Wolfram Barfuss, Guido Previde Massara, T. Di Matteo and Tomaso Aste

Parsimonious modeling with information filtering networks

Article (Published version) (Refereed)

Original citation:
Barfuss, Wolfram, Massara, Guido Previde, Di Matteo, T. and Aste, Tomaso (2016) *Parsimonious modeling with information filtering networks.* Physical Review E, 94 (6). ISSN 1539-3755

DOI: 10.1103/PhysRevE.94.062306

PHYSICAL REVIEW E **94**, 062306 (2016)

Parsimonious modeling with information filtering networks

Wolfram Barfuss,[1,*,†] Guido Previde Massara,[2] T. Di Matteo,[2,3,4] and Tomaso Aste[2,4]
[1]*Department of Physics, FAU Erlangen-Nürnberg, Nägelsbachstrasse 49b, 91052 Erlangen, Germany*
[2]*Department of Computer Science, University College London, Gower Street, London, WC1E 6BT, United Kingdom*
[3]*Department of Mathematics, King's College London, The Strand, London, WC2R 2LS, United Kingdom*
[4]*Systemic Risk Centre, London School of Economics and Political Sciences, London, WC2A2AE, United Kingdom*
(Received 25 June 2016; published 13 December 2016)

We introduce a methodology to construct parsimonious probabilistic models. This method makes use of information filtering networks to produce a robust estimate of the global sparse inverse covariance from a simple sum of local inverse covariances computed on small subparts of the network. Being based on local and low-dimensional inversions, this method is computationally very efficient and statistically robust, even for the estimation of inverse covariance of high-dimensional, noisy, and short time series. Applied to financial data our method results are computationally more efficient than state-of-the-art methodologies such as Glasso producing, in a fraction of the computation time, models that can have equivalent or better performances but with a sparser inference structure. We also discuss performances with sparse factor models where we notice that relative performances decrease with the number of factors. The local nature of this approach allows us to perform computations in parallel and provides a tool for dynamical adaptation by partial updating when the properties of some variables change without the need of recomputing the whole model. This makes this approach particularly suitable to handle big data sets with large numbers of variables. Examples of practical application for forecasting, stress testing, and risk allocation in financial systems are also provided.

DOI: 10.1103/PhysRevE.94.062306

I. INTRODUCTION

This paper addresses the following question: how can one construct, from a set of observations, the model that most meaningfully describes the underlying system? This is a general question at the core of scientific research. Indeed, the so-called scientific method has been devised around a combination of observation, model, and prediction in a circular way, where hypotheses are formulated and tested with further observations, iteratively refining or changing the models to obtain better predictions; all within the principle of parsimony where a simpler model with less parameters and less assumptions should be preferred to a more complex one.

In the context of the present paper the "model" is the multivariate probability distribution that best describes the set of observations. The problem of finding such a distribution becomes particularly challenging when the number of variables, p, is large and the number of observations, q, is small. Indeed, in such a multivariate problem, the model must take into account at least an order $O(p^2)$ of interrelations between the variables and therefore the number of model-parameters scales at least quadratically with the number of variables. A parsimonious approach requires us to discover the model that best reproduces the statistical properties of the observations while keeping the number of parameters as small as possible. Using a maximum entropy approach, up to the second order in the moments of the distribution, the model becomes the multivariate normal distribution. In the multivariate normal case there is a simple relationship between the sparsity pattern of the inverse of the covariance matrix (the precision matrix, henceforth denoted by $\mathbf{J}$) and the underlying partial correlation structure (referred to as "graphical model" in the literature [1]): two nodes i and j are linked in the graphical model if and only if the corresponding precision matrix element J_{ij} is different from zero. Therefore, the problem of estimating a sparse precision matrix is equivalent to the problem of learning a sparse multivariate normal graphical model (known in the literature as Gaussian Markov random field (GMRF) [2]). Once the sparse precision matrix has been estimated, a number of efficient tools—mostly based on research in sparse numerical linear algebra—can be used to sample from the distribution, calculate conditional probabilities, calculate conditional statistics, and forecast [2,3]. GMRFs are of great importance in many applications spanning computer vision [4], sparse sensing [5], finance [6–11], gene expression [12–14]; biological neural networks [15], climate networks [16,17]; geostatistics and spatial statistics [18–20]. Almost universally, applications require modeling a large number of variables with a relatively small number of observations, and therefore the issue of the statistical significance of the model parameters is very important.

The problem of finding meaningful and parsimonious models, sometimes referred to as sparse structure learning [21], has been tackled by using a number of different approaches. Let us hereafter briefly account for some of the most relevant in the present context.

Constraint based approaches recover the structure of the network by testing the local Markov property. Usually the algorithm starts from a complete model and adopts a *backward selection* approach by testing the independence of nodes conditioned on subsets of the remaining nodes (algorithms SGS and PC [21]) and removing edges associated to nodes that are conditionally independent; the algorithm stops when some criteria are met—e.g., every node has less than a

*Now at Potsdam Institute for Climate Impact Research, Telegrafenberg A31, 14473 Potsdam, Germany.
†Present address: Department of Physics, Humboldt University, Newtonstrasse 15, 12489 Berlin, Germany.

given number of neighbors. Conversely, *forward selection* algorithms start from a sparse model and add edges associated to nodes that are discovered to be conditionally dependent. A hybrid model is the GS algorithm where candidate edges are added to the model (the "grow" step) in a forward selection phase and subsequently reduced using a backward selection step (the "shrinkage" step) [21]. However, the complexity of checking a large number of conditional independence statements makes these methods unsuitable for moderately large graphs. Furthermore, aside from the complexity of measuring conditional independence, these methods do not generally optimize a global function, such as likelihood or the Akaike information criterion [22,23], but they rather try to exhaustively test all the (local) conditional independence properties of a set of data and therefore are difficult to use in a probabilistic framework.

Score based approaches learn the inference structure trying to optimize some global function: likelihood, Kullback-Leibler divergence [24], Bayesian information criterion (BIC) [25], minimum description length [26], or the likelihood ratio test statistics [27]. In these approaches, the main issue is that the optimization is generally computationally demanding and some sort of greedy approach is required. Statistical physics methodologies for the discovery of inference networks by maximization of entropy or minimization of cost functions have been used in biologically motivated studies to extract gene regulatory networks and signaling networks (see, for instance, Refs. [28,29]).

In the field of *decomposable models* there are a number of methods that efficiently explore the graphical structure (directed, in the case of Bayesian models, or undirected in the case of log-linear or multivariate Gaussian models) by using advanced graph structures such as junction tree or clique graph [27,30,31], with the goal of producing sparse models (so-called "thin junction trees" [32–34]).

Other approaches [35–37] treat the problem as a constrained optimization problem to recover the sparse covariance matrix. Within this line, *regression-based approaches* generally try to minimize some loss function, which enforces parsimony and sparsity by using penalization to constrain the number and size of the regression parameters. Specifically, ridge regression uses a ℓ_2-norm penalty; instead the *lasso* method [38] uses an ℓ_1-norm penalty and the elastic-net approach uses a convex combination of ℓ_2 and ℓ_1 penalties on the regression coefficients [39]. These approaches are among the best performing regularization methodologies presently available. The ℓ_1-norm penalty term favors solutions with parameters with zero value leading to models with sparse inverse covariances. Sparsity is controlled by regularization parameters $\lambda_{ij} > 0$; the larger the value of the parameters the more sparse the solution becomes. This approach has become extremely popular and, around the original idea, a large body of literature has been published with several novel algorithmic techniques that are continuously advancing this method [36,38,40–43], among these the popular implementation *Glasso* (Graphical-lasso) [44], which uses lasso to compute sparse graphical models. However, Glasso methods are computationally intensive and, although they are sparse, the nonzero parameters interaction structure tends to be noisy and not significantly related with the true underlying interactions between the variables.

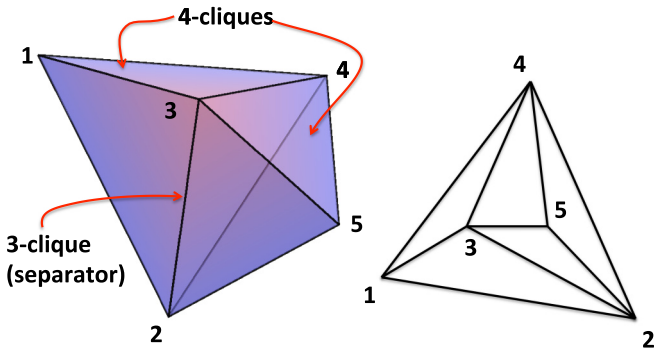


FIG. 1. Two schematic representations of a TMFG network. The network is made of two cliques of four nodes (4-clique) and one separator of three nodes (3-clique). The cliques are the tetrahedra {1,2,3,4} and {2,3,4,5} and the separator is the triangle {2,3,4}. This is a chordal graph.

