Inverse Graph Learning over Optimization Networks

Vincenzo Matta, Augusto Santos, and Ali H. Sayed

Abstract

Many inferential and learning tasks can be accomplished efficiently by means of distributed optimization algorithms where the network topology plays a critical role in driving the local interactions among neighboring agents. There is a large body of literature examining the effect of the graph structure on the performance of optimization strategies. In this article, we examine the inverse problem and consider the reverse question: How much information does observing the behavior at the nodes convey about the underlying network structure used for optimization? Over large-scale networks, the difficulty of addressing such inverse questions (or problems) is compounded by the fact that usually only a limited portion of nodes can be probed, giving rise to a second important question: Despite the presence of several unobserved nodes, are partial and local observations still sufficient to discover the graph linking the probed nodes? The article surveys recent advances on this inverse learning problem and related questions. Examples of applications are provided to illustrate how the interplay between graph learning and distributed optimization arises in practice, e.g., in cognitive engineered systems such as distributed detection, or in other real-world problems such as the mechanism of opinion formation over social networks and the mechanism of coordination in biological networks. A unifying framework for examining the reconstruction error will be described, which allows to devise and examine various estimation strategies enabling successful graph learning. The relevance of specific network attributes, such as sparsity versus density of connections, and node degree concentration, is discussed in relation to the topology inference goal. It is shown how universal (i.e., data-driven) clustering algorithms can be exploited to solve the graph learning problem. Some revealing and perhaps unexpected behavior emerges: for example, comparison of different graph estimators highlights how strategies that are optimal for the full-observability regime are not necessarily the best ones for the limited-observability regime.

V. Matta is with DIEM, University of Salerno, via Giovanni Paolo II, I-84084, Fisciano (SA), Italy (e-mail: vmatta@unisa.it).
A. Santos was with the Adaptive System Laboratory, EPFL, CH-1015 Lausanne, Switzerland (e-mail: augusto.pt@gmail.com).
A. H. Sayed is with the École Polytechnique Fédérale de Lausanne EPFL, School of Engineering, CH-1015 Lausanne, Switzerland (e-mail: ali.sayed@epfl.ch).

The work of A. H. Sayed was supported in part by grant 205121-184999 from the Swiss National Science Foundation.
Index Terms

Distributed optimization, graph learning, topology inference, network tomography, Granger estimator, diffusion network, Erdős-Rényi graph.

I. INTRODUCTION

Optimization lies at the core of engineering design. In most man-made engineered systems, rational designs rely on meeting certain well-defined optimality and resource constraints. Even physical behavior in Nature offers admirable instances of optimization. For example, a water drop takes on a spherical shape in order to reach an optimal (i.e., minimal) energy configuration.

This work deals with the solution of optimization problems over complex systems whose evolution is dictated by interactions among a large number of elementary units or agents. Two fundamental attributes of such multi-agent systems are: i) the locality of information exchange between the individual units; and ii) the capability of such systems to solve rather effectively a range of demanding tasks (such as optimization, learning, inference) that would be unattainable by stand-alone isolated agents. Optimizing over these types of multi-agent networks gives rise to distributed cooperative solutions where each agent aims to attain the \textit{globally optimal} solution by means of \textit{local cooperation} with its neighbors [1]–[9].

There is a large body of literature that examines how the graph topology linking the agents affects the performance of distributed optimization methods. This article focuses on the reverse question, namely, what information the optimization solution conveys about the underlying topology. Specifically, assuming that we are able to observe the evolution of the signals at a subset of the agents over the network, we would like to examine what type of information can be extracted from these measurements in relation to the interconnections between the agents.

Rather than focus on \textit{what} the agents learn (which is the goal of the \textit{direct} learning problem), we focus instead on a \textit{dual} learning problem that deals with \textit{how} the agents learn (i.e., with the graph of connections through which the information propagated to feed the direct learning process). The dual learning problem is illustrated in Fig. [1] In the direct problem, we start from a graph topology, run a distributed optimization algorithm, and analyze its performance (such as convergence rate and closeness to optimal solution) and the dependence of this performance on the graph. In the dual problem, we start from observing the signals generated by the agents and focus instead on discovering the underlying graph that led to the observed signal evolution.

The inverse problem has many challenging aspects to it, as we explain below. Nevertheless, it is a problem of fundamental importance because it can provide answers to many useful questions of interest. For instance, by observing the evolution of signals at a subset of the agents, can one establish which
Fig. 1. Illustration of the dual learning problem considered in this work. A multi-agent network performs a distributed optimization task (the direct learning problem). The network graph influences the way each agent exchanges information with its neighbors. The online output of the optimization process, at time $i$ and agent $k$, is denoted by $w_k(i)$. An inferential engine can probe the subset $\{j, k, \ell, m\}$ of the network, and collect the pertinent outputs. Based on these outputs, the goal of the dual learning problem is to estimate the subgraph of connections between agents $j, k, \ell, m$.

agents are sharing information with each other? Or how privacy is reflected in the agents’ signals? Also, by observing the convergence behavior of some agents, can one discover which agents are having a magnified influence on the overall behavior of the network? Applications that can benefit from such answers are numerous. For example, discovering who is communicating with whom over the Internet or how information flows is an important step towards enhanced cyber-security [15]–[18]. Likewise, tracing the route of information flow over a social network can help understand the source of fake news or the mechanism of opinion formation [19], [20].

There exist some useful survey articles related to the topic of inverse graph modeling [21]–[23]. However, most prior works assume that all agents in a network can be observed. This is usually not the case. For example, in probing signals from the brain, only certain localities are examined. Also, in probing signal flows over a social network with millions of members, only a limited number of observations may be available. It is therefore critical to study how these limitations influence the inverse problem of graph modeling. As such, a core feature of the analysis in this article is that we deal with large networks. Over these networks, due to different forms of physical limitations, it is not practical to assume that data can be collected from all agents. This is seldom the case and our standing assumption in this presentation will be that observations are collected from only a subset of the agents. We refer to this scenario as the partial observation regime. As a result, the graph learning task becomes more complicated than usual, since the observations collected at the monitored nodes are influenced (through information propagation)
by the unobserved (or latent) nodes. It is then natural to inquire whether this partial-observation setting leads to an ill-posed inverse problem or can still provide sufficient information to learn the underlying graph linking the observed agents. In particular, we are interested in establishing whether the topology of the monitored subnetwork can be consistently retrieved despite the (massive, since the network is large) influence of the latent (unobservable) part. It should be clear that this is a hard problem, which will be not feasible in general. It should also be clear that the feasibility of the problem (i.e., achievability of consistent graph learning), the type and complexity of the estimation, will be dependent on the type of dynamical evolution that describes the network. In other words, we will not be focusing on graph learning for arbitrary multi-agent graphs. Rather, we will be focusing on graph learning for networks involved in the solution of a distributed optimization task. This is relevant because, to operate properly, many optimization networks place constraints on the exchange of information, and these constraints usually translate into regularity conditions on the combination matrix used to weight the signals exchanged by the agents. In this article we survey recent results on graph learning under partial observations and regular combination matrices. One fundamental conclusion arises: consistent graph learning is possible for large optimization networks running first-order diffusion models and relying on popular combination matrices. Learning is possible despite the presence of a massive latent (unobservable) component of the network.

**Notation.** We use boldface letters to denote random variables, and normal font letters for their realizations. Matrices are denoted by capital letters, and vectors by small letters. This convention can be occasionally violated, for example, the total number of network nodes is denoted by $N$. A random vector $\mathbf{x}$ that depends on a spatial (i.e., agent) index $k$ and a time index $i$ will be denoted by $\mathbf{x}_{k,i}$. A scalar random variable that depends on a spatial index $k$ and a time index $i$ will be denoted by $x_{k}(i)$.

The symbol $\overset{p}{\to}$ denotes convergence in probability as $N \to \infty$. When we say that an event occurs “w.h.p.” we mean that it occurs “with high probability” as $N \to \infty$.

Sets and events are denoted by upper-case calligraphic letters, whereas the corresponding normal font letter will denote the set cardinality. For example, the cardinality of $S$ is $\vert S \vert$. The complement of $S$ is denoted by $S'$. For a $K \times K$ matrix $Z$, the submatrix spanning the rows of $Z$ indexed by set $S \subseteq \{1,2,\ldots,K\}$ and the columns indexed by set $T \subseteq \{1,2,\ldots,K\}$, is denoted by $Z_{ST}$, or alternatively by $[Z]_{S:T}$. When $S = T$, the submatrix $Z_{SS}$ is abbreviated as $Z_{S}$. The symbol $\log$ denotes the natural logarithm.
II. EXAMPLES OF OPTIMIZATION NETWORKS

A. Distributed Stochastic Gradient

A relevant class of distributed optimization problems can be described through a unifying formulation that we are going to illustrate briefly [14], [24], [25]. Let us consider a network with $N$ agents, and assume that the objective of the network is the following optimization problem:

$$\min_{w \in \mathbb{R}^M} J(w), \quad J(w) \triangleq \sum_{k=1}^{N} J_k(w), \quad J_k(w) \triangleq \mathbb{E}[Q_k(w; x_k)]$$  (1)

In (1), the index $k = 1, 2, \ldots, N$ denotes the individual agent, the quantity $J_k(w)$ is an individual cost function, and $J(w)$ is the global cost function. Each agent collects some random data $x_k$, and the cost function is assumed to be expressible as the mean (i.e., $\mathbb{E}[-]$ denotes the expectation operator) of a certain loss function, $Q_k(w; x_k)$. For the purpose of this motivating example, it is sufficient to assume that the individual cost functions satisfy classic smoothness conditions (e.g., that they are $\nu$-strongly and first-order differentiable convex functions [14]). We denote the global minimizer by

$$w^o \triangleq \arg \min_w J(w).$$  (2)

Under the smoothness conditions, minimization can be safely attained via the steepest descent algorithm, i.e., by updating estimates for $w^o$ along the negative direction of the overall gradient, $\nabla J(w) = \sum_{k=1}^{N} \nabla J_k(w)$ [26], [27]. Unfortunately, it is seldom the case that the agents have sufficient knowledge to evaluate the true gradient, and, for this reason, each agent will use instead some instantaneous approximation of the gradient, which is typically obtained by taking the gradient of the (actual, rather than averaged) loss function:

$$\nabla J_k(w; x_k) \triangleq \nabla Q_k(w; x_k)$$  (3)

Moreover, the agents can approach the solution of (2) by means of a distributed strategy. At time instants $i = 0, 1, \ldots$, every individual agent $k = 1, 2, \ldots, N$ collects streaming data vectors $\{x_{k,i}\}$, and manages to compute a state vector $w_{k,i}$ to approximate the global minimum $w^o$. Several distributed implementations have been proposed to perform multiagent optimization [1]–[9]. The most popular algorithms underlying these optimization strategies are consensus [10], [11], gossip algorithms [12], [13], or diffusion algorithms [14], [24], [28], [29]. In this article we will focus on a diffusion implementation known as Combine-Then-Adapt (CTA), which is particularly suited for learning from streaming data.

