Counting and Exploring Sizes of Markov Equivalence Classes of Directed Acyclic Graphs

Yangbo He heyb@pku.edu.cn
Jinzhu Jia jzjia@math.pku.edu.cn
LMAM, School of Mathematical Sciences, LMEQF, and Center of Statistical Science, Peking University
Bin Yu binyu@stat.berkeley.edu
Departments of Statistics and EECS, UC Berkeley

Editor: Isabelle Guyon and Alexander Statnikov

Abstract

When learning a directed acyclic graph (DAG) model via observational data, one generally cannot identify the underlying DAG, but can potentially obtain a Markov equivalence class. The size (the number of DAGs) of a Markov equivalence class is crucial to infer causal effects or to learn the exact causal DAG via further interventions. Given a set of Markov equivalence classes, the distribution of their sizes is a key consideration in developing learning methods. However, counting the size of an equivalence class with many vertices is usually computationally infeasible, and the existing literature reports the size distributions only for equivalence classes with ten or fewer vertices.

In this paper, we develop a method to compute the size of a Markov equivalence class. We first show that there are five types of Markov equivalence classes whose sizes can be formulated as five functions of the number of vertices respectively. Then we introduce a new concept of a rooted subclass. The graph representations of rooted sub-classes of a Markov equivalence class are used to partition this class recursively until the sizes of all rooted subclasses can be computed via the five functions. The proposed size counting is efficient for Markov equivalence classes of sparse DAGs with hundreds of vertices. Finally, we explore the size and edge distributions of Markov equivalence classes and find experimentally that, in general, (1) most Markov equivalence classes are half completed and their sizes are small, (2) the sizes of sparse classes grow approximately exponentially with the numbers of vertices, and (3) for the classes that are sparser than half completed graph, the denser of a graph, the smaller of the size of the corresponding class.

Keywords: Directed acyclic graphs; Markov equivalence class; Size distribution; Causality

1. Introduction

Graphical models based on directed acyclic graphs (DAGs) are commonly used to derive the dependent or causal relationships in many fields such as sociology, epidemiology, and biology (Finegold and Drton, 2011; Friedman, 2004; Heckerman et al., 1999; Jansen et al., 2003; Maathuis et al., 2009). A DAG can be used to represent causal relationships of variables, where the directed edges connect the causes and their direct effects. In general, observational data is not sufficient to distinguish the underlying DAG from its statistically equivalent DAGs; however, it is possible to learn the Markov equivalence class that contains these equivalent DAGs (Pearl, 2000; Spirtes et al., 2001). This has led to many works that try to learn a Markov equivalence class or to learn causality based on a given Markov equivalence class from observational or experimental data (Castelo and Perlman, 2004; Chickering, 2002; He and Geng, 2008; Maathuis et al., 2009; Meganck et al., 2006; Perlman, 2001).

The size of a Markov equivalence class is the number of DAGs in the class. This size has been used in papers to design causal learning approaches or to evaluate the "complexity" of a Markov equivalence class in causal learning. For examples, He and Geng (2008) proposes several criterions, all of which are defined on the sizes of Markov equivalence classes, to minimize the number of interventions; this minimization makes helpful but expensive interventions more efficient. Based on observational data, Maathuis et al. (2009) introduces a method to estimate the average causal effects of the covariates on the response by considering each DAG in the equivalence class; the size of the class determines the complexity of the estimation. Chickering (2002) shows that the causal structure searching in the space of Markov equivalence class models could be substantially more efficient than the searching in the space of DAG models if most sizes of Markov equivalence classes are large.

The size of a small Markov equivalence class is usually counted via traversal methods that list all DAGs in the Markov equivalence class (Gillispie and Perlman, 2002). However, if the class is large, it may be infeasible to list all DAGs. For example, as we will show later in our experiments, the size of a Markov equivalence class with 50 vertices and 250 edges could be greater than 10^{24}. To our knowledge, there are no efficient methods to compute the size of a large Markov equivalence class; approximate proxies, such as the number of vertices, have been used instead of the exact size in the literature (Chickering, 2002; He and Geng, 2008).

Computing the size of a Markov equivalence class is the focus of this article. We first discuss Markov equivalence classes whose sizes can be calculated just through the numbers of vertices and edges. Five explicit formulas are given to obtain the sizes for five types of Markov equivalence classes respectively. Then, we introduce rooted sub-classes of a Markov equivalence class and discuss the graphical representations of these sub-classes. Finally, for a general Markov equivalence class, we introduce a counting method by recursively partitioning the Markov equivalence class into smaller rooted sub-classes until all rooted sub-classes can be counted with the five explicit formulas. Consequently, the proposed algorithms allow us to efficiently compute the sizes of Markov equivalence classes and to explore size distributions of sparse classes with hundreds of vertices.