Meaningful structures associated with the relevant network of interactions in complex systems are instead retrieved by information filtering networks, which were first introduced by Mantegna [45] and some of the authors of the present paper [6,46] for the study of the structure of financial markets and biological systems [47]. There is now a large body of literature demonstrating that information filtering networks such as maximum spanning tree (MST) or planar maximally filtered graphs (PMFG) constructed from correlation matrices retrieve meaningful structures, extracting the relevant interactions from multivariate data sets in complex systems [7,8,47]. Recently, a new family of information filtering networks, the triangulated maximal planar graph (TMFG) [48], was introduced. These are planar graphs, similar to the PMFG, but with the advantages to be generated in a computationally efficient way and, more importantly for this paper, they are decomposable graphs (see example in Fig. 1). A decomposable graph has the property that every cycle of length greater than three has a chord, an edge that connects two vertices of the cycle in a smaller cycle of length three. Decomposable graphs, also called chordal or triangulated, are clique forests, made of k cliques (complete subgraphs of k vertices) connected by separators. Separators are also cliques of smaller sizes with the property that the graph becomes divided into two or more disconnected components when the vertices of the separator are disconnected. For example, in the schematic representation of the TMFG reported in Fig. 1 the cliques are the tetrahedra {1,2,3,4} and {2,3,4,5}, whereas the separator is the triangle {2,3,4}.

The novelty of the method presented in this paper is the combination of decomposable information filtering networks [6–10] with Gaussian Markov random fields [1,2] to produce parsimonious models associated with a meaningful structure of dependency between the variables. The strength of this methodology is that the global sparse inverse covariance matrix is produced from a simple sum of local inversions. This makes the method computationally very efficient and statistically robust. Given the local-global nature of its construction, in the following we shall refer to this method as *LoGo*.

In this paper, we demonstrate that the structure provided by information filtering networks is also extremely effective to generate high-likelihood sparse probabilistic models. In the linear case, the LoGo sparse inverse covariance has only $O(p)$

parameters but, despite its sparsity, the associated multivariate normal distribution can still retrieve high likelihood values yielding, in practical applications such as financial data, comparable or better results than state-of-the-art Glasso penalized inversions.

The methodology introduced in this paper contributes to the literature on graphical modeling, sparse inverse covariances, and machine learning. But it is of even greater relevance to physical sciences, particularly for what concerns complex systems research and network theory. Indeed, physical sciences are increasingly engaged in the study of complex systems where the challenge is to elaborate models able to make predictions based on observational data. With LoGo, for the first time we combine a successful tool to describe complex systems structure (namely the information filtering networks) with a parsimonious probabilistic model that can be used for quantitative predictions. Such a combination is of relevance to any context where parsimonious statistical modeling are applicable.

The rest of the paper is organized as follows: In Sec. II we describe our methodology providing algorithms for sparse covariance inversion for two graph topologies: MST and the TMFG. We then show in Sec. III that our method yields comparable or better results in maximum likelihood compared to lasso-type and ridge regression estimates of the inverse covariances from financial time series. Subsequently, we discuss how our approach can be used for time series prediction, financial stress testing, and risk allocation. With Sec. IV we end with possible extensions for future work and conclusive remarks.

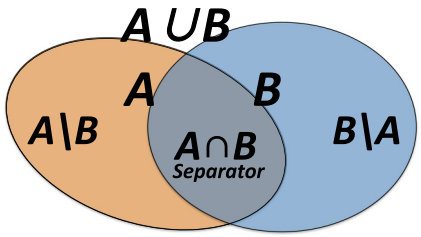
II. PARSIMONIOUS MODELING WITH INFORMATION FILTERING NETWORKS

A. Factorization of the multivariate probability distribution function

Let us start by demonstrating how a decomposable information filtering network can be associated with a convenient factorization of the multivariate probability distribution. Let us consider, in general, two sets A and B of variables with nonempty intersection $A \cap B \neq \emptyset$. Let us also assume that the variables are mutually dependent within their own ensemble A or B, but when one variable belongs to set $A \setminus B$ and the other variable belongs to set $B \setminus A$, then they are independent conditioned to $A \cap B$. We can now use the Bayes formula: $f(A \cup B) = f(A \setminus B|B)f(B)$, where $f(A \cup B)$ is the joint probability distribution function of all variable in A and B, $f(A \setminus B|B)$ is the conditional probability distribution function for the variables in A minus the subset in common with B conditioned to all variables in B, and $f(B)$ is the marginal probability distribution function of all variables in B (see Fig. 2). From the Bayes formula we also have the following identity: $f(A) = f(A \setminus B|B)f(A \cap B)$, which combined with the previous gives the following factorization for the joint probability distribution function of all variable in A and B [1]:

$$f(A \cup B) = \frac{f(A)f(B)}{f(A \cap B)}. \tag{1}$$

Let us now apply this formula to a set of variables associated with a decomposable information filtering network $\mathcal{G}$ made of M_c cliques, $\mathcal{C}_m$, with $m = 1,..,M_c$, and M_s complete separators



$$f(A \cup B) = f(A \setminus B \,|\, B) f(B)$$
$$f(A) = f(A \setminus B \,|\, B) f(A \cap B)$$

FIG. 2. Decomposition of the joint probability distribution function. Bayes formulas for two sets of variables A and B with a separating set $A \cap B$. Variables within sets A or B are assumed conditionally dependent whereas variables belonging to the two separated sets $A \setminus B$ and $B \setminus A$ are assumed independent conditionally to $A \cap B$. By combining the two formulas one obtains $f(A \cup B) = f(A)f(B)/f(A \cap B)$.

$\mathcal{S}_n$, with $n = 1,...,M_s$. In such a network, the vertices represent the p variables $X_1,...,X_p$ and the edges represent couples of *conditionally* dependent variables (condition being with respect to all other variables). Conversely, variables that are not directly connected with a network edge are conditionally independent. Given such a network, in the same way as for Eq. (1), one can write the joint probability density function for the set of p variables $\boldsymbol{X} = (X_1, X_2, ..., X_p)^{\mathsf{T}}$ in terms of the following factorization into cliques and separators [1]:

$$f(\boldsymbol{X}) = \frac{\prod_{m=1}^{M_c} f_{\mathcal{C}_m}(\boldsymbol{X}_{\mathcal{C}_m})}{\prod_{n=1}^{M_s} f_{\mathcal{S}_n}(\boldsymbol{X}_{\mathcal{S}_n})^{k(\mathcal{S}_n)-1}}, \tag{2}$$

where $f_{\mathcal{C}_m}(\mathbf{X}_{\mathcal{C}_m})$ and $f_{\mathcal{S}_n}(\boldsymbol{X}_{\mathcal{S}_n})$ are the marginal probability density functions of the variables constituting $\mathcal{C}_m$ and $\mathcal{S}_n$, respectively [1]. The term $k(\mathcal{S}_n)$ counts the number of disconnected components produced by removing the separator $\mathcal{S}_n$, and it is therefore the degree of the separator in the clique tree. Given the graph $\mathcal{G}$, Eq. (2) is exact, it is a direct consequence of the Bayes formula and it is therefore very general and applicable to both linear, nonlinear as well as parametric or nonparametric modeling.

B. Functional form of the multivariate probability distribution function

We search for the functional form of the multivariate probability distribution function, $f(\boldsymbol{X})$. To find the functional form of the distribution f and the values of its parameters $\mathbf{J}$, we use the maximum entropy method [49,50], which constrains the model to have some given expectation values while maximizing the overall information entropy $-\int f(\boldsymbol{X}) \log f(\boldsymbol{X}) d^p\boldsymbol{X}$. At the second order, the model distribution that maximizes entropy while constraining moments at given values is

$$f(\boldsymbol{X}) = \frac{1}{Z} \exp\left(-\sum_{ij} \frac{1}{2}(X_i - \mu_i) J_{i,j} (X_j - \mu_j) \right), \tag{3}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{p \times 1}$ is the vector of expectation values with coefficients $\mu_i = \mathbb{E}[X_i]$ and $J_{i,j}$ are the matrix elements of $\mathbf{J} \in$

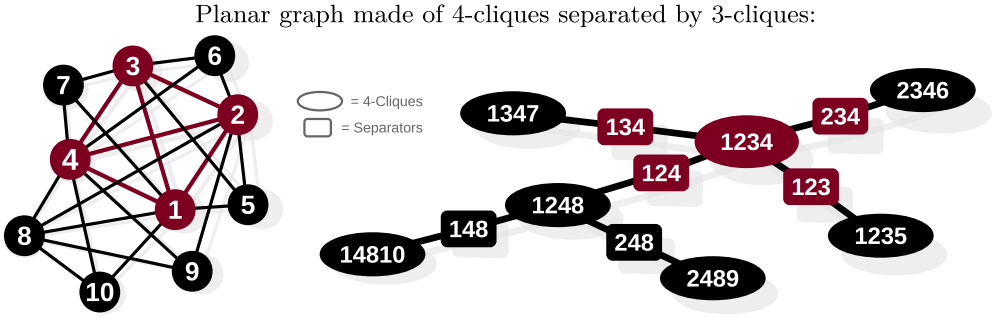


FIG. 3. Local-global inversion of the covariance matrix. Example for a system of $p = 10$ variables associated with a decomposable TMFG graph with $M_c = 7$ cliques and $M_s = 6$ separators.

