ISIT'2013 1569720229

Optimal Bounded-Degree Approximations of Joint Distributions of Networks of Stochastic Processes

Christopher J. Quinn
Dep. of Electrical & Computer Eng.
University of Illinois
Urbana, Illinois
Email: quinn7@illinois.edu

Ali Pinar
Sandia National Laboratories
Livermore, California
Email: apinar@sandia.gov

Negar Kiyavash
Dep. of Industrial and Enterprise Systems Eng.
University of Illinois
Urbana, Illinois
Email: kiyavash@illinois.edu

***Abstract*—We propose two algorithms to identify approximations for joint distributions of networks of stochastic processes. The approximations correspond to low-complexity network structures - connected, directed graphs with bounded indegree. The first algorithm identifies an optimal approximation in terms of KL divergence. The second efficiently finds a near-optimal approximation. Sufficient conditions are introduced to guarantee near-optimality.**

I. Introduction

There is a large body of research on identifying the structure of large networks from data in various domains such as social sciences, biology, and engineering. In some applications, it is not practical to identify the full network structure. The computation required might be prohibitively intense or only the most significant influences might be necessary for analysis purposes. Thus, in many practical settings, researchers seek good approximations of the full network structure.

Suppose a researcher is studying paths of influence in a large online social network. As users have tens to hundreds of friends, the researcher might want to consider only the five or ten most important friends. Thus, he would want the best bounded-indegree approximation of the full network. Since he is studying paths of influence, the network approximation should be connected. Tree approximations, with indegree one, would be too restrictive. Lastly, depending on the analysis, the researcher might be willing to trade optimality of the approximation for reduced computation.

In this paper, we consider the problem of finding the best approximation for a joint distribution of a network of interacting stochastic processes, where the approximation's structure is connected and has bounded indegree. We present an algorithm to find an optimal such structure in polynomial time. We also present an algorithm to efficiently identify a near-optimal approximation.

For the problem of identifying the full structure of a network of stochastic processes, algorithms are discussed in [1]. [2] presents an efficient algorithm to find the best directed spanning tree approximation. That is analogous to the problem of finding the best undirected spanning tree approximation for a network of random variables [3]. For networks of random variables, in general it is NP-hard to find approximations with indegree higher than one. Some heuristic approaches are reviewed in [4].

This paper is organized as follows. Section II describes the problem of identifying optimal approximations with corresponding low-complexity graph structures. Section III proposes an algorithm that identifies the best approximation in terms of KL divergence. Section IV discusses sufficient conditions for a greedy algorithm to efficiently determine a near-optimal approximation. Lastly, Section V describes the computational complexity of the algorithms and the storage complexity of the approximations.

II. Setup

Let $\underline{\mathbf{X}} = \{\mathbf{X}_1, \ldots, \mathbf{X}_m\}$ be a network of m stochastic processes, each of length n. Let $P_{\underline{\mathbf{X}}}$ denote its joint distribution. Let $K < m$ be an integer bound on the indegree of the approximations. For a given process $\mathbf{X}_i$, let $\underline{\mathrm{A}}(i) \subseteq \{1, \ldots, m\} \backslash \{i\}$ denote a set of indices of other processes, with $|\underline{\mathrm{A}}(i)| \leq K$. Let $\mathcal{G}$ denote the set of all graphs on m nodes that contain a directed spanning tree and with indegrees at most K. For a particular $G \in \mathcal{G}$, the set $\underline{\mathrm{A}}(i)$ is the set of indices of parents of $\mathbf{X}_i$ in structure G. Using causal conditioning notation, where for two processes Y^n and Z^n,

$$P_{Y^n \| Z^n}(y^n \| z^n) \triangleq \prod_{t=1}^{n} P_{Y_t | Y^{t-1}, Z^{t-1}}(y_t | y^{t-1}, z^{t-1}), \qquad (1)$$

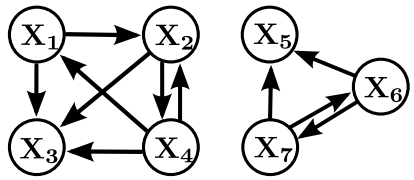
an approximating distribution for network $\underline{\mathbf{X}}$ has the form

$$\widehat{P}_{\underline{\mathbf{X}}}(d\underline{\mathbf{x}}) \triangleq \prod_{i=1}^{m} P_{\mathbf{X}_i \| \mathbf{X}_{\underline{\mathrm{A}}(i)}}(d\mathbf{x}_i \parallel \mathbf{x}_{\underline{\mathrm{A}}(i)}). \qquad (2)$$

