2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Tree-Structured Statistical Modeling via Convex Optimization

James Saunderson, Venkat Chandrasekaran, Pablo A. Parrilo and Alan S. Willsky

Abstract— **We develop a semidefinite-programming-based approach to stochastic modeling with multiscale autoregressive (MAR) processes—a class of stochastic processes indexed by the vertices of a tree. Given a tree and the covariance matrix of the variables corresponding to the leaves of the tree, our procedure aims to construct an MAR process with small state dimensions at each vertex that approximately realizes the given covariance at the leaves. Our method does not require prior specification of the state dimensions at each vertex. Furthermore, we establish a large class of MAR processes for which, given only the index tree and the leaf covariance of the process, our method can recover a parametrization that matches the leaf-covariance and has the correct state dimensions. Finally we demonstrate, using synthetic examples, that given i.i.d. samples of the leaf variables our method can recover the correct state dimensions of an underlying MAR process.**

I. Introduction

Modeling complex data as samples of a structured stochastic process is a common approach that underlies many techniques in signal processing, system identification, and machine learning. Among the stochastic processes that have received a great deal of attention in this context are those that are tree-structured, in the sense that the process is indexed by the vertices of a tree and the conditional independence structure among the variables is related to the edge structure of the tree. Models of this type admit very efficient estimation algorithms, and yet possess considerable statistical modeling power.

A natural sub-class of these tree-structured models are multiscale autoregressive (MAR) processes [2]. These can be represented as state space models driven by white Gaussian noise, where the states are indexed by a tree and the state variables are related by affine dynamics operating from the root to the leaves of the tree. MAR models have seen application in areas including computer vision, image processing, remote sensing, and geophysics [19]. In many of these applications, the signal we wish to model (for example the image in image processing applications) corresponds to the leaf-variables of the MAR process, with the other variables, which capture longer-range correlations in the signal, modeled as unobserved or *latent*.

In this paper we consider the problem of building *parsimonious* MAR processes where the leaf-variables model observed data. In this work we assume the tree that indexes the MAR process is given. As such, by parsimonious MAR processes we mean those for which the state space at each latent variable has small dimension. It is an interesting and challenging problem to learn a tree structure given only data at the leaves. This problem has been the subject of much work in the context of phylogenetics [6] and machine learning [14]. We could use any previously developed techniques for learning such a tree as input for our methods.

Our approach to the modeling problem is based on

1) reformulating the modeling problem as a matrix decomposition problem and
2) developing an approach to this matrix decomposition problem based on semidefinite programming.

One way to assess a modeling procedure is to consider its consistency properties. In this paper we analyze our modeling procedure with respect to the following recovery property.

Definition 1: Suppose that Σ is the covariance matrix among the leaf variables of an MAR process indexed by a tree $\mathcal{T}$. We say that a modeling procedure *recovers* the MAR process from Σ and $\mathcal{T}$ if it produces an MAR process indexed by $\mathcal{T}$ that has leaf covariance Σ and the same state dimensions as the original MAR process.

Note that we do not care about the particular parametrization of the MAR process, only that it has the correct 'complexity' and realizes the correct leaf covariance. Indeed there are many different (parametrizations of) MAR processes with the same state dimensions and the same leaf covariance. We focus only on recovering those aspects of the model that are identifiable given only information about the leaf-variables of the process.

The central problem addressed in this paper is to determine conditions on MAR processes that ensure they can be recovered by our modeling procedure. Our main result, Thm. 2, establishes that a large class of MAR processes that have scalar valued variables can be recovered by our modeling procedure.

Although our main recovery result deals with underlying models that have scalar variables, a feature of our modeling procedure is that we do not make any hard prior choices about the state dimensions of the MAR process. Our semidefinite program (SDP) naturally favors models with small state dimensions.

The stochastic realization problem for MAR processes has been considered by a number of authors including Irving et al. [8] who developed a technique based on the notion of canonical correlations, and Frakt et al. [7] who proposed a computationally efficient method for learning internal MAR processes, an important subclass of MAR processes. Both of these are computationally local methods that require prior

This research was funded in part by AFOSR under Grant FA9550-08-1-1080, and in part by Shell International Exploration and Production, Inc.

