Parameter and Structure Learning in Nested Markov Models

Ilya Shpitser **Thomas S. Richardson** **James M. Robins** **Robin J. Evans**

Abstract

The constraints arising from DAG models with latent variables can be naturally represented by means of acyclic directed mixed graphs (ADMGs). Such graphs contain directed ($\rightarrow$) and bidirected ($\leftrightarrow$) arrows, and contain no directed cycles. DAGs with latent variables imply independence constraints in the distribution resulting from a 'fixing' operation, in which a joint distribution is divided by a conditional. This operation generalizes marginalizing and conditioning. Some of these constraints correspond to identifiable 'dormant' independence constraints, with the well known 'Verma constraint' as one example. Recently, models defined by a set of the constraints arising after fixing from a DAG with latents, were characterized via a recursive factorization and a nested Markov property. In addition, a parameterization was given in the discrete case. In this paper we use this parameterization to describe a parameter fitting algorithm, and a search and score structure learning algorithm for these nested Markov models. We apply our algorithms to a variety of datasets.

1 Introduction

Many data-generating process correspond to distributions that factorize according to a directed acyclic graph (DAG). Such models also have an intuitive causal interpretation: an arrow from a variable X to a variable Y in a DAG model can be interpreted, in a way that can be made precise, to mean that X is a "direct cause" of Y.

In many contexts we do not observe all of the variables in the data-generating process. This creates major challenges for structure learning and for identifying causal intervention distributions. While existing machinery based on DAGs with latent variables can be applied to such settings, this creates a number of problems. First, there will in general be an infinite number of DAG models such that a particular margin of these models may represent the observed distribution; this is still true if we require the graph to be faithful. Second, prior knowledge about latent variables is often scarce, which implies any modeling assumptions made by explicitly representing latents leaves one open to model misspecification bias. An alternative approach, is to consider graphical models represented by graphs containing directed and bidirected edges, called Acyclic Directed Mixed Graphs (ADMGs). In a companion paper we define a 'nested' Markov property for ADMGs that encodes independence constraints under a 'fixing' operation that divides the joint distribution by a conditional density. Given a DAG $\mathcal{G}$ with latent variables there is an ADMG $\mathcal{G}^*$ naturally associated with $\mathcal{G}$ via the operation of 'latent projection' [19]; the vertices of $\mathcal{G}^*$ are solely the subset of vertices in $\mathcal{G}$ that are observed. We show that the observed distribution resulting from the DAG with latent variables $\mathcal{G}$ obeys the nested Markov property associated with the corresponding latent projection $\mathcal{G}^*$.

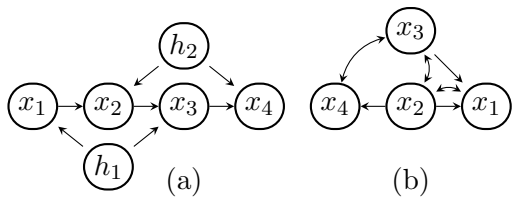

Figure 1: (a) A latent variable DAG not entailing any d-separation statements on x_1, x_2, x_3, x_4. (b) An ADMG $\mathcal{G}$ with missing edges representing a saturated nested Markov model.

Previous work [4] has given a discrete parameterization and ML fitting algorithms, as well as a characterization of model equivalence for mixed graph models representing conditional independences [1]. It is well-known, however, that models representing DAG marginals can contain nonparametric constraints which cannot be represented as conditional independence constraints. For instance, in any density P Markov relative to a DAG with latents represented by the graph shown in Fig. 1 (a), it is known (see [19, 13]) that

$$\frac{\partial}{\partial x_1} \sum_{x_2} P(x_4 \mid x_1, x_2, x_3) P(x_2 \mid x_1) = 0 \qquad (1)$$

This constraint can be viewed as stating that X_4 is independent of X_1 in the distribution obtained from $P(x_1, x_2, x_3, x_4)$ after dividing by a conditional $P(x_3|x_2, x_1)$ [12]. Also note that the expression (1) is an instance of the g-formula of [13]. If the graph shown in Fig. 1 (a) is causal, then this constraint can be interpreted as an (identifiable) dormant independence constraint [15], which states that X_4 is independent of X_1 given $do(x_3)$, where $do(.)$ denotes an intervention [7].

Since the DAG in Fig. 1 (a) implies no conditional independences on the 4 observable variables, the set of marginal distributions on x_1, x_2, x_3, x_4 obtained from densities over $x_1, x_2, x_3, x_4, h_1, h_2$ Markov relative to this DAG is a saturated model when viewed as a model of conditional independence. Thus, any structure learning algorithm which only relies on conditional independence constraints will return a (maximally uninformative) unoriented complete graph when given data sampled from one of such marginal distributions.

Nevertheless, it is possible to use constraints such as (1), which we call post-truncation independences or reweighted independences to distinguish between models, with appropriate assumptions. In [16], such constraints were used to test for the presence of certain direct causal effects (represented by directed arrows in graphical causal models). A recent paper [11] has given a nested factorization for mixed graph models which implies, along with the standard conditional independences in mixed graphs, post-truncation independences of the type shown in (1). Furthermore, a parameterization for discrete models based on this factorization was given. Another recent paper [17] has taken advantage of this parameterization to give a general algorithm for efficiently computing causal effects. In this paper, we take advantage of this parameterization to give a maximum likelihood parameter fitting algorithm for mixed graph models of post-truncation independence. Furthermore, we use this algorithm to construct a search and score algorithm based on BIC [14] for learning mixed graph structure while taking advantage of post-truncation independences.

The paper is organized as follows. In section 2, we introduce the graphical, and probabilistic preliminaries necessary for the remainder of the paper. In section 3, we introduce nested Markov models. In section 4, we give a parameterization of discrete nested Markov models. In section 5, we describe the parameter fitting algorithm. In section 6, we describe the search and score algorithm. Section 7 contains our experiments. Section 8 gives our conjecture for the characterization of equivalence classes of mixed graphs over four nodes, and gives experimental evidence in favor of our conjecture. Section 9 contains the discussion and concluding remarks.