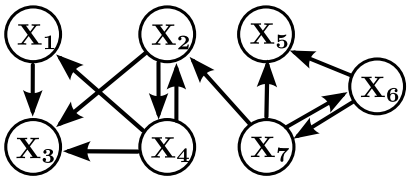
The objective is to select the approximating structure $G \in \mathcal{G}$ whose corresponding approximation $\widehat{P}_{\underline{\mathbf{X}}}$ (2) minimizes the KL divergence $\mathrm{D}(P_{\underline{\mathbf{X}}} \| \widehat{P}_{\underline{\mathbf{X}}})$. Minimizing the KL divergence $\mathrm{D}(P_{\underline{\mathbf{X}}} \| \widehat{P}_{\underline{\mathbf{X}}})$ corresponds to maximizing a sum of directed informations.

Theorem 1.

$$\arg\min_{G \in \mathcal{G}} \mathrm{D}(P_{\underline{\mathbf{X}}} \parallel \widehat{P}_{\underline{\mathbf{X}}}) = \arg\max_{G \in \mathcal{G}} \sum_{i=1}^{m} \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}(i)} \to \mathbf{X}_i). \qquad (3)$$



(a) A disconnected structure.



(b) A structure with multiple directed spanning trees.

Fig. 1. The structures of two approximations. The first is disconnected. The other contains multiple directed spanning trees, with roots $\mathbf{X}_6$ or $\mathbf{X}_7$. Both structures have the same indegrees.

Directed information was defined by Marko [5] as[1]:

$$I(\underline{\mathbf{Z}} \to \mathbf{Y}) \;=\; \int_{\underline{\mathbf{z}}} D\big(P_{\mathbf{Y}\|\underline{\mathbf{Z}}=\underline{\mathbf{z}}} \| P_{\mathbf{Y}}\big)\, P_{\underline{\mathbf{Z}}}(d\underline{\mathbf{z}}). \tag{4}$$

Theorem 1 was proved for the case that $\mathcal{G}$ was the set of all directed spanning trees on m nodes, with $K = 1$ [2]. The proof for the more general case is similar and not included for space constraints.

In [2], an efficient algorithm was proposed to identify the spanning tree with maximal sum (3), called the maximal weight directed spanning tree (MWDST). Pairwise directed informations $\{\mathrm{I}(\mathbf{X}_j \to \mathbf{X}_i)\}$ for all ordered pairs (j, i) were computed. Then value $\mathrm{I}(\mathbf{X}_j \to \mathbf{X}_i)$ was assigned as the weight of edge (j, i) in the complete graph on m nodes. Next, an $\mathcal{O}(m^2)$ MWDST algorithm [6] was used to identify an optimal approximation. This result is analogous to a problem solved by Chow and Liu [3] for networks of random variables.

We next discuss algorithms to identify the best connected approximation.

III. Optimal Connected Approximation

If the set of approximating structures $\mathcal{G}$ is not restricted to connected structures, (3) can be easily solved. In this case, the maximization can be taken inside the summation (3). Thus, for an optimal structure, each process $\mathbf{X}_i$ would have parent set $\underline{\mathrm{A}}(i)$ that had maximal directed information $\underline{\mathrm{A}}(i) = \arg\max_{\underline{\mathrm{A}}'(i)} \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}'(i)} \to \mathbf{X}_i)$. However, there is no guarantee of connectivity. In some cases, such as when edges correspond to influence or information flow, it is useful to have paths between nodes to characterize their relationship. See Figure 1.

[1]As elaborated in [1], Marko's original definition [5] is consistent with Granger causality. Later works, in the specific context of communication channels with instantaneous outputs, included conditioning on Z_t in (1) to account for synchronization.