The authors are with the Department of Electrical Engineering and Computer Science, Laboratory of Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

Emails: {jamess, venkatc, parrilo, willsky}@mit.edu

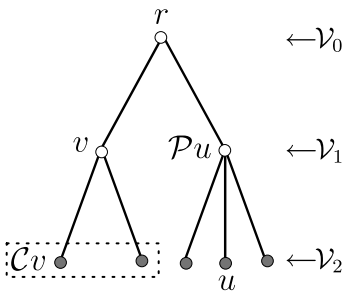

Fig. 1. Summary of notation related to trees. Note that $\mathcal{V}_2$, for example, refers to all of the vertices at distance 2 from the root r. Note that all of the leaves of this tree are at the same distance from the root, and so this tree satisfies our standing assumption.

assumptions on the dimensions of the state spaces at each vertex.

A standard approach to choosing parameters for any statistical model with latent variables is to use the expectation-maximization (EM) algorithm [5] which has been specialized to the case of learning parameters of MAR processes [10]. The EM algorithm, however, does not offer any consistency guarantees, and does not (in its most basic form) learn the state dimensions of the latent variables along with the parameters.

In the case where the index tree is a 'star', consisting only of a root vertex and leaves, then the corresponding MAR model is exactly what is known as a factor analysis model [18] in statistics and the (algebraic) Frisch Scheme [9] in system identification. In this case, the covariance among the leaf variables decomposes as the sum of a diagonal and a low rank matrix. A generalization of this decomposition plays a central role in this work. Furthermore, the SDP we describe in Sec. IV is a non-trivial generalization of a well-known SDP-based heuristic for factor analysis, known as minimum trace factor analysis [16].

The rest of the paper is organized as follows. In Sec. II we introduce notation and terminology related to trees and MAR processes. We then highlight, in Sec. III a particular decomposition that the covariance among the leaf-variables admits. In Sec. IV we propose SDPs to perform exact and approximate covariance decomposition, and analyze the exact decomposition SDP in Sec. V-A. Finally in Sec. VI we apply this method in experiments on synthetic data.

Due to space constraints, we omit detailed proofs throughout the paper.

II. Preliminaries

We introduce notation to allow us to work with random processes on trees. Throughout the paper we will use the tree shown in Fig. 1 to provide a concrete example of much of our notation.

A. Trees

Let $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ be a tree with a distinguished vertex $r \in \mathcal{V}$ called the *root*. We divide the vertices into *scales* depending on their distance from the root. Explicitly, $\mathcal{V}_s$ denotes the set of vertices at distance s from the root.

When it is convenient we can think of $\mathcal{T}$ as a directed tree with edges oriented away from the root. Given a vertex $v \in \mathcal{V}$ let $\mathcal{P}v$ be the parent of v, the (unique) vertex such that $(\mathcal{P}v, v)$ is a directed edge in $\mathcal{T}$. Similarly the children of v, denoted $\mathcal{C}v$, are those vertices whose (common) parent is v. The *leaves* of the tree are those vertices with no children. The *descendants* of a vertex v are the vertices connected to v by a directed path. Finally we use the notation $\mathcal{V}_{\backslash r}$ instead of $\mathcal{V} \setminus \{r\}$ for the set of vertices excluding the root. We summarize these notational conventions in Fig. 1.

We restrict ourselves to a particular class of trees in this paper. We assume that trees are rooted and have *all of their leaves at the same scale* with respect to the root.

B. Multiscale Autoregressive Models

1) Definition and Notation: Consider a zero-mean Gaussian process $(x_v)_{v\in\mathcal{V}}$ where each x_v takes values in $\mathbb{R}^{n_v}$ for some n_v. We refer to $\mathbb{R}^{n_v}$ as the *state space* at v and do *not* fix the dimension of these state spaces *a priori*.

Define $(x_v)_{v\in\mathcal{V}}$ by $x_r \sim \mathcal{N}(0, R)$ and if $v \in \mathcal{V}_{\backslash r}$,

$$x_v = A_v x_{\mathcal{P}v} + w_v \tag{1}$$

where A_v is an $n_v \times n_{\mathcal{P}v}$ matrix, $w_v \sim \mathcal{N}(0, Q_v)$, w_v and w_u are independent if $u \neq v$, and for each $v \in \mathcal{V}_{\backslash r}$, w_v is independent of x_r. We refer to the process $(x_v)_{v\in\mathcal{V}}$ as a *multiscale autoregressive (MAR) process*. Such a process can be parametrized by the matrices $(A_v, Q_v)_{v\in\mathcal{V}_{\backslash r}}$, R, and the tree $\mathcal{T}$. To avoid certain non-identifiability issues we assume, throughout, that R and each A_v and Q_v have full rank.