2 Preliminaries

A *directed mixed graph* $\mathcal{G}(V, E)$ is a graph with a set of vertices V and a set of edges E which may contain directed ($\rightarrow$) and bidirected ($\leftrightarrow$) edges. A directed cycle is a path of the form $x \rightarrow \cdots \rightarrow y$ along with an edge $y \rightarrow x$. An *acyclic* directed mixed graph (ADMG) is a mixed graph containing no directed cycles.

2.1 Conditional ADMGs

A *conditional* acyclic directed mixed graph (CADMG) $\mathcal{G}(V, W, E)$ is an ADMG with a vertex set $V \cup W$, where $V \cap W = \emptyset$, subject to the restriction that for all $w \in W$, $\mathrm{pa}_{\mathcal{G}}(w) = \emptyset = \mathrm{sp}_{\mathcal{G}}(w)$.

Whereas an ADMG with vertex set V represents a joint density $p(x_V)$, a conditional ADMG is a graph with two disjoint sets of vertices, V and W that is used to represent the Markov structure of a 'kernel' $q_V(x_V|x_W)$. Following [6, p.46], we define a *kernel* to be a non-negative function $q_V(x_V|x_W)$ satisfying:

$$\sum_{x_V \in \mathfrak{X}_V} q_V(x_V \mid x_W) = 1 \qquad \text{for all } x_W \in \mathfrak{X}_W. \tag{2}$$

We use the term 'kernel' and write $q_V(\cdot|\cdot)$ (rather than $p(\cdot|\cdot)$) to emphasize that these functions, though they satisfy (2) and thus most properties of conditional densities, will not, in general, be formed via the usual operation of conditioning on the event $X_W = x_W$. To conform with standard notation for densities, we define for every $A \subseteq V$, $q_V(x_A|x_W) \equiv \sum_{V \setminus A} q_V(x_V|x_W)$, and $q_V(x_{V \setminus A}|x_{W \cup A}) \equiv \frac{q_V(x_V|x_W)}{q_V(x_A|x_W)}$.

For a CADMG $\mathcal{G}(V, W, E)$ we consider collections of random variables $(X_v)_{v \in V}$ taking values in probability spaces $(\mathfrak{X}_v)_{v \in V}$ conditional on variables $(X_w)_{w \in W}$ with state spaces $(\mathfrak{X}_w)_{w \in W}$. In all the cases we consider the probability spaces are either real finite-dimensional vector spaces or finite discrete sets. For $A \subseteq V \cup W$ we let $\mathfrak{X}_A \equiv \times_{u \in A}(\mathfrak{X}_u)$, and $X_A \equiv (X_v)_{v \in A}$. We use the usual shorthand notation: v denotes a vertex and a random variable X_v, likewise A denotes a vertex set and X_A. It is because we will always condition on the variables in W that we do not permit edges between vertices in W.

An ADMG $\mathcal{G}(V, E)$ may be seen as a CADMG in which $W = \emptyset$. In this manner, though we will state subsequent definitions for CADMGs, they will also apply to ADMGs.

The induced subgraph of a CADMG $\mathcal{G}(V, W, E)$ given by set A, denoted $\mathcal{G}_A$ consists of $\mathcal{G}(V \cap A, W \cap A, E_A)$, where E_A is the set of edges in $\mathcal{G}$ with both endpoints in A. Note that in forming $\mathcal{G}_A$, the status of the vertices in A with regard to whether they are in V or W is preserved.

2.2 Districts

A set C is *connected* in $\mathcal{G}$ if every pair of vertices in C are connected by a path with every vertex on the path contained in C. A connected set C in an ADMG $\mathcal{G}$ is *inclusion maximal* if no superset of C is connected.

For a given CADMG $\mathcal{G}(V, W, E)$, the *induced bidirected graph* $(\mathcal{G})_{\leftrightarrow}$ is the CADMG formed by removing all directed edges from $\mathcal{G}$. Similarly, $(\mathcal{G})_{\rightarrow}$ is formed by removing all bidirected edges. A set connected in $(\mathcal{G})_{\leftrightarrow}$ is called bidirected connected.

For a given vertex $x \in V$ in $\mathcal{G}$, the district (c-component) of x, denoted by $dis_{\mathcal{G}}(x)$ is the connected component of $(\mathcal{G})_{\leftrightarrow}$. Districts in an ADMG $\mathcal{G}(V, E)$ form a partition of V. In a DAG $\mathcal{G}(V, E)$ the set of districts is the set of all single element node sets of V. In a CADMG, all districts are subsets of V, the nodes of W are not included by definition. For an induced subgraph $\mathcal{G}_A$, we write $dis_A(x)$ as a shorthand for $dis_{\mathcal{G}_A}(x)$.

2.3 The fixing operation and fixable vertices

We now introduce a 'fixing' operation on an ADMG or CADMG that has the effect of transforming a random vertex into a fixed vertex, thereby changing the graph. However, in general this operation may only be applied to a subset of the vertices in the graph, which we term the set of (potentially) fixable vertices.

Definition 1 *Given a CADMG $\mathcal{G}(V, W)$ the set of* fixable vertices*,*

$$\mathbb{F}(\mathcal{G}) \equiv \{v \mid v \in V, \mathrm{dis}_{\mathcal{G}}(v) \cap \mathrm{de}_{\mathcal{G}}(v) = \{v\}\}.$$

In words, a vertex v is fixable in $\mathcal{G}$ if there is no vertex v^* that is both a descendant of v and in the same district as v in $\mathcal{G}$.