The CTA algorithm evolves by iterating the following two steps for every time $i$. First, during the combination step, every agent $k$ computes an intermediate vector $\psi_{k,i-1}$ as a weighted linear combination
(through some nonnegative combination coefficients \( \{c_{\ell k}\} \) of the states \( \{w_{\ell,i-1}\} \) coming from its neighbors at previous time \( i-1 \):

\[
\psi_{k,i-1} = \sum_{\ell=1}^{N} c_{\ell k} w_{\ell,i-1}, \quad \text{[combination step]}
\]  

(4)

We see from (4) that the structure of the combination matrix \( C = [c_{\ell k}] \) is critical in determining how agent \( k \) incorporates information coming from agent \( \ell \). In particular, the skeleton of \( C \) (i.e., the support graph given by the locations of the strictly positive entries of \( C \)) encodes the possible paths that the information can follow through during the evolution of the diffusion algorithm. It is often assumed that \( C \) is doubly stochastic, meaning that the entries on each of its rows and on each of its columns add up to one. Second, during the adaptation step, each agent uses its locally available current data \( x_{k,i} \) to update the intermediate state from \( \psi_{k,i-1} \) to the new state \( w_{k,i} \):

\[
w_{k,i} = \psi_{k,i-1} - \mu \nabla Q_k(\psi_{k,i-1}; x_{k,i}), \quad \text{[adaptation step]}
\]  

(5)

In (5), the positive scalar \( \mu \) is the step-size parameter and is usually kept small. The adaptation step has the purpose of taking into account the effect of the streaming data \( x_{k,i} \). Under appropriate conditions, the CTA algorithm is able to solve the optimization problem, in the sense that the individual states of all agents, \( w_{k,i} \), will approach a small neighborhood of size \( O(\mu) \) around the desired solution \( w^* \), provided that sufficient time for learning is given. A fundamental tradeoff arises: the smaller \( \mu \) is, the smaller the size of the oscillations around the true solution (more accurate learning), and the longer the time it takes to attain a given accuracy (slower adaptation) [28], [29].

Formulations similar to (4)–(5) form the basis of several other distributed optimization algorithms. It was shown in prior work that such updates can emulate behavior by some biological networks [30], [31], such as the movements of a school of fish evading predators. Adjacent members of the school exchange continually information by means of local interactions, giving rise to a diffusion mechanism that can be modeled through the aforementioned distributed optimization algorithm. The objective of the optimization is to estimate the predator location. Moreover, since the predator can vary its position, the fish school must be reactive in order to track drifts in the predator position, and for this reason the diffusion mechanism must be inherently adaptive. The topology of interactions plays a fundamental role in the way the information propagates and in the capacity of the fish school to evade the predator attack.

### B. Distributed Detection

A second relevant application of distributed optimization is distributed detection [32], [33]. We are given a collection of streaming data \( \{x_{k,i}\} \), where \( k \) and \( i \) are agent and time indices, respectively. The
data are both spatially and temporally independent and identically distributed (i.i.d.) according to two possible mutually exclusive hypotheses: the null hypothesis \( H_0 \) and the alternative hypothesis \( H_1 \), which correspond respectively to probability functions \( \pi_0 \) and \( \pi_1 \) (densities for continuous variables, or mass functions for discrete variables).

It is useful to relate the distributed detection application to the general formulation employed in Sec. II-A. We start from a well known result: for the considered i.i.d. model, in the classic centralized scenario, the optimal way to perform detection is to compare against a threshold the sum (across \( k \) and \( i \)) of the log-likelihood ratios. Motivated by this observation, we will attempt to choose the individual costs \( J_k(w) \) in (1) in such a way that the CTA algorithm is able to mimic a distributed computation of the log-likelihood ratio statistic. To this end, we set (here \( w \) needs to be scalar):

\[
Q_k(w; x_{k,i}) = \frac{1}{2} \left( w - \log \frac{\pi_1(x_{k,i})}{\pi_0(x_{k,i})} \right)^2.
\]

The corresponding CTA algorithm will be:

\[
\begin{align*}
\mathbf{w}_{k,i} &= \sum_{\ell=1}^{N} c_{\ell k} \mathbf{w}_{\ell,i-1} - \mu \nabla Q_k \left( \sum_{\ell=1}^{N} c_{\ell k} \mathbf{w}_{\ell,i-1}; \mathbf{x}_{k,i} \right) \\
&= (1 - \mu) \sum_{\ell=1}^{N} c_{\ell k} \mathbf{w}_{\ell,i-1} + \mu \log \frac{\pi_1(\mathbf{x}_{k,i})}{\pi_0(\mathbf{x}_{k,i})},
\end{align*}
\]

Equation (7) can be conveniently cast into vector form by introducing the \( N \times 1 \) vectors:

\[
\mathbf{z}_i = \begin{bmatrix} 
\mu \log \frac{\pi_1(x_{1,i})}{\pi_0(x_{1,i})}, & \ldots, & \mu \log \frac{\pi_1(x_{N,i})}{\pi_0(x_{N,i})}
\end{bmatrix}^T,
\]

yielding:

\[
\mathbf{w}_i = (1 - \mu) \mathbf{C}^T \mathbf{w}_{i-1} + \mathbf{z}_i.
\]

Evaluating now from (6) the exact (scalar) gradient we get:

\[
\nabla J_k(w) = \mathbb{E}[\nabla Q_k(w; x_{k,i})] = w - \mathbb{E} \left[ \log \frac{\pi_1(x_{k,i})}{\pi_0(x_{k,i})} \right].
\]

Computing the expectation under the different hypotheses we obtain:

\[
\begin{align*}
\nabla J_k(w) &= w + \mathcal{D}_{01} \quad \text{under } H_0, \\
\nabla J_k(w) &= w - \mathcal{D}_{10} \quad \text{under } H_1,
\end{align*}
\]

where \( \mathcal{D}_{hj} \) denotes the Kullback-Leibler (KL) divergence between hypothesis \( h \) and hypothesis \( j \), for \( h, j = 0, 1 \). We see from (11), (12) that the minimum of the true gradient is attained at value \(-\mathcal{D}_{01}\) under \( H_0 \), and value \( \mathcal{D}_{10} \) under \( H_1 \). We conclude that a distributed implementation such as the CTA algorithm would allow each agent to fluctuate (for sufficiently small \( \mu \)) around the negative value.
−D_{01} under H_0, and around the positive value D_{10} under H_1. In practice, effective discrimination between the hypotheses can be attained through a decision rule that compares the output of the optimization routine against a threshold γ ∈ (−D_{01}, D_{10}). Observe that in the distributed detection case, the adaptation/learning tradeoff is also governed by the step-size µ. As µ becomes smaller, at the price of slower adaptation, we reach an increasingly large detection precision, with error probabilities scaling exponentially fast with 1/µ — see [35], [36] for a detailed asymptotic analysis.

Remarkably, there is yet another useful interpretation of the distributed detection algorithm in terms of a distributed optimization algorithm. As a matter of fact, after the threshold comparison, the individual agents are able to converge to the true underlying hypothesis that governs the data. As a result, they are solving the following optimization problem: they are choosing, with probability that tends to one as µ goes to zero, the distribution (π_0 or π_1) that minimizes the KL distance with the empirical data.

C. Social Learning

We illustrate a third relevant application of distributed optimization in the context of social networks. In the classical (non-Bayesian) social learning paradigm [37], there is a set of admissible hypotheses θ ∈ Θ. The agents of a social network gather some private data \{x_{k,i}\} (distributed according to a certain probability function f_k) that they do not want to share. They are instead open to sharing their own belief (i.e., credibility) associated with each possible hypothesis. The beliefs of agent k are stored in a belief vector μ_k(θ), for k = 1, 2, ..., N and θ ∈ Θ. Accordingly, the agents implement a social learning algorithm that operates as follows. Agent k at time i computes some local likelihood L_k(x_{k,i}|θ) relying solely on its private data x_{k,i}. Then, agent k computes an intermediate belief \(ψ_{k,i}(θ)\) through the following update rule [38]–[40]:

\[
ψ_{k,i}(θ) = \frac{\mu_{k,i-1}(θ) L_k(x_{k,i}|θ)}{\sum_{θ' ∈ Θ} \mu_{k,i-1}(θ') L_k(x_{k,i}|θ')}
\]

Subsequently, during a social interaction stage, each agent exchanges with its neighbors these intermediate beliefs, which are then combined as [39], [40]:

\[
μ_{k,i}(θ) = \frac{\exp \left\{ \sum_{ℓ=1}^{N} c_{ℓk} \log ψ_{ℓ,i}(θ) \right\}}{\sum_{θ' ∈ Θ} \exp \left\{ \sum_{ℓ=1}^{N} c_{ℓk} \log ψ_{ℓ,i}(θ') \right\}}
\]

Exploiting Eqs. (13) and (14), it is straightforward to obtain the following recursion, for any θ ≠ θ':

\[
\log \frac{ψ_{k,i}(θ)}{ψ_{k,i}(θ')} = \sum_{ℓ=1}^{N} c_{ℓk} \log \frac{ψ_{ℓ,i-1}(θ)}{ψ_{ℓ,i-1}(θ')} + \log \frac{L_k(x_{k,i}|θ)}{L_k(x_{k,i}|θ')},
\]

December 19, 2019 DRAFT
which, upon introducing the $N \times 1$ vectors:

$$w_i = \begin{bmatrix} \log \frac{\psi_{1,i}(\theta)}{\psi_{1,i}(\theta')} , \ldots, \log \frac{\psi_{N,i}(\theta)}{\psi_{N,i}(\theta')} \end{bmatrix}^\top,$$

$$z_i = \begin{bmatrix} \log \frac{L_1(x_{1,i}|\theta)}{L_1(x_{1,i}|\theta')} , \ldots, \log \frac{L_N(x_{N,i}|\theta)}{L_N(x_{N,i}|\theta')} \end{bmatrix}^\top,$$

is conveniently represented in vector-matrix form as:

$$w_i = C^\top w_{i-1} + z_i. \quad (17)$$

Developing the recursion in (17), it is possible to show that the aforementioned algorithm possesses some interesting convergence properties [39]–[41]. In particular, let us introduce the following averaged KL divergence

$$\mathcal{D}_k(\theta) \triangleq \sum_{\ell \in S} [C^\infty]_{\ell k} D[f_\ell||L_\ell(\theta)] \quad (18)$$

where $C^\infty$ denotes the limit (supposed to exist) of the power matrix $C^i$ as $i \to \infty$. It can be shown that the belief of the individual agent $k$ collapses to 1 at the value [41]:

$$\theta^*_k \triangleq \arg \min_{\theta \in \Theta} \mathcal{D}_k(\theta) \quad (19)$$

In other words, agent $k$ is able to minimize the average divergence $\mathcal{D}_k(\theta)$.