Since we do not specify the dimensions n_v of the state spaces *a priori*, almost all of our discussion is at the level of block matrices, where each block is indexed by a pair of vertices (u, v) and has dimension $n_u \times n_v$ as a matrix. If X is a block matrix indexed by subsets $\mathcal{U}$ and $\mathcal{W}$ of vertices, we abuse notation and terminology slightly and call X a $|\mathcal{U}| \times |\mathcal{W}|$ matrix. For example we call A_v, in the definition of an MAR process, a $|v| \times |\mathcal{P}v|$ matrix.

2) MAR processes as Markov chains: If we collect together all of the x_v for $v \in \mathcal{V}_s$ as the variable x_s (and similarly define w_s) we can think of an MAR process as a Markov chain indexed by scale with $x_0 = x_r$ and

$$x_s = A_s x_{s-1} + w_s. \tag{2}$$

for $s = 1, 2, \ldots, t$ where $\mathcal{V}_t$ corresponds to the leaves of the index tree. The matrices A_s for $s = 1, 2, \ldots, t$ are $|\mathcal{V}_s| \times |\mathcal{V}_{s-1}|$ (block) matrices defined by

$$[A_s]_{u,v} = \begin{cases} A_v & \text{if } v = \mathcal{P}u \\ 0 & \text{otherwise.} \end{cases}$$

The relationship between the A_s and the A_v is illustrated in Fig. 2

Products of the form $A_t A_{t-1} \cdots A_{s+1}$ have support related to the structure of $\mathcal{T}$, with only the entries $[A_t A_{t-1} \cdots A_{s+1}]_{u,v}$ where $u \in \mathcal{V}_t$ is a descendant of $v \in \mathcal{V}_s$ being non-zero.

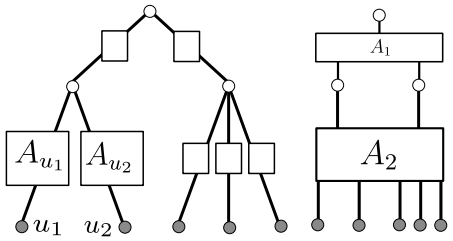

Fig. 2. The left figure shows how the matrices A_v for $v \in \mathcal{V}_{\setminus r}$ relate to the tree structure of an MAR process. The right figure is the Markov chain version of the same MAR process, parametrized in terms of the matrices A_s for $s = 1, 2, \ldots, t$.

III. Covariance Decompositions

As for time-indexed linear state space models we can solve for the leaf variables in terms of the inputs w_s as

$$x_t = (A_t \cdots A_1)x_0 + (A_t \cdots A_2)w_1 + \cdots + A_t w_{t-1} + w_t. \tag{3}$$

Let Σ_t be the covariance of x_t and Q_s be the covariance of w_s, noting that Q_s is diagonal as a block matrix. Taking covariances of (3) yields a decomposition of Σ_t as

$$\begin{aligned}\Sigma_t &= (A_t \cdots A_1)R(A_t \cdots A_1)^T + (A_t \cdots A_2)Q_1(A_t \cdots A_2)^T \\ &\quad + \cdots + A_t Q_{t-1} A_t^T + Q_t \\ &=: L_0 + L_1 + \cdots + L_{t-1} + L_t.\end{aligned} \tag{4}$$

This decomposition is illustrated in Fig. 4. Let us highlight the defining properties of this decomposition.

P1 *Block Diagonal Structure:* For $s = 1, 2, \ldots, t$ the terms $L_s = (A_t \cdots A_{s+1})Q_s(A_t \cdots A_{s+1})^T$ are block diagonal with a support structure that arises from the support pattern of the product $A_t \cdots A_{s+1}$. Since this structure arises often in the sequel we introduce notation to express it compactly, writing $\mathcal{B}_t^s$ for the map that given a symmetric $|\mathcal{V}_t| \times |\mathcal{V}_t|$ matrix X is defined by

$$[\mathcal{B}_t^s(X)]_{u,v} = \begin{cases} [X]_{u,v} & \text{if } u \text{ and } v \text{ are both} \\ & \text{descendants of some } w \in \mathcal{V}_s \\ 0 & \text{otherwise,} \end{cases}$$

where u and v take values in $\mathcal{V}_t$. These block diagonal structures are illustrated in Fig. 3.