Next, we also report new results about the size and edge distributions of Markov equivalence classes for sparse graphs with hundreds of vertices. By using the proposed size counting method in this paper and an MCMC sampling method recently developed by He et al. (2013), we experimentally explore the size distributions of Markov equivalence classes with large numbers of vertices and different levels of edge sparsity. In the literature, the size distributions are studied in detail just for Markov equivalence classes with up to 10 vertices by traversal methods (Gillispie and Perlman, 2002).

The rest of the paper is arranged as follows. In Section 2, we provide a brief review of the concept of a Markov equivalence class. In Section 3, we propose efficient algorithms to calculate the size of a Markov equivalence class. In Section 4, we study the sizes of Markov equivalence classes experimentally. We conclude in Section 5 and finally present all proofs in the Appendix.

2. Markov Equivalence Class

A graph $\mathcal{G}$ consists of a vertex set V and an edge set E. A graph is directed (undirected) if all of its edges are directed (undirected). A sequence of edges that connect distinct vertices in V, say $\{v_1, \cdots, v_k\}$, is called a path from v_1 to v_k if either $v_i \rightarrow v_{i+1}$ or $v_i - v_{i+1}$ is in E for $i = 1, \cdots, k-1$. A path is *partially directed* if at least one edge in the path is directed. A path is directed (undirected) if all edges are directed (undirected). A *cycle* is a path from a vertex to itself.

A *directed acyclic graph* (DAG) $\mathcal{D}$ is a directed graph without any directed cycle. Let V be the vertex set of $\mathcal{D}$ and τ be a subset of V. The induced subgraph $\mathcal{D}_\tau$ of $\mathcal{D}$ over τ, is defined to be the graph whose vertex set is τ and whose edge set contains all of those edges of $\mathcal{D}$ with two end points in τ. A *v-structure* is a three-vertex induced subgraph of $\mathcal{D}$ like $v_1 \rightarrow v_2 \leftarrow v_3$. A graph is called a *chain graph* if it contains no partially directed cycles. The isolated undirected subgraphs of the chain graph after removing all directed edges are the chain components of the chain graph. A *chord* of a cycle is an edge that joins two nonadjacent vertices in the cycle. An undirected graph is *chordal* if every cycle with four or more vertices has a chord.

A graphical model is a probabilistic model for which a DAG denotes the conditional independencies between random variables. *A Markov equivalence class* is a set of DAGs that encode the same set of conditional independencies. Let the *skeleton* of an arbitrary graph $\mathcal{G}$ be the undirected graph with the same vertices and edges as $\mathcal{G}$, regardless of their directions. Verma and Pearl (1990) proves that two DAGs are *Markov equivalent* if and only if they have the same skeleton and the same v-structures. Moreover, Andersson et al. (1997) shows that a Markov equivalence class can be represented uniquely by an *essential graph*.

Definition 1 (Essential graph) *The* essential graph *of a DAG $\mathcal{D}$, denoted as $\mathcal{C}$, is a graph that has the same skeleton as $\mathcal{D}$, and an edge is directed in $\mathcal{C}$ if and only if it has the same orientation in every equivalent DAG of $\mathcal{D}$.*

It can be seen that the essential graph $\mathcal{C}$ of a DAG $\mathcal{D}$ has the same skeleton as $\mathcal{D}$ and keeps the v-structures of $\mathcal{D}$. Andersson et al. (1997) also introduces some properties of an essential graph.

Lemma 2 (Andersson et al. (1997)) *Let $\mathcal{C}$ be an essential graph of $\mathcal{D}$. Then $\mathcal{C}$ is a chain graph, and each chain component $\mathcal{C}_\tau$ of $\mathcal{C}$ is an undirected and connected chordal graph, where τ is the vertex set of the chain component $\mathcal{C}_\tau$.*