$\mathbb{R}^{p\times p}$. They are the Lagrange multipliers associated with the second moments of the distribution $\mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)] = \Sigma_{i,j}$, which are the coefficients of the covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{p\times p}$ of the set of p variables $\boldsymbol{X}$. It is clear that Eq. (3) is a multivariate normal distribution with $Z = \sqrt{(2\pi)^p \det(\mathbf{\Sigma})}$. If we require the model $f(\boldsymbol{X})$ to reproduce exactly all second moments $\Sigma_{i,j}$, then the solution for the distribution parameters is $\mathbf{J} = \mathbf{\Sigma}^{-1}$. Therefore, in order to construct the model, one could estimate empirically the covariance matrix $\hat{\mathbf{\Sigma}}$ from a set of q observations and then invert it in order to estimate the inverse covariance. However, in the case when the observation length q is smaller than the number of variables p the empirical estimate of the covariance matrix $\hat{\mathbf{\Sigma}}$ cannot be inverted. Furthermore, also in the case when $q > p$, such a model has $p(p+3)/2$ parameters and this might be an overfitting solution describing noise instead of the underlying relationships between the variables resulting in poor predictive power [13,51]. Indeed, we shall see in the following that, when uncertainty is large (q small), models with a smaller number of parameters can have stronger predictive power and can better describe the statistical variability of the data [52]. Here, we consider a parsimonious modeling that fixes only a selected number of second moments and leaves the others unconstrained. This corresponds to model the multivariate distribution by using a *sparse inverse covariance* where the unconstrained moments are associated with zero coefficients in the inverse. Let us note that this in turn implies zero partial correlation between the corresponding couples of variables.

C. Sparse inverse covariance from decomposable information filtering networks

From Eq. (2) it follows that, in the case of the multivariate normal distribution, the network $\mathcal{G}$ coincides with the structure of nonzero coefficients, $J_{i,j}$ in Eq. (3) and their values can be computed from the local inversions of the covariance matrices associated with the cliques and separators, respectively [1]:

$$J_{i,j} = \sum_{\mathcal{C}\ s.t.\ \{i,j\}\in\mathcal{C}} \left(\mathbf{\Sigma}^{-1}_{\mathcal{C}}\right)_{i,j} - \sum_{\mathcal{S}\ s.t.\ \{i,j\}\in\mathcal{S}} (k(\mathcal{S}) - 1)\left(\mathbf{\Sigma}^{-1}_{\mathcal{S}}\right)_{i,j}, \tag{4}$$

and $J_{i,j} = 0$ if $\{i,j\}$ are not both part of a common clique.

This is a very simple formula that reduces the global problem of a $p \times p$ matrix inversion into a sum of local inversions of matrices of the sizes of the cliques and separators (no more than 3 and 4 in the case of TMFG graphs [48,53]). This means that, for TMFG graphs, only four observations would be enough to produce a nonsingular global estimate of the inverse covariance. An example illustrating this inversion procedure is provided in Fig. 3.

D. Construction of the maximum likelihood network

We are now facing two related problems: (1) how to choose the moments to retain and (2) how to verify that the parsimonious model is describing well the statistical properties of the system of variables. The solutions of these two problems are related because we aim to develop a methodology that chooses the nonzero elements of the inverse covariance in such a way as to best model the statistical properties of the real system under observation. In order to construct a model that is closest to the real phenomenon we search for the set of parameters, $\mathbf{J}$, associated with the largest likelihood, i.e., with the largest probability of observing the actual observations: $\{x_{1,1},\ldots,x_{1,q}\}, \{x_{2,1},\ldots,x_{2,q}\}\ldots\{x_{p,1},\ldots,x_{p,q}\}$. The logarithm of the likelihood from a model distribution function, $f(\boldsymbol{X})$ [Eq. (3)], with parameters $\mathbf{J}$, is associated to the empirical estimate of the covariance matrix, $\hat{\Sigma}$, by [54]

$$\ln \mathcal{L}(\mathbf{J}) = \frac{q}{2}[\ln\det\mathbf{J} - \mathrm{Tr}(\hat{\mathbf{\Sigma}}\mathbf{J}) - p\ln(2\pi)]. \tag{5}$$

The network $\mathcal{G}$ that we aim to discover must be associated with largest log-likelihood and it can be constructed in a greedy way by adding in subsequent steps elements with maximal log-likelihood. In this paper we propose two constructions: (1) the maximum spanning tree (MST) [56], which builds a spanning tree which maximizes the sum of edge weights, and (2) a variant of the TMFG [48], which builds a planar graph that aims to maximize the sum of edge weights. In both cases edge weights are associated with the log-likelihood. One can show that for all decomposable graphs, following Eq. (4), the middle term in Eq. (5) is $\mathrm{Tr}(\hat{\mathbf{\Sigma}}\mathbf{J}) = p$. Hence, to maximize log-likelihood, only $\log\det\mathbf{J}$ must be maximized; from Eq. (2),

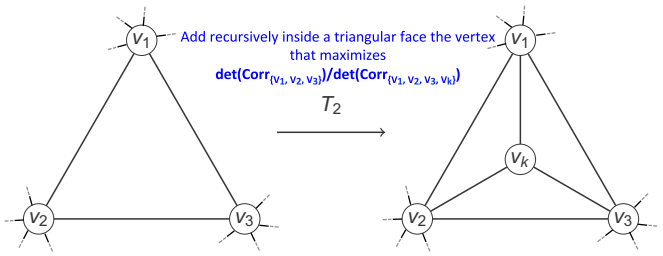


FIG. 4. TMFG construction. The TMFG graph is generated by adding vertices (e.g., v_k) inside triangular faces (e.g., $\{v_1,v_2,v_3\}$) maximizing the ratio of the determinants between separator and clique $\det(\hat{\mathbf{R}}_{\{v_1,v_2,v_3\}})/\det(\hat{\mathbf{R}}_{\{v_1,v_2,v_3,v_k\}})$. This move generates a new 4-clique $\mathcal{C}=\{v_1,v_2,v_3,v_k\}$ and transforms the triangular face into a separator $\mathcal{S}=\{v_1,v_2,v_3\}$.

this is [1]

$$\log\det\mathbf{J}=\sum_{n=1}^{M_s}[k(\mathcal{S}_n)-1]\log\det\hat{\Sigma}_{\mathcal{S}_n}-\sum_{m=1}^{M_c}\log\det\hat{\Sigma}_{\mathcal{C}_m}. \tag{6}$$

For the LoGo-MST, the construction is simplified because in a tree the cliques are the edges $e=(i,j)$, the separators are the nonleaf vertices v_i, and $k(\mathcal{S}_i)=k_i$ are the vertex degrees. In this case, Eq. (6) becomes $\log\det\mathbf{J}=\sum_{i=1}^{p}\log\hat{\sigma}_i^{2(k_i-1)}-\sum_{e\in MST}\log\det\hat{\mathbf{\Sigma}}_e$, with $\hat{\sigma}_i^2$ the sample variance of variable "i." Given that $\det\hat{\Sigma}_e=\hat{\sigma}_i^2\hat{\sigma}_j^2(1-\hat{R}_{i,j}^2)$, with $\hat{R}_{i,j}$ the Pearson's correlation matrix element i,j, then $\log\det\mathbf{J}=-\sum_{i=1}^{p}\log\hat{\sigma}_i^2-\sum_{e\in MST}\log(1-\hat{R}_{i,j}^2)$. Therefore, the MST can be built through the standard Prim's [55] or Kruskal's [56] algorithms from a matrix of weights $\mathbf{W}$ with coefficients $W_{i,j}=\hat{R}_{i,j}^2$. The *LoGo-MST* inverse covariance estimation, $\mathbf{J}$, is then computed by the local inversion, Eq. (4), on the MST structure. Note that the MST structure depends only on the correlations not the covariance.

The LoGo-TMFG construction requires a specifically designed procedure. Also in this case, only correlations matter; indeed, the structure of the inverse covariance network reflects the partial correlations, i.e., the correlation between two variables given all others. LoGo-TMFG starts with a tetrahedron, $\mathcal{C}_1=\{v_1,v_2,v_3,v_4\}$, with smallest correlation determinant $\det\hat{\mathbf{R}}_{\mathcal{C}_1}$, and then iteratively introduces, inside existing triangular faces, the vertex that maximizes $\log\det\hat{\mathbf{R}}_{\mathcal{S}}-\log\det\hat{\mathbf{R}}_{\mathcal{C}}$, where $\mathcal{C}$ and $\mathcal{S}$ are the new clique and separator created by the vertex insertion. The *LoGo-TMFG* procedure is schematically reported in Algorithm I and in Fig. 4. The TMFG is a computationally efficient algorithm [48] that produces a decomposable (chordal) graph, with $3(p-2)$ edges, which is a clique tree constituted by four-clique connected with three-clique separators. Note that for TMFG $k(\mathcal{S}_n)=2$ always.