We now describe an algorithm to solve (3) for the case where $\mathcal{G}$ is the set of all connected structures with maximum indegree[2] $K \geq 1$. For every pair of processes $\{\mathbf{X}_i, \mathbf{X}_j\}$, calculate the best set of K parents $\underline{\mathrm{A}}(i, j)$ for $\mathbf{X}_i$ that contains the edge $\mathbf{X}_j \to \mathbf{X}_i$. Then assign weight $\mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}(i,j)} \to \mathbf{X}_i)$ to edge $\mathbf{X}_j \to \mathbf{X}_i$ in the complete graph on m nodes and run the MWDST algorithm to find the best directed spanning tree. For each process $\mathbf{X}_i$, the set $\underline{\mathrm{A}}(i, j)$ picked determines the parent set. This process is described in Algorithm 1. It takes as input joint distributions of subsets of processes

$$\mathcal{R} = \{P_{\underline{\mathbf{X}}_{\mathrm{W}}} :\; \underline{\mathrm{W}} \subseteq \{1, \ldots, m\} \text{ and } |\underline{\mathrm{W}}| \leq K + 1\}.$$

Algorithm 1. Optimal Connected

Input: $\mathcal{R}$, K

1. **For** $i \in \{1, \ldots, m\}$
2. $\quad \underline{\mathrm{A}}(i) \leftarrow \emptyset$
3. $\quad$ **For** $j \in \{1, \ldots, m\} \backslash \{i\}$
4. $\quad\quad$ **For** $\underline{\mathrm{W}} \in \{\underline{\mathrm{W}}' : \underline{\mathrm{W}}' \subseteq \{1, \ldots, m\} \backslash \{i\}$ and $j \in \underline{\mathrm{W}}'$ and $|\underline{\mathrm{W}}'| = K\}$
5. $\quad\quad\quad$ **Compute** $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{W}}} \to \mathbf{X}_i)$
6. $\quad\quad$ $\underline{\mathrm{A}}(i, j) \leftarrow \arg\max_{\underline{\mathrm{W}}} \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{W}}} \to \mathbf{X}_i)$
7. $\{\underline{\mathrm{A}}(i)\}_{i=1}^{m} \leftarrow$ **MWDST** $(\{\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}(i,j)} \to \mathbf{X}_i)\}_{1 \leq i \neq j \leq m})$

Theorem 2. *Algorithm 1 returns an optimal approximation* $\widehat{P}_{\underline{\mathbf{X}}}$ *to* (3)*.*

Proof. The MWDST algorithm will pick the directed spanning tree with the largest sum of edge weights. In Algorithm 1, edge $\mathbf{X}_j \to \mathbf{X}_i$ is given weight $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}(i,j)} \to \mathbf{X}_i)$. $\underline{\mathrm{A}}(i, j)$ is the best set of parents for $\mathbf{X}_i$ that includes $\mathbf{X}_j$. The sum of edge weights picked thus corresponds to the sum of directed informations in Theorem 1. The MWDST maximizes that sum over all possible directed spanning trees. □

Algorithm 1 finds an optimal approximation in terms of KL divergence $\mathrm{D}(P_{\underline{\mathbf{X}}} \| \widehat{P}_{\underline{\mathbf{X}}})$. It only uses distributions over $K + 1$ processes instead of the full joint distribution. However, it computes $m\binom{m-1}{K}$ directed informations involving K processes. If K is large, this could be computationally difficult. For some applications, instead of reducing K, it is more beneficial to efficiently identify a near-optimal approximation.

IV. Efficient Near-Optimal Connected

In this section, we examine conditions when a near-optimal solution to (3) can be found in time polynomial in K.

A. Greedy Submodularity

Consider the following greedy procedure. Set $\mathbf{X}_i$'s parent set as the best individual parent $\mathbf{Z} = \arg\max_j \mathrm{I}(\mathbf{X}_j \to \mathbf{X}_i)$. Then look for the second best parent $\arg\max_j \mathrm{I}(\mathbf{X}_j \to \mathbf{X}_i \| \mathbf{Z})$. Repeat in this manner K times, adding one parent at each iteration.

[2]K can vary for different nodes. To simplify presentation, we consider the case where K is uniform.

In general, greedy algorithms are not provably good. We next describe sufficient conditions to guarantee near-optimality.

Definition 1. *A joint distribution $P_{\underline{\mathbf{X}}}$ is called* greedily-submodular *if there exists an $\alpha > 0$, such that for any process $\mathbf{Y}$ and any subset $\underline{\mathbf{X}}_{\underline{\mathrm{W}}}$ of other processes,*

$$\mathrm{I}(\mathbf{X}_{i+1} \to \mathbf{Y} \| \mathbf{X}_1, \ldots, \mathbf{X}_i) \leq \alpha \, \mathrm{I}(\mathbf{X}_i \to \mathbf{Y} \| \mathbf{X}_1, \ldots, \mathbf{X}_{i-1}), \quad (5)$$

for all $1 \leq i < |\underline{\mathrm{W}}|$ where the processes in $\underline{\mathbf{X}}_{\underline{\mathrm{W}}}$ are indexed according to the order in which they are selected by the greedy algorithm.

