51st IEEE Conference on Decision and Control
December 10-13, 2012. Maui, Hawaii, USA

Network reconstruction of dynamical polytrees with unobserved nodes

Donatello Materassi Murti V. Salapaka

Abstract— **The paper deals with the problem of unveiling the link structure of a network of linear dynamical systems. A technique is provided guaranteeing an exact detection of the links of a network of dynamical systems with no undirected cycles (Linear Dynamic Polytrees). In particular, the presence of unobserved (latent) nodes is taken into account. Knowledge on the specific number of hidden processes is not required. It is proven that the topology can be consistently reconstructed, as long the degree of each latent node is at least three with outdegree of at least two. The result extends previous work that was limited to a more restricted class of dynamical systems (Rooted Trees).**

I. Introduction

In many diverse areas, determining cause-effect relationships among various entities in a network is of significant interest. Interconnections of simple systems are used to understand the emergence of complicated phenomena (see, for example, [1]) and have provided novel modeling approaches in many fields, such as Economics (see e.g. [2]), Biology (see e.g. [3]), Cognitive Sciences (see e.g. [4]), Ecology (see e.g. [5]) and Geology (see e.g. [6]), especially when the investigated phenomena are characterized by spatial distributions where a multivariate analysis is involved.

Given the widespread interest in unraveling the interconnectedness of complex networks, the necessity for general tools is rapidly increasing (see [7] and [8] and the bibliography therein for recent results). Indeed, even though there is considerable work in this area (see [7], [8], [9], [10]), deriving information about a network topology remains a formidable task with many theoretical and practical challenges [11].

Most present techniques offer methods to identify a network structure based on heuristic considerations, where theoretical guarantees about the correctness of the reconstruction are usually not provided. For example, in [7] different techniques for quantifying and evaluating the modular structure of a network are compared and a new one is proposed trying to combine both the topological and dynamic information of the complex system. However, the network topology is only qualitatively estimated.

In this paper we address the problem of reconstructing a network of dynamical systems using only passive observations. One of the most difficult challeges in the reconstruction of a network of dynamical systems is given by the intractability presented by cycles. This is the reason why most techniques focus on identifying acyclic structures (see for example [12], [3], [2], [13]). However, even though an acyclic topology may seem quite a reductive choice, given an intricate and connected link structure, one may be interested in "approximating" it with a tree scheme. Such an approximation could be considered "satisfactory" if the most important connections were captured.

For example, tree topologies have been successfully employed in [3] for the study of gene regulatory networks even though the underlying structure is considerably more complicated.

Another such an application is in the identification of a tree network in a complex scenario is developed in [2] for the analysis of a stock portfolio. The authors identify a tree structure according to the following procedure: i) a metric based on the correlation index is defined among the nodes; ii) such a metric is employed to extract the Minimum Spanning Tree [14] which yields the reconstructed topology. In [15] limitations of these strategies are highlighted, where it is shown that, even though the actual network is a tree, the presence of dynamical connections or delays can lead to the identification of a wrong topology. In [13] a similar strategy, where the correlation metric is replaced by a metric based on the coherence function, is numerically shown to provide an exact reconstruction for rooted tree topologies.

One important issue in the identification of structures of dynamical systems is given by the processes that can be observed. Indeed, in many scenarios it is not possible to obtain measurements from all the processes involved in a complex system and part of the variables can be latent. This scenario, in the case of a structure of random variables has been investigated in [16] and, more recently, in [17] in the case of rooted trees.

In particular the results in [17] provide a foundational basis for this article. In this paper we extend theoretical guarantees that were provided in [17] for rooted tree networks of random variables. We show that a modified version of coherence metric used in [13] provides an exact reconstruction of polytrees of dynamical systems. Thus, the main contribution of the paper is in the synthesis of the results of [13] and [17].

The paper is organized as follows. In Section II we provide the preliminary definitions; in Section III we give a formal description of the problem; in Section IV we provide the preliminary notions necessary for the development of the main result in Section V, namely an algorithm for the reconstruction of polytrees of dynamical systems; in Section VI we illustrate how such an algorithm works using simple examples.

Notation

- $E[\cdot]$: expectation operator
- $\mathcal{Z}\{\cdot\}$: z-transform operator
- $\Phi_{xy}(z)$: cross power spectral density of two jointly stationary stochastic vectors
  $\Phi_{xy}(z) := \mathcal{Z}\{E[x(0)y^T(\tau)]\}$

Donatello Materassi is with Laboratory for Information and Decision Systems, Massachusetts Institute of Technology

Murti Salapaka is with Department of Electrical and Computer Engineering, University of Minnesota.