This behavior has some useful implications. For example, when the network is strongly connected, and both the likelihoods and true distribution are equal across the network (say, $L_k = L$ and $f_k = f$), the agents are finding the likelihood $L(\cdot|\theta^*)$ that provides the best approximation (in KL divergence) to the true distribution $f$ [39], [40]. Another relevant application is that of weakly-connected networks [42], [43]. These networks arise frequently over social platforms, and have a highly asymmetric structure featuring sending subnetworks and receiving subnetworks, with the latter being prevented from sending data to the former. Over these networks, it is possible to show that the final beliefs are ruled only by the opinion promoted by the sending subnetworks. When the sending subnetworks have conflicting objectives, the final opinion chosen by a particular agent depends critically on the way the information of the different sending subnetworks percolates. For this reason, a useful interplay emerges between the graph learning and the social learning problem, and estimating the topology of interaction becomes critical [20].

III. DYNAMICS MODEL

In the previous section we motivated the problem of inverse graph learning through some examples. We are now ready to abstract the problem and obtain a formal model that will be useful to illustrate the
main results. We are given a connected network of $N$ agents, which implement a distributed diffusion algorithm. The output of agent $k = 1, 2, \ldots, N$ at time $i \geq 0$ will be henceforth assumed to be a scalar random variable denoted by $w_k(i)$. For a given time instant, the outputs of all agents are stacked into an $N \times 1$ column vector:

$$w_i = [w_1(i), w_2(i), \ldots, w_N(i)]^\top. \quad (20)$$

Likewise, a second scalar random variable (such as the data or some function thereof) $\{z_k(i)\}$ will be stacked into a vector:

$$z_i = [z_1(i), z_2(i), \ldots, z_N(i)]^\top. \quad (21)$$

We assume that the data are i.i.d., both spatially and temporally. Motivated by the previous examples of Secs. II-B and II-C — see $(9)$ and $(17)$ — we now focus on the following diffusion model, a.k.a. first-order Vector AutoRegressive (VAR) model:

$$w_i = Aw_{i-1} + z_i \quad (22)$$

The model in $(22)$ is a simple example of distributed optimization, which will be nevertheless very useful to convey fundamental results and give useful insights into the challenging problem of inverse graph learning addressed in this work. We remark however that there is still lot of work to be done for extending the results presented in this survey to more general optimization settings where, e.g., the gradient in $(5)$ can assume some more general forms.

According to many distributed optimization settings, in the following treatment the matrix $A$ will be assumed to be nonnegative, symmetric, and a scaled (stable) version of a doubly-stochastic matrix, namely,

$$a_{\ell k} \geq 0, \quad A = A^\top, \quad \sum_{\ell=1}^N a_{\ell k} = \rho, \quad 0 < \rho < 1 \quad (23)$$

For the distributed detection example, the conditions in $(23)$ are automatically met by setting $A = (1 - \mu)C^\top$ in $(9)$, with $C$ being symmetric and doubly stochastic. For the social learning example, setting $A = C^\top$ in $(17)$ would not yield a stable $A$, since a doubly-stochastic matrix has maximum eigenvalue equal to 1. A stable matrix can be obtained by considering an adaptive implementation of the social learning algorithm, with the introduction of a constant step-size.

For ease of presentation, in the forthcoming treatment we will always assume, without loss of generality, that the random variables $\{z_k(i)\}$ are zero mean and unit variance. Finally, we introduce the limiting correlation matrix and the limiting one-lag correlation matrix, defined respectively as:

$$R_0 \triangleq \lim_{i \to \infty} \mathbb{E} \left[ w_i w_i^\top \right], \quad R_1 \triangleq \lim_{i \to \infty} \mathbb{E} \left[ w_i w_{i-1}^\top \right]. \quad (24)$$
IV. GRAPH LEARNING

In the graph learning problem, the main goal is to estimate the support graph of $A$ from observation of the agents’ output samples $w_i$ collected across time $i$. It is natural to believe that the statistical correlation between the output of two agents can provide an indication on whether they are connected or not. On closer reflection, one finds that over a connected network with cooperative agents, pairwise correlation between two agents is also affected by data streaming from other agents through the successive local interactions: agents interact with their neighbors, which in turn interact with their neighbors, and so forth. As a result, if agent $\ell$ is connected to $k$ through an intermediate agent $m$, the outputs of $\ell$ and $k$ will be correlated even if there is no direct link between them. Therefore, using correlations to deduce graph connections is not a reliable method over such multi-agent networks.

Indeed, it is not true in general that the graph is a function of pairwise correlations. This is true only for special networks that are called correlation networks, but many other possibilities exist. For example, in Gaussian a graphical model [44]: $i$) the measurements at the network nodes obey a multivariate normal distribution with a certain correlation matrix; and $ii$) the nonzero entries of the inverse of the correlation matrix (a.k.a. concentration matrix) correspond to the support graph of the network. But it should be remarked that even this result is not general enough, and that effective estimators for the graph must necessarily depend on how the form of the signal dynamics over the graph. Our presentation will help clarify these observations, as well as the challenges that arise from relying solely on partial observations.

Let us now focus on the diffusion model in (22). Multiplying both sides by $w_i^\top$ and taking expectations, we have straightforwardly:

$$
\mathbb{E} \left[ w_i w_i^\top \right] = A \mathbb{E} \left[ w_{i-1} w_{i-1}^\top \right] + \mathbb{E} \left[ z_i w_i^\top \right],
$$

(25)

where the last term is zero because the sequence $\{z_i\}$ is formed by independent and zero-mean random vectors. From (25) we immediately argue that the matrix $A$ can be expressed as a function of the correlation matrix $R_0$ and the one-lag correlation matrix $R_1$ defined in (24):

$$
A = R_1 R_0^{-1}.
$$

(26)

In particular, this solution can be interpreted as searching for the coefficients $\{a_{\ell k}\}$ that provide the best (in mean-square-error) linear prediction of $w_i$ given the past sample $w_{i-1}$ — see, e.g., [45]. In the context of Granger causality, this solution is also known as the Granger predictor or Granger estimator [46]. Equation (26) is relevant for graph learning because the correlation matrices can be estimated from samples, with increasingly large precision as the number of samples increases.
However, in order to evaluate $R_0$ and $R_1$, the solution in (26) requires probing the entire network. Accordingly, this solution is not useful under the partial observation setting adopted here. As a matter of fact, the multi-agent systems encountered in real-world applications are typically made of a large number of individual units. For this reason, it is often the case that only a limited subset of agents can be probed. We will denote the set of probed agents by $S$ (of cardinality $S$), and the set of unobserved or latent agents by $S'$ (of cardinality $S'$). The goal of the graph learning then becomes to estimate the support graph of the monitored portion, i.e., the support graph of $A_{S}$ (recall that this notation refers to restricting $A$ to the columns and rows defined by the indexes in $S$). One approach to estimate this subgraph could be by applying (26) to the submatrices corresponding to the probed set $S$. This approach would correspond to determining the coefficients $\{a_{\ell k}\}$ (for $\ell, k \in S$) that provide the minimum-mean-square-error linear prediction of the sub-vector containing the elements of $\{w_k(i)\}$ for $k \in S$, given the sub-vector of the past samples $\{w_k(i-1)\}$ for $k \in S$. Unfortunately, matrix analysis tells us that

$$A_S = [R_1 R_0^{-1}]_S \neq [R_1]_S [R_0]_S^{-1}$$

(27)

The middle term corresponds to extracting the $S$ component from the product $R_1 R_0^{-1}$, where the last component corresponds to first extracting the $S$ components from the individual correlations $R_1$ and $R_0$. The inequality sign is because the term $[R_1 R_0^{-1}]_S$ takes into account the effect of the latent agents in $S'$ before projection onto the set $S$. Therefore, a Granger predictor that ignores the latent variables is not necessarily good. In particular, the elementary result in (27) provides an immediate hint on the fact that the inverse graph learning problem is not necessarily feasible under partial observation.

### A. Main Issues in Graph Learning

It is useful to illustrate three fundamental issues arising in the context of graph learning.

**Feasibility.** The first fundamental issue of graph learning is to establish whether the problem is feasible. In other words, we want to establish whether the support graph of interest can be consistently retrieved, disregarding complexity constraints. In practice, we can assume that we can collect as many samples as desired, and that the computational complexity associated, e.g., with matrix inversion or search algorithms is not of concern. As an example, consider model (22) under full observation. From (26) we see that the problem is feasible, since there is a closed-form relationship that allows retrieving $A$ from $R_0$ and $R_1$, and since we assume the latter correlation matrices can be estimated perfectly from the data as the number of samples goes to infinity. In our partial observation setting, feasibility is a critical and challenging issue, due to the assumption that we can collect data from only a limited portion of the network, whereas the size of the unobserved network component scales to infinity.
Fortunately, we will show that it is possible to establish feasibility in the partial-observations regime. Even when this is done, there are at least two other aspects to be considered. These aspects are related to general practical constraints associated to the graph learning task, and are usually referred to as hardness and sample complexity.

**Hardness.** When examining hardness, we continue to disregard the complexity associated with the number of samples. That is, we continue to assume that an infinite volume of samples is available, such that the statistical quantities of interest are perfectly known. For instance, with reference to the model in (22), this amounts to saying that $R_0$ and $R_1$ are perfectly known. The concept of hardness is then related to the processing required to compute the support graph from $R_0$ and $R_1$. Hence, in this example hardness is due to the need to invert a large matrix. Hardness can be a serious issue since, e.g., in some graph learning problems an NP-search would be required to draw the estimated graph [48]–[52].

**Sample Complexity.** This concept refers to the number of samples that are required to perform accurate graph learning. The main questions related to this problem are: i) devising learning algorithms with reduced sample complexity; and ii) ascertaining the fundamental limits of the graph learning problem in terms of sample complexity, i.e., how the best attainable learning performance scales with the number of samples. In relation to these issues, most useful results are available in the context of graphical models. These models, as will be explained in the forthcoming section, do not natively match dynamical systems like the one in (22). Less is known for the latter category of systems, especially under the challenging regime of partial observations [22].

The results illustrated in this work focus on the feasibility issue, which becomes highly nontrivial with partial observations, due to the massive presence of unobserved agents implied by the large-network setting.

**B. Related Work on Graph Learning**

In this section we review briefly some relevant works on graph learning that have appeared in the recent literature. Owing to the nature of the system considered in (22), we will mainly focus on linear system dynamics, but hasten to add that there exist works on graph reconstruction over nonlinear dynamical systems as well [21], [53]–[58]. The first part of our summary deals with the problem of graph learning under full observation.

One useful work on topology inference over linear systems is [59], which considers a fairly general class of systems (including non-causal systems and VAR models of any order). The main contribution of [59] is to devise an inferential strategy relying on Wiener filtering to retrieve the network graph. Such strategy is shown to guarantee exact reconstruction for the so-called self-kin networks. For more arbitrary
network structures, the reconstruction of the smallest self-kin network embodying the true network is guaranteed.