Algorithm 1: LoGo-TMFG algorithm. Construction of sparse LoGo-TMFG sparse inverse covariance $\mathbf{J}$ starting from a covariance matrix $\hat{\Sigma}$. The nonzero elements of $\mathbf{J}$ are the edges of $\mathcal{G}$, which is a chordal graph, a clique tree, made of tetrahedra separated by triangles.

input: A covariance matrix $\hat{\Sigma}\in\mathbb{R}^{p\times p}$ and associated correlation matrix $\hat{\mathbf{R}}\in\mathbb{R}^{p\times p}$ from a set of observations $\{x_{1,1},...,x_{1,q}\}$, $\{x_{2,1},...,x_{2,q}\}$.... $\{x_{p,1},...,x_{p,q}\}$
output: $\mathbf{J}$ a sparse estimation of Σ^{-1}
$\mathbf{J}\leftarrow\mathbf{0}$ Initialize $\mathbf{J}$ with zero elements;
$\mathcal{C}_1\leftarrow$ Tetrahedron, $\{v_1,v_2,v_3,v_4\}$, with smallest $\det\hat{\mathbf{R}}_{\mathcal{C}_1}$;
$\mathcal{T}\leftarrow$ Assign to $\mathcal{T}$ the four triangular faces in $\mathcal{C}_1$: $\{v_1,v_2,v_3\}$, $\{v_1,v_2,v_4\}$, $\{v_1,v_4,v_3\}$, $\{v_4,v_2,v_3\}$;
$\mathcal{V}\leftarrow$ Assign to $\mathcal{V}$ the remaining $p-4$ vertices not in $\mathcal{C}_1$;
while $\mathcal{V}$ *is not empty* **do**
  find the combination of $\{v_a,v_b,v_c\}\in\mathcal{T}$ and $v_d\in\mathcal{V}$ with largest $\det(\hat{\mathbf{R}}_{\{v_a,v_b,v_c\}})/\det(\hat{\mathbf{R}}_{\{v_a,v_b,v_c,v_d\}})$;
  // $\{v_a,v_b,v_c,v_d\}$ is a new 4-clique $\mathcal{C}$, $\{v_a,v_b,v_c\}$ becomes a separator $\mathcal{S}$, three new triangular faces, $\{v_a,v_b,v_d\}$, $\{v_a,v_c,v_d\}$ and $\{v_b,v_c,v_d\}$ are created
  Remove v_d from $\mathcal{V}$;
  Remove $\{v_a,v_b,v_c\}$ from $\mathcal{T}$;
  Add $\{v_a,v_b,v_d\}$, $\{v_a,v_c,v_d\}$, $\{v_b,v_c,v_d\}$ to $\mathcal{T}$;
  Compute $J_{i,j}=J_{i,j}+\left(\mathbf{\Sigma}^{-1}_{\{v_a,v_b,v_c,v_d\}}\right)_{i,j}-\left(\mathbf{\Sigma}^{-1}_{\{v_a,v_b,v_c\}}\right)_{i,j}$;
end
return $\mathbf{J}$;

Let us note that by expanding to the second order in the correlation coefficients, the logarithms of the determinants are well approximated by a constant minus the sum of the square correlation coefficients associated with the edges in the cliques or separators. This can simplify the algorithm and the TMFG could be simply computed from a set of weights given by the squared correlation coefficients matrix, as described in Ref. [48].

Further, let us note that, for simplicity, in this paper we only consider likelihood maximization. A natural, straightforward extension of the present work is to consider Akaike information criterion [22,23] instead. However, we have verified that, for the cases studied in this paper, the two approaches give very similar results.

III. RESULTS

A. Inverse covariance estimation

We investigated stock prices time series from a U.S. equity market computing the daily log returns $(x_i(t)=\log\text{Price}_i(t)-\log\text{Price}_i(t-1)$ with $i=1,..,342$ and $t=1,...,T$ with $T=4025$ days during a period of 15 years from

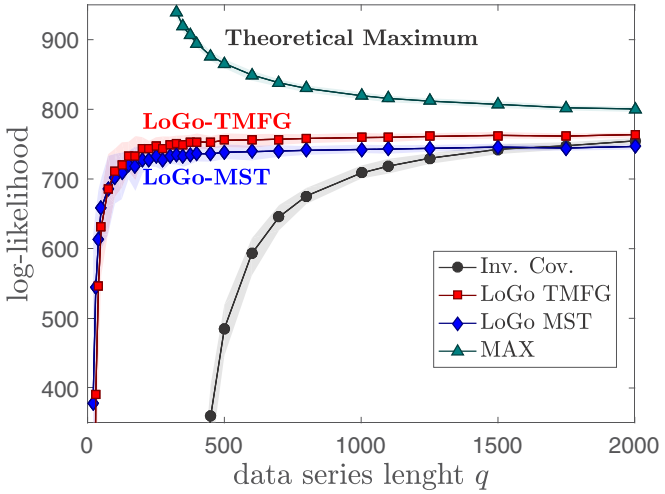


FIG. 5. Demonstration that LoGo sparse inverse covariance represents the dependency structure better than the complete inverse covariance. The figure reports comparisons between log-likelihood for models constructed by using sparse inverse LoGo-TMFG, LoGo-MST and the complete inverse of the empirical covariance matrix (Inv. Cov.). These measures are on $p = 300$ off-sample test data series of different lengths q varying from 20 to 2000. The inverse matrices are computed on training data sets of the same length. Data are log returns sampled from 342 stocks prices of equities traded on the U.S. market during the period 1997–2012. The statistics is made stationary by random shuffling the time order. Symbols correspond to averages over 100 samples generated by picking at random 300 series over the 342 and assigning training and testing sets by choosing at random two nonoverlapping time windows of length q, the shaded bands are the 95% quantiles. The line on the top, labeled "MAX," is the theoretical maximum, which is the log-likelihood obtained from the inverse covariance of the testing set.

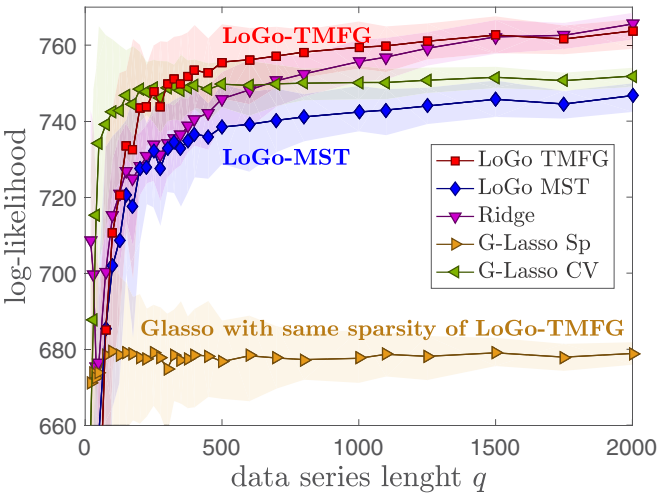


FIG. 6. Demonstration that LoGo sparse modeling has better performances than Glasso models with same sparsity. This figure reports the same log-likelihood as described in the caption of Fig. 5 compared with log-likelihood from state-of-the-art Glasso ℓ_1 penalized sparse inverse covariance models (cross validated, G-Lasso-CV, and of the same sparsity of TMFG, G-Lasso-Sp) and Ridge ℓ_2 penalized inverse model (Ridge).

1997 to 2012 [9]). We build 100 different data sets by creating stationary time series of different lengths selecting returns at random points in time and randomly picking $p = 300$ series out of the 342 in total. Each data set was divided into two temporal nonoverlapping windows with q elements constituting the "training set" and other q elements the "testing set."

In order to quantify the goodness of the methodology we computed the log likelihood of the testing data set using the inverse covariance estimates from the training set. Larger log-likelihood indicate models that better describe the testing data. Figure 5 reports the results for time series of different lengths from $q = 25$ to $q = 2000$. Smaller values of q mean shorter number of observations in the training data set used to construct the model and therefore correspond to larger uncertainties. Note that, the green upward triangles in Fig. 5, denoted with "MAX," are the theoretical maximum from the inverse sample covariance matrix calculated on the testing set, which is reported as a reference indicating the upper value for the attainable likelihood. Let us first observe from this figure that, for these stationary financial time series study, *LoGo-TMFG* outperforms the likelihood from the inverse covariance solution $\mathbf{J} = \hat{\Sigma}^{-1}$ (denoted with "Inv. Cov." in Fig. 5). For $q < p = 300$ the inverse covariance is not computable and therefore comparison cannot be made; when $q > p = 300$, the inverse covariance is computable but it performs very poorly for small sample sizes $q \sim p$ becoming comparable to *LoGo-TMFG* only after $q \sim 1500$ with both approaching the theoretical maximum at $q \to \infty$. Note that also *LoGo-MST* outperforms the inverse covariance solution in most of the range of q. We then compared the log-likelihood from *LoGo-MST* and *LoGo-TMFG* sparse inverse covariance with state-of-the-art Glasso ℓ_1-norm penalized sparse inverse covariance models (using the implementation by Ref. [57]) and Ridge ℓ_2-norm penalized inverse model. Glasso method depends on the regularization parameters which were estimated by using two standard methods: (i) *G-Lasso-CV* uses a twofold cross validation method [58]; (ii) *G-Lasso-Sp* fixes the regularization parameter to the value that creates in the training set a sparse inverse with sparsity equal to *LoGo-TMFG* network [$3(p-2)$ parameters]. *Ridge* inverse penalization parameter was also computed by cross validation method [58]. Figure 6 reports a comparison between these methods for various values of q. We can observe that *LoGo-TMFG* outperforms the Glasso methods achieving larger likelihood from $q > 100$. Results are detailed in Table I, where we compare also with the null model ("NULL"), which is a completely disconnected network corresponding to a diagonal $\mathbf{J}$.