Let SizeMEC($\mathcal{C}$) denote the size of the Markov equivalence class represented by $\mathcal{C}$ (size of $\mathcal{C}$ for short). Clearly, SizeMEC($\mathcal{C}$) $= 1$ if $\mathcal{C}$ is a DAG; otherwise $\mathcal{C}$ may contain more than one chain component, denoted by $\mathcal{C}_{\tau_1}, \cdots, \mathcal{C}_{\tau_k}$. From Lemma 2, each chain component is an undirected and connected chordal graph (UCCG for short); and any UCCG is an essential graph that represents a Markov equivalence class (Andersson et al., 1997). We can calculate the size of $\mathcal{C}$ by counting the DAGs in Markov equivalence classes represented by its chain components using the following equation (Gillispie and Perlman, 2002; He and Geng, 2008):

$$\text{SizeMEC}(\mathcal{C}) = \prod_{i=1}^{k} \text{SizeMEC}(\mathcal{C}_{\tau_i}). \tag{1}$$

To count the size of Markov equivalence class represented by a UCCG, we can generate all equivalent DAGs in the class. However, when the number of vertices in the UCCG is large, the number of DAGs in the corresponding Markov equivalence class may be huge, and the traversal method proves to be infeasible to count the size. This paper tries to solve this counting problem for Markov equivalence classes of DAGs with hundred of vertices.

3. The Size of Markov Equivalence Class

In order to obtain the size of a Markov equivalence class, it is sufficient to compute the size of Markov equivalence classes represented by undirected and connected chordal graph (UCCGs) according to Lemma 2 and Equation (1). In Section 3.1, we discuss Markov equivalence classes represented by UCCGs whose sizes are functions of the number of vertices. Then in Section 3.2.1, we provide a method to partition a Markov equivalence class into smaller subclasses. Using these methods, finally in Section 3.2.2, we propose a recursive approach to calculate the size of a general Markov equivalence class.

3.1 Size of Markov Equivalence Class Determined by the Number of Vertices

Let $\mathcal{U}_{p,n}$ be an undirected and connected chordal graph (UCCG) with p vertices and n edges. Clearly, the inequality $p-1 \leq n \leq p(p-1)/2$ holds for any UCCG $\mathcal{U}_{p,n}$. When $\mathcal{U}_{p,n}$ is a tree, $n = p-1$ and when $\mathcal{U}_{p,n}$ is a completed graph, $n = p(p-1)/2$. Given p and n, in some special cases, the size of a UCCG $\mathcal{U}_{p,n}$ is completely determined by p. For example, it is well known that a Markov equivalence class represented by a completed UCCG with p vertices contains $p!$ DAGs. Besides the Markov equivalence classes represented by completed UCCGs, there are five types of UCCGs whose sizes are also functions of p. We present them as follows.

Theorem 3 *Let $\mathcal{U}_{p,n}$ be a UCCG with p vertices and n edges. In the following five cases, the size of the Markov equivalence class represented by $\mathcal{U}_{p,n}$ is determined by p.*

1. *If $n = p-1$, we have SizeMEC$(\mathcal{U}_{p,n}) = p$.*

2. *If $n = p$, we have SizeMEC$(\mathcal{U}_{p,n}) = 2p$.*

3. *If $n = p(p-1)/2 - 2$, we have* $\text{SizeMEC}(\mathcal{U}_{p,n}) = (p^2 - p - 4)(p-3)!$.

4. *If $n = p(p-1)/2 - 1$, we have* $\text{SizeMEC}(\mathcal{U}_{p,n}) = 2(p-1)! - (p-2)!$.

5. *If $n = p(p-1)/2$, we have SizeMEC$(\mathcal{U}_{p,n}) = p!$.*

For the UCCGs other than the above five cases, it seems that the sizes of the corresponding Markov equivalence classes cannot be completely determined by the numbers of vertices and edges; the sizes of these Markov equivalence classes may depend on the exact essential graphs. Below, we display several classes of this kind for $n = p + 1$ or $n = p(p-1)/2 - 3$ in Example 1.

Example 1. Figure 1 displays four UCCGs. Both $\mathcal{U}_{5,6}$ and $\mathcal{U}'_{5,6}$ have 6 edges, and both $\mathcal{U}_{5,7}$ and $\mathcal{U}'_{5,7}$ have 7 edges. We have that SizeMEC($\mathcal{U}_{5,6}$) = 13, SizeMEC($\mathcal{U}'_{5,6}$) = 12, SizeMEC($\mathcal{U}_{5,7}$) = 14 and SizeMEC($\mathcal{U}'_{5,7}$) = 30. Clearly, in these cases, the sizes of Markov equivalence classes are not completely determined by the numbers of vertices and edges.