In the context of graph signal processing [60]–[64], recent works focused on autoregressive diffusion models of arbitrary order [65]–[67]. As a common feature of many of these works, the proposed estimation algorithms leverage some prior knowledge on the graph structure, which is then translated into appropriate structural constraints. Typical constraints are in terms of sparsity of the connections, or smoothness (in the graph signal terminology) of the signals defined at the graph nodes. In [65], a two-step inferential process is proposed, where: i) a graph shift operator [68]–[70] is estimated through the agents’ signals that arise from the diffusion process; and ii) given the spectral templates obtained from this estimation, the eigenvalues that would identify the graph are then estimated by adding proper structural constraints (e.g., sparsity) that could render the problem well-posed. In [66], the same concept of a two-step procedure is considered, with the main goal being to characterize the space of valid graphs, namely, the graphs that can explain the signals measured at the network agents. In [67], a model for causal graph processes is proposed, which exploits both inter-relations among agents’ signals and their intra-relations across time. Capitalizing on these relations, a viable algorithm for graph structure recovery is designed, which is shown to converge under reasonable technical assumptions.

It is worth mentioning that there also exist works on graph learning over dynamical systems other than the linear diffusion systems considered so far. In [71], a graphical model is proposed to represent networks of stochastic processes. Under suitable technical conditions, it is shown that such graphs are consistent with the so-termed directed information graphs, which are based on a generalization of Granger causality. It is also proved how directed information quantifies Granger causality in a specific sense and efficient algorithms are devised to estimate the topology from the data. In [72], a novel measure of causality is introduced, which is able to capture functional dependencies exhibited by certain (possibly nonlinear) network dynamical systems. These dependencies are then encoded in a functional dependency graph, which becomes a representation of possibly directed (i.e., causal) influences that are more sophisticated than the classical types of influences encoded in linear network dynamical systems.

In summary, the aforementioned works (which we list with no pretense of exhaustiveness) address under various settings the problem of feasibility and complexity of graph learning under the full observation regime. From these works we observe the following. The general procedure amounts to: i) computing some empirical covariance from the time-series diffusing across the graph; and ii) identifying the candidate solutions for the support graph of the diffusion matrix by leveraging relationships between the covariance matrices and the diffusion matrix. Then, some constraints are usually necessary to reduce the number of candidate solutions, i.e., to render the problem feasible, and also to reduce the complexity...
in the algorithmic search. We will follow a similar line of reasoning in our work — see Fig. 1. For example, and as explained before, the estimator in (26) is motivated by such an approach. However, we must recall that in our setting we focus on the partial observation setting where an infinitely large portion of the network is not accessible. Most challenges in terms of feasibility of the graph learning problem will in fact stem from this additional complication.

C. Graph Learning under Partial Observations

In the presence of unaccessible network agents, there are results allowing proper graph learning when the topology is of some pre-assigned type (polytrees) [73], [74]. For fairly arbitrary graph structures, some results about the possibility of correct graph retrieval are provided in [75], [76]. One limitation of these results resides in the fact that the sufficient conditions for graph learning depend on some “microscopic” details about the model (e.g., about the local structure of the topology or the pertinent statistical model). For this reason, over large-scale networks, which are the focus of our analysis, a different approach is necessary.

A promising approach over large networks can be based on an asymptotic analysis carried out as the network size $N$ scales to infinity. In order to cope with the large network size in a way that enables a tractable analysis, we will model the network graph as a random graph. An asymptotic analysis can then become affordable, letting emerge the thermodynamic properties of the graph, with the conditions for graph learning being summarized in some macroscopic (i.e., average) indicators, such as the probability that two agents of the random graph are connected.

Similar forms of asymptotic analysis were recently performed for high-dimensional graphical models with latent variables. In [77], the focus is on Gaussian graphical models, and consistent graph learning is proved (along with a viable algorithmic solution) under an appropriate local separation criterion. In [78] results of consistent learning are instead provided for locally-tree graphs. Graph learning under the so termed “sparsity+low-rank” condition is examined in [79]. Under this condition (where the observed subnetwork is sparse and the unobserved subnetwork is low-rank in an appropriate sense), it is proved that the graph and the amount of latent variables can be jointly estimated. In [80], a graphical model consisting of a ferromagnetic restricted Boltzmann machine with bounded degree is considered. It is shown that such class of graphical models can be effectively learned through the usage of a novel influence-maximization metric.

However, classical graphical models (such as the ones used in the aforementioned references) do not assume that there are signals evolving over time at the network nodes. In contrast, classical graphical models assume a still picture of the network, where the data measured at the individual agents are modeled
as random variables characterized by a certain joint distribution. The inter-agent statistical dependencies
are encoded in the joint distribution through an underlying graph. Under this framework, estimation of
the graph from the data defined at the nodes is performed assuming that the inferential engine has access
to i.i.d. samples of these data, and there is no model of the evolution of the data across time.

For this reason, the results obtained in the aforementioned references on graph learning in the presence
of latent variables do not apply to the dynamical system considered in (22). Moreover, given the specific
constraints arising from our setting (e.g., regularity of the combination matrices used for the distributed
optimization algorithm), more powerful results can be obtained. For instance, we will see that, different
from what happens in other contexts, our problem can become feasible also for densely connected
networks. For these models, results for graph learning under partial observation have been recently
obtained in [81]–[88]. In the following, we will summarize these recent advances in some detail.

V. RANDOM GRAPH MODEL

As explained in the previous section, over large networks it is necessary to perform some asymptotic
analysis to obtain useful analytical results, and to establish the fundamental thermodynamic properties that
emerge with high probability over the network. One typical way to tackle this problem is to randomize the
network structure, i.e., to work with random graphs. One useful class of random graphs is the celebrated
model proposed by Erdős and Rényi [89], [90], which is an (undirected) graph where the probability
that nodes $\ell$ and $k$ are connected is a Bernoulli random variable characterized by a certain connection
probability $p$, and where all edges are drawn independently and with the same connection probability.

An important graph descriptor is the degree of a node. The degree of node $i$ is the number of its
neighbors (including node $i$ itself), and is denoted by $d_i$. Owing to the Bernoulli model, the average
degree $D_{av}$ of every node of the Erdős-Rényi (ER) graph is equal to $1 + (N - 1)p$.

Let us examine the evolution of the random graph when $N$ grows. When the connection probability
is a constant $p > 0$, the number of neighbors increases linearly with $N$ (in the following, the notation $\sim$
means “scales as”, when $N \to \infty$):

$$D_{av} \sim Np \quad \text{[dense regime]}$$

(28)

It is not difficult to figure out that, since in this case any node has a number of neighbors growing
as $N$, the graph exhibits a dense connection structure, and for sufficiently large $N$, is likely to be a
connected graph, i.e., a graph where there always exists an (undirected) path connecting any pair of
nodes. However, a fundamental result of random graph theory states that, in order to ensure a graph is
connected with high probability as $N$ grows, it is sufficient that the average degree grows logarithmically with $N$, formally [89], [90]:

$$D_{av} = \log N + O(\log N) \quad \text{[log-sparse regime]}$$

(29)

where $O(\log N)$ is a sequence diverging to $+\infty$ at most logarithmically and, hence, the connection probability $D_{av}/N$ vanishes. The logarithmic growth corresponds in fact to a phase transition, since it represents the minimal growth that ensures a connected graph.

There is yet a third (sparse) connected regime, which is intermediate between the log-sparse and the dense regimes introduced so far. This intermediate regime occurs when the average degree grows faster than logarithmically (while the connection probability still vanishes), formally when:

$$D_{av} = \omega_N \log N \quad \text{[intermediate-sparse regime]}$$

(30)

where $\omega_N \to \infty$ in an arbitrary fashion, but sufficiently slow so as to ensure that the connection probability $D_{av}/N$ vanishes.

There is one fundamental property that holds under the sparse and dense regimes, but not under the log-sparse regime, and is the following statistical concentration of the minimal and maximal degrees of the graph:

$$\frac{d_{\min}}{D_{av}} \xrightarrow{p} 1, \quad \frac{d_{\max}}{D_{av}} \xrightarrow{p} 1 \quad \text{[uniform degree concentration]}$$

(31)

This means that, under (30), the minimal and maximal degree concentrate around the expected degree.

The overall taxonomy comprising the different elements of sparsity, density, and degree concentration, is reported in Fig. 2 along with an example of evolution, as $N$ grows, of the ER graphs corresponding to the different regimes. For each regime, we consider an ER graph of increasing size ($N = 25, 100, 1000$), and for each value of $N$ we display the behavior of a subgraph (for clarity of visualization) of fixed cardinality $S = 25$. For all regimes we start with a connection probability equal to $1/2$. Accordingly, the top panels have similar shape. Then, as $N$ increases, the connection probability obeys the scaling law relative to the particular regime. In the leftmost panels (sparse regime), we see that the displayed subgraph becomes progressively sparser.

1We note that the term “concentration” does not refer to the number of node connections, but, according to a standard terminology adopted in statistics, will be used to refer to statistical quantities that collapse to some deterministic value as $N \to \infty$ [91].

2We remark that the overall graph, which is too large to be displayed, remains connected even if the shown subgraph becomes progressively disconnected. In fact, on the overall graph with $N$ nodes, we can leverage the increasing number of nodes to find a path between any two nodes (with high probability) provided that the connection probability scales appropriately.
s some more structure is preserved. Finally, in the rightmost panels (dense regime), the subgraph has an invariant behavior.

We see that the union of the log-sparse and intermediate-sparse regimes identifies the sparse (as opposed to the dense) regime. Likewise, the union of the intermediate-sparse and dense regimes identifies the regime of uniform degree concentration.

A. Partial observation settings

The main challenge of the graph learning problem considered in this article is related to the partial observation setting, where only a subset $S$ of the network can be probed. In order to deal with the
Fig. 3. Partial observation settings considered in this work. The probed agents forming the subgraph of interest are highlighted in blue. The evolution for the plain ER regime is illustrated in the top panels. Here, as the network size grows (from $N = 20$ to $N = 40$) the number of probed agents grows as well, with the fraction of probed agents $\xi = S/N = 0.2$ staying constant, and with the subgraph of probed nodes varying. The partial ER regime is illustrated in the bottom panels. Here the number of probed agents stays constant ($S=4$) as the network size grows, and the structure of the probed subgraph is deterministically fixed.

asymptotic regime, it is necessary to define how the cardinality $S$ scales with the overall network size $N$. In particular, we introduce the asymptotic fraction of probed nodes $\xi$:

$$\frac{S}{N} \xrightarrow{N \to \infty} \xi$$

(32)

The extreme case where the cardinality of probed nodes is fixed while $N \to \infty$ corresponds to a low-observability regime ($\xi = 0$) where the subset of unaccessible nodes becomes dominant and infinitely larger than the subset of accessible nodes. However, when the size of $S$ is fixed and finite, it is not useful to model the connections within $S$ through an ER model, because in the sparse regime every edge in $S$ would trivially disappear as $N$ gets large!