Definition 2 *Given a CADMG $\mathcal{G}(V, W, E)$, and a kernel $q_V(X_V \mid X_W)$, for every $r \in \mathbb{F}(\mathcal{G})$ we associate a* fixing transformation *ϕ_r on the pair $(\mathcal{G}, q_V(X_V \mid X_W))$ defined as follows:*

$$\phi_r(\mathcal{G}) \equiv \mathcal{G}^*(V \setminus \{r\}, W \cup \{r\}, E_r),$$

where E_r is the subset of edges in E that do not have arrowheads into r, and

$$\phi_r(q_V(x_V \mid x_W); \mathcal{G}) \equiv \frac{q_V(x_V \mid x_W)}{q_V(x_r \mid x_{\mathrm{mb}_{\mathcal{G}}(r, \mathrm{an}_{\mathcal{G}}(\mathrm{dis}_{\mathcal{G}}(r)))})}. \tag{3}$$

We use $\circ$ to indicate composition of operations in the natural way, so that: $\phi_r \circ \phi_s(\mathcal{G}) \equiv \phi_r(\phi_s(\mathcal{G}))$ and $\phi_r \circ \phi_s(q_V(X_V|X_W); \mathcal{G}) \equiv \phi_r\left(\phi_s\left(q_V(X_V|X_W); \mathcal{G}\right); \phi_s(\mathcal{G})\right)$.

2.4 Reachable and Intrinsic Sets

In order to define our factorization, we will need to define special classes of vertex sets in ADMGs.

Definition 3 *A CADMG $\mathcal{G}(V, W)$ is* reachable *from an ADMG $\mathcal{G}^*(V \cup W)$ if there is an ordering of the vertices in $W = \langle w_1, \ldots, w_k \rangle$, such that for $j = 1, \ldots, k$,*

$$w_1 \in \mathbb{F}(\mathcal{G}^*) \text{ and for } j = 2, \ldots, k,$$
$$w_j \in \mathbb{F}(\phi_{w_{j-1}} \circ \cdots \circ \phi_{w_1}(\mathcal{G}^*)).$$

In words, a subgraph is reachable if, under some ordering, each of the vertices in W may be fixed, first in $\mathcal{G}^*$, and then in $\phi_{w_1}(\mathcal{G}^*)$, then in $\phi_{w_2}(\phi_{w_1}(\mathcal{G}^*))$, and so on. If a CADMG $\mathcal{G}(V, W)$ is reachable from $\mathcal{G}^*(V \cup W)$, we say that the set V is reachable in $\mathcal{G}^*$. Note that a reachable set R in $\mathcal{G}$ may be obtained by fixing vertices using more than one valid sequence. We will denote any valid composition of fixing operations that fixes a set A by ϕ_A if applied to the graph, and by ϕ_{X_A} if applied to a kernel. Note that with a slight abuse of notation (though justified as we will later see) we suppress the precise fixing sequence chosen.

Definition 4 *A set C is* intrinsic *in $\mathcal{G}$ if it is a district in a reachable subgraph of $\mathcal{G}$. The set of intrinsic sets in an ADMG $\mathcal{G}$ is denoted by $\mathcal{I}(\mathcal{G})$.*

Note that in any DAG $\mathcal{G}(V, E)$, $\mathcal{I}(\mathcal{G}) = \{\{x\} | x \in V\}$, while in any bidirected graph $\mathcal{G}$, $\mathcal{I}(\mathcal{G})$ is equal to the set of all connected sets in $\mathcal{G}$.

3 Nested Markov Models

We define a factorization on probability distributions represented by ADMGs via intrinsic sets.

Definition 5 (nested factorization) *Let $\mathcal{G}(V, E)$ be an ADMG. A distribution $p(X_V)$ obeys the* nested factorization according to $\mathcal{G}(V, E)$ *if for every reachable subset $A \subseteq V$, $\phi_{X_{V \setminus A}}(p(x_V); \mathcal{G}) = \prod_{D \in \mathcal{D}(\phi_A(\mathcal{G}))} f_D(x_D | x_{\mathrm{pa}_{\mathcal{G}}(D) \setminus D})$.*

A distribution $p(x_V)$ that obeys the nested factorization with respect to $\mathcal{G}$ is said to be in the nested Markov model of $\mathcal{G}$.

Theorem 6 *If $p(x_V)$ is in the nested Markov model of $\mathcal{G}$, then for any reachable set A in $\mathcal{G}$, any valid fixing sequence on $V \setminus A$ gives the same CADMG over A, and the same kernel $q_A(x_A | x_{V \setminus A})$ obtained from $p(x_V)$.*

Due to this theorem, our decision to suppress the precise fixing sequence from the fixing operation is justified.

It is known that nested Markov factorization implies the global Markov property for ADMGs.

Theorem 7 *If a distribution $p(X_V)$ is in the nested Markov model for $\mathcal{G}(V, E)$ then $p(X_V)$ obeys the global Markov property for $\mathcal{G}(V, E)$.*

The proof appears in [11]. This result implies nested Markov models capture all conditional independence statements normally associated with mixed graphs. In addition, nested Markov models capture additional independence constraints that manifest after truncation operations. For example, all distributions contained in the model that factorizes according to the graph shown in Fig. 1 (a), obey the constraint that X_1 is independent of X_4 after "truncating out" (that is, dividing by) $P(x_3 \mid x_{2,1})$.

4 Parameterization of Binary Nested Markov Models

We now give a parameterization of nested Markov models. The approach generalizes in a straightforward way to finite discrete state spaces.