978-1-4673-2066-5/12/$31.00 ©2012 IEEE

- $\Phi_x(z) := \Phi_{xx}(z)$
- $\delta^-(x)$: the indegree of a node x in a graph
- $\delta^+(x)$: the outdegree of a node x in a graph
- $\delta(x) := \delta^-(x) + \delta^+(x)$: the degree of a node x in a graph

II. Preliminary definitions and notions

The following three definitions are functional to the development of theoretical results.

Definition 1: Let $\mathcal{E}$ be a set containing time-discrete scalar, zero-mean, jointly wide-sense stationary random processes such that, for any $e_i, e_j \in \mathcal{E}$, the power spectral density $\Phi_{e_i e_j}(z)$ exists, is real rational with no poles on the unit circle and given by

$$\Phi_{e_i e_j}(z) = \frac{A(z)}{B(z)},$$

where $A(z)$ and $B(z)$ are polynomials with real coefficients such that $B(z) \neq 0$ for any $z \in \mathbf{C}$, with $|z| = 1$. Then, $\mathcal{E}$ is a set of rationally related random processes.

Definition 2: The set $\mathcal{F}$ is defined as the set of real-rational single-input single-output (SISO) transfer functions that are analytic on the unit circle $\{z \in \mathbf{C} | \ \ |z| = 1\}$.

Definition 3: Let $\mathcal{E}$ be a set of rationally related random processes. The set $\mathcal{FE}$ is defined as

$$\mathcal{FE} := \left\{ x = \sum_{k=1}^{m} H_k(z) e_k \mid e_k \in \mathcal{E}, H_k(z) \in \mathcal{F}, m \in \mathbb{N} \right\}.$$

The following definition provides a class of models for a network of dynamical systems. It is assumed that the dynamics of each agent (node) in the network is represented by a scalar random process $\{x_j\}_{j=1}^n$ that is given by the superposition of a noise component e_j and the "influences" of other "parent nodes" through dynamic links. The noise acting on each node is assumed to be unrelated to other noise components. If a certain agent "influences" another one a directed edge can be drawn and a directed graph can be obtained.

Definition 4 (Linear Dynamic Graph): A Linear Dynamic Graph $\mathcal{G}$ is defined as a pair $(H(z), e)$ where

- $e = (e_1, .., e_n)^T$ is a vector of n rationally related random processes such that $\Phi_e(z)$ is diagonal
- $H(z)$ is a $n \times n$ matrix of transfer functions in $\mathcal{F}$ such that $H_{jj}(z) = 0$, for $j = 1, ..., n$.

The "node processes" $\{x_j\}_{j=1}^n$ of the LDG are the processes defined as

$$x_j = e_j + \sum_{i=1}^{n} H_{ji}(z) x_i,$$

or in a more compact way

$$x(t) = e(t) + H(z)x(t). \tag{1}$$

Let $V := \{x_1, ..., x_n\}$ and let $A := \{(x_i, x_j) | H_{ji}(z) \neq 0\}$. The pair $G = (V, A)$ is the associated directed graph of the LDG. Nodes and edges of a LDG will mean nodes and edges of the graph associated with the LDG. Also, we say that a LDG is *topologically identifiable* if $\Phi_e(e^{i\omega})$ is positive definite for every ω.

A LDG is an interconnection of stochastic processes via linear transfer functions $H_{ji}(z)$ according to a graph G and forced by stationary additive mutually uncorrelated noise.

The generative class of models: Linear Dynamic Polytrees

In this paper we will consider only acyclic structures. Consistent estimators for the structure of a LDG will be provided assuming that data are generated according to an interconnection of dynamical systems that has no loops (disregarding the orientation of the edges).

Definition 5 (Polytrees and rooted trees): A *polytree* is a directed graph where each pair of nodes is connected by a unique undirected path. Each node of a polytree with indegree equal to zero is a "root". A *rooted tree* is a polytree with exactly one root.

From the definition of LDG, we obtain the following definitions for acyclic dynamic graphs.

Definition 6 (Linear Dynamic Trees): The LDG $(H(z), e)$ is a Linear Cascade Model Tree (LCMT) if the associated graph is a rooted tree (see [13]). The LDG $(H(z), e)$ is a Linear Dynamic Polytree (LDP) if the associated graph is a polytree.