In order to deal with the graph learning problem under the low-observability regime in a meaningful fashion, the following partial ER model was introduced in [82]: i) the subgraph of interest, $S$, is deterministic and arbitrary; ii) the latent nodes act like a noisy disturbance, in that the connections outside $S$, and between $S'$ and $S$, are drawn according to an ER model.

The distinction between the plain and the partial ER models is illustrated in Fig.3. In the top panels, a plain ER model with $\xi = 0.2$ is considered. We see that the subset $S$ of probed nodes (displayed in blue) increases from $S = 4$ to $S = 8$ when $N$ increases from 20 to 40. Moreover, the subgraph associated to $S$ (as well as the overall graph) changes randomly its shape according to an ER model. In comparison, the
partial ER model is displayed in the bottom panels. In this case the subset $S$ of probed nodes has fixed cardinality and structure. The edges (displayed in gray) between nodes belonging to the unobserved set $S'$, as well as between $S'$ and $S$, are randomly drawn according to an ER model.

B. Combination Matrices

The combination rule plays a critical role in the distributed optimization process, since it sets the way by which agents aggregate the information exchanged with their neighbors. Obviously, a combination matrix depends heavily on the underlying graph of connections, since the combination weights corresponding to disconnected agents’ pairs are necessarily zero. In this work we will impose some regularity conditions on the combination matrices, and focus in particular on two classes of combination matrices, which are now introduced. We will always assume that (23) holds true.

Assumption 1 (Class $C_1$). The nonzero entries of the combination matrix, scaled by the average degree $D_{av}$, do not vanish, namely, given that $\ell$ and $k$ are connected, a certain $\tau > 0$ exists such that, with high probability for large $N$:

$$a_{\ell k} > \frac{\tau}{D_{av}}$$  \hspace{1cm} (33)

Condition (33) is motivated by the following observation. For typical choices for the combination matrices, agent $\ell$ will distribute the overall available weight mass across its neighbors, in a rather homogeneous way. Thus, we will have typically, over connected pairs $(\ell, k)$, the following approximate proportionality:

$$a_{\ell k} \propto \frac{1}{D_{av}},$$  \hspace{1cm} (34)

which explains why the quantity $D_{av}a_{\ell k}$ does not vanish, and why condition (33) is meaningful.

Assumption 2 (Class $C_2$). We assume that, for connected pairs $\ell$ and $k$:

$$\frac{\kappa}{d_{max}} \leq a_{\ell k} \leq \frac{\kappa}{d_{min}}$$  \hspace{1cm} (35)

for some $\kappa$, with $0 < \kappa \leq \rho$.

We see from (35) that, when an edge exists linking $\ell$ to $k$, the variation of the (nonzero) matrix entries is defined in terms of the (reciprocal of the) maximal and minimal graph degrees. Also this condition can be motivated by the observation that the agents tend to distribute the weights across their neighbors in some homogeneous way. However, it is possible to show that, under the connectivity regimes for the ER model considered here, the leftmost inequality in (35) implies (33), namely, we can conclude that (81):

$$C_2 \subset C_1,$$  \hspace{1cm} (36)
i.e., that the conditions for a matrix to be in class $C_2$ are more stringent than the conditions required to be in class $C_1$.

As a matter of fact, the most popular combination matrices used in distributed optimization belong to class $C_2$ and, hence, to $C_1$. Two notable instances are the Laplacian and the Metropolis combination rules, which can be defined as follows. For $\ell \neq k$, with $\ell$ and $k$ connected:

$$a_{\ell k} = \frac{\rho \lambda}{d_{\text{max}}}, \quad \text{[Laplacian rule]}$$

$$a_{\ell k} = \frac{\rho}{\max\{d_i, d_j\}}, \quad \text{[Metropolis rule]}$$

For both rules, the self-weights are determined by the leftmost condition in (35), yielding $a_{kk} = \rho - \sum_{\ell \neq k} a_{\ell k}$. For the Laplacian rule, the parameter $\lambda$ fulfills the inequalities $0 < \lambda \leq 1$.

**VI. CONSISTENT GRAPH LEARNING**

In the following, the term “consistency” refers to the possibility of learning the graph correctly as $N \to \infty$. We will see that different notions of consistency are possible. We start from the weakest one.

We denote by $\hat{A}_S$ a certain estimate of the combination (sub)matrix corresponding to the subset $S$. We explain in the next section several ways by which such an estimate can be computed. The consistency result presented in the forthcoming theorems holds for *undirected* graphs and symmetric combination matrices. This notwithstanding, it is useful to formulate the general theory (e.g., the estimators and related metrics) to handle also *directed* graphs and asymmetric combination matrices. For this reason, when we refer to agent pairs we will actually refer to *ordered* pairs, with $(\ell, k)$ being distinct from $(k, \ell)$, because a *directed* link could in principle exist from $k$ to $\ell$ and not vice versa.

Next we introduce a general thresholding rule to classify connected/disconnected pairs. We will declare that the ordered $(\ell, k)$ pair is connected (i.e., that the $(\ell, k)$-th entry of the *true* combination matrix is nonzero) if the corresponding *estimated* matrix entry, $\hat{a}_{\ell k}$, exceeds some threshold $\tau$. Accordingly, let us introduce the following error quantities:

$$E_0(\tau) \triangleq \frac{\text{no. of entries where } a_{\ell k} = 0 \text{ and } \hat{a}_{\ell k} > \tau}{\text{no. of entries where } a_{\ell k} = 0},$$

$$E_1(\tau) \triangleq \frac{\text{no. of entries where } a_{\ell k} > 0 \text{ and } \hat{a}_{\ell k} \leq \tau}{\text{no. of entries where } a_{\ell k} > 0},$$

(38)
where we assume $(\ell, k) \in S$ with $i \neq j$. More informally, Eqs. (38) can be rephrased as:

\[
\begin{align*}
\mathcal{E}_0(\tau) & \triangleq \frac{\text{no. of mistakenly classified disconnected pairs}}{\text{no. of disconnected pairs}}, \\
\mathcal{E}_1(\tau) & \triangleq \frac{\text{no. of mistakenly classified connected pairs}}{\text{no. of connected pairs}}.
\end{align*}
\]  

(39)

**Definition 1** (Weak Consistency). We say that the subgraph in $S$ can be learned weakly if there exist $\hat{A}_S$ and $\tau$ such that:

\[
\mathcal{E}_0(\tau) + \mathcal{E}_1(\tau) \xrightarrow{p} 0
\]  

(40)

The notion of consistency in (40) ensures that the average fraction of mistakenly classified edges goes to zero. Obviously, when the cardinality $S$ of probed agents is fixed (as in the low-observability regime with partial ER model), an average number of mistakes that goes to zero implies that the subgraph of $S$ is perfectly recovered. In contrast, when the cardinality $S$ grows with $N$, ensuring a small average fraction of mistakes can be unsatisfactory, which motivates the qualification “weak”. Let us clarify this issue through a simple example. Consider a reconstruction that is perfect, except for 100 edges that are estimated by the learning algorithm but that are actually not present in the true graph. Clearly, the average number of mistakes ($100/S$) goes to zero as the subnetwork size $S$ goes to infinity, but due to the 100 spurious edges, we will never end up with perfect reconstruction. The presence of (even a small number of) spurious edges can be penalizing especially under the sparse regime, where the number of true edges is small, and a reconstructed network where the number of spurious edges is comparable with the number of true edges might be unsatisfactory.

From these observations, we argue that stronger notions of consistency are desirable. To this aim, we now introduce the useful concepts of **margins** and **identifiability gap** [83].

**Definition 2** (Margins). For a given matrix estimator $\hat{A}_S$, we introduce the lower and upper margins corresponding to the disconnected pairs:

\[
\delta_N \triangleq \min_{\ell, k \in S: a_{\ell k} = 0 \atop i \neq j} \hat{a}_{\ell k}, \quad \bar{\delta}_N \triangleq \max_{\ell, k \in S: a_{\ell k} = 0 \atop i \neq j} \hat{a}_{\ell k},
\]  

and the lower and upper margins corresponding to the connected pairs:

\[
\Delta_N \triangleq \min_{\ell, k \in S: a_{\ell k} > 0 \atop i \neq j} \hat{a}_{\ell k}, \quad \bar{\Delta}_N \triangleq \max_{\ell, k \in S: a_{\ell k} > 0 \atop i \neq j} \hat{a}_{\ell k}.
\]  

(41)

(42)
The physical meaning of the margins is to identify upper and lower bounds on the entries corresponding to agent pairs of a given type (i.e., connected/disconnected). For example, the lower and upper margins for the disconnected pairs identify a region (see Fig. 4) where we can find all the entries of the estimated matrix corresponding to the disconnected pairs. A similar interpretation holds for the connected pairs. Now, one could expect that a good estimator exhibits the desirable property that \( \hat{a}_{\ell k} \) goes to zero if agents \( \ell \) and \( k \) are not connected. While it is legitimate to ask for this property, a more careful analysis reveals that correct classification can occur even if, over disconnected pairs \((\ell, k)\), the entries \( \hat{a}_{\ell k} \) go to some nonzero value (i.e., if they have a bias). The important property to enable correct classification is that the region of disconnected pairs stays clear from the region of connected pairs, which means that some gap must exist between the lower margin over connected pairs and the upper margin over disconnected pairs. This observation leads naturally to the definitions of bias and gap, and to the associated concept of strong consistency.

**Definition 3 (Strong Consistency).** Let \( \hat{A}_S \) be an estimated combination matrix. If there exists a sequence \( s_N \), a real value \( \eta \), and a strictly positive value \( \Gamma \), such that, for a small \( \epsilon > 0 \):

\[
\begin{align*}
    s_N \overline{\delta}_N &< \eta + \epsilon & \text{w.h.p.} \\
    s_N \Delta_N &> \eta + \Gamma - \epsilon & \text{w.h.p.}
\end{align*}
\]

we say that \( \hat{A}_S \) achieves local structural consistency, with a bias at most equal to \( \eta \), an identifiability gap at least equal to \( \Gamma \), and with a scaling sequence \( s_N \). \( \square \)

We remark that the latter concept of consistency is strong because it entails the possibility of recovering asymptotically without errors the true graph in \( S \). In fact, comparing the scaled estimated matrix entries against some thresholds comprised between \( \eta \) and \( \eta + \Gamma \) (we neglected the small \( \epsilon \)), for sufficiently large \( N \), will end up with correct classification.

It is nevertheless evident from (43) that, in order to evaluate the classification threshold, certain system parameters should be known beforehand. For example, if \( s_N = D_{\text{av}} \), one should be able to predict the average number of neighbors to set a proper threshold. For this reason, in practical applications it could be very useful to have at disposal a blind data-driven mechanism (such as some clustering algorithm) to determine a proper threshold automatically from the data. We will use the qualification “universal” to

---

We see that the definition of consistency includes a scaling sequence \( s_N \). This scaling, which might look rather technical at first glance, admits a straightforward interpretation. For example, if we assume some homogeneity in the way the weights are distributed across the neighbors, the combination matrix entries scale roughly as \( 1/D_{\text{av}} \) and, hence, they vanish as \( N \to \infty \). Accordingly, it is necessary to scale them by \( s_N = D_{\text{av}} \) to get a stable asymptotic behavior.
denote these data-driven techniques. Accordingly, we can strengthen once more the notion of consistency to embody the universality requirement.