4.1 Heads and Tails of Intrinsic Sets

Definition 8 *For an intrinsic set $C \in \mathcal{I}(\mathcal{G})$ of a CADMG $\mathcal{G}$, define the recursive head (rh) as: $rh(C) \equiv \{x | x \in C; ch_{\mathcal{G}_C}(x) = \emptyset\}$.*

Definition 9 *The* tail *associated with a recursive head H of an intrinsic set C in a CADMG $\mathcal{G}$ is given by: $tail(H) \equiv (C \setminus H) \cup pa_{\mathcal{G}}(C)$.*

4.2 Binary Parameterization

Multivariate binary distributions which obey the nested factorization with respect to an CADMG $\mathcal{G}$ may be parameterized by the following parameters:

Definition 10 *The* binary parameters *associated with a CADMG $\mathcal{G}$ are a set of functions: $\mathfrak{Q}_{\mathcal{G}} \equiv \left\{q_C(X_H = \mathbf{0}|x_{tail(H)})|H = rh(C), C \in \mathcal{I}(\mathcal{G})\right\}$.*

Intuitively, a parameter $q_C(X_H = \mathbf{0}|x_{tail(H)})$ is the probability that the variable set X_H assumes values $\mathbf{0}$ in a kernel obtained from $p(x_V)$ by fixing $X_{V\setminus C}$, and conditioning on $X_{tail(H)}$. As a shorthand, we will denote the parameter $q_C(X_H = \mathbf{0}|x_{tail(H)})$ as $\theta_H(x_{tail(H)})$.

Definition 11 *Let $\nu : V \cup W \mapsto \{0, 1\}$ be an assignment of values to the variables indexed by $V \cup W$. Define $\nu(T)$ to be the values assigned to variables indexed by a subset $T \subseteq V \cup W$. Let $\nu^{-1}(0) = \{v \mid v \in V, \nu(v) = 0\}$.*

A distribution $P(X_V \mid X_W)$ is said to be parameterized by *the set $\mathfrak{Q}_{\mathcal{G}}$, for CADMG $\mathcal{G}$ if:*

$$p(X_V = \nu(V) \mid X_W = \nu(W)) = \sum_{B\,:\,\nu^{-1}(0)\cap V \subseteq B \subseteq V} (-1)^{|B\setminus\nu^{-1}(0)|} \times \prod_{H \in [\![B]\!]_{\mathcal{G}}} \theta_H(X_{tail(H)} = \nu(tail(H))),$$

where the empty product is defined to be 1, and $[\![B]\!]_{\mathcal{G}}$ is a partition of nodes in B given in [17].

Note that this parameterization maps θ_H parameters to probabilities in a CADMG via an inverse Möbius transform. Note also that this parameterization generalizes both the standard Markov parameterization of DAGs in terms of parameters of the form $p(x_i = 0|pa(x_i))$, and the parameterization of bidirected graph models given in [3].

4.3 Example

Consider an ADMG $\mathcal{G}$ shown in Fig. 1 (b). The parameters associated with a binary model represented by this graph are:

$$\theta_{1,4}(x_2, x_3), \theta_1(x_2, x_3), \theta_{2,3}, \theta_2, \theta_3, \theta_4(x_2), \theta_{3,4}(x_2)$$

Each of these parameters are functions which map binary values to probabilities, which implies this binary model contains 15 parameters, in other words it is saturated. This is the case even though $\mathcal{G}$ is not a complete graph. A similar situation arises in mixed graph models of conditional independence. In such models a model represented by a graph with missing edges may be saturated if nodes which are not direct neighbors are connected by an *inducing path* [18]. In particular, the mixed graph shown in Fig. 1 represents a saturated model of conditional independence because there is an inducing path between X_1 and X_4. The reason this graph does not represent a saturated nested Markov model is because truncations allow us to test independence of some pairs of non-adjacent nodes, even if they are connected by an inducing path. However, there are some inducing paths which are "dense" enough such that truncations cannot be used to test independence between node pairs connected by such a path. Such a dense inducing path exists in the graph in Fig. 1 (b) between X_1 and X_4.

As an illustration of our parameterization, for the graph in Fig. 1 (b), we have the following:

$$\begin{aligned} p(x_1 = 0, x_3 = 0, x_4 = 0, x_2 = 1) = \\ \theta_{1,4}(x_2 = 1, x_3 = 0) * \theta_3 - \theta_{1,4}(x_2 = 1, x_3 = 0) * \theta_{2,3} \\ p(x_1 = 0, x_3 = 0, x_2 = 1, x_4 = 1) = \\ \theta_1(x_2 = 1, x_3 = 0) * \theta_3 - \theta_1(x_2 = 1, x_3 = 0) * \theta_{2,3} \\ -\theta_{1,4}(x_2 = 1, x_3 = 0) * \theta_3 + \theta_{1,4}(x_2 = 1, x_3 = 0) * \theta_{2,3} \end{aligned}$$

5 Parameter Fitting for Binary Nested Markov Models

We now describe a parameter fitting algorithm based on the parameterization in definition 11, which relates the parameters of an nested Markov model and standard multinomial probabilities via the Möbius inversion formula. We first describe this mapping in more detail.

For a given CADMG $\mathcal{G}$, define $\mathcal{P}_{\mathcal{G}}$ be the set of all multinomial probability vectors in the simplex $\Delta_{2^{|V|}-1}$ which obey the nested factorization according to $\mathcal{G}$, let $\mathcal{Q}_{\mathcal{G}}$ be the set of all vectors of

parameters which define coherent nested Markov models.

The mapping $\rho_{\mathcal{G}} : \mathcal{Q}_{\mathcal{G}} \mapsto \mathcal{P}_{\mathcal{G}}$ in definition 11 can be viewed as a composition $\mu_{\mathcal{G}} \circ \tau_{\mathcal{G}}$ of two mappings. Here $\tau_{\mathcal{G}}$ maps $\mathcal{Q}_{\mathcal{G}}$ to the set of all terms of the form $\prod_{H \in [\![B]\!]_{\mathcal{G}}} \theta_H(X_{tail(H)} = \nu(tail(H)))$ composed of parameters in $\mathcal{Q}_{\mathcal{G}}$. We denote this set by $\mathcal{T}_{\mathcal{G}}$. The second mapping $\mu_{\mathcal{G}}$ maps $\mathcal{T}_{\mathcal{G}}$ to $\mathcal{P}_{\mathcal{G}}$ via an inverse Möbius transform. In [4] these mappings were defined via element-wise matrix operations, with $\tau_{\mathcal{G}}$ defined via a matrix P containing 0 and 1 entries, and $\mu_{\mathcal{G}}$ defined via a matrix M containing $0, 1, -1$ entries. There may be more efficient representations of these mappings. In particular $\mu_{\mathcal{G}}$ may be evaluated via the fast Möbius transform [5]. Such an efficient mapping was given in [17].