For any root in a LDP it is possible to define an associated LCMT.

Definition 7 (Subtrees of LCMT): Given a LDP, the LCMT given by a root and all its "descendants" is a *subtree* of the LDP.

A graphical representation of a polytree with three roots and the associated subtrees is given in Figure 1. We also provide

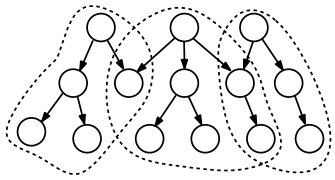

Fig. 1. A polytree and the subtrees associated with its roots.

the definition of a latent LDG, namely a LDG where only a subset of its nodes can be *observed*.

Definition 8: A latent LDG (or LCMT, LDP) is a LDG (or LCMT, LDP) where the set of nodes, V, is partitioned in two sets O and H. The nodes in the set O are called "observable nodes" while the nodes in the set H are called "hidden nodes".

The sufficient and necessary conditions under which a latent undirected tree (of Gaussian random variables) can be successfully reconstructed only by knowing the statistics of the "observable nodes" are given in [16]. These "minimality" conditions amount to

1) no pair of random variables is perfectly correlated or uncorrelated (that is $(\rho - 1)\rho \neq 0$, where ρ is the correlation coefficient of the two random variables)

2) each hidden variable, as represented in the graphical model, has at least degree 3.

Following [16], we extend the definition of minimality to LDPs (in our case the graph is directed and consists of stochastic processes).

Definition 9 (Minimal Linear Dynamic Polytrees): A latent LDP is minimal if the following three conditions are met

- the LDP is topologically identifiable
- each hidden node has degree greater than or equal to 3
- each hidden node has outdegree greater than or equal to 2

Finally we provide the definition of terminal node.

Definition 10 (Terminal node): Given a polytree (or a rooted tree), we say that its node x is terminal if its degree is exactly one, that is $\delta(x) = 1$.

III. Problem Formulation

Consider a minimal latent LDP with n nodes $\{x_j\}_{j=1}^{n}$. Let $\{x_j\}_{j=1}^{n_o}$ be the n_o observed processes, with $n_o \leq n$. Assume no information about the presence or absence of hidden nodes. Only the (cross)-spectral densities of the observed processes $\Phi_{x_i x_j}(e^{i\omega})$, with $i, j = 1, ..., n_o$, are known. Determine the structure of the graph associated with the LDP.

IV. Preliminary results

We start introducing a metric on the nodes of a LDP in the following way.

Definition 11 (Log-Coherence Distance): We define the log-coherence distance as

$$I(x_i, x_j) = -\int \log\left(\frac{|\Phi_{ij}(e^{i\omega})|^2}{\Phi_i(e^{i\omega})\Phi_j(e^{i\omega})}\right). \tag{2}$$

Observe that, if a LDP is topologically identifiable, the log-coherence distance between two nodes is always strictly positive. The following proposition provides a test to determine if two nodes belong to a common subtree.

Proposition 12: Consider two nodes x_i and x_j in a LDP. They belong to a common subtree if and only if $I(x_i, x_j) < +\infty$.

Proof: The proof is straightforward and left to the reader. ∎

The following proposition states that the log-coherence distance is additive along a path of a LDP if it remains in the same subtree.

Proposition 13: Consider two nodes x_{i_0} and x_{i_p} in the same subtree of a Linear Dynamic Polytree, and let $P = \{(x_{i_0}, xi_1), ...x_{(i_{p-1}, x_{i_p})}\}$ be the unique (undirected) path linking them. The log-coherence distance is additive on the path, that is

$$I(x_{i_0}, x_{i_p}) = \sum_{j=1}^{p} I(x_{i_{j-1}}, x_{i_j}). \tag{3}$$

Proof: The proof of this statement is left to the reader. An analogous proof with a metric based on the correlation index for graphical models of random variables can be found in [16].

∎

The following lemma allows one to test if a pair of nodes is connected through a direct link with one of the two nodes being also a terminal node for all the subtrees the pair belongs to.