P2 *Low-Rank Structure:* For parsimonious MAR models the dimensions n_v of the state spaces at each vertex are all small, so for $0 \le s \le t-1$ the L_s have *low rank*. Specifically $L_0 = (A_t \cdots A_1)R(A_t \cdots A_1)^T$ has rank n_r and the terms $L_s = (A_t \cdots A_{s+1})Q_s(A_t \cdots A_{s+1})^T$ for $1 \le s \le t-1$ have rank $\sum_{v \in \mathcal{V}_s} n_v$.

P3 *Nested Column Spaces:* The column spaces of the terms in the decomposition are nested, satisfying

$$\mathcal{R}(L_0) \subset \mathcal{R}(L_1) \subset \cdots \subset \mathcal{R}(L_t)$$

where $\mathcal{R}(X)$ denotes the column space of the matrix X. This nesting of column spaces arises because of the way the L_s factor into products of the A_τ for $s < \tau \le t$.

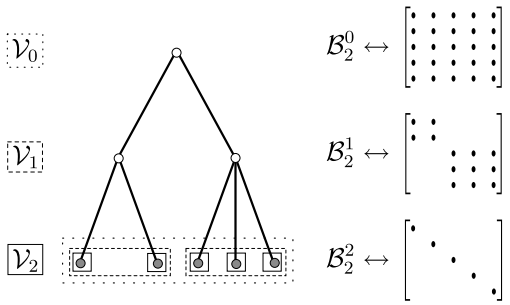

Fig. 3. An illustration of the block diagonal projections $\mathcal{B}_t^s$ defined in property P1 of Sec. III. For example, the vertices $\mathcal{V}_1 = \{u, v\}$ induce a partition of $\mathcal{V}_2$ given by the descendants of u and the descendants of v shown by the dashed boxes. This partition defines the block pattern of $\mathcal{B}_2^1$.

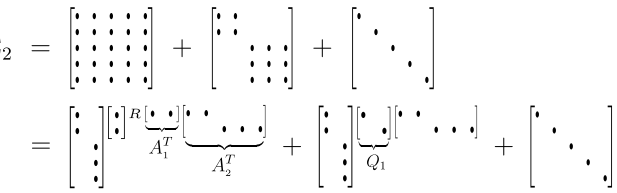

Fig. 4. An illustration of the leaf-covariance decomposition described in Sec. III for the tree in Fig. 1. The first equality represents the block diagonal structure of the terms, the second equality represents the low-rank and nested column space structures of the terms.

Conversely, any decomposition of Σ_t as a sum of positive semidefinite matrices satisfying P1, P2, and P3 is the leaf covariance of an MAR process indexed by $\mathcal{T}$.

From now on we usually think of an MAR process in terms of the implicit parameterization given by the terms of the decomposition (4). It is straightforward to recover an explicit parameterization of an MAR process from the L_s if necessary.

IV. SDP Formulations for Covariance Decomposition

A. An Exact Covariance Decomposition SDP

In this section we formulate an SDP to decompose Σ_t according to (4) as

$$\Sigma_t = L_0 + L_1 + \cdots + L_t$$

where each $L_s \succeq 0$ satisfies P1, P2, and P3 from Sec. III. Our final formulation will specifically address each of these structural issues.

The constraints that each L_s is positive semidefinite and has a particular block diagonal structure are straightforward to incorporate into a semidefinite programming framework.

The constraint that the column spaces of the L_s for $s = 0, 1, \ldots, t-1$ are nested can be imposed by choosing a positive constant M and enforcing that $(L_0, L_1, \ldots, L_{t-1}) \in$

$\mathcal{K}_M$ where

$$\mathcal{K}_M := \{(L_0, \ldots, L_{t-1}) : L_0 \succeq 0, ML_s \succeq L_{s-1} \text{ for } s = 1, \ldots, t-1\}. \quad (5)$$

Since we fix M, this constraint clearly excludes some valid models. For the purposes of our analysis in Sec. V-A, we will, for simplicity of exposition, assume that M is sufficiently large that the model of interest is not excluded.