Note that $\rho_{\mathcal{G}}$ is smooth with respect to each parameter. This implies we can solve many optimization problems for functions expressed in terms of $\rho_{\mathcal{G}}$ using standard iterative methods. The difficulty is that the fitting algorithm must be defined in such a way that each step that starts in the parameter space $\mathcal{Q}_{\mathcal{G}}$ stays in $\mathcal{Q}_{\mathcal{G}}$. We use the approach taken in [4], where a single step of the fitting algorithm updates the estimates for all and only parameters which refer to a particular vertex $v \in V$. For a particular vector $\mathbf{q} \in \mathcal{Q}_{\mathcal{G}}$, let $\mathbf{q}(v)$ be the set of parameters whose heads contain v. Let $\mathcal{Q}_{\mathcal{G}}(v)$ be the subset of $\mathcal{Q}_{\mathcal{G}}$ containing only such vectors. Then the restriction of $\rho_{\mathcal{G}}$ to $\mathcal{Q}_{\mathcal{G}}(v)$ is a linear function since any such parameter occurs at most once in in a term in $\mathcal{T}_{\mathcal{G}}$. This implies that $\rho_{\mathcal{G}}$ can be expressed as $A_v * \mathcal{Q}_{\mathcal{G}}(v) - \mathbf{b}_v$ for some matrix A_v and vector $\mathbf{b}_v$. To remain within $\mathcal{Q}_{\mathcal{G}}$ it suffices to maintain the constraint that $A_v \mathbf{q}(v) \geq \mathbf{b}_v$.

5.1 Maximum Likelihood Parameter Fitting

We are now ready to describe our parameter fitting algorithm. Our scheme closely follows that in [4], albeit with a different parameterization. The algorithm iteratively updates parameters $\mathbf{q}(v)$ for every vertex v in turn, and at each step maximizes the log likelihood via gradient ascent. For the purposes of this paper, we assume strictly positive counts in our data. The case of zero counts gives rise to certain statistical complications, and will be handled in subsequent work.

Q-FIT$(\mathcal{G}, \mathbf{q}, L(\rho_{\mathcal{G}}))$
INPUT: $\mathbf{G}$ an ADMG, $\mathbf{q}$ a set of parameters defining a model which obeys the nested factorization wrt $\mathcal{G}, L(\rho_{\mathcal{G}})$ a concave function defined in terms of $\rho_{\mathcal{G}}$.
OUTPUT: $\hat{\mathbf{q}}$, a local maximum in the surface defined on $\mathcal{Q}_{\mathcal{G}}$ via $L(\rho_{\mathcal{G}})$.

Cycle through each vertex v in $\mathcal{G}$, and do

1 Construct the constraint matrices $A_v, \mathbf{b}_v$.
2 Fit $\mathbf{q}(v)$ to obtain new estimate $\mathbf{q}^*$ maximizing $L(\rho_{\mathcal{G}})$ subject to $A_v \mathbf{q}(v) \geq \mathbf{b}_v$.
3 If $\mathbf{q}^*$ sufficiently close to $\mathbf{q}$, return $\mathbf{q}^*$.
4 Otherwise, set $\mathbf{q}$ to $\mathbf{q}^*$.

Figure 2: A parameter fitting algorithm for nested Markov models.

For a particular vertex v, the function we are optimizing has the form $\log \mathcal{L}_{\mathcal{G}}(\mathbf{q}(v)) = \sum_i n_i \log \rho_{\mathcal{G}}(\mathbf{q}(v))$ where $\rho_{\mathcal{G}}$ is restricted to $\mathcal{Q}_{\mathcal{G}}(v)$ and is thus a linear function in $\mathbf{q}(v)$.

Our fitting algorithm is given in Fig. 2. Our choice of L is the log likelihood function which is strictly concave in $\mathbf{q}(v)$ by above, while our initial guess for $\mathbf{q}$ are the parameters which define a fully independent model. The optimization problem in line 2 can be solved by standard gradient ascent methods.

6 Structure Learning in Nested Markov Models

A fitting algorithm which maximizes likelihood allows us to do structure learning in nested Markov models, using standard search and score methods which use likelihood-based scoring criteria such as BIC [14].

The algorithm is a standard greedy local search augmented with a tabu meta-heuristic. We found a meta-heuristic necessary for our search procedure because a complete theory of equivalence of

models with respect to post-truncation independences is not yet available. Because we do not yet understand equivalence in this setting, we are unable to define efficient local steps which always move across equivalence classes as in the GES algorithm for DAGs [2]. Without such steps, in order to achieve reasonable local minima in the score surface, the algorithm must be able to move across score plateaus.

For the purposes of our experiments, we used the BIC scoring function, although our approach does not require this, and any competing scoring method could have been used. We chose BIC due to its desirable asymptotic properties.

We interpret the output of our search procedure to be the "best" mixed graph model under the assumption that every post-truncation independence observed in the data has a structural explanation. This assumption is a natural generalization of the faithfulness [18], or stability [7] assumption from the conditional independence setting to the post-truncation independence setting. We do not pursue the precise statement of this assumption in this paper, since doing so entails defining a strong global Markov property for nested Markov models (the post-truncation analogue of d-separation in DAGs and m-separation in mixed graphs). This property is sufficiently intricate that its definition and properties are developed in a companion paper.