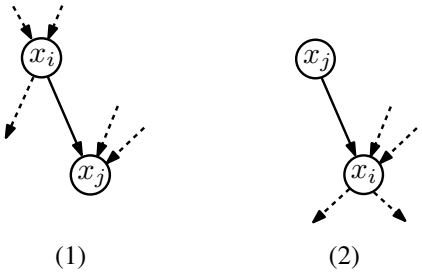

Fig. 2. Graphical representation of the configurations for two nodes x_i and x_j that are detected by the "one-hop terminal test"

Lemma 14 (One-hop terminal test): In an identifiable LDP, consider three observable nodes x_i, x_j, x_k and define

$$\Psi_{x_i x_j x_k} := I(x_i, x_k) - I(x_j, x_k) \tag{4}$$

for every x_k such that $I(x_i, x_k)I(x_j, x_k) < +\infty$. It follows that

$$\Psi_{x_i x_j x_k} = I(x_i, x_j) \tag{5}$$

for every x_k such that $I(x_i, x_k)I(x_j, x_k) < +\infty$, if and only if x_i and x_j are connected and one of the following conditions is met

1) $\delta^+(x_j) = 0$ or
2) $\delta^+(x_j) = 1$, $\delta^-(x_j) = 0$

Proof: See the Appendix. ∎

The possible scenarios described by Lemma 14 are depicted in Figure 2.

The following lemma provides a test to check if two nodes, that are both terminal, in one rooted subtree are directly connected to the same hidden node.

Lemma 15 (Two-hop terminal test): In an identifiable LDP, consider three observable nodes x_i, x_j, x_k and define

$$\Psi_{x_i x_j x_k} := I(x_i, x_k) - I(x_j, x_k) \tag{6}$$

for every x_k such that $I(x_i, x_k)I(x_j, x_k) < +\infty$. It follows that

$$|\Psi_{x_i x_j x_k}| = C < |I(x_i, x_j)| \tag{7}$$

for all k such that $I(x_i, x_k)I(x_j, x_k) \neq \infty$, if and only if the path between x_i and x_j has length 2, x_i and x_j are separated by a hidden node and one of the following conditions is met

1) $\delta^+(x_i) = 1$, $\delta^-(x_i) = 0$, $\delta^+(x_j) = 0$
2) $\delta^+(x_j) = 1$, $\delta^-(x_j) = 0$, $\delta^+(x_i) = 0$
3) $\delta^+(x_i) = \delta^+(x_j) = 0$

Proof: See the Appendix. ∎

The possible scenarios described by Lemma 15 are depicted in Figure 3.

The two-hop terminal test is a useful tool to detect hidden nodes that are connected to terminal nodes: if the test is positive on a pair of observed nodes, then there must a hidden node connected to them. Thus, the hidden node detected in this way can be added to the graph. In certain situations, though, the introduction in the graph of hidden nodes detected using the two-hop terminal test could lead to

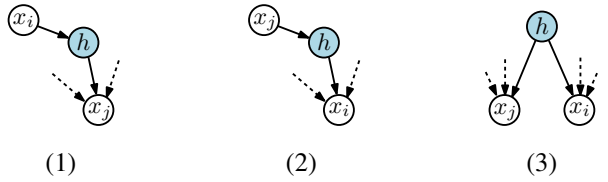

Fig. 3. Graphical representation of the configurations for two nodes x_i and x_j that are detected by the "two-hop terminal test"

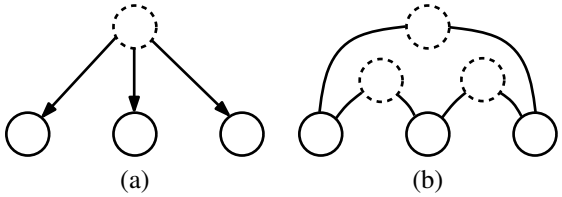

Fig. 4. (a) A polytree with the a hidden root and three observable children. (b) The application of the two-hop terminal test leads to the detection of a hidden node separating each pair of observable nodes. Since the addition of the hidden nodes and the relative edges creates a loop, it is possible to conclude that all the detected hidden nodes are actually the same.

the creation of loops in the reconstructed graph. A scenario like this is depicted in Figure 4. In the connection of Figure 4(a), the two-hop terminal test is positive on every pair of observed nodes. As a result, three hidden nodes are detected, even though only one is actually present. As the following lemma illustrates, if the introduction of the detected hidden nodes creates a cycle, then all the hidden nodes involved in the same cycle are the same hidden node.