We want to choose the objective function of our SDP to favor low rank solutions for $L_0, \ldots, L_{t-1}$. A long-known heuristic, put on firm theoretical ground in recent years, is that one can minimize the trace of positive semidefinite matrices as a convex surrogate for minimizing the rank. See, for example, [12] and [15] and the references therein for much more on this topic.

This leads us to an SDP that aims to decompose Σ_t into the sum of low-rank (except for L_t) block diagonal terms.

$$\begin{aligned} (\hat{L}_s)_{s=0}^t = \arg\min & \sum_{s=0}^t \lambda_s \mathrm{tr}(L_s) \\ \text{s.t.} \quad \Sigma_t &= \sum_{s=0}^t \mathcal{B}_t^s(L_s) \\ L_t &\succeq 0, \quad (L_0, \ldots, L_{t-1}) \in \mathcal{K}_M \end{aligned} \quad (6)$$

where $\lambda_s \geq 0$ are parameters of the convex program. Our analysis requires that if $\bar{s} \leq s$ then $\lambda_{\bar{s}} \geq \lambda_s$. This is intuitive as we want to encourage $L_{\bar{s}}$ to have lower rank than L_s, so we should penalize its rank (via the convex surrogate for rank, namely the trace) more by having $\lambda_{\bar{s}} \geq \lambda_s$. For convenience, and by way of normalization, we set $\lambda_0 = 1$.

It can be shown that the SDP (6) has a unique solution $(\hat{L}_0, \ldots, \hat{L}_t)$ which, of course, depends only on Σ_t and the tree $\mathcal{T}$.

B. An Approximate Covariance Decomposition SDP

In the context of modeling we aim to *approximately* decompose the given matrix Σ_t according to (4) as Σ_t typically will not admit a (non-trivial) exact decomposition. A natural modification of the SDP is to replace the constraint $\Sigma_t = \sum_{s=0}^t \mathcal{B}_t^s(L_s)$ with a penalty on the size of $\Sigma_t - \sum_{s=0}^t \mathcal{B}_t^s(L_s)$. Using a squared Frobenius norm penalty, for example, gives the following modification of (6)

$$\min \gamma \sum_{s=0}^t \lambda_s \mathrm{tr}(L_s) + \frac{1}{2}\|\Sigma_t - \sum_{s=0}^t \mathcal{B}_t^s(L_s)\|_F^2 \quad (7)$$

$$\text{s.t.} \quad L_t \succeq 0, \quad (L_0, \ldots, L_{t-1}) \in \mathcal{K}_M \quad (8)$$

where $\gamma > 0$ is a regularization parameter that balances the competing objectives of building a model that matches the observations (that is Σ_t) and has low complexity in the sense of low total state dimension. Note that the convex program defined by (7) and (8) can be formulated as an SDP [3].

V. Analysis and Discussion of the Covariance Decomposition SDPs

A. Recovery Properties of the Exact Decomposition SDP

Throughout this section we assume that Σ_t refers to the covariance at scale t of an MAR process parameterized by $(A_v, Q_v)_{v \in \mathcal{V}_{\setminus r}}$ and R (as in Sec. II). Furthermore, we define $L_0^\star, \cdots L_t^\star$ to be the terms in the decomposition (4) of Σ_t, and assume that M is large enough so that $(L_0^\star, \ldots, L_{t-1}^\star) \in \mathcal{K}_M$.

Our aim, here, is to establish sufficient conditions on the parameters $(A_v, Q_v)_{v \in \mathcal{V}_{\setminus r}}$, and R so that the optimal point $(\hat{L}_0, \ldots, \hat{L}_t)$ of the exact covariance decomposition SDP (6) is precisely $(L_0^\star, \ldots, L_t^\star)$. This is exactly the notion of recovery in Def. 1 of the introduction.

The usual optimality conditions for semidefinite programming (see [3], for example) give necessary and sufficient conditions for recovery, but involve the cone $\mathcal{K}_M^*$, complicating the analysis somewhat. Because $\mathcal{K}_M^*$ contains the product of $t-1$ copies of the positive semidefinite cone, we obtain sufficient conditions for recovery by replacing $\mathcal{K}_M^*$ with a product of positive semidefinite cones in the optimality conditions.

Proposition 1: The SDP (6) correctly decomposes Σ_t if there exists a dual certificate Y such that $\lambda_s I - \mathcal{B}_t^s(Y) \succeq 0$ and $L_s^\star(\lambda_s I - \mathcal{B}_t^s(Y)) = 0$ for $s = 0, 1, \ldots, t$.