This is a weaker condition than submodularity, a discrete analog of concavity [7]. If $P_{\underline{\mathbf{X}}}$ had submodular directed information values, then for all pairs of processes $\{\mathbf{X}_j, \mathbf{X}_i\}$ and sets of processes $\underline{\mathbf{X}}_S \subseteq \underline{\mathbf{X}}_{S'} \subseteq \underline{\mathbf{X}} \backslash \{\mathbf{X}_j, \mathbf{X}_i\}$,

$$\mathrm{I}(\mathbf{X}_j \to \mathbf{X}_i \| \underline{\mathbf{X}}_{S'}) \leq \mathrm{I}(\mathbf{X}_j \to \mathbf{X}_i \| \underline{\mathbf{X}}_S). \quad (6)$$

Submodularity implies conditioning does not increase directed information. Note that (6) with $j = i+1$, $S = \{1, \ldots, i-1\}$ and $S' = S \cup \{i\}$ implies (5) holds with $\alpha \leq 1$ since $\mathbf{X}_{i+1}$ is chosen after $\mathbf{X}_i$. Entropy is submodular [8]. However, in general mutual information and directed information are not, as shown in the following example.

Example 1. *Consider a network of three zero-mean, jointly Gaussian processes $\{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$. Let $\mathbf{N}$ denote noise. Let $\{\mathbf{N}, \mathbf{X}, \mathbf{Z}\}$ be mutually independent i.i.d. Gaussian processes. Let $Y_{t+1} = X_t + Z_t + N_t$. Then from [9] pg. 256,*

$$\begin{aligned} \mathrm{I}(\mathbf{X} \to \mathbf{Y}) &= \frac{1}{n} \sum_{t=1}^{n} \mathrm{I}(X^{t-1}; Y_t | Y^{t-1}) = \mathrm{I}(X_1; Y_2) \\ &= \frac{1}{2} \log \left(1 + \frac{var(X_1)}{var(Z_1) + var(N_1)} \right). \end{aligned}$$

First note that

$$\mathrm{I}(\mathbf{X} \to \mathbf{Y}) < \mathrm{I}(\mathbf{X} \to \mathbf{Y} \| \mathbf{Z}),$$

so directed information is not submodular. Suppose $var(X_1) = var(Z_1) \gg var(N_1)$. Then α in (5) *would be large. If instead the variances were equal, $\alpha = 1.71$ would suffice.*

Remark 1. *The authors are not aware of this property being discussed in the literature previously. Two other conditions that are weaker than submodularity are [10] and [11]. The former uses submodularity up to an additive error. The latter uses submodularity up to multiplicative error. Both measure the increase in conditioning of the terms in* (6)*, unlike* (5) *which only bounds sequential increases while greedily selecting a parent set.*

Assumption 1. *We assume that $P_{\underline{\mathbf{X}}}$ is greedily-submodular.*

Let $\underline{\mathrm{A}}$ denote the set of indices for an optimal set of K parents and $\underline{\mathrm{B}}$ the indices for the greedily selected set of $L \leq K$ parents.

Theorem 3. *Under Assumption 1,*

$$\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{B}}} \to \mathbf{Y}) \geq \left(1 - \exp\left(\frac{-L}{\sum_{i=0}^{K-1} \alpha^i} \right) \right) \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}} \to \mathbf{Y}).$$

Remark 2. *The proof is similar to the proof for a related bound for submodular functions [7]. We include it for completeness and to use notation specific to directed information.*

Proof. For simplicity, we prove the case $\underline{\mathrm{A}} \bigcap \underline{\mathrm{B}} = \emptyset$. The other case is almost identical and results in a tighter bound. Note that the greedy algorithm selects each element of $\underline{\mathrm{B}}$ before any element of $\underline{\mathrm{A}}$. Let $\underline{\mathrm{A}}_l$ be the set $\underline{\mathrm{A}}$ but ordered according to how the greedy algorithm would pick elements from $\underline{\mathrm{A}}$ after picking $\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}$.