**Definition 4** (Universal strong consistency). Let $\hat{A}_S$ be an estimated combination matrix. If there exist a sequence $s_N$, a real value $\eta$, and a strictly positive value $\Gamma$, such that:

\[
\begin{align*}
s_N \delta_N & \overset{p}{\to} \eta, \\
s_N \Delta_N & \overset{p}{\to} \eta + \Gamma \\
s_N \delta_N & \overset{p}{\to} \eta, \\
s_N \Delta_N & \overset{p}{\to} \eta + \Gamma
\end{align*}
\]  

we say that $\hat{A}_S$ achieves uniform local structural consistency, with a bias $\eta$, an identifiability gap $\Gamma$, and with a scaling sequence $s_N$. \hfill \square

We see from (44) that the notion of universal strong consistency adds to the notion of strong consistency an inherent clustering ability. This is because the (scaled) lower margins, $s_N \delta_N$ and $s_N \delta_N$, converge to one and the same value, $\eta$, whereas the (scaled) upper margins, $s_N \Delta_N$ and $s_N \Delta_N$, converge to one and the same value, $\eta + \Gamma$. In light of this behavior, the estimated entries corresponding to disconnected pairs are squeezed to the bias $\eta$, and the estimated entries corresponding to connected pairs are squeezed to the higher value $\eta + \Gamma$, giving rise to two well-separated clusters that allow (asymptotically) faithful classification by means of a universal clustering algorithm — see Fig. 4.
VII. RELEVANT MATRIX ESTIMATORS

A general matrix estimator \( \hat{A}_S \) can be always written as:

\[
\hat{A}_S = A_S + E,
\]

where \( E \) is an error matrix. We see from the decomposition in (45) that there are two main ingredients to establish consistency for the graph learning problem. One is the asymptotic behavior of the true matrix \( A_S \) (how do its entries scale when \( N \) goes to infinity?). Assume that there is a scaling sequence \( s_N \) ensuring that the true entries over the connected pairs converge somewhere. Then, the asymptotic behavior of the error matrix \( E \) becomes critical. If the error (scaled by \( s_N \)) converges to zero, then we can hope to recover the true graph. But as it should be clear from the previous discussion, a nonzero error introduces some bias that does not necessarily impair graph learning.

In this article we consider three types of matrix estimators. Even if they are distinct, through the common description in (45) we will be able to recognize some commonalities in the way the error matrix behaves asymptotically. In particular, we will show that, for all the estimators, this matrix depends on suitable powers of the combination matrix, which is critical to determine the asymptotic properties of the error.

Preliminarily, we notice that the steady-state correlation and one-lag matrices in (24) can be evaluated in closed form as follows [14]:

\[
\begin{align*}
R_0 &= (I - A^2)^{-1}, \\
R_1 &= AR_0 = A(I - A^2)^{-1},
\end{align*}
\]

(46)

where \( I \) is the \( N \times N \) identity matrix, and where we remark that the bold notation for \( R_0 \) and \( R_1 \) is due to the randomness of the matrix \( A \), which inherits the randomness of the underlying ER graph.

A. Granger Estimator

The Granger estimator, as discussed in the introduction, is simply obtained by replacing (26) with its counterpart over the monitored subset \( S \), i.e., by accounting only for the probed nodes while neglecting the effect of the latent nodes:

\[
\hat{A}^{(\text{Gra})}_S = [R_1]_S ([R_0]_S)^{-1}.
\]

(47)

It is possible to show that the error of the Granger estimator can be written as [81]:

\[
E^{(\text{Gra})} = A_{SS'} (I_{S'} - [A^2]_{S'})^{-1} [A^2]_{S'S'}
\]

(48)

where \( I_{S'} \) is the submatrix of the \( N \times N \) identity matrix \( I \), relative to subset \( S' \).
B. One-Lag Correlation Estimator

Another possibility is to consider $[R_1]_S$ as an estimator for the combination matrix. One reason behind this choice is that (46) can be written as the matrix of interest, $A_S$, plus some higher-order powers of $A$, namely,

$$\hat{A}^{(1\text{-lag})}_S = [R_1]_S = A_S + [A^3]_S + [A^5]_S + \ldots$$  \hspace{1cm} (49)

It is convenient to rewrite (49) as:

$$\hat{A}^{(1\text{-lag})}_S = A_S + E^{(1\text{-lag})},$$  \hspace{1cm} (50)

which yields the following series for the error matrix:

$$E^{(1\text{-lag})} = \sum_{h=1}^{\infty} [A^{2h+1}]_S$$  \hspace{1cm} (51)

C. Residual Estimator

The (scaled) difference between consecutive time samples is sometimes referred to as the residual:

$$r_i \triangleq \frac{w_i - w_{i-1}}{\sqrt{2}}.$$  \hspace{1cm} (52)

Observing that $E[r_i r_i^\top] = R_0 - R_1 = (I_N + A)^{-1}$, we can introduce the matrix estimator:

$$\hat{A}^{(\text{res})}_S = [R_1]_S - [R_0]_S + I_S = A_S - [A^2]_S - [A^3]_S + \ldots,$$  \hspace{1cm} (53)

which implies that the error matrix for the residual estimator takes on the form:

$$E^{(\text{res})} = \sum_{h=1}^{\infty} ([A^{2h+1}]_S - [A^{2h}]_S)$$  \hspace{1cm} (54)

As anticipated, we see from (48), (51) and (54) that the error matrices associated with these three estimators admit a series representation.

The different behavior of these series gives rise to different characteristics of the graph learners. For example, if we compare (51) against (54), we observe that the latter error is in the form of an alternating series. It has been shown in [83], [88] that this feature can reduce the error and, hence, can improve the performance of the graph learner. This result is interesting for the following reason. While the Granger estimator gives the exact combination matrix under the full-observation regime, the residual and the one-lag estimators are always affected by an error. This notwithstanding, the simulations conducted in [83], [88] highlight that in the partial-observations regime the latter two estimators can outperform the Granger estimator.
VIII. RESULTS

In the recent literature, useful results for the problem of graph learning under partial observations have been obtained. Results for different settings are available, e.g., for different observability regimes (nonzero $\xi$ as opposed to the low-observability regime where $\xi = 0$) different connectivity regimes (sparse, dense, with uniform degree concentration) and/or different conditions on the combination matrices. Aside from very technical details, the bottom line of the ensemble of these results is that inverse graph learning under partial observations is possible, even if in different guises (e.g., with weak or strong consistency), also depending on which particular setting is considered.

Our objective is to present these results in some unified way. Accordingly, we find it appropriate to avoid “highly” technical details and focus instead on the main insights. For each result, we will direct the reader to the references where the technical details can be found.

A common feature underlying all the methods used to prove the forthcoming results is that all pertinent theorems have been proved working directly in the graph edge domain. This is in contrast with the mainstream approach of graph signal processing, where results are normally obtained in the graph spectral domain. We remark that the edge-domain analysis is beneficial to get a more immediate interpretation of the behavior of the matrix estimators, which is in turn useful to examine identifiability.

A. Results Under the Sparse Regime

The first result below was proved in [81], and can be stated as follows.

**Theorem 1** (Weak consistency under sparse ER model [81]). *If the combination matrix belongs to class $C_1$, for a sparsely-connected ER graph model where the fraction of probed nodes is $\xi > 0$, the Granger estimator achieves weak consistency.*

This result shows for the first time that inverse graph learning under partial observations is in fact possible. The result pertains to the sparse regime (either log-sparse or intermediate-sparse), and for the case where the number of probed nodes $S$ grows with $N$, giving rise to a strictly nonzero fraction of probed nodes $\xi$. The techniques used in [81] rely on basic matrix analysis tools.

As already explained after (40), the notion of weak consistency has some limitations. In order to overcome these limitations, a refined analysis is presented in [81] to examine the error rate, and to show that the number of edges introduced artificially by the estimation algorithm is asymptotically smaller than the number of true edges. However, this analysis relies on some approximation and, in any case, does not allow to conclude that all the subgraph of interest is perfectly reconstructed as $N \to \infty$. 

December 19, 2019
B. Low-Observability Regime

The results of Theorem 1 do not hold for the low-observability regime ($\xi = 0$). Such regime is considered in [82], which offers the following result of strong consistency (i.e., asymptotically perfect reconstruction).

**Theorem 2** (Strong consistency under low-observability [82]). Consider the low-observability regime ($\xi = 0$) where the subgraph in $S$ is deterministic and arbitrary, the cardinality $S$ is fixed, and the connection probability of the underlying ER model corresponds either to log-sparse regime or to an intermediate-sparse regime with:

$$\frac{(\log D_{av})^2}{\log N} \to 0$$  \hspace{1cm} (55)

Then, if the combination matrix belongs to class $C_1$, the Granger estimator achieves strong consistency. □

The techniques used in [82] to prove Theorem 2 differ substantially from the techniques used in [81]. Interestingly, these new techniques are based on suitable graph constructs and analysis of the asymptotic scaling of distances over the graphs that allow extending the results in [81] in two fundamental directions. First, the more challenging regime of low-observability is addressed, where the latent part becomes dominant (i.e., infinitely larger) than the monitored part. Second, the result proved in [82] provides the first result of exact reconstruction, since strong consistency is proved.

C. Results Under the Uniform Concentration Regime

The results of Theorems 1 and 2 pertain to the sparse regime. They seem to suggest that, according to a common belief, the sparsity of connections implies that the effect of the latent nodes can be somehow controlled and enable faithful graph learning.

However, in recent works [83], [87], [88], graph learning under partial observations has been examined under a different perspective, with the emphasis being put on the graph regime of uniform concentration. We recall that the regime of uniform concentration is neither simply sparse nor dense, since it is defined as the union of the intermediate-sparse regime and the dense regime. The following result has been proved.

**Theorem 3** (Universal strong consistency under uniform degree concentration [83]). Assume that the combination matrix belongs to class $C_2$. Under both the ER model with $\xi > 0$, or the partial ER model

$^4$Condition (55) poses a slight limitation on the growth-rate of $D_{av}$, which implies that the result holds provided that some minimal sparsity holds, i.e., it holds for a subset of the intermediate-sparse regime.
TABLE I

| Estimator    | Bias $\eta$                                      | Gap $\Gamma$ |
|--------------|--------------------------------------------------|--------------|
| Granger      | $\kappa^2 p \frac{(2\rho - \kappa)(1 - \xi)}{1 - (\rho^2 - 2\rho\kappa\xi + \kappa^2\xi)}$ | $\kappa$     |
| one-lag      | $\kappa^2 p \frac{\rho + \kappa \xi^2 + 2 \xi}{(1 - \rho^2)(1 - \xi^2)^2}$                  | $\kappa(1 + \xi^2)$ | $(1 - \xi^2)^2$ |
| residual     | $-\frac{\kappa^2 p}{(1 + \rho)(1 + \xi)^2}$                                              | $\kappa$ | $(1 + \xi)^2$ |

TABLE I

**Theorem 3** For all considered estimators, the scaling sequence is $s_N = D_{av}$, and we defined $\zeta = \rho - \kappa$. The parameter $p$ is the limiting connection probability, which is equal to zero in the sparse regime.

with low-observability ($\xi = 0$), the Granger, the one-lag, and the residual estimators achieve universal strong consistency, with a scaling sequence equal to $D_{av}$, and with the error biases and identifiability gaps listed in Table I.