It is not particularly obvious how to construct a Y with the properties stated in Prop. 1 as these properties are rather global in nature. It turns out that we can simplify the task of constructing Y by combining dual certificates that concern only the interactions between a parent and all of its children. This is the main technical lemma of this paper.

Lemma 1: Suppose that for each non-leaf vertex v there is a $|\mathcal{C}v| \times |\mathcal{C}v|$ symmetric positive semidefinite matrix Y_v such that

1) $[Y_v]_{uu} = I$ for all $u \in \mathcal{C}v$
2) $Y_v A_{\mathcal{C}v} = 0$

where $A_{\mathcal{C}v} = \begin{bmatrix} A_{u_1} & \cdots & A_{u_m} \end{bmatrix}^T$ and $\mathcal{C}v = \{u_1, \ldots, u_m\}$. Then there exists Y with the properties stated in Prop. 1 and so the SDP (6) correctly decomposes Σ_t.

Rather than supplying a detailed proof, we only explain how to construct Y from the Y_v. First, for each $1 \leq s \leq t$, we define a $|\mathcal{V}_s| \times |\mathcal{V}_s|$ matrix Y_s by

$$Y_s = I - \mathrm{diag}(Y_{v_1}, \ldots, Y_{v_m})$$

where $\mathcal{V}_{s-1} = \{v_1, \ldots, v_m\}$. Then we take Y to be the $|\mathcal{V}_t| \times |\mathcal{V}_t|$ matrix

$$\begin{aligned} Y = &(\lambda_0 - \lambda_1)(A_t \cdots A_2) Y_1 (A_t \cdots A_2)^T + \\ &(\lambda_1 - \lambda_2)(A_t \cdots A_3) Y_2 (A_t \cdots A_3)^T + \cdots \\ &+ (\lambda_{t-2} - \lambda_{t-1}) A_t Y_{t-1} A_t^T + (\lambda_{t-1} - \lambda_t) Y_t. \end{aligned} \quad (9)$$

It can be shown that as long as $\lambda_t < \lambda_{t-1} < \cdots < \lambda_0$, the matrix Y in (9) has the required properties.

Given this result, we need only consider the apparently simpler situation of finding 'local' dual certificates. Any

results about constructing matrices Y_v satisfying the assumptions of Lem. 1 translate into results about the success of the covariance decomposition SDP (6).

One such result, on which we will focus in this paper, deals with the case where x_v is scalar valued (i.e. $n_v = 1$) for all v. In this case all of our block matrices reduce to matrices with scalar entries. Delorme and Poljak [4], in the context of analyzing an approximation algorithm for the MAX-CUT problem, gave a characterization of when a matrix Y_v satisfying the assumptions of Lem. 1 exists.

Definition 2: A vector $u \in \mathbb{R}^n$ is *balanced* if for all $i = 1, 2, \ldots, n$

$$|u_i| \leq \sum_{j \neq i} |u_j|.$$

Theorem 1 (Delorme and Poljak [4]): Given $u \in \mathbb{R}^n$, there exists a matrix $Z \succeq 0$ such that $Z_{ii} = 1$ for $i = 1, 2, \ldots, n$ and $Zu = 0$ if and only if u is balanced.

We would like to point out that there is no known simple characterization of the existence of a matrix Y_v satisfying the assumptions of Lem. 1 where $n_v > 1$ except for the few special cases considered in [1].

The following is our main result, and summarizes the discussion in this section. It gives simple conditions under which the SDP (6) correctly decomposes the covariance among the leaves of an MAR model where all of the variables x_v are scalar valued.

Theorem 2: Suppose that for all $v \in \mathcal{V}$, $n_v = 1$ and that for all children u of v,

$$|A_u| \leq \sum_{w \in \mathcal{C}v \setminus \{u\}} |A_w|.$$

Then the covariance decomposition SDP (6) (with sufficiently large M) correctly decomposes Σ_t.

Remark: The balance condition imposes an interesting structural restriction on the trees $\mathcal{T}$ that can index an MAR process if we hope to identify that process using the SDP (6). Suppose $v \in \mathcal{V}$ has just two children, u_1 and u_2. Then the balance condition says that we must have

$$|A_{u_1}| = |A_{u_2}|$$

a condition that does not hold generically. As such, in order for the parameters of an MAR process to be generically balanced, we need every vertex in the tree $\mathcal{T}$ to have at least three children. Even at this qualitative level, Thm. 2 gives us insight about how we ought *not* to go about choosing our tree $\mathcal{T}$ when trying to solve a modeling problem as our procedure will clearly not be effective on trees having vertices with only two children.