LoGo models can achieve better performances than Glasso models with fewer coefficients and are computationally more efficient. This is shown in Fig. 7, where we report the comparison between the number of nonzero off-diagonal coefficients in the precision matrix $\mathbf{J}$ in Glasso-CV and LoGo models. These results show that the number of coefficients for G-Lasso-CV is 3 to 6 times larger than for LoGo-TMFG, while the computation time for LoGo-TMFG is about three orders of magnitude smaller than the computation time for G-Lasso-CV. Note that LoGo-TMFG has a constant number of coefficients equal to $3p - 6$ corresponding to the number of edges in the TMFG network. A further comparison between performance,

TABLE I. Demonstration that LoGo sparse modeling has better or comparable performances than state-of-the-art models. Comparison between log-likelihood for LoGo, Glasso, Ridge, Complete inverse, and Null models. Measures are on $p = 300$ off-sample test data series of lengths $q = 50, 100, 300, 500, 1000, 2000$. Data are the same as in Fig. 6: log returns sampled from 342 stocks prices made stationary by random shuffling the time order in the time series. The values reported are the averages of 100 samples and the standard deviations are reported between brackets. "MAX" is the theoretical maximum log-likelihood obtained from the inverse covariance of the testing set.

q	50	100	300	500	1000	2000
Inv. Cov.	–	–	112 (53)	485 (29)	709 (8)	755 (4)
LoGo TMFG	631 (52)	711 (28)	753 (9)	756 (8)	759 (5)	764 (3)
LoGo MST	658 (43)	702 (28)	737 (9)	738 (8)	742 (5)	747 (3)
Ridge	676 (17)	715 (11)	741 (6)	746 (6)	756 (5)	766 (3)
G-Lasso Sp	674 (23)	679 (18)	678 (10)	677 (8)	678 (5)	679 (2)
G-Lasso CV	734 (25)	743 (15)	750 (5)	750 (5)	750 (4)	752 (2)
MAX	–	–	895 (6)	866 (5)	820 (4)	801 (3)
NULL	−276 (0.02)	−276 (0.02)	−276 (0.01)	−276 (0.01)	−276 (0.01)	−276 (0.00)

sparsity and execution time is provided in Fig. 8, where the top plot reports the fraction of log-likelihood for Glasso (implemented by using [57]) and for LoGo-TMFG versus the fraction of nonzero off-diagonal coefficients of **J** for data series lengths of $q = 500$. In the bottom plot we report instead the fraction of computational time versus the fraction of nonzero off-diagonal coefficients of **J** for Glasso and for LoGo-TMFG. We can observe that at the same sparsity (value 1 in the x axis indicated with the vertical line) Glasso underperforms LoGo by 10% and Glasso is about 50 times slower than LoGo on the same machine. We verified that eventually the maximum performance of Glasso can become 1.5% better than LoGo but with 10 times more parameters and a computation time 2000 times longer. Let us note that, in this example, the best Glasso performance are measured *a posteriori* on the testing set, they are therefore hypothetical maxima, which cannot be reached with cross validation methods that instead result in average performances of a few percent inferior to LoGo (see Fig. 6). All these results refer to the same simulations as for the results in Figs. 5. The computation time of LoGo can be decreased even further by parallelizing the algorithm.

The previous results are for time series made stationary by random selecting log returns at different times. In practice,

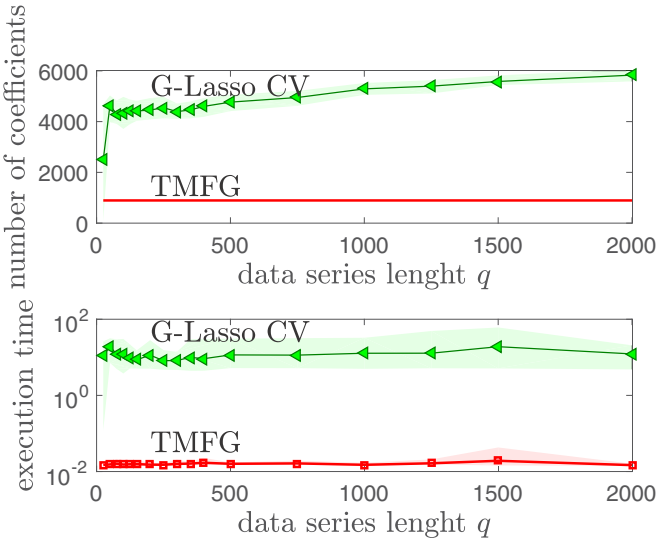


FIG. 7. Demonstration that LoGo models are sparser than Glasso models and are computationally more efficient. The plot on the top reports the number of nonzero off-diagonal coefficients in the precision matrix **J** for G-Lasso-CV and LoGo-TMFG. The plot on the bottom reports the computation times (seconds) for G-Lasso-CV and LoGo-TMFG. These data refer to the same simulations as for the results in Figs. 5 and 6. Note that TMFG-LoGo has a constant number of coefficients equal to $3(p-2) = 894$.

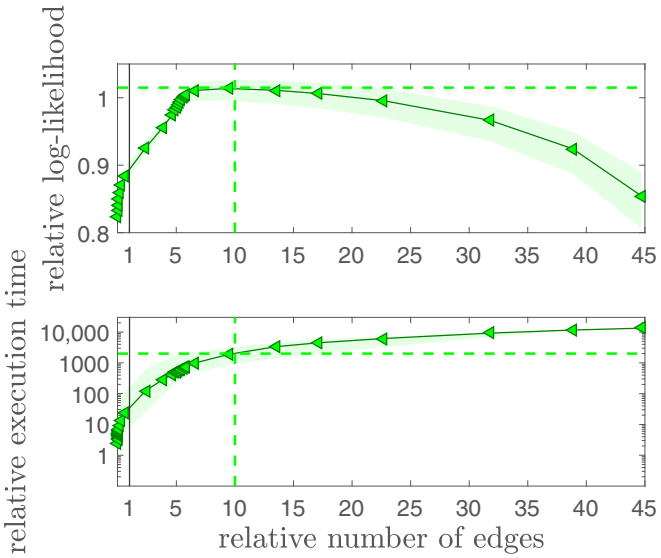


FIG. 8. Demonstration that LoGo sparse models with comparable performances with respect to best-performing Glasso methods produces sparser models in a fraction of the computation time. The top plot reports the fraction of log-likelihood for Glasso and for LoGo-TMFG vs. the fraction of nonzero off-diagonal coefficients of **J** for Glasso and for LoGo-TMFG. The bottom plot reports the fraction of computational time vs. the fraction of nonzero off-diagonal coefficients of **J** for Glasso and for LoGo-TMFG. The measures are on $p = 300$ off-sample test data series of length $q = 500$. Inverse matrices are computed on training data sets of the same length. Data are log returns sampled from 342 stocks prices of equities traded on the U.S. market during the period 1997–2012. The statistics is made stationary by random shuffling the time order. Symbols correspond to averages over 100 samples generated by picking at random 300 series over the 342 and assigning training and testing sets by choosing at random two nonoverlapping time windows of length q; the shaded bands are the 95% quantiles.

TABLE II. LoGo performances on historic data. Same analysis as in Table I performed for historic data where the training sets are past log returns and the testing sets are future log returns from two nonoverlapping adjacent windows of length q.

q	50	100	300	500	1000	2000
Inv. Cov.	–	–	–	253 (835)	481 (335)	678 (15)
LoGo TMFG	679 (212)	757 (120)	674 (247)	665 (312)	605 (211)	694 (12)
LoGo MST	699 (202)	747 (120)	656 (255)	641 (324)	583 (221)	678 (11)
Ridge	753 (113)	746 (86)	721 (113)	721 (155)	684 (115)	710 (10)
G-Lasso Sp	722 (110)	704 (107)	664 (113)	627 (159)	625 (97)	659 (4)
G-Lasso CV	769 (134)	761 (89)	718 (110)	700 (178)	666 (120)	711 (6)
MAX	–	–	–	919 (63)	869 (40)	851 (3)
NULL	−276 (0.14)	−276 (0.07)	−276 (0.08)	−276 (0.07)	−276 (0.06)	−276 (0.00)

financial time series—and other real-world signals—are nonstationary having statistical properties that change with time. Table II reports the same analysis as in Fig. 6 and Table I but with data sets taken in temporal sequence with the training set being the q data points preceding the testing set. Let us note that, considering the time period analyzed, in the case of large q, the training set has most data points in the period preceding the 2007–2008 financial crisis, whereas the testing set has data in the period following the crisis. Nonetheless, we see that the results are comparable with the one obtained for the stationary case. Surprisingly, we observe that for relatively small time-series lengths the values of the log-likelihood achieved by the various models is larger than in the stationary case. This counterintuitive fact can be explained by the larger temporal persistence of real data with respect to the randomized series. Further results and a plot of the log likelihood for this nonstationary case are given in Appendix A (Fig. 9).

We also investigated artificial data sets of $p = 300$ multivariate variables generated from factor models with 3 and 30 common factors, respectively. Results for the average log-likelihood and the standard deviations computed over 100 samples at different values of q are reported in Table III. In this case we note that while for a number of factors equal to 3, LoGo is again performing consistently better than the inverse covariance and better than Glasso models of equal sparsity and comparably well with cross-validated Glasso; conversely, when the number of factors is set to 30, LoGo is underperforming with respect to cross-validated Glasso and even the inverse covariance (for $q > 400$). However, we note that LoGo is doing better than the Glasso model with equal sparsity. This seems to indicate that factor models with more

TABLE III. LoGo performances on artificial data. Same analysis as in Table I performed for artificial data generated with two factor models with 3 (a) and 30 (b) factors, respectively.