We first note two inequalities. For all $l < L$,

$$\mathrm{I}(\mathbf{X}_{\underline{\mathrm{B}}(l+1)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}}) \geq \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}_l(1)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}}), \quad (7)$$

which holds since $\mathbf{X}_{\underline{\mathrm{B}}(l+1)}$ is the process that the greedy algorithm selects after $\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}$, and

$$\alpha^{i-1} \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}_l(1)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}}) \geq \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}_l(i)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l), \underline{\mathrm{A}}_l(1), \ldots, \underline{\mathrm{A}}_l(i-1)\}}), \quad (8)$$

which follows from Assumption 1 for the set $\underline{\mathrm{A}} \cup \{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}$.

We now compare an optimal solution to the first l elements in the greedy solution.

$$\begin{aligned} &\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}} \to \mathbf{Y}) - \mathrm{I}(\underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}} \to \mathbf{Y}) \\ &\quad \leq \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}} \cup \{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}} \to \mathbf{Y}) - \mathrm{I}(\underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}} \to \mathbf{Y}) \\ &\quad = \mathrm{I}(\underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}} \to \mathbf{Y}) \\ &\qquad + \sum_{i=1}^{K} \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}_l(i)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\} \cup \{\underline{\mathrm{A}}_l(1), \ldots, \underline{\mathrm{A}}_l(i-1)\}}) \\ &\qquad - \mathrm{I}(\underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}} \to \mathbf{Y}) \quad (9) \\ &\quad \leq \sum_{i=1}^{K} \alpha^{i-1} \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}_l(1)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}}) \quad (10) \\ &\quad \leq \sum_{i=1}^{K} \alpha^{i-1} \mathrm{I}(\mathbf{X}_{\underline{\mathrm{B}}(l+1)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}}). \quad (11) \end{aligned}$$

Equation (9) follows from the chain rule applied in the order the greedy algorithm would select from $\underline{\mathrm{A}} \cup \{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}$. Equations (10) and (11) follow from (8) and (7) respectively.

Let $\delta_l \triangleq \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}} \to \mathbf{Y}) - \mathrm{I}(\underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}} \to \mathbf{Y})$. We can then write $\delta_l - \delta_{l+1} = \mathrm{I}(\mathbf{X}_{\underline{\mathrm{B}}(l+1)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1), \ldots, \underline{\mathrm{B}}(l)\}})$. Also denote $\beta \triangleq \sum_{i=1}^{K} \alpha^{i-1}$. From (11) we have

$$\delta_l \leq \beta \left(\delta_l - \delta_{l+1} \right),$$

which implies $\delta_{l+1} \leq \left(1 - \frac{1}{\beta} \right) \delta_l$. Thus

$$\delta_l \leq \left(1 - \frac{1}{\beta} \right)^l \delta_0 \leq e^{-\frac{l}{\beta}} \delta_0.$$

The last step uses the bound $(1-p) \leq e^{-p}$, which holds for all p. For $0 < p < 1$, both sides are positive so powers can be

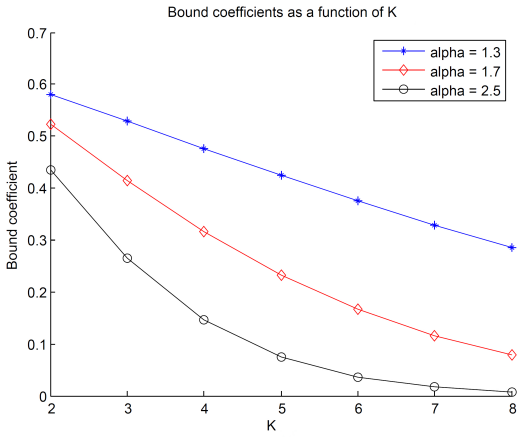


Fig. 2. A plot of the values of the bound in Theorem 3, for $\alpha \in \{1.3, 1.7, 2.5\}$ and $L = K \in \{2, \ldots, 8\}$.

taken. Since $\delta_0 = \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}} \to \mathbf{Y}) - \mathrm{I}(\emptyset \to \mathbf{Y}) = \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}} \to \mathbf{Y})$, this gives

$$\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}} \to \mathbf{Y}) - \mathrm{I}(\underline{\mathbf{X}}_{\{\underline{\mathrm{B}}(1),\ldots,\underline{\mathrm{B}}(l)\}} \to \mathbf{Y}) \leq e^{-\frac{l}{\beta}} \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}} \to \mathbf{Y}),$$

which after rearranging gives the theorem. □

Figure 2 shows the bound coefficient for $\alpha \in \{1.3, 1.7, 2.5\}$ and $L = K$ for $K \in \{2, \ldots, 8\}$.