B. Discussion of the Approximate Covariance Decomposition SDP

Suppose Σ_t is a good approximation of the covariance at scale t of an MAR model that can be recovered using the exact decomposition SDP. One would hope that the optimum of the approximate covariance decomposition SDP is close in some sense to the 'true' decomposition of the underlying MAR model if we choose γ and M suitably with respect to the error between Σ_t and the underlying leaf covariance.

Similar results hold for a number of related problems including sparse and low-rank decompositions and matrix completion (see [13], for example). These results typically require a suitable tightening of the conditions on the model required for exact recovery (in our case, a suitable tightening of the balance conditions in Thm. 2). We leave this as an avenue for further investigation.

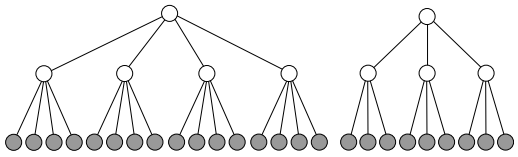

Fig. 5. The two trees that index the MAR processes in our synthetic experiments. All the variables have state dimension one.

VI. EXPERIMENT

In this section we demonstrate by simulation the performance of our method for learning the state dimensions and parameters of an MAR process given i.i.d. samples of the leaf variables, rather than the exact covariance matrix. We consider two MAR processes, defined with respect to the trees in Fig. 5. All of the variables in both processes are scalar-valued. For both processes, the underlying model are generated by taking $R = 1$ and, for $v \in \mathcal{V}_{\setminus r}$, $Q_v = 1$ and $A_v = 1 + 0.01 n_v$ where n_v are i.i.d. standard normal random variables. These choices ensure that (with high probability) the balance condition of Thm. 2 is satisfied.

For each value of N in a given range, and each of the two underlying processes, we run the following procedure fifty times. We form the sample covariance matrix corresponding to N i.i.d. samples of the leaf-variables and run the approximate covariance decomposition SDP, recording on each trial whether all the state dimensions of the underlying MAR process were correctly recovered. We choose the parameters $\lambda_s = 0.9^s$ (for $s = 0, 1, 2$) and $M = 1.1$ (large enough so that the underlying model is in the model class). In both cases we chose $\gamma = \min\{M, 6\sqrt{p/N}\}$, where p is the number of observed variables ($p = 9$ and $p = 16$ for the trees on the right and left respectively in Fig. 5). We note that similar results are obtained for values of γ in a range around these values. We solve the SDP (7) using a combination of the modeling language YALMIP [11] and SDPT3 [17]. Fig. 6 shows the results, indicating that with sufficiently many samples, the method is successful in both cases in recovering the state dimensions of the model.

VII. CONCLUSIONS

In this paper we propose a semidefinite programming-based method for modeling with MAR processes. Our approach naturally favors parsimonious models yet does not require prior choices to be made about the state dimensions of the model. We prove that under certain natural conditions on the underlying MAR process, our method can exactly

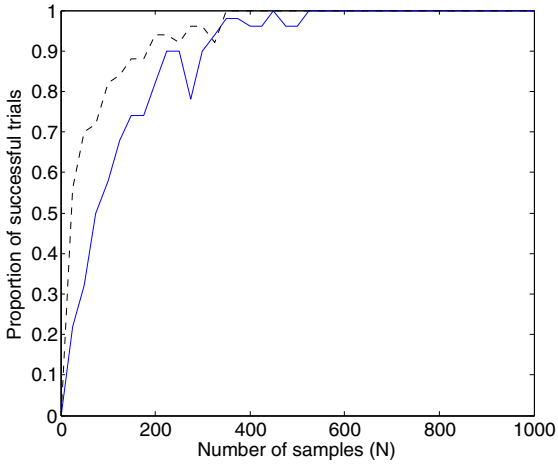

Fig. 6. For each of the two trees shown in Fig. 5 and each N we repeat the following procedure 50 times. We form a sample covariance matrix from N i.i.d. samples of the leaf variables of an MAR process defined on the given tree. We use our method to model the given data with a parsimonious MAR process and check whether the state dimensions of the learned model match those of the underlying model. On the vertical axis we plot the proportion of trials in which all state dimensions were recovered correctly. We display the number of samples on the horizontal axis. The blue solid line displays results for the tree on the left in Fig. 5 and the black dashed line displays results for the tree on the right in Fig. 5.

recover the model parameters given the tree structure and the covariance matrix among the variables at the finest scale.

