexity for topology estimation in networks of LTI systems," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on, Orlando, Florida*, 2011, pp. 187–192.

[12] A. Bolstad, B. D. Van Veen, and R. Nowak, "Causal network inference via group sparse regularization," *IEEE Transactions on Signal Processing*, vol. 59, no. 6, pp. 2628–2641, 2011.

[13] S. Basu, A. Shojaie, and G. Michailidis, "Network Granger causality with inherent grouping structure," *Journal of Machine Learning Research*, vol. 16, pp. 417–453, 2015.

[14] C. J. Quinn, N. Kiyavash, and T. P. Coleman, "Efficient methods to compute optimal tree approximations of directed information graphs," *IEEE Trans. on Signal Processing*, vol. 61, no. 12, pp. 3173–3182, 2013.

[15] C. J. Quinn, A. Pinar, and N. Kiyavash, "Bounded degree approximations of stochastic networks," *arXiv preprint arXiv:1506.04767*, 2015.

[16] C. J. Quinn, N. Kiyavash, and T. P. Coleman, "Directed information graphs," *Information Theory, IEEE Transactions on*, vol. 61, no. 12, pp. 6887–6909, Dec 2015.

[17] M. Koivisto and K. Sood, "Exact Bayesian structure discovery in Bayesian networks," *The Journal of Machine Learning Research*, vol. 5, pp. 549–573, 2004.

[18] P. Parviainen and M. Koivisto, "Finding optimal Bayesian networks using precedence constraints," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1387–1415, 2013.

[19] G. Kramer, "Directed Information For Channels With Feedback," Ph.D. dissertation, Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland, 1998.

[20] H. Marko, "The bidirectional communication theory–a generalization of information theory," *IEEE Transactions on Communications*, vol. 21, no. 12, pp. 1345–1351, Dec 1973.