We can also bound how close an optimal solution on K elements is to an optimal solution on $K+1$ elements. Let $\underline{\mathrm{A}}^K_{\mathrm{opt}}$ denote an optimal solution of K elements.

Lemma 1. *Under Assumption 1, with* $\alpha \neq 1$,

$$\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K+1}_{opt}} \to \mathbf{Y}) \leq \left(\frac{\alpha^{K+1}-1}{\alpha^K - 1}\right) \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K}_{opt}} \to \mathbf{Y}).$$

Proof. Let $\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}$ and $\underline{\mathrm{A}}^{K}_{\mathrm{opt}}$ each be ordered according to the greedy order for those sets separately. We consider making $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}} \to \mathbf{Y})$ as large as possible given

$$\mathrm{I}(\underline{\mathbf{X}}_{\{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1),\ldots,\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(K)\}} \to \mathbf{Y}) \leq \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K}_{\mathrm{opt}}} \to \mathbf{Y}).$$

Note that if for any $0 < i < K$,

$$\mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(i+1)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1),\ldots,\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(i)\}}) < \alpha\, \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(i)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1),\ldots,\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(i-1)\}}), \quad (12)$$

then with the difference between the terms in (12),

$$\mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(i)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1),\ldots,\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(i-1)\}})$$

could be decreased, and for $j \geq i+1$,

$$\mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(j)} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1),\ldots,\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(j-1)\}})$$

could be increased, thus increasing how large $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}} \to \mathbf{Y})$ could be. Assume (12) is met with equality for each i in the set $\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}$. This gives

$$\begin{aligned} \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K}_{\mathrm{opt}}} \to \mathbf{Y}) &\geq \mathrm{I}(\underline{\mathbf{X}}_{\{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1),\ldots,\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(K)\}} \to \mathbf{Y}) \\ &= \sum_{i=0}^{K-1} \alpha^i\, \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1)} \to \mathbf{Y}), \end{aligned} \quad (13)$$

which implies

$$\mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1)} \to \mathbf{Y}) \leq \frac{1}{\sum_{i=0}^{K-1} \alpha^i}\, \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K}_{\mathrm{opt}}} \to \mathbf{Y}).$$

Next,

$$\begin{aligned} \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}} \to \mathbf{Y}) &= \sum_{i=0}^{K} \alpha^i\, \mathrm{I}(\mathbf{X}_{\underline{\mathrm{A}}^{K+1}_{\mathrm{opt}}(1)} \to \mathbf{Y}) \\ &\leq \frac{\sum_{i=0}^{K} \alpha^i}{\sum_{i=0}^{K-1} \alpha^i}\, \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K}_{\mathrm{opt}}} \to \mathbf{Y}) \\ &= \left(1 + \frac{\alpha^K}{\sum_{i=0}^{K-1} \alpha^i}\right) \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}^{K}_{\mathrm{opt}}} \to \mathbf{Y}). \end{aligned} \quad (14)$$

If $\alpha \neq 1$, then we can use the geometric series formula $\sum_{i=0}^{K-1} \alpha^i = \frac{1-\alpha^K}{1-\alpha}$ to conclude

$$1 + \frac{\alpha^K}{\sum_{i=0}^{K-1} \alpha^i} = \frac{\alpha^{K+1}-1}{\alpha^K - 1}.$$

□

B. The Algorithm

We now discuss a procedure to greedily find a near-optimal solution to (3), described in Algorithm 2. Similar to Algorithm 1, it precomputes parent sets for each possible directed edge, though in a greedy fashion. Then a MWDST algorithm is called. Denote the parent sets as $\{\underline{\mathrm{B}}(i)\}$.

Algorithm 2. Near-optimal Connected

Input: $\mathcal{R}$, K

1. **For** $i \in \{1, \ldots, m\}$
2. $\quad \underline{\mathrm{B}}(i) \leftarrow \emptyset$
3. $\quad$ **For** $j \in \{1, \ldots, m\} \backslash \{i\}$
4. $\quad\quad \underline{\mathrm{B}}(i,j) \leftarrow \{j\}$
5. $\quad\quad\quad$ **For** $L = 2, \ldots, K$
6. $\quad\quad\quad\quad$ **For** $l \in \{1, \ldots, m\} \backslash \underline{\mathrm{B}}(i,j)$
7. $\quad\quad\quad\quad\quad$ **Compute** $\mathrm{I}(\mathbf{X}_l \to \mathbf{X}_i \| \underline{\mathbf{X}}_{\underline{\mathrm{B}}(i,j)})$
8. $\quad\quad\quad\quad$ $\underline{\mathrm{B}}(i,j) \leftarrow \underline{\mathrm{B}}(i,j) \bigcup \arg\max_l \mathrm{I}(\mathbf{X}_l \to \mathbf{X}_i \| \underline{\mathbf{X}}_{\underline{\mathrm{B}}(i,j)})$
9. $\{\underline{\mathrm{B}}(i)\}_{i=1}^m \leftarrow$ **MWDST** $(\{\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{B}}(i,j)} \to \mathbf{X}_i)\}_{1 \leq i \neq j \leq m})$