In future work we would like to prove statistical consistency of our method, providing guidelines for choosing the parameters γ and M, as well as considering more general choices of loss function in our approximate covariance decomposition SDP. Furthermore, the problem of learning the tree structure in addition to model parameters and state dimensions given data is a natural and challenging generalization of problems considered in this paper. It would be interesting to extend the convex optimization-based approach of this paper to that setting.

REFERENCES

[1] W. Barrett and S. Pierce, "Null Spaces of Correlation Matrices," *Linear Algebra and its Applications*, vol. 368, pp. 129–157, 2003.

[2] M. Basseville, A. Benveniste and A. S. Willsky, "Multiscale Autoregressive Processes I. Schur-Levinson parametrizations," *IEEE Trans. Signal Process.*, vol. 40, no. 8, pp. 1915–1934, 1992.

[3] S. P. Boyd and L. Vandenberghe, "Convex Optimization," *Cambridge University Press*, 2004.

[4] C. Delorme and S. Poljak, "Combinatorial Properties and the Complexity of a Max-Cut Approximation," *European Journal of Combinatorics*, vol. 14, no. 4, pp. 313–333, 1993.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum-Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Stat. Soc. Ser. B*, vol. 38, no. 1, pp. 1–38, 1977.

[6] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, "Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids," *Cambridge Univ. Press*, 1999.

[7] A. B. Frakt and A. S. Willsky, "Computationally Efficient Stochastic Realization for Internal Multiscale Autoregressive Models," *Multidimensional Systems and Signal Processing*, vol. 12, no. 2, pp. 109–142, 2001.

[8] W. W. Irving and A. S. Willsky, "A Canonical Correlations Approach to Multiscale Stochastic Realization," *IEEE Trans. Automatic Control*, vol. 46, no. 10, pp. 1514–1528, 2001.

[9] R. E. Kalman, "Identification of Noisy Systems," *Russian Mathematical Surveys*, vol. 40, no. 4, pp. 25–42, 1985.

[10] A. Kannan, M. Ostendorf, W. C. Karl, D. A. Castanon and R. K. Fish, "ML Parameter Estimation of Multiscale Stochastic Processes using the EM Algorithm," *IEEE Trans. Signal Process.*, vol. 48, no. 6, pp. 1836–1847, 2000.

[11] J. Löfberg, "YALMIP: A Toolbox for Modeling and Optimization in MATLAB." *Proc. CACSD Conf.*, Taipei, Taiwan, 2004.

[12] M. Mesbahi and G. P. Papavassilopoulos, "On the Rank Minimization Problem over a Positive Semidefinite Linear Matrix Inequality," *IEEE Trans. Automatic Control*, vol. 42, no. 2, pp. 239–243, 1997.

[13] S. Negahban, P. Ravikumar, M. J. Wainwright, B. Yu, "A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers," arXiv:1010.2731v1.

[14] J. Pearl, "Probabilistic Reasoning in Intelligent Systems: Network of Plausible inference," *Morgan Kaufmann*, 1988.

[15] B. Recht, M. Fazel, P. A. Parrilo, "Guaranteed Minimum-Rank Solutions of Linear Matrix Equations via Nuclear Norm Minimization," *SIAM Review*, vol. 52, no. 3, pp. 471–501, 2010.

[16] A. Shapiro, "Rank-Reducibility of a Symmetric Matrix and Sampling Theory of Minimum Trace Factor Analysis," *Psychometrika*, vol. 47, no. 2, pp. 187–199, 1982.

[17] K. C. Toh, M. J. Todd and R. H. Tutuncu, "SDPT3 — a Matlab Software Package for Semidefinite Programming," *Optimization Methods and Software*, vol. 11, no. 12, pp. 545–581, 1999.

[18] L. L. Thurstone, "Multiple Factor Analysis," *Psychological Review*, vol. 38, no. 5, pp. 406–427, 1931.

[19] A. S. Willsky, "Multiresolution Markov Models for Signal and Image Processing," *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1396–1458, 2002.