The result of Theorem 3 relies on exploiting the asymptotic properties arising from the uniform degree concentration (31), coupled with the structure of combination matrices of class $C_2$, to characterize the asymptotic behavior of the error series in (48), (51) and (54). It is useful to contrast the results in Theorem 3 against the results in Theorems 1 and 2. We recall that Theorem 3 holds for both the ER model and the low-observability (i.e., partial ER) model:

- Theorem 3 provides the first result pertaining to the dense case, which is not addressed in Theorems 1 and 2. Under this regime universal strong consistency is proved.

- On the other hand, Theorem 3 cannot handle the log-sparse regime, which is instead handled by Theorems 1 and 2.

- In the intermediate-sparse regime, Theorem 3 proves universal strong consistency, whereas Theorems 1 and 2 cannot provide universality. However, it is not correct to state that Theorem 3 is more general than Theorems 1 and 2, since it holds for a more restricted class of matrices (class $C_2$).

- Finally, Theorem 3 proves consistency for two additional matrix estimators (which can be relevant in practice since they can deliver a performance superior to the Granger estimator).

One fundamental conclusion stemming from Theorem 3 is that, contrary to a widespread belief, sparsity
is not necessarily the fundamental enabler for consistent graph learning. One fundamental element is instead uniform concentration of the graph degrees, which coupled with the regular combination matrices in class $C_2$ and the randomness of the ER model, give rise to universally strong graph learning under partial observability.

IX. ILLUSTRATIVE EXAMPLES

In this section we present a couple of examples that are useful to illustrate how the estimators and tools examined in the previous sections work in practice.

A. Dual Graph Learning over a Detection Network

Let us consider a network of agents engaged in solving a binary detection problem by means of a diffusion strategy, as illustrated in Sec. II-B. In particular, we consider a Gaussian shift-in-mean problem, where the data are i.i.d. unit-variance Gaussian random variables, whose mean is equal to $-1$ under the null hypothesis $H_0$, and to $+1$ under the alternative hypothesis $H_1$. The step-size of the stochastic gradient algorithm is set equal to $\mu = 0.1$

In Fig. 5, we assume that all agents initially (i.e., at time zero) believe that the true hypothesis is $H_1$, while, in contrast, the data that they start observing are actually generated according to $H_0$. In the network topology on the leftmost panels, the agents that can be probed are highlighted by different colors, whereas the agents that are not accessible are displayed in gray. In the ten leftmost panels, we display the output of the distributed optimization problem (i.e., the direct inferential problem), namely, the signals $w_k$, for $k \in S$ that are collected by the graph learner in order to solve the dual learning problem. The color of the particular signal refers to the color of the corresponding agent in the graph topology.

First, we see that the distributed optimization algorithm is effective in accomplishing the detection task. In fact, after a relatively short transient all agents are able to fluctuate around a negative value, which will allow them to decide in favor of the correct hypothesis $H_0$.

Second, despite the apparent similarity between the signals at different agents, we see that there is significant information contained in these data streams about the agent interactions, i.e., about the network subgraph in $S$. As a matter of fact, inverse graph learning is possible, as we can appreciate from the boxes on the right, which highlight the correct reconstruction of the subgraph of probed agents.

B. Sequential Graph Reconstruction

The inherent locality of the graph reconstruction that is implied by Theorems 1, 2 and 3 suggests that the whole network can be reconstructed through a sequence of tomographic experiments that considers
Fig. 5. Example of inverse graph learning over an optimization network. A network of agents solves a distributed detection problem. The signals on the left represent the output variables of the distributed optimization algorithm, evolving at the probed agents (with color codes corresponding to the agents highlighted in the topologies displayed on the right). The dual learning problem (graph learning) is shown in the rightmost panels, where the graph of probed nodes is correctly retrieved by the estimation algorithm.

only small patches of the overall network [82], [85]. This sequential scheme is of great interest over large networks, where one could be eventually able to cover all nodes, but not simultaneously. For example, for various types of constraints (i.e., computation, accessibility) it might be largely unpractical to measure all signals from the network. Nevertheless, integrating the partial results coming from each patch examined in a single probe, we can eventually estimate the entire graph.

An example of this sequential reconstruction is offered in Fig. 6 where the boxes are numbered progressively to denote the current patches under test. For each probe, the graph learning algorithm produces an estimate of the subgraph (displayed in green) linking the currently probed agents. In the shown example, we assume that the network is partitioned into a certain number of non-overlapping equal-sized patches, and that the agents belonging to each individual patch are chosen at random. The overall ensemble of patches covers the whole network. Moreover, we consider that at each probe, a pair of these patches is chosen, and that after all probes all possible pairs are tested. In the second to last bottom box, we display the overall network graph that is learnt by aggregating the information relative to the individual patches. In the last bottom box we display the true graph. Comparing these latter two boxes, we see that the true graph is ultimately learnt by the sequential reconstruction algorithm.
Fig. 6. Example of sequential graph reconstruction. Successive local tomography experiments are shown, where each experiment corresponds to probe a subset of agents (highlighted in green). For each probe, the estimated edges linking the subset of currently probed agents are displayed in green as well. In the last two (bottom) panels, we display (second to last panel) the total estimated graph learnt by aggregating the 6 local experiments; and (last panel) the true graph. We see that the true graph is eventually learnt.

REFERENCES

[1] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.
[2] A. Nedić and D. P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” *SIAM J. Optim.*, vol. 12, no. 1, pp. 109–138, 2001.
[3] A. Nedić and A. Ozdaglar, “Cooperative distributed multi-agent optimization,” in *Convex Optimization in Signal Processing and Communications*, Y. Eldar and D. Palomar Eds. Cambridge University Press, 2010, pp. 340–386.
[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.
[5] S. Lee and A. Nedić, “Distributed random projection algorithm for convex optimization,” *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 221–229, Apr. 2013.
[6] C. Xi and U. A. Khan, “Distributed subgradient projection algorithm over directed graphs,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3986–3992, Oct. 2016.
[7] C. Xi, V. S. Mai, R. Xin, E. Abed, and U. A. Khan, “Linear convergence in optimization over directed graphs with row-stochastic matrices,” *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3558–3565, Oct. 2018.
[8] M. G. Rabbat and A. Ribeiro, “Multiagent distributed optimization,” in *Cooperative and Graph Signal Processing*, P. Djuric and C. Richard, Eds. Elsevier, 2018, pp. 147–167.
[9] M. Nokleby and W. U. Bajwa, “Stochastic optimization from distributed streaming data in rate-limited networks,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 152–167, Mar. 2019.
[10] M. H. DeGroot, “Reaching a consensus,” *J. Amer. Statist. Assoc.*, vol. 69, no. 345, pp. 118–121, 1974.
[11] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” *Systems and Control Letters*, vol. 53, no. 1, pp. 65–78, Sep. 2004.

[12] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.

[13] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.

[14] A. H. Sayed, “Adaptation, Learning, and Optimization over Networks,” *Found. Trends Mach. Learn.*, vol. 7, no. 4-5, pp. 311–801, 2014.

[15] A. Ganesh, L. Massoulié, and D. Towsley, “The effect of network topology on the spread of epidemics,” in *Proc. IEEE INFOCOM*, vol. 2, Mar. 2005, pp. 1455–1466.

[16] P. C. Pinto, P. Thiran, and M. Vetterli, “Locating the source of diffusion in large-scale networks,” *Physical Review Letters*, vol. 109, pp. 068702-1–068702-5, Aug. 2012.

[17] P. Venkitasubramaniam, T. He, and L. Tong, “Anonymous networking amidst eavesdroppers,” *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2770–2784, Jun. 2008.

[18] S. Marano, V. Matta, T. He, and L. Tong, “The embedding capacity of information flows under renewal traffic,” *IEEE Trans. Inf. Theory*, vol. 59, no. 3, pp. 1724–1739, Mar. 2013.

[19] S. Mahdizadehaghdam, H. Wang, H. Krim, and L. Dai, “Information diffusion of topic propagation in social media,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 569–581, Dec. 2016.

[20] V. Matta, V. Bordignon, A. Santos, and A. H. Sayed, “Interplay between topology and social learning over weak graphs” *submitted for publication*, Oct. 2019, available online as [arXiv:1910.13905v1 [cs.MA]].

[21] G. B. Giannakis, Y. Shen, and G. V. Karanikolas, “Topology Identification and Learning over Graphs: Accounting for Nonlinearities and Dynamics,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 787–807, May 2018.

[22] G. Mateos, S. Segarra, A. Marques, and A. Ribeiro, “Connecting the dots: Identifying network structure via graph signal processing,” *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 16–43, May 2019.

[23] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, “Learning Graphs From Data: A Signal Representation Perspective,” *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019.

[24] A. H. Sayed, S. Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, “Diffusion strategies for adaptation and learning over networks,” *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.

[25] A. H. Sayed, “Adaptive networks,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[26] D. P. Bertsekas, *Convex Analysis and Optimization*. Athena Scientific, MA, 2003.

[27] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, MA, 2004.

[28] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks — part I: Transient analysis,” *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3487–3517, Jun. 2015.

[29] J. Chen and A. H. Sayed, “On the learning behavior of adaptive networks — part II: Performance analysis,” *IEEE Trans. Inf. Theory*, vol. 61, no. 6, pp. 3518–3548, Jun. 2015.

[30] I. D. Couzin, “Collective cognition in animal groups,” *Trends in Cognitive Sciences*, vol. 13, no. 1, pp. 36–43, Jan. 2009.

[31] B. L. Partridge, “The structure and function of fish schools,” *Scientific American*, vol. 246, no. 6, pp. 114–123, Jun. 1982.

[32] F. S. Cattivelli and A. H. Sayed, “Distributed detection over adaptive networks using diffusion adaptation,” *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 1917–1932, May 2011.

[33] V. Matta and A. H. Sayed, “Estimation and detection over adaptive networks,” in *Cooperative and Graph Signal Processing*, P. Djuric and C. Richard, Eds. Elsevier, 2018, pp. 69–106.
[34] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, NY, 1991.

[35] V. Matta, P. Braca, S. Marano, and A. H. Sayed, “Diffusion-based adaptive distributed detection: Steady-state performance in the slow adaptation regime,” *IEEE Trans. Inf. Theory*, vol. 62, no. 8, pp. 4710–4732, Aug. 2016.