(a)				
q	50	400	1000	2000
Inv. Cov.	–	−394 (14)	−63 (17)	−46 (18)
LoGo TMFG	−98 (13)	−51 (17)	−51 (18)	−58 (20)
LoGo MST	−142 (10)	−116 (13)	−116 (11)	−121 (13)
Ridge	−163 (8)	−82 (16)	−53 (15)	−48 (17)
G-Lasso Sp	−452 (9)	−447 (4)	−447 (3)	−447 (2)
G-Lasso CV	−128 (7)	−62 (13)	−61 (11)	−55 (15)
MAX	–	59 (18)	2 (17)	−20 (18)
NULL	−427 (12)	−425 (6)	−426 (3)	−425 (2)
(b)				
q	50	400	1000	2000
Inv. Cov.	–	−484 (12)	−143 (12)	−115 (11)
LoGo TMFG	−440 (7)	−376 (2)	−373 (1)	−372 (2)
LoGo MST	−432 (5)	−400 (3)	−393 (1)	−392 (1)
Ridge	−324 (11)	−158 (9)	−129 (12)	−113 (11)
G-Lasso Sp	−430 (3)	−425 (1)	−424 (1)	−424 (1)
G-Lasso CV	−326 (5)	−151 (6)	−130 (14)	−118 (11)
MAX	–	−18 (9)	−78 (12)	−88 (11)
NULL	−425 (4)	−426 (1)	−425 (1)	−426 (1)

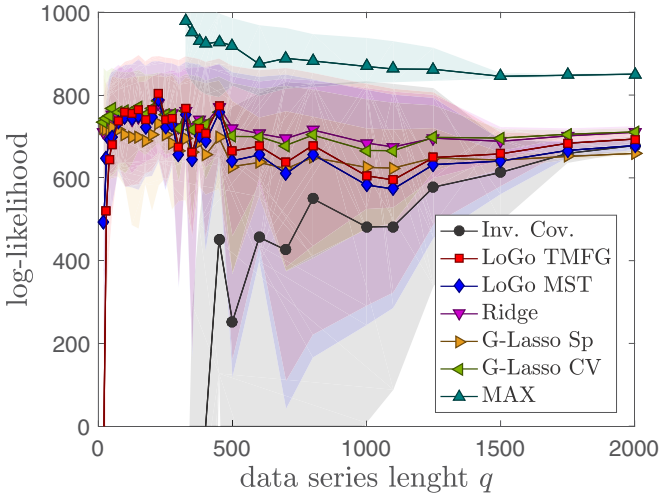


FIG. 9. Demonstration that also for nonstationary financial data LoGo sparse inverse covariance represents the dependency structure better than the complete inverse covariance. The figure reports comparisons between log-likelihood for models constructed by using sparse inverse LoGo-TMFG, LoGo-MST, and the complete inverse of the empirical covariance matrix (Inv. Cov.). The measures are on $p = 300$ off-sample test data series of different lengths q varying from 20 to 2000. Inverse matrices are computed on training data sets of the same length. Data are log returns sampled from 342 stocks prices of equities traded on the U.S. market during the period 1997–2012. Symbols correspond to averages over 100 samples generated by picking at random 300 series over the 342 and assigning training and testing sets by choosing at random two consecutive nonoverlapping time windows of length q; the shaded bands are the 95% quantiles. Testing set is the time window preceding the training set. The line on the top, labeled "MAX," is the theoretical maximum, which is the log-likelihood obtained from the inverse covariance of the testing set.

than a few factors require denser models with larger numbers of nonzero elements in the inverse covariance than the one provided by MST and TMFG networks used in the present LoGo construction. Note that by increasing the number of factors, performances of all models become worse. Let us finally note that ridge inverse covariance performs well in all the cases studied. It is indeed well known that this is a powerful estimator for the inverse covariance; however, the purpose of the present investigation concerns sparse modeling and ridge inverse covariance is dense with all coefficients different from zero. Furthermore, let us remark that LoGo can outperform ridge in several cases, as we can notice from Fig. 6 and Tables I, II, and III. Further results and a plot of the log-likelihood for factor models are given in Appendix B (Fig. 10).

B. Time series prediction: LoGo for regressions and causality

LoGo estimates the joint probability distribution function yielding the set of parameters for the model system's dependency structure. In this section we demonstrate how this model can be used also for forecasting. Information filtering networks have been proven to be powerful tools to characterize the structure of complex systems comprising several variables—such as financial markets and biological systems [6,7,9,47,65]. They have also been proven effective to understand how financial risk is distributed in markets and how to construct performing portfolios [8,10]. However, so far, they were not associated to probabilistic models able to make use of their capability of meaningful representation of the market structure. LoGo provides this instrument and in particular we here show how information filtering networks can be used to compute sparse regressions. Indeed, generally speaking, a regression consists in estimating the expectation values of a set of variables, $\boldsymbol{X}_2$, conditioned to the values of another set of variables $\boldsymbol{X}_1$:

$$\boldsymbol{\mu}_{2|1} = \mathbb{E}[\boldsymbol{X}_2|\boldsymbol{X}_1]. \tag{7}$$

When multivariate normal statistics is used, this is the linear regression. If LoGo sparse inverse covariance $\mathbf{J}$ is used, then Eq. (7) computes a sparse linear regression. If the set of variables $\boldsymbol{X}_1$ are "past" variables and the variables $\boldsymbol{X}_2$ are "future" variables, then the regression becomes a forecasting instrument where values of variables in the future can be inferred from past observations. Here we consider to have p_1 variables in the past ($\boldsymbol{X}_1$) and p_2 variables in the future ($\boldsymbol{X}_2$). These variables can either be the same variables at different lags or different variables. For simplicity of notation, and without loss of generality, we consider centered variables with zero expectation values. We can consider $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ as two distinct sets of variables that, of course, have some dependency relation. The conditional expectation values $\boldsymbol{\mu}_{2|1}$ can be calculated from the conditional joint distribution function, which, from the Bayes theorem, is $f(\boldsymbol{X}_2|\boldsymbol{X}_1) = f(\boldsymbol{X}_2,\boldsymbol{X}_1)/f(\boldsymbol{X}_1)$. From this expression one obtains

$$\mathbf{J}_{2,2}\boldsymbol{\mu}_{2|1} = -\mathbf{J}_{2,1}\boldsymbol{X}_1, \tag{8}$$

where we have written the precision matrix $\mathbf{J}$ as a block matrix,

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_{1,1} & \mathbf{J}_{1,2} \\ \mathbf{J}_{2,1} & \mathbf{J}_{2,2} \end{pmatrix}, \tag{9}$$

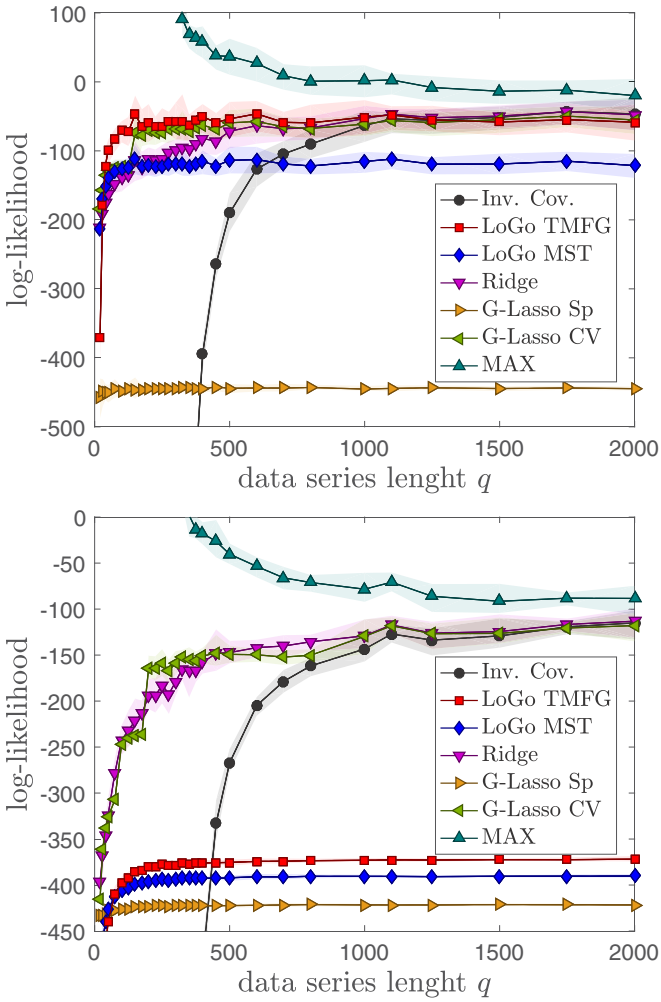


FIG. 10. Demonstration that also for sparse factor models LoGo sparse inverse covariance can perform well when the number of factors is low but it underperforms for larger number of factors. The figure reports comparisons between log-likelihood for models constructed by using sparse inverse LoGo-TMFG, LoGo-MST and the complete inverse of the empirical covariance matrix (Inv. Cov.). The measures are on $p = 300$ off-sample test data series of different lengths q varying from 20 to 2000. Inverse matrices are computed on training data sets of the same length. Plots refer to sparse factor models simulations with 3 (top) and 30 (bottom) factors, respectively. Symbols correspond to averages over 100 samples generated by picking at random 300 series over the 342 and assigning training and testing sets by choosing at random two consecutive nonoverlapping time windows of length q, the shaded bands are the 95% quantiles. Testing set is the time window preceding the training set. The line on the top, labeled "MAX," is the theoretical maximum, which is the log-likelihood obtained from the inverse covariance of the testing set.

with $\mathbf{J}_{2,2}$ being the $p_2 \times p_2$ part of the precision matrix in the lower right and $\mathbf{J}_{2,1}$ the $p_2 \times p_1$ part of the precision matrix in the lower left. As pointed out previously, Eq. (8) is just a different way to write the linear regression, which, in a more conventional form, reads $\boldsymbol{X}_2 = \boldsymbol{\beta}\boldsymbol{X}_1 + \boldsymbol{\epsilon}$ with the coefficients $\boldsymbol{\beta} = -\mathbf{J}_{2,2}^{-1}\mathbf{J}_{2,1}$ and the residuals given by $\boldsymbol{\epsilon} = \boldsymbol{X}_2 - \boldsymbol{\mu}_{2|1}$. However, by using LoGo to estimate $\mathbf{J}$, Eq. (8) becomes a sparse predictive model associated with a meaningful

inference structure. Now, through Eq. (8) we can quantify the effect of past values of a set of variables over the future values of another set. Indeed, Eq. (8) is a map describing the impact of variables in the past onto the future; nonzero elements of $\mathbf{J}_{2,1}$ single out the subset of variables of $\boldsymbol{X}_1$ that has direct future impact on a subset of variables of $\boldsymbol{X}_2$. With LoGo we can identify the channels that spread instability through the network and we can quantify their effects.