Lemma 16: Consider an identifiable and minimal LDP. The two-hop terminal test is positive for the pairs of terminal nodes (x_1, x_2), (x_2, x_3),... (x_{n-1}, x_n), (x_n, x_1) if and only if there is a unique hidden node connected to x_1, x_2,..., x_n.

Proof: Only the necessity of the lemma needs to be proven since the sufficiency follows from the two-hop terminal test. By contradiction, consider that the node is not unique. Then, there would be a a cycle in the polytree. ■

Finally, we introduce a result that allows to compute the distance of a detected hidden node from the other nodes of the network.

Proposition 17: Let x_h be a hidden node of a LDP connected to two terminal nodes x_i and x_j. Let x_k be a node of the network for which the distances $I(x_i, x_k) < +\infty$ and $I(x_j, x_k) < +\infty$ are known. The distance between x_k and x_h is given by

$$I(x_h, x_k) = [I(x_j, x_k + I(x_i, x_k) - I(x_j, x_i)]/2. \quad (8)$$

Proof: From the two-hop test and the additivity of the metric, we have that

$$I(x_h, x_i) - I(x_h, x_j) = \Psi_{x_i x_j x_k} \quad (9)$$

$$I(x_h, x_i) + I(x_h, x_j) = I(x_j, x_i) \quad (10)$$

Thus, we have

$$I(x_h, x_i) = [I(x_j, x_i) + I(x_k, x_i) - I(x_k, x_j]/2. \quad (11)$$

Since the distance is additive, we have that

$$I(x_h, x_k) = I(x_i, x_k) - I(x_h, x_i) \quad (12)$$

$$= I(x_i, x_k) - [I(x_j, x_i) + I(x_k, x_i) - I(x_k, x_j]/2. \quad (13)$$

■

The importance of this proposition is in the fact that once a hidden node is detected using the two-hop test, its distance from all the other detected nodes of the network can be computed using other known distances.

V. Generalized Recursive Regrouping Algorithm for Polytrees

In [17] an algorithm for the reconstruction of rooted trees is described. Such an algorithm (Recursive Regrouping Algorithm) consistently reconstructs minimal rooted trees (indeed it is defined for graphical model of random variables). The algorithm that is now proposed is a generalization of [17] to the case of polytrees of dynamical systems. The key point is that the tests defined in [17] and generalized in Section IV as the one-hop and the two-hop terminal tests only detect nodes that are terminal in a subtree of the polytree. In [17], a node identified as terminal can be "eliminated" from the set of nodes and the algorithm can be applied recursively on the remaining nodes. In this way a guaranteed reconstruction of the topology is achieved. In the case of polytrees, this "elimination" procedure can not be performed because the node to be eliminated in one subtree could still have some edges that have not been detected linking it to other subtrees. Thus, the generalization of the Recursive Regrouping Algorithm to the polytree case has to perform a "selective elimination" of a node only in certain subtrees. This is obtained in the following algorithm by setting the distance of the node to be eliminated to $+\infty$ only with the nodes of the subtree it has to be removed from.

Algorithm

1) Initialize V with the observed nodes $V = \{x_1, ...x_{n_o}\}$
2) Initialize $I_{i,j} \leftarrow I(x_i, x_j)$ for all pairs of observed nodes
3) Repeat, until the reconstructed topology is connected,
   a) Compute $\Psi_{ijk} = I_{ik} - I_{jk}$ for any triplet (x_i, x_j, x_k)
   b) Run the 1-hop test to determine if a pair (x_i, x_j) is directly connected with x_j being a terminal node in a subtree. In such a case redefine $I_{j,k} = I_{k,j} = +\infty$ for every k such that $I_{j,k} I_{k,i} < \infty$.
   c) Run the 2-hop test and determine if a pair (x_i, x_j) is directly connected to a hidden node. In such a case
      - introduce the new detected hidden nodes in the graph with the detected edges
      - collapse the hidden nodes forming a loop into the same node x_h and add x_h to to V
      - determine the distance between every newly added hidden node x_h and every other detected node x_k in V using (8) as in Proposition 17.
      - for each node x_i identified as terminal at this pass, update $I_{i,k} = I_{k,i} = I_{j,k} = I_{k,j} = +\infty$ for every x_k such that $I(j,k) I_{k,i} < \infty$.
   d) check if a pair (x_i, x_j) is such that $I_{i,j} < +\infty$, and $I_{j,k} = I_{i,k} = +\infty$ for every $x_k \neq x_i, x_j$ and in such a case connect them

Theorem 18: The generalized regrouping algorithm exactly reconstructs an identifiable polytree.