[36] V. Matta, P. Braca, S. Marano, and A. H. Sayed, “Distributed detection over adaptive networks: Refined asymptotics and the role of connectivity,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 442–460, Dec. 2016.

[37] A. Jadabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, “Non-Bayesian social learning,” *Games and Economic Behavior*, vol. 76, no. 1, pp. 210–225, Sep. 2012.

[38] X. Zhao and A. H. Sayed, “Learning over social networks via diffusion adaptation,” in *Proc. Asilomar Conference on Signals, Systems and Computers*, Nov. 2012, pp. 709–713.

[39] A. Nedić, A. Olshhevsky, and C. A. Uribe, “Fast convergence rates for distributed non-Bayesian learning,” *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5538–5553, Nov. 2017.

[40] A. Lalitha, T. Javidi, and A. D. Sarwate, “Social learning and distributed hypothesis testing,” *IEEE Trans. Inf. Theory*, vol. 64, pp. 6161–6179, Sep. 2018.

[41] V. Matta, A. Santos, and A. H. Sayed, “Exponential collapse of social beliefs over weakly-connected heterogeneous networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, May 2019, pp. 5267–5271.

[42] B. Ying and A. H. Sayed, “Information exchange and learning dynamics over weakly connected adaptive networks,” *IEEE Trans. Inf. Theory*, vol. 62, no. 3, pp. 1396–1414, Mar. 2016.

[43] H. Salami, B. Ying, and A. H. Sayed, “Social learning over weakly connected graphs,” *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 2, pp. 222–238, Jun. 2017.

[44] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons, NY, 1990.

[45] N. Wiener, “The theory of prediction,” in *Modern Mathematics for the Engineer*, E. F. Beckenbach, Ed. McGraw-Hill, New York, 1956, pp. 165–190.

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

[47] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge University Press, New York, 1985.

[48] S. E. Shimony, “Finding MAPs for belief networks is NP-hard,” *Artificial Intelligence*, vol. 68, no. 2, pp. 399–410, Aug. 1994.

[49] D. M. Chickering, D. Heckerman, and C. Meek, “Large-sample learning of Bayesian networks is NP-hard,” *Journal of Machine Learning Research*, vol. 5, pp. 1287–1303, Dec. 2004.

[50] A. Bogdanov, E. Mossel, and S. Vadhan, “The complexity of distinguishing Markov random fields,” in *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, A. Goel, K. Jansen, J. D. P. Rolim, and R. Rubinfeld, Eds. Springer-Verlag Berlin Heidelberg, 2008, pp. 331–342.

[51] J. Bento and A. Montanari, “Which graphical models are difficult to learn?” in *Proc. Neural Information Processing Systems (NIPS)*, Vancouver, Canada, Dec. 2009, pp. 1303–1311.

[52] G. Bresler, D. Gamarnik, and D. Shah, “Hardness of parameter estimation in graphical models,” in *Proc. Neural Information Processing Systems (NIPS)*, Montréal, Canada, Dec. 2014, pp. 1062–1070.

[53] D. Napoletani and T. D. Sauer, “Reconstructing the topology of sparsely connected dynamical networks,” *Physical Review E*, vol. 77, no. 2, pp. 026103-1–026103-5, Feb. 2008.

[54] J. Ren, W.-X. Wang, B. Li, and Y.-C. Lai, “Noise bridges dynamical correlation and topology in coupled oscillator networks,” *Physical Review Letters*, vol. 104, no. 5, pp. 058701-1–058701-4, Feb. 2010.
[55] A. Mauroy and J. Goncalves, “Linear identification of nonlinear systems: A lifting technique based on the Koopman operator,” in Proc. IEEE Conference on Decision and Control (CDC), Las Vegas, NV, USA, Dec. 2016, pp. 6500–6505.

[56] E. S. C. Ching and H. C. Tam, “Reconstructing links in directed networks from noisy dynamics,” Physical Review E, vol. 95, no. 1, pp. 010301–1–010301–5, Jan. 2017.

[57] P.-Y. Lai, “Reconstructing network topology and coupling strengths in directed networks of discrete-time dynamics,” Physical Review E, vol. 95, no. 2, pp. 022311–1–022311–13, Feb. 2017.

[58] Y. Yang, T. Luo, Z. Li, X. Zhang, and P. S. Yu, “A robust method for inferring network structures,” in Scientific Reports, vol. 7, no. 5221, pp. 1–12, Jul. 2017.

[59] D. Materassi and M. V. Salapaka, “On the problem of reconstructing an unknown topology via locality properties of the Wiener filter,” IEEE Trans. Autom. Control, vol. 57, no. 7, pp. 1765–1777, Jul. 2012.

[60] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” IEEE Signal Process. Mag., vol. 30, no. 3, pp. 83–98, May 2013.

[61] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, “Discrete signal processing on graphs: Sampling theory,” IEEE Trans. Signal Process., vol. 63, no. 24, pp. 6510–6523, Dec. 2015.

[62] M. Tsitsvero, S. Barbarossa, and P. D. Lorenzo, “Signals on graphs: Uncertainty principle and sampling,” IEEE Trans. Signal Process., vol. 64, no. 18, pp. 4845–4860, Sep. 2016.

[63] N. Perraudin and P. Vandergheynst, “Stationary signal processing on graphs,” IEEE Trans. Signal Process., vol. 65, no. 13, pp. 3462–3477, Jul. 2017.

[64] S. P. Chepuri and G. Leus, “Graph sampling for covariance estimation,” IEEE Trans. Signal Inf. Process. Netw., vol. 3, no. 3, pp. 451–466, Sep. 2017.

[65] S. Segarra, M. T. Schaub, and A. Jadbabaie, “Network inference from consensus dynamics,” in Proc. IEEE Conference on Decision and Control (CDC), Dec. 2017, pp. 3212–3217.

[66] B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, and M. G. Rabbat, “Characterization and inference of graph diffusion processes from observations of stationary signals,” IEEE Trans. Signal Inf. Process. Netw., vol. 4, no. 3, pp. 481–496, Sep. 2018.

[67] J. Mei and J. Moura, “Signal processing on graphs: Causal modeling of unstructured data,” IEEE Trans. Signal Process., vol. 65, no. 8, pp. 2077–2092, Apr. 2017.

[68] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs,” IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[69] A. Sandryhaila and J. M. F. Moura, “Discrete signal processing on graphs: Frequency analysis,” IEEE Trans. Signal Process., vol. 62, no. 12, pp. 3042–3054, Jun. 2014.

[70] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Stationary graph processes and spectral estimation,” IEEE Trans. Signal Process., vol. 65, no. 22, pp. 5911–5926, Nov. 2017.

[71] C. J. Quinn, N. Kiyavash, and T. P. Coleman, “Directed information graphs,” IEEE Trans. Inf. Theory, vol. 61, no. 12, pp. 6887–6909, Dec. 2015.

[72] J. Etesami and N. Kiyavash, “Measuring causal relationships in dynamical systems through recovery of functional dependencies,” IEEE Trans. Signal Inf. Process. Netw., vol. 3, no. 4, pp. 650–659, Dec. 2017.

[73] D. Materassi and M. V. Salapaka, “Network reconstruction of dynamical polytrees with unobserved nodes,” in Proc. IEEE Conference on Decision and Control (CDC), Maui, HI, USA, Dec. 2012, pp. 4629–4634.
[74] J. Etesami, N. Kiyavash, and T. Coleman, “Learning minimal latent directed information polytrees,” *Neural Computation*, vol. 28, no. 9, pp. 1723–1768, Aug. 2016.

[75] P. Geiger, K. Zhang, B. Schölkopf, M. Gong, and D. Janzing, “Causal inference by identification of vector autoregressive processes with hidden components,” in *Proc. International Conference on Machine Learning (ICML)*, vol. 37, Lille, France, Jul. 2015, pp. 1917–1925.

[76] D. Materassi and M. V. Salapaka, “Identification of network components in presence of unobserved nodes,” in *Proc. IEEE Conference on Decision and Control (CDC)*, Osaka, Japan, Dec. 2015, pp. 1563–1568.

[77] A. Anandkumar, V. Y. F. Tan, F. Huang, and A. S. Willsky, “High-dimensional Gaussian graphical model selection: Walk summability and local separation criterion,” *Journal of Machine Learning Research*, vol. 13, pp. 2293–2337, Jan. 2012.

[78] A. Anandkumar and R. Valluvan, “Learning loopy graphical models with latent variables: Efficient methods and guarantees,” *The Annals of Statistics*, vol. 41, no. 2, pp. 401–435, Apr. 2013.

[79] V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky, “Latent variable graphical model selection via convex optimization,” *The Annals of Statistics*, vol. 40, no. 4, pp. 1935–1967, Aug. 2012.

[80] G. Bresler, F. Koehler, A. Moitra, and E. Mossel, “Learning restricted Boltzmann machines via influence maximization,” in *Proc. ACM Symposium on Theory of Computing (STOC)*, Phoenix, AZ, USA, Jun. 2019.

[81] V. Matta and A. H. Sayed, “Consistent tomography under partial observations over adaptive networks,” *IEEE Trans. Inf. Theory*, vol. 65, no. 1, pp. 622–646, Jan. 2019.

[82] A. Santos, V. Matta, and A. H. Sayed, “Local tomography of large networks under the low-observability regime,” to appear in *IEEE Trans. Inf. Theory*, Oct. 2019, available online in early access, doi: 10.1109/TIT.2019.2945033.

[83] V. Matta, A. Santos, and A. H. Sayed, “Learning Erdős-Rényi graphs under partial observations: Concentration or sparsity?” submitted for publication, May 2019, available online as arXiv:1904.02963v1 [math.ST].

[84] V. Matta and A. H. Sayed, “Tomography of adaptive multi-agent networks under limited observation,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, Canada, Apr. 2018, pp. 6638–6642.

[85] A. Santos, V. Matta, and A. H. Sayed, “Divide-and-conquer tomography for large-scale networks,” in *Proc. IEEE Data Science Workshop (DSW)*, Lausanne, Switzerland, Jun. 2018, pp. 170–174.

[86] A. Santos, V. Matta, and A. H. Sayed, “Consistent tomography over diffusion networks under the low-observability regime,” in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Vail, CO, USA, Jun. 2018, pp. 1839–1843.

[87] V. Matta, A. Santos, and A. H. Sayed, “Tomography of large adaptive networks under the dense latent regime,” in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Oct. 2018, pp. 2144–2148.

[88] V. Matta, A. Santos, and A. H. Sayed, “Graph learning with partial observations: Role of degree concentration,” in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Paris, France, Jul. 2019, pp. 1–5.

[89] P. Erdős and A. Rényi, “On random graphs I,” *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.

[90] B. Bollobás, *Random Graphs*. Cambridge University Press, 2001.

[91] S. Boucheron, G. Lugosi, and P. Massart, *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.