Risk and systemic vulnerability are described instead by the structure of $\mathbf{J}_{2,2}$. Indeed, the expected conditional fluctuations of the variables $\boldsymbol{X}_2$ are quantified by the conditional covariance:

$$\mathbf{Cov}(\boldsymbol{X_2}|\boldsymbol{X_1}) = \mathbf{J}_{2,2}^{-1}, \tag{10}$$

which involves the term $\mathbf{J}_{2,2}$ only, which therefore describes propagation of uncertainty across variables. Through this term we can link information filtering network with causality relations; indeed, Granger causality and transfer entropy are both associated to the ratio of the determinants of two conditional covariances between past and future variables [63,64]. This introduces a way to associate causal directionality to information filtering networks.

C. Financial applications: Stress testing and risk allocation

1. Financial applications: Stress testing

A typical stress test for financial institutions, required by regulatory bodies, consists in forecasting the effect of severe financial and economic shocks on the balance sheet of a financial institution. In this context let's reformulate the previous results by considering $\boldsymbol{X}_1$ the set of economic and financial variables that can be shocked and $\boldsymbol{X}_2$ the set of the securities held in an institution's portfolio. Assuming that all the changes in the economic and financial variables and in the assets of the portfolio can be modeled as a GMRF, then Eq. (8) represents the distribution of the returns of the portfolio ($\boldsymbol{X}_2$) conditional on the realization of the economic and financial shocks ($\boldsymbol{X}_1$). An approach along similar lines was proposed in Refs. [11,59]. We note that with the LoGo approach we have a sparse relationship between the financial variables and the securities. This makes calibration more robust and it can be insightful to identify mechanisms of impact and vulnerabilities.

2. Risk allocation

A second application is the calculation of conditional statistics in the presence of linear constraints (see Ref. [2]). In this case we indicate with $\boldsymbol{X}$ a set of p random variables associated with the returns in a portfolio of p assets and with $\mathbf{J}$ the associated sparse inverse covariance matrix. Let $\mathbf{w} \in \mathbb{R}^{p\times 1}$ be the vector of holdings of the portfolio, then $\mathbf{w}^{\mathsf{T}} \cdot \boldsymbol{X}$ is the return of the portfolio. An important question in portfolio management is to allocate profits and losses to different assets conditional on a given level of profit or loss, which is equivalent to knowing the distribution of returns conditional on a given level of loss $\boldsymbol{X}|\mathbf{w}^{\mathsf{T}} \cdot \boldsymbol{X} = \boldsymbol{L}$. More generally we want to estimate $\boldsymbol{X}|\boldsymbol{A} \cdot \boldsymbol{X} = \boldsymbol{z}$ where $\boldsymbol{A} \in \mathbb{R}^{k\times p}$ is generally a low rank k ($k = 1$ in our example) matrix that specifies k hard linear constraints. Using the Lagrange multipliers method (see Ref. [60] for an introduction), the conditional mean is calculated as ([2])

$$\mathbb{E}(\boldsymbol{X}|\boldsymbol{A} \cdot \boldsymbol{X} = \boldsymbol{z}) = \boldsymbol{A}\mathbf{J}^{-1}(\boldsymbol{A}\mathbf{J}^{-1}\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{z}, \tag{11}$$

and the conditional covariance is

$$\mathbf{Cov}(\boldsymbol{X}|\boldsymbol{A} \cdot \boldsymbol{X} = \boldsymbol{z}) = \mathbf{J}^{-1} - \mathbf{J}^{-1}\boldsymbol{A}^{\mathsf{T}}(\boldsymbol{A}\mathbf{J}^{-1}\boldsymbol{A}^{\mathsf{T}})^{-1}\boldsymbol{A}\mathbf{J}^{-1}. \tag{12}$$

In case $\mathbf{J}$ is estimated using decomposable information filtering networks (MST or TMFG), then it can be written as a sum of smaller matrices (as in algorithm I) involving cliques and separators:

$$\mathbf{J} = \sum_{\mathcal{C}\in \text{Cliques}} \mathbf{J}_{\mathcal{C}} - \sum_{\mathcal{S}\in \text{Separators}} \mathbf{J}_{\mathcal{S}}. \tag{13}$$

This decomposition allows for a sparse and potentially parallel evaluation of the matrix products in Eqs. (11) and (12).

This framework can therefore be used to build the profit and loss (P/L) distribution of a portfolio, conditionally on a number of explanatory variables, and to allocate the P/L to the different assets conditional on the realization of a given level of profit and loss. The solution is analytical and therefore extremely quick. Besides, given the decomposability of the portfolio, Eq. (13) allows us to calculate important statistics in parallel, by applying the calculations locally to the cliques and to the separators. For instance, it is a simple exercise to show that the unconditional expected P/L and the unconditional volatility can be calculated in parallel by adding the contributions of the cliques and subtracting the contributions of the separators. In summary, LoGo provides the possibility to build a basic risk management framework that allows risk aggregation, allocation, stress testing, and scenario analysis in a multivariate Gaussian framework in a quick and potentially parallel fashion.

IV. CONCLUSIONS

We have introduced a methodology, LoGo, that makes use of information filtering networks to produce probabilistic models that are sparse, are associated with high likelihood, and are computationally fast, making possible their use with very large data sets. It has been established that information filtering networks produce sparse structures that well reflect the properties of the underlying complex system [6]; however, so far, information filtering networks have been only used for descriptive purposes; now LoGo provides us an instrument to build predictive models from information filtering networks opening an entirely new range of perspectives that we have just started exploring.

LoGo produces high-dimensional sparse inverse covariances by using low-dimensional local inversions only, making the procedure computationally very efficient and little sensitive to the curse of dimensionality. The construction through a sum of local inversion, which is at the basis of LoGo, makes this method very suitable for parallel computing and dynamical adaptation by local, partial updating. We discussed the wide potential applicability of LoGo for sparse inverse covariance estimation, for sparse forecasting models, and for financial applications, such as stress-testing and risk allocation.

By comparing the results of LoGo modeling for financial data with a state-of-the-art Glasso procedure, we demonstrated that LoGo can obtain, in a fraction of the computation time, equivalent or better results with a sparser network structure.

However, we observed that when applied to factor models with more than a few factors, LoGo is underperforming with respect to less sparse or dense models. This is probably the consequence of the present LoGo implementations that use information filtering networks (MST and TMFG) with constrained sparsity (respectively, $p-1$ and $3p-6$ edges). Such a limitation can be easily overcome by constructing more complex networks with larger maximum cliques and a larger number of edges. A natural extension beyond MST and TMFG would be to use, instead of the present greedy local likelihood maximization under topological constraints, an information criterion (such as Akaike information criterion [22,23]), which let a chordal graph be constructed through local moves without constraining *a priori* its final topological properties. This would produce clique forests, which generalize the MST and TMFG studied in this paper. Further extensions could be along the lines of the biologically motivated work [29] where ensemble of inference network were explored. These extensions could increase model robustness; this, however, would be unavoidably at the expense of computational efficiency. The tradeoff between efficiency and performance for the choice of information filtering networks for LoGo will be the topic of future investigations.

The model introduced in this paper is a second-order solution of the maximum entropy problem, resulting in linear, normal multivariate modeling. It is, however, well known that many real systems follow nonlinear probability distributions. Linearity is a severe limitation that can, however, be overcome in several ways. For instance, LoGo can be extended to a much broader class of nonlinear models by using the so-called kernel trick [61]. Other generalizations to nonlinear transelliptical models [62] can also be implemented. Furthermore, the probability decomposition at the basis of LoGo [Eq. (2)] is general and can be even applied to nonparametric, nonlinear models. These would be, however, the topics of future works.

ACKNOWLEDGMENTS

T.A. acknowledges support of the UK Economic and Social Research Council (ESRC) in funding the Systemic Risk Centre (Grant No. ES/K002309/1). T.D.M. thanks the COST Action TD1210 for partially supporting this work. We acknowledge Bloomberg for providing the data.