Proof: ["Sketch of the proof"] The proof follows from the results in [17] where the Recursive Regrouping Algorithm is shown to successfully reconstruct a rooted tree. Indeed, the algorithm we propose runs the Recursive Regrouping Algorithm in a "parallel way" on all the rooted subtrees of the polytree. Due to limits of space, we just provide an intuitive explanation of this fact omitting the technical details. First, we observe that if a polytree is minimal, then every subtree is minimal. The one-hop-terminal and the two-hop-terminal tests detect nodes that are terminal in a subtree of the polytree. Also, they determine the nodes (hidden or not) of the polytree where the terminal nodes happen to be directly connected. As in the standard Recursive Regrouping Algorithm, the new hidden nodes and the new detected edges are added to the current estimate of the graph. The distances between the newly introduced hidden nodes and the other nodes are computed again as in the standard Recursive Regrouping Algorithm. The validity of Equation (8) to compute the distance of a hidden node x_h from the other nodes is proven in [16] in the case of the correlation metric $d(x_i, x_j) = -log(\rho(x_i, x_j))$. However, the proof only makes use of the additivity property of the metric, and thus it applies to our case, as well.

The main difference is that in the standard Recursive Regrouping Algorithm the nodes that are detected as terminal do not need to be tested anymore: all their connection have been identified. They can simply get removed. In the case of polytrees this can not be done because a node that is terminal in a subtree is possibly connected to nodes of other subtrees through edges that have not been identified yet. The solution to this impasse is provided at steps 3b) and 3d) of the algorithm where the quantities $I_{i,k}$, $I_{k,i}$, $I_{j,k}$, and $I_{k,j}$ are reinitialized to $+\infty$. At these two steps, for all the nodes identified as terminal in a subtree during the iterative step, the distances are reinitialized to $+\infty$, but only for nodes that are in subtrees for which they are detected as terminal nodes. As a result, this generalization of the algorithm follows the same computational steps of the standard Recursive Regrouping Algorithm within each subtree by "eliminating" the terminal nodes from a subtree when they are detected. Each terminal node is "eliminated" only in each subtree where it is terminal, but not from the other subtrees. As a consequence the generalized algorithm provides the reconstruction of each subtree of the polytree, and therefore the whole polytree. ■

Observe that the algorithm is not capable of reconstructing the directionality of the links, but only their presence

VI. Examples

A. Step by step execution of the algorithm

We illustrate the steps of the algorithm a simple example, in order to clarify how the technique works. A polytree of nine nodes (of which two are roots) is depicted in Figure 5(a). Only seven nodes are observed, thus we know only the existence of the limited subset shown in Figure 5(b) with the relative distances. By using the one-hop-terminal test we find that the nodes 4 and 5 and the nodes 6 and 7 are directly linked. Also, it is known that 5 is a terminal node in every subtree where also 4 is present. The distance of 5 from any node that is in every subtree shared with 4 is reinitialized to $+\infty$ Similarly 7 is terminal in every subtree where 6 is present and its distance from all nodes in every subtree shared with 6 are reinitialized to $+\infty$. After this step 5 and 7 have distance equal to $+\infty$ from all the nodes of the graph. By using the two-hop terminal test we find that there is a hidden node between 1 and 2, a hidden node between 2 and 3 and a hidden node between 1 and 3. If these three nodes were distinct, there would be a cycle in the graph, so they collapse into the same hidden node 8. The distances of 1, 2, and 3 from all the nodes in the common subtree are reinitialized to $+\infty$. As a result nodes 1 and 2 have distance equal to $+\infty$ from all the nodes of the polytree. Nodes 1, 2, 5 and 7 are not going to be involved in any other test and are effectually not active anymore. Instead node 3 is still active because it has a finite distance from 4 and 6. The intermediate configuration obtained after this first pass of the algorithm is depicted in Figure 5(c). At the second pass of the algorithm, nodes 3, 4 and 6 are found to be connected through a hidden node (9) using the two-hop terminal test. This configuration obtained after the second pass of the algorithm is depicted in Figure 5(c). Since now the graph is connected, the algorithm terminates.