In Algorithm 2, $\underline{\mathrm{B}}(i,j)$ is the best set of parents for $\mathbf{X}_i$ with $\mathbf{X}_j$ as one of the parents, selected in a greedy fashion. The value $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{B}}(i,j)} \to \mathbf{X}_i)$ is the weight of edge $\mathbf{X}_j \to \mathbf{X}_i$ given to the MWDST algorithm.

Remark 3. *The edge weight* $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{B}}(i,j)} \to \mathbf{X}_i)$ *can be computed by the chain rule, using the processes picked in line 8. Let* $\{j_1, j_2, \ldots, j_K\}$ *denote* $\underline{\mathrm{B}}(i,j)$. *Then*

$$\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{B}}(i,j)} \to \mathbf{X}_i) = \sum_{l=1}^{K} \mathrm{I}(\mathbf{X}_{j_l} \to \mathbf{X}_i \| \mathbf{X}_{j_1}, \mathbf{X}_{j_2}, \ldots \mathbf{X}_{j_{l-1}}).$$

Let $\{\underline{\mathrm{A}}(i)\}_{i=1}^m$ denote the parent sets returned by Algorithm 1.

Theorem 4. *Under Assumption 1, for Algorithm 2*

$$\sum_{i=1}^{m} \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{B}}(i)} \to \mathbf{X}_i) \geq \left(1 - \exp\left(\frac{K}{\sum_{i=0}^{K-1} \alpha^i}\right)\right) \sum_{i=1}^{m} \mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}(i)} \to \mathbf{X}_i).$$

Proof. Let T_1 denote the MWDST picked in Algorithm 1. For an edge $e \in \{\mathbf{X}_j \to \mathbf{X}_i : 1 \leq j \neq i \leq m\}$ in the complete graph on m nodes, let $w_1(e)$ denote the weight $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{A}}(i,j)} \to \mathbf{X}_i)$ assigned by Algorithm 1. Define T_2 and $w_2(e)$ for Algorithm 2 likewise. Also, let $c \triangleq \left(1 - \exp\left(\frac{K}{\sum_{i=0}^{K-1} \alpha^i}\right)\right)$. For each edge e in the complete graph,

$$w_2(e) \geq c w_1(e), \tag{15}$$

which follows from Theorem 3. Furthermore,

$$\sum_{e \in T_2} w_2(e) \geq \sum_{e \in T_1} w_2(e) \tag{16}$$

$$\geq c \sum_{e \in T_1} w_1(e). \tag{17}$$

Equation (16) follows since in Algorithm 2, T_2 was selected as the MWDST, and (17) follows from (15). □

V. Complexity

We now examine the computational complexity of the proposed algorithms. We then consider storage complexity of the approximations.

First consider the complexity of computing a directed information involving multiple processes. Calculating $\mathrm{I}(\mathbf{X}, \mathbf{Z} \to \mathbf{Y})$ without any assumptions has exponential complexity. Note that

$$\mathrm{I}(\mathbf{X}, \mathbf{Z} \to \mathbf{Y}) = \sum_{t=1}^{n} \mathrm{I}(Y_t; X^{t-1}, Z^{t-1} | Y^{t-1}).$$

The last term in particular, $\mathrm{I}(Y_n; X^{n-1}, Z^{n-1} | Y^{n-1})$, involves a sum over all realizations of $3n-2$ random variables. Thus, the last term has complexity $\mathcal{O}(|\mathsf{X}|^{3n})$. We will assume Markovicity of a fixed order l, so

$$\mathrm{I}(\mathbf{X}, \mathbf{Z} \to \mathbf{Y}) = \sum_{t=1}^{n} \mathrm{I}(Y_t; X_{t-l}^{t-1}, Z_{t-l}^{t-1} | Y_{t-l}^{t-1}).$$