APPENDIX A: COMPARISON BETWEEN LOGO AND STATE-OF-THE-ART SPARSE GLASSO MODEL FOR NONSTATIONARY FINANCIAL DATA

We investigated the comparison between LoGo and state-of-the-art sparse Glasso model from Ref. [57] for the same financial data used for Fig. 6 and Table I but using the real temporal sequence of the financial data. These sequences are nonstationary having varying statistical properties across the time windows. This unavoidably must affect the capability of the model to describe statistically test data from the study of the training data being the two associated with different temporal states where different events affect the market dynamics. Results for the log-likelihood are reported in Fig. 9, these are the same results reported in Table II. By comparing with Fig. 6 we observe a much greater overall noise with an interesting collapse of performances with similar values for all models. We also observe larger log-likelihood values for shorter time windows. This fact is commented on in Sec. III. Notice that these results are in agreement with the finding for the stationary case with LoGo still performing better or comparably well with respect to Glasso-CV. Computational times and sparsity value are very similar to what was reported for the stationary case.

APPENDIX B: COMPARISON BETWEEN LOGO AND STATE-OF-THE-ART SPARSE GLASSO MODEL FOR SPARSE FACTOR MODELS

Plots for the log-likelihood versus the time-series length for factor models are reported in Fig. 10, which correspond to the results reported in the paper in Table II. We observe that LoGo still performs well and relatively similar to the real data performances when the number of factors is small. Conversely, when the number of factors increases, LoGo underperforms even the inverse correlation for $q > 400$. Let us note that LoGo is still over performing the Glasso with the same number of parameters.

[1] S. L. Lauritzen, *Graphical Models* (Clarendon, Oxford, 1996).
[2] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications*, Vol. 104 of Monographs on Statistics and Applied Probability (Chapman & Hall, London, 2005).
[3] T. A. Davis, *Direct Methods for Sparse Linear Systems*, Vol. 2 (Siam, Philadelphia, 2006).
[4] C. Wang, N. Komodakis, and N. Paragios, Comput. Vis. Image Underst. **117**, 1610 (2013).
[5] A. Montanari, in *Compressed Sensing: Theory and Applications*, edited by Y. C. Eldar and G. Kutyniok (Cambridge University Press, Cambridge, 2012), p. 394.
[6] M. Tumminello, T. Aste, T. Di Matteo, and R. N. Mantegna, Proc. Natl. Acad. Sci. USA **102**, 10421 (2005).
[7] T. Aste, W. Shaw, and T. Di Matteo, New J. Phys. **12**, 085009 (2010).
[8] F. Pozzi, T. Di Matteo, and T. Aste, Sci. Rep. **3**, 1665 (2013).
[9] N. Musmeci, T. Aste, and T. Di Matteo, PLoS ONE **10**, e0116201 (2015).
[10] N. Musmeci, T. Aste, and T. Di Matteo, J. Network Theory Finance **1**, 77 (2015).
[11] A. Denev, *Probabilistic Graphical Models: A New Way of Thinking in Financial Modeling* (Risk Books, New York, 2015).
[12] X. Wu, Y. Ye, K. R. Subramanian, and L. Zhang, in *BIOKDD*, ACM SIGKDD Explorations Newsletter (ACM, New York, 2003), Vol. 3, p. 63.
[13] J. Schäfer and K. Strimmer, Bioinformatics **21**, 754 (2005).

[14] T. R. Lezon, J. R. Banavar, M. Cieplak, A. Maritan, and N. V. Fedoroff, Proc. Natl. Acad. Sci. USA **103**, 19033 (2006).
[15] E. Schneidman, M. J. Berry, R. Segev, and W. Bialek, Nature **440**, 1007 (2006).
[16] T. Zerenner, P. Friederichs, K. Lehnertz, and A. Hense, Chaos: Interdisc. J. Nonlin. Sci. **24**, 023103 (2014).
[17] I. Ebert-Uphoff and Y. Deng, Geophys. Res. Lett. **39**, 1 (2012).
[18] M. Haran, *Handbook of Markov Chain Monte Carlo* (Chapman & Hall/CRC, Boca Raton, 2011), p. 449.
[19] D. T. Hristopulos and S. N. Elogne, IEEE Trans. Signal Process. **57**, 3475 (2009).
[20] D. T. Hristopulos, SIAM J. Sci. Comput. **24**, 2125 (2003).
[21] Y. Zhou, arXiv:1111.6925.
[22] H. Akaike, IEEE Trans. Auto. Control **19**, 716 (1974).
[23] H. Akaike, in *Selected Papers of Hirotugu Akaike* (Springer, Berlin, 1998), p. 199.
[24] S. Kullback and R. A. Leibler, Ann. Math. Stat. **22**, 79 (1951).
[25] G. Schwarz *et al.*, Ann. Stat. **6**, 461 (1978).
[26] J. Rissanen, Automatica **14**, 465 (1978).
[27] F. Petitjean, G. Webb, A. E. Nicholson *et al.*, in *IEEE 13th International Conference on Data Mining (ICDM), 2013* (IEEE, Dallas, Texas, 2013), p. 597.
[28] L. Diambra Physica A: Stat. Mech. Appl. **390**, 2198 (2011).
[29] Evan J. Molinelli *et al.*, PLoS Comput Biol. **9**, e1003290 (2013).
[30] P. Giudici and P. J. Green, Biometrika **86**, 785 (1999).
[31] A. Deshpande, M. Garofalakis, and M. I. Jordan, in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence* (Morgan Kaufmann Publishers Inc., New York, 2001), p. 128.
[32] F. R. Bach and M. I. Jordan, in *Advances in Neural Information Processing Systems 14* (MIT Press, Cambridge, MA, 2001), p. 569.
[33] N. Srebro, in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence* (Morgan Kaufmann Publishers Inc., New York, 2001), p. 504.
[34] A. Chechetka and C. Guestrin, in *Advances in Neural Information Processing Systems 20* (MIT Press, Cambridge, MA, 2008), p. 273.
[35] A. d'Aspremont, O. Banerjee, and L. El Ghaoui, SIAM J. Matrix Anal. Appl. **30**, 56 (2008).
[36] O. Banerjee, L. El Ghaoui, and A. d'Aspremont, J. Machine Learning Res. **9**, 485 (2008).
[37] O. Banerjee, L. E. Ghaoui, A. d'Aspremont, and G. Natsoulis, in *Proceedings of the 23rd International Conference on Machine Learning* (ACM, New York, 2006), p. 89.
[38] R. Tibshirani, J. Roy. Stat. Soc. Ser. B **58**, 267 (1996).
[39] H. Zou and T. Hastie, J. Roy. Stat. Soc. Ser. B **67**, 301 (2005).
[40] N. Meinshausen and P. Bühlmann, Ann. Stat. **34**, 1436 (2006).
[41] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, Electronic J. Stat. **5**, 935 (2011).
[42] C. Hsieh, I. S. Dhillon, P. K. Ravikumar, and M. A. Sustik, in *Advances in Neural Information Processing Systems 24*, edited by J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger (Curran Associates, Inc., Cambridge, MA, 2011), p. 2330.
[43] F. Oztoprak, J. Nocedal, S. Rennie, and P. A. Olsen, in *Advances in Neural Information Processing Systems 25*, edited by P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K. Weinberger (MIT Press, Cambridge, MA, 2012), p. 755.
[44] J. Friedman, T. Hastie, and R. Tibshirani, Biostatistics **9**, 432 (2008).
[45] R. N. Mantegna, Eur. Phys. J. B Condensed Matter Complex Sys. **11**, 193 (1999).
[46] T. Aste and T. Di Matteo, Physica A **370**, 156 (2006).
[47] W.-M. Song, T. Di Matteo, and T. Aste, PLoS ONE **7**, e31929 (2012).
[48] G. P. Massara, T. Di Matteo, and T. Aste, J. Complex Networks (2016), arXiv:1505.02445.
[49] E. T. Jaynes, Phys. Rev. **106**, 620 (1957).
[50] E. T. Jaynes, Phys. Rev. **108**, 171 (1957).
[51] H. Hotelling, J. Roy. Stat. Soc. Ser. B **15**, 193 (1953).
[52] A. P. Dempster, Biometrics **28**, 157 (1972).
[53] R. Diestel, *Graph Theory*, vol. 173 of *Graduate Texts in Mathematics* (Springer, Heidelberg, 2010).
[54] P. G. Hoel *et al.*, *Introduction to Mathematical Statistics* (John Wiley & Sons, Inc., New York, 1954).
[55] R. Prim, Bell Syst. Tech. J. **36**, 1389 (1957).
[56] J. B. Kruskal, Jr., Proc. Am. Math. Soc. **7**, 48 (1956).
[57] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. D. Ravikumar, J. Machine Learning. **15**, 2911 (2014).
[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, J. Machine Learning Res. **12**, 2825 (2011).
[59] R. Rebonato, *Coherent Stress Testing* (John Wiley & Sons, Inc., Hoboken, NJ, USA, 2010).
[60] G. Strang, *Introduction to Applied Mathematics* (Wellesley-Cambridge Press, Cambridge, 1986).
[61] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University Press, Cambridge, 2004).
[62] H. Liu, F. Han, and C.-h. Zhang, in *Advances in Neural Information Processing Systems 2* (MIT Press, Cambridge, MA, 2012), p. 809.
[63] C. W. J. Granger, J. Econometr. **39**, 199 (1988).
[64] L. Barnett, A. B. Barrett, and A. K. Seth, Phys. Rev. Lett. **103**, 238701 (2009).
[65] N. Musmeci, T. Aste, and T. Di Matteo, Sci. Rep. **6**, 36320 (2016).