6.1 Implementation

We implemented fitting and search algorithms using the R language [8]. Our implementation was based on an older implementation of fitting and search for mixed graph models of conditional independence [4].

7 Experiments

To illustrate our search and score method, we used simulated data.

7.1 Simulated Data from DAG Marginal Models With Post-Truncation Constraints

To demonstrate that our algorithm can successfully learn "interesting" graphs distinct from known DAG and MAG equivalence classes, we have used search and score on simulated data obtained from DAG models with latent variables shown in Fig. 4. Both of these models are known to contain post-truncation independences. The observable nodes X_1, X_2, X_3, X_4 in the graph shown in Fig. 4 (a) correspond to binary random variables, while the latent node U corresponds to a discrete random variable with 16 possible values. Similarly, the observable nodes X_1, X_2, X_3, X_4, X_5 in the graph shown in Fig. 4 (b) correspond to binary random variables, while the latent nodes U_1, U_2 correspond to discrete random variables with 8 possible values.

The model shown in the graph in Fig. 4 (a) contains two independence constraints over observable variables. The first is an ordinary conditional independence constraint ($X_1 \perp\!\!\!\perp X_3 | X_2$). The second is a post-truncation independence which states that $X_1 \perp\!\!\!\perp X_4$ after truncating out $P(x_3 \mid x_2)$. The model shown in the graph in Fig. 4 (b) contains three independence constraints over observable variables. The first two are ordinary independence constraints which state that ($X_3 \perp\!\!\!\perp X_1 | X_2$) and ($X_4 \perp\!\!\!\perp X_1, X_2 | X_3$). The last is a post-truncation independence which states that $X_4, X_5 \perp\!\!\!\perp X_1$ after truncating out $P(x_3 \mid x_2)$. Note that X_4 and X_5 can both be made conditionally independent of X_1, but not by using the same conditioning set.

7.2 Results

We chose parameters of the DAG models shown in Fig. 4 in such a way as to ensure "approximately faithful" models. We then generated samples from our models and retained the values of only X_1, X_2, X_3, X_4 in the first model, and of only X_1, X_2, X_3, X_4, X_5 in the second model. We evaluated the performance of our structure learning algorithm on datasets ranging from 500 to 5000 samples (in 500 sample increments). For each dataset size, we generated 1000 datasets randomly

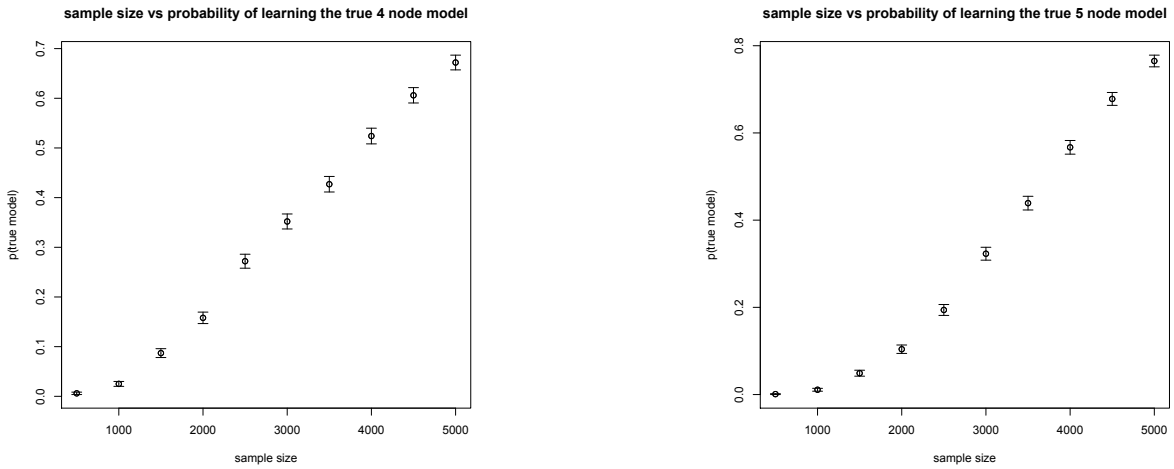

Figure 3: (a) Probability of learning the true 4 node model vs sample size. (b) Probability of learning the true 5 node model vs sample size.

from the true models.

Figures 3 (a) and (b) show our results. The probability of learning the true model grows linearly with sample size, with 672 4 node models out of 1000 correctly recovered, and 765 5 node models out of 1000 correctly recovered, from 5000 sample datasets. By "correctly recovered" we mean that our search procedure returned one of the graphs shown in Fig. 7 in datasets obtained from a DAG model in Fig. 4 (a), and one of the graphs in the appropriate equivalence class in datasets obtained from a DAG model in Fig. 4 (b). Although there is currently no complete theory of observational equivalence of nested Markov models, as there is for Markov factorizing models, we do provide evidence for the characterization of all equivalence classes of 4 node models in the next section.

8 Equivalence Conjecture for Mixed Graphs of Four Nodes

When describing the search and score algorithm for nested Markov models, we mentioned that no complete theory of equivalence of models with respect to post-truncation independence currently exists. Characterization of equivalence does exist for DAGs [19], and MAGs [1]. In this section, we present an investigation of the issue of equivalence with respect to post-truncation independence for the special case of mixed graphs of four nodes.

It is known that there are exactly 543 four node DAGs (sequence A003024 in OEIS). This implies that there are $543 * 2^6 = 34752$ ADMGs with four nodes. These ADMGs are arranged into 185 equivalence classes representing DAG models of conditional independence, and 63 equivalence classes representing models of independence which can be represented by a mixed graph, but not a DAG, for a total of 248 equivalence classes. If we consider nested Markov models, which in addition to conditional independences imply post-truncation independences, we expect the number of equivalence classes to expand, since two mixed graphs may agree on all conditional independences, but disagree on post-truncation independences. For instance, this is the case for the ADMG shown in Fig. 5 (a) and a complete DAG.