B. Application of the algorithm to non minimal trees

As a second example we show the result of the application of the algorithm to a non-minimal polytree. A non-minimal polytree is depicted in Figure 6(a). The structure obtained applying the algorithm proposed in this article is shown in Figure 6(b). Node 11 satisfies the conditions of minimality (outdegree greater or equal to 2 and degree great or equal to 3) and it gets correctly detected. Nodes 12 and 13 do not meet the minimality conditions and pass undetected. Indeed, as a result node 12, that has degree equal to 2, is "bypassed", while node 13, that has degree equal to 1 is "ignored". All the other links, instead, are correctly identified.

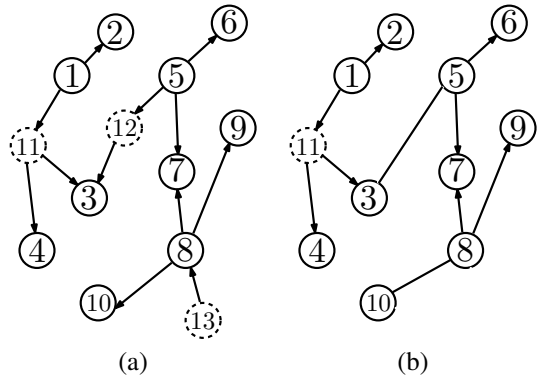

Fig. 6. (a) A non-minimal polytree. (b) The result of the reconstruction using the proposed algorithm.

VII. Conclusions

In this paper we have formulated the problem of reconstructing an acyclic structure of stochastic processes where

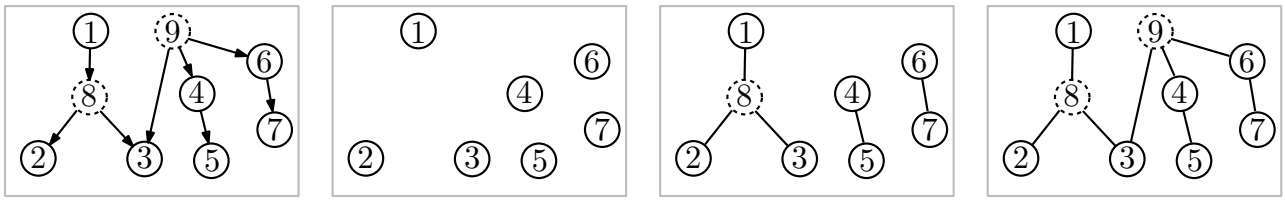

Fig. 5. (a) True configuration with 7 observable nodes $\{1, ..., 7\}$ and 2 latent nodes $\{8, 9\}$; (b) Starting configuration with only observable nodes. (c) At the first step the pairs (1, 2), (2, 3) and (1, 3) are identified by the two-hop-terminal test as pairs of terminal nodes connected to a hidden node. The node is unique, otherwise there would a loop in the topology. Thus, node 8 is detected. At the same time the one-hop-terminal test detects the links $4-5$ and $6-7$ determining also that 5 and 7 are terminal. From this step on nodes 1, 2 and 3 will not be tested with any node that is a descendant of node 1, and nodes 5 and 7 will not be tested anymore with descendants of node 9. (d) At the second step the two-hop-terminal test finds that 3, 4 and 6 are connected to the same hidden node (node 9). The algorithm ends at this point since the structure is connected.

not all the processes are directly observed. Under mild assumptions on the degrees of the nodes in the graph structure, the exact reconstruction of the topology is guaranteed.

VIII. Acknowledgement

M.S. acknowledges the support from the NSF Grant CMMI 0900113.

References

[1] J. Liu, V. Yadav, H. Sehgal, J. M. Olson, H. Liu, and N. Elia, "Phase transitions on fixed connected graphs and random graphs in the presence of noise," *IEEE Transactions on Automatic Control*, vol. 53, p. 1817, 2008.

[2] R. Mantegna and H. Stanley, *An Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge UK: Cambridge University Press, 2000.

[3] M. Eisen, P. Spellman, P. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Natl. Acad. Sci. USA*, vol. 95, no. 25, pp. 14 863–8, 1998.

[4] A. Brovelli, M. Ding, A. Ledberg, Y. Chen, R. Nakamura, and S. L. Bressler, "Beta oscillations in a large-scale sensorimotor cortical network: directional influences revealed by Granger causality." *Proc Natl Acad Sci USA*, vol. 101, no. 26, pp. 9849–9854, June 2004.

[5] D. Urban and T. Keitt, "Landscape connectivity: A graph-theoretic perspective," *Ecology*, vol. 82, no. 5, pp. 1205–1218, 2001.