The complexity of computing $\mathrm{I}(\mathbf{X}, \mathbf{Z} \to \mathbf{Y})$ then becomes $\mathcal{O}(n|\mathsf{X}|^{3l}) = \mathcal{O}(n)$. More generally, computing $\mathrm{I}(\underline{\mathbf{X}}_{\mathrm{W}} \to \mathbf{Y} \| \underline{\mathbf{X}}_{\mathrm{W}'})$, where $|\underline{\mathrm{W}}| + |\underline{\mathrm{W}}'| = K$, has $\mathcal{O}(n|\mathsf{X}|^{(K+1)l})$ complexity assuming Markovicity.

Assumption 2. *We assume Markovicity of order l.*

1) Algorithm 1: For each process $\mathbf{X}_i$ and each set $\underline{\mathrm{W}} \subseteq \{1, \ldots, m\} \backslash \{i\}$ of K indices, $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{W}}} \to \mathbf{X}_i)$ will be computed. There are $m\binom{m-1}{K}$ such values, and computing each requires complexity $\mathcal{O}(n|\mathsf{X}|^{(K+1)l})$. Note that $\mathrm{I}(\underline{\mathbf{X}}_{\underline{\mathrm{W}}} \to \mathbf{X}_i)$ only needs to be computed once and then shared between the $j \in \underline{\mathrm{W}}$. Algorithm 1 also performs a MWDST step, which runs in $\mathcal{O}(m^2)$ time [6]. Under Assumption 2, the total complexity is $\mathcal{O}(m\binom{m-1}{K} n|\mathsf{X}|^{(K+1)l})$. If K is fixed, the complexity becomes $\mathcal{O}(m^{K+1} n)$.

2) Algorithm 2: For each ordered pair of processes $(\mathbf{X}_i, \mathbf{X}_j)$, first there are $(m-2)$ directed informations computed involving three processes. Next there are $(m-3)$ computed involving four processes, and so on. After those computations, a MWDST algorithm is called. The complexity is thus

$$\mathcal{O}(m^2 + m(m-1)\sum_{i=1}^{K-1}(m-1-i)n|\mathsf{X}|^{(i+2)l}) = \mathcal{O}(Km^3 n|\mathsf{X}|^{(K+1)l}).$$

For constant K this becomes $\mathcal{O}\left(m^3 n\right)$.

3) Storage complexity: The full joint distribution has mn random variables, and so requires $\mathcal{O}(|\mathsf{X}|^{mn})$ storage. Under Assumption 2, the storage complexity of the full joint distribution and approximations are $\mathcal{O}(mn|\mathsf{X}|^{ml})$ and $\mathcal{O}(mn|\mathsf{X}|^{(K+1)l}) = \mathcal{O}(mn)$ respectively if K is fixed.

Acknowledgment

Christopher Quinn was supported by the Department of Energy Computational Science Graduate Fellowship under grant number DE-FG02-97ER25308. This work was supported in part by AFOSR under grant FA 9550-10-1-0573; and by NSF grants CCF 10-54937 CAR and CNS 08-31488. Ali Pinar was supported by the DOE ASCR Complex Distributed Interconnected Systems (CDIS) program.

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] C. J. Quinn, N. Kiyavash, and T. P. Coleman, "Directed information graphs," *ArXiv e-prints*, Apr. 2012.

[2] C. J. Quinn, T. P. Coleman, and N. Kiyavash, "Efficient methods to compute optimal tree approximations of directed information graphs," *IEEE Transactions on Signal Processing*, 2013, forthcoming.

[3] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE transactions on Information Theory*, vol. 14, no. 3, pp. 462–467, 1968.

[4] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

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

[6] J. Edmonds, "Optimum branchings," *J. Res. Natl. Bur. Stand., Sect. B*, vol. 71, pp. 233–240, 1967.

[7] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions I," *Mathematical Programming*, vol. 14, pp. 265–294, 1978.

[8] S. Fujishige, "Polymatroidal dependence structure of a set of random variables," *Information and Control*, vol. 39, no. 1, pp. 55–72, 1978.

[9] T. Cover and J. Thomas, *Elements of information theory*. Wiley-Interscience, 2006.

[10] V. Cevher and A. Krause, "Greedy dictionary selection for sparse representation," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 5, pp. 979–988, 2011.

[11] A. Das and D. Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," *ArXiv e-prints*, Feb. 2011.