We conjecture that for a given ADMG $\mathcal{G}$, if the model of conditional independence [9] and a nested Markov model agree on the parameter count, then they define the same model [10]. This conjecture would imply that it is sufficient to characterize equivalence in ADMGs where the model in [9] and the nested Markov model give different parameter counts. There are exactly 228 such ADMGs.

We conjecture that these 228 ADMGs are arranged in 84 equivalence classes. These 84 classes fall in a small number of graph patterns with multiple classes having the same pattern but different vertex labeling. Specifically, of the 84 classes, 24 are of type (a), shown in Fig. 5 (a), 12 are of

type (b), shown in Fig. 5 (b), 24 contain graphs with the patterns shown in Fig. 6, and 24 contain graphs with the patterns shown in Fig. 7.

The model contained in one of 24 singleton equivalence classes shown in Fig. 5 (a) has a single post-truncation independence which states (up to node relabeling) that $X_4 \perp\!\!\!\perp X_2 | X_3$ in the distribution obtained from $P(x_1, x_2, x_3, x_4)$ after truncating out $P(x_3 \mid x_2, x_1)$.

The model contained in one of 12 singleton equivalence classes shown in Fig. 5 (b) has a single post-truncation independence which states (up to node relabeling) that $X_4 \perp\!\!\!\perp X_3 | X_2$ in the distribution obtained from $P(x_1, x_2, x_3, x_4)$ after truncating out $P(x_2 \mid x_1)$. The advantage of exploiting post-truncation independence is clear in the cases shown in Fig. 5. If the best scoring model lies in these classes, then we can recover the model structure exactly just from observing a single post-truncation independence, whereas if we restricted ourselves to conditional independence we would be unable to distinguish models in these classes from saturated models.

The model contained in one of 24 equivalence classes shown in Fig. 6 (which contains 5 ADMGs) has a single post-truncation independence which states (up to node relabeling) that $X_4 \perp\!\!\!\perp X_1$ in the distribution obtained from $P(x_1, x_2, x_3, x_4)$ after truncating out $P(x_3 \mid x_2, x_1)$. In the case of models shown in Fig. 6, even though the equivalence class contains 5 graphs, these graphs agree on many interesting (from a causal point of view) structural features. In particular, in all elements of a particular class the following edges are present (up to node relabeling): $X_3 \rightarrow X_4, X_2 \rightarrow X_3, X_2 \leftrightarrow X_4$. As before, these models are indistinguishable from the saturated model with respect to standard conditional independence constraints.

The model contained in one of 24 equivalence classes shown in Fig. 7 (which contains 3 ADMGs) has one conditional independence which states (up to node relabeling) that $X_3 \perp\!\!\!\perp X_1 | X_2$, and one post-truncation independence which states (up to node relabeling) that $X_4 \perp\!\!\!\perp X_1$ in the distribution obtained from $P(x_1, x_2, x_3, x_4)$ after truncating out $P(x_3 \mid x_2)$. Similarly, models shown in Fig. 6 are members of equivalence classes contains 3 graphs, yet these graphs agree on many interesting structural features. As before, in all elements of a particular class the following edges are present (up to node relabeling): $X_3 \rightarrow X_4, X_2 \rightarrow X_3, X_2 \leftrightarrow X_4$. These models are indistinguishable from the (DAG) model asserting a single conditional independence $X_3 \perp\!\!\!\perp X_1 | X_2$, with respect to standard conditional independence constraints.

To confirm our conjecture, we have verified that log-likelihood values of nested Markov models obtained by **Q-FIT** from datasets generated from a four node saturated model are always the same within our conjectured classes.

Finally, we note that if our conjecture is correct, post-truncation independences occur in about 25% (84/(248 + 84)) of four node mixed graph equivalence classes. This suggests that, far from being "rare and exotic," these constraints may be fairly common in latent variable models. This is particularly encouraging since post-truncation constraints seem to be quite informative for causal discovery.

9 Discussion

We described a new class of graphical models called nested Markov models, which can be viewed as the "closure" of DAG marginal models which preserves all equality constraints. These constraints include standard conditional independence constraints, and less well-understood constraints which manifest after a truncation operation, which corresponds to dividing by a conditional distribution. We have given a nested factorization of these models which generalizes the standard Markov factorization of DAG models, and the factorization of bidirected graph models [3]. We have given a parameterization for discrete models, and used this parameterization to give a parameter fitting and structure learning algorithm for nested Markov models. Together with results in [17], our parameter fitting scheme gives an MLE for any identifiable causal effect in discrete nested Markov models.

We have applied our structure learning algorithm to simulated data. We have shown that our al-

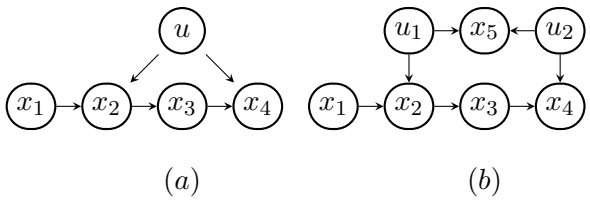

Figure 4: DAG models used in our simulation experiments.

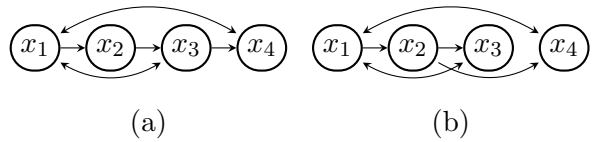

Figure 5: (a) An equivalence class pattern containing 24 equivalence classes which in turn contain 1 graph each. (b) An equivalence class pattern containing 12 equivalence classes which in turn contain 1 graph each.

gorithm can correctly distinguish models based on post-truncation independences, which no other currently known discovery algorithm is capable of doing. Finally, we used our fitting procedure to justify a conjecture which characterizes model equivalence with respect to post-truncation independence in four node mixed graph models.