[6] J.-S. Bailly, P. Monestiez, and P. Lagacherie, "Modelling spatial variability along drainage networks with geostatistics," *Mathematical Geology*, vol. 38, no. 5, pp. 515–539, 2006.

[7] S. Boccaletti, M. Ivanchenko, V. Latora, A. Pluchino, and A. Rapisarda, "Detecting complex network modularity by dynamical clustering," *Phys. Rev. E*, vol. 75, p. 045102, 2007.

[8] D. Napoletani and T. Sauer, "Reconstructing the topology of sparsely connected dynamical networks," *Phys. Rev. E*, vol. 77, p. 026103, 2008.

[9] M. Ozer and M. Uzuntarla, "Effects of the network structure and coupling strength on the noise-induced response delay of a neuronal network," *Physics Letters A*, vol. 375, pp. 4603–4609, 2008.

[10] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, 2004.

[11] E. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. Berlin, Germany: Springer-Verlag, 2009.

[12] C. Michener and R. Sokal, "A quantitative approach to a problem of classification," *Evolution*, vol. 11, pp. 490–499, 1957.

[13] D. Materassi and G. Innocenti, "Topological identification in networks of dynamical systems," *IEEE Trans. Aut. Control*, vol. 55, no. 8, pp. 1860–1871, August 2010.

[14] R. Diestel, *Graph Theory*. Berlin, Germany: Springer-Verlag, 2006.

[15] G. Innocenti and D. Materassi, "A modeling approach to multivariate analysis and clusterization theory," *Journal of Physics A*, vol. 41, no. 20, p. 205101, 2008.

[16] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[17] M. Choi, V. Tan, A. Anandkumar, and A. Willsky, "Learning latent tree graphical models," *Journal of Machine Learning Research*, vol. 12, pp. 1771–1812, 2011.

Appendix

Proof: [Proof of Lemma 14] First we consider the case of a minimal LCMT (rooted tree structure). In such a case $I(x_i, x_j)$, $I(x_i, x_k)$, $I(x_j, x_k) < +\infty$ is true for all x_i, x_j, x_k belonging to the tree. The necessity of the test follows from the additivity property of the distance. If x_j is a terminal node directly connected to x_i, then $\Psi(x_i, x_j, x_k) = I(x_i, x_j)$ for every x_k. The sufficiency is proved by contradiction. If x_i and x_j are not connected, then there is another node in the path linking them. If such a node is observable let it be x_k, otherwise, because of the minimality of the LCMT, there is an observable node x_k in the connected subgraph that separates x_i and x_j. Then, we obtain

$$\Psi_{x_i,x_j,x_k} = I(x_i, x_k) - I(x_j, x_k) < I(x_i, x_k) + I(x_j, x_k) \leq I(x_i, x_j)$$

that is a contradiction.

For the general polytree case, we have that $I(x_i, x_j)$, $I(x_i, x_k)$, $I(x_j, x_k) < +\infty$ if and only if x_i, x_j and x_k belong to the same rooted tree. Hence, the condition in (5) is equivalent to the condition that, in all the rooted trees x_j belongs to, x_j is a terminal node and x_i is directly connected to it. Thus, by examining the possible scenarios, only one of the following conditions is met

1) $\delta^{(+)}(x_j) = 0$
2) $\delta^{(+)}(x_j) = 1$ and $\delta^{(-)}(x_j) = 0$.

■

Proof: [Proof of Lemma 15] First we consider the case of a minimal LCMT (rooted tree structure). In such a case $I(x_i, x_j)$, $I(x_i, x_k)$, $I(x_j, x_k) < +\infty$ is true for all x_i, x_j, x_k, since they belong to the same subtree. Let x_h be the hidden node separating x_i and x_j. From the additivity of the distance we have that $\Psi_{x_i,x_j,x_k} = d(x_i, x_h) + d(x_h, x_k) - d(x_k, x_h) - d(x_h, x_j) = C$. The constant C does not depend on x_k and, given the topological identifiability of the polytree, $C < d(x_i, x_j)$

We have that $I(x_i, x_j)$, $I(x_i, x_k)$, $I(x_j, x_k) < +\infty$ if and only if x_i, x_j and x_k belong to the same rooted tree. Hence, the condition is equivalent to the condition that, in all the rooted trees x_i and x_j belong to, there is a hidden node separating them. ■