The advantage of our approach is twofold. First, by representing latent variables implicitly, we are able to reason over a potentially infinite set of DAG models which can give rise to a particular pattern of constraints. Second, our machinery explicitly incorporates post-truncation independence, which is a kind of equality constraint which generalizes conditional independence, and which can be used to distinguish models which are not distinguishable with respect to standard conditional independence. We have shown cases where discovering a single post-truncation independence is sufficient to recover the full structure of an ADMG without any ambiguity, though the corresponding model has no standard conditional independence constraints.

Both our nested factorization and the post-truncation independences this factorization implies have an intuitive causal interpretation. The factorization can be thought of as decomposing the joint distribution into tractable pieces corresponding to joint direct effects on bidirected connected sets, while the post-truncation independence correspond to (identifiable) dormant independence constraints [15], which can be viewed as either an absence of some direct effect, or a decomposition of a joint direct effect into multiple smaller joint direct effects.

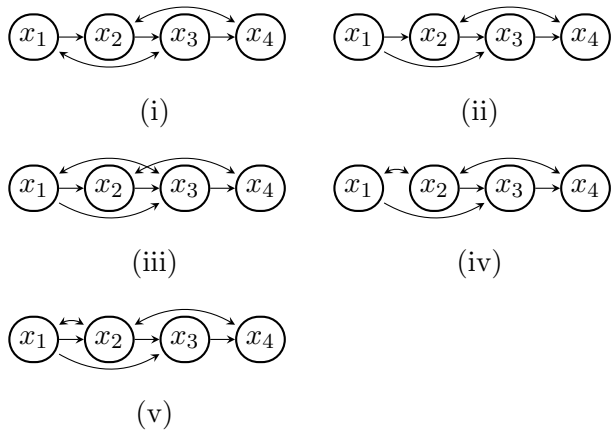

Figure 6: (i)-(v) Five graph patterns together representing 24 equivalence classes, each class containing 5 graphs, one from each pattern.

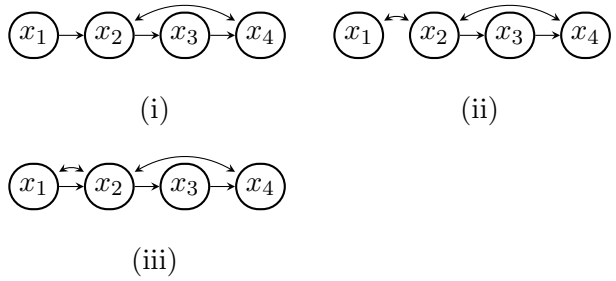

Figure 7: (i)-(iii) Three graph patterns together representing 24 equivalence classes, each class containing 3 graphs, one from each pattern.

References

[1] Ayesha Ali, Thomas S. Richardson, and Peter Spirtes. Markov equivalence for ancestral graphs. *Annals of Statistics*, 37:2808–2837, 2009.

[2] David Maxwell Chickering. Optimal structure identifiation with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2002.

[3] M. Drton and T.S. Richardson. Binary models for marginal independence. *J. Roy. Statist. Soc. Ser. B*, 70(2):287–309, 2008.

[4] Robin J. Evans and Thomas S. Richardson. Maximum likelihood fitting of acyclic directed mixed graphs to binary data. In *Proceedings of the Twenty Sixth Conference on Uncertainty in Artificial Intelligence*, volume 26, 2010.

[5] R. Kennes. Computational aspects of the moebius transform of a graph. *IEEE Transactions on Systems, Man, and Cybernetics*, 22:201–223, 1991.

[6] S.L. Lauritzen. *Graphical Models*. Oxford, U.K.: Clarendon, 1996.

[7] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

[8] R Development Core Team. *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing, 2004.

[9] Thomas S. Richardson. A factorization criterion for acyclic directed mixed graphs. In *25th Conference on Uncertainty in Artificial Intelligence*, 2009.

[10] Thomas S. Richardson, James M. Robins, and Ilya Shpitser. Generalized Markov models associated with acyclic directed mixed graphs. Unpublished manuscript, 2011.

[11] Thomas S. Richardson, James M. Robins, and Ilya Shpitser. Nested markov properties for acyclic directed mixed graphs. In *28th Conference on Uncertainty in Artificial Intelligence (UAI-12)*. AUAI Press, 2012.

[12] James M. Robins. Testing and estimation of direct effects by reparameterizing directed acyclic graphs with structural nested models. In C. Glymour and G. Cooper, editors, *Computation, Causation, and Discovery*, pages 349–405. MIT Press, Cambridge, MA, 1999.

[13] J.M. Robins. A new approach to causal inference in mortality studies with sustained exposure periods – application to control of the healthy worker survivor effect. *Mathematical Modeling*, 7:1393–1512, 1986.

[14] Gideon E. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[15] Ilya Shpitser and Judea Pearl. Dormant independence. Technical Report R-340, Cognitive Systems Laboratory, University of California, Los Angeles, 2008.

[16] Ilya Shpitser, Thomas S. Richardson, and James M. Robins. Testing edges by truncations. In *International Joint Conference on Artificial Intelligence*, volume 21, pages 1957–1963, 2009.

[17] Ilya Shpitser, Thomas S. Richardson, and James M. Robins. An efficient algorithm for computing interventional distributions in latent variable causal models. In *27th Conference on Uncertainty in Artificial Intelligence (UAI-11)*. AUAI Press, 2011.

[18] P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer Verlag, New York, 1993.

[19] T. S. Verma and Judea Pearl. Equivalence and synthesis of causal models. Technical Report R-150, Department of Computer Science, University of California, Los Angeles, 1990.