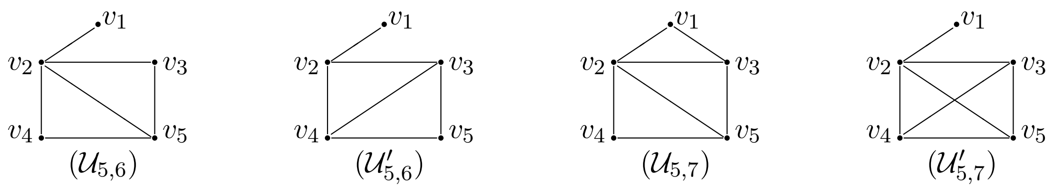

Figure 1: Examples that UCCGs with the same number of edges have different sizes.

3.2 Size of a General Markov Equivalence Class

In this section, we introduce a general method to count the size of a Markov equivalence class. We have shown in Theorem 3 that there are five types of Markov equivalence classes whose sizes can be calculated with five formulas respectively. For one any other Markov equivalence class, we will show in this section that it can be partitioned recursively into smaller subclasses until the sizes of all subclasses can be calculated with the five formulas above. We first introduce the partition method and the graph representation of each subclass in Section 3.2.1. Then provide a size counting algorithm for one arbitrary Markov equivalence class in Section 3.2.2. The proofs of all results in this section can be found in the Appendix.

3.2.1 Methods to Partition a Markov Equivalence Class

Let $\mathcal{U}$ be a UCCG, τ be the vertex set of $\mathcal{U}$ and let $\mathcal{D}$ be a DAG in the equivalence class represented by $\mathcal{U}$. A vertex v is a *root* of $\mathcal{D}$ if all directed edges adjacent to v are out of v, and $\mathcal{D}$ is *v-rooted* if v is a root of $\mathcal{D}$. To count DAGs in the class represented by $\mathcal{U}$, below, we show that all DAGs can be divided into different groups according to the roots of the DAGs and then we calculate the numbers of the DAGs in these groups separately. Each group is called as a rooted sub-class defined as follows.

Definition 4 (*a rooted sub-class*) *Let $\mathcal{U}$ be a UCCG over τ and $v \in \tau$. We define the v-rooted sub-class of $\mathcal{U}$ as the set of all v-rooted DAGs in the Markov equivalence class represented by $\mathcal{U}$.*

The following theorem provides a partition of a Markov equivalence class represented by a UCCG and the proof can be found in Appendix.

Theorem 5 (a rooted partition) *Let $\mathcal{U}$ be a UCCG over $\tau = \{v_1, \cdots, v_p\}$. For any $i \in \{1, \cdots, p\}$, the v_i-rooted sub-class is not empty and this set of p sub-classes forms a disjoint partition of the set of all DAGs represented by $\mathcal{U}$.*

Below we describe an efficient graph representation of v-rooted sub-class. One reason to this graph representation is that for any $v \in \tau$, the number of DAGs in the v-rooted sub-class might be extremely huge and it is computationally infeasible to list all v-rooted DAGs in this sub-class. Using all DAGs in which v is a root, we construct a rooted essential graph in Definition 6.

Definition 6 (rooted essential graph) *Let $\mathcal{U}$ be a UCCG. The v-rooted essential graph of $\mathcal{U}$, denoted by $\mathcal{U}^{(v)}$, is a graph that has the same skeleton as $\mathcal{U}$, and an edge is directed in $\mathcal{U}^{(v)}$ if and only if it has the same orientation in every v-rooted DAG of $\mathcal{U}$.*

From Definition 6, a rooted essential graph has more directed edges than the essential graph $\mathcal{U}$ since the root introduces some directed edges. Algorithm 3 in Appendix shows how to generate the v-rooted essential graph of a UCCG $\mathcal{U}$. We display the properties of a rooted essential graph in Theorem 7 and the proof can be found in Appendix.

Theorem 7 *Let $\mathcal{U}$ be a UCCG and $\mathcal{U}^{(v)}$ be a v-rooted essential graph of $\mathcal{U}$ defined in Definition 6. The following three properties hold for $\mathcal{U}^{(v)}$:*

1. *$\mathcal{U}^{(v)}$ is a chain graph,*
2. *every chain component $\mathcal{U}^{(v)}_{\tau'}$ of $\mathcal{U}^{(v)}$ is chordal, and*
3. *the configuration $v_1 \to v_2 - v_3$ does not occur as an induced subgraph of $\mathcal{U}^{(v)}$.*

Moreover, there is a one-to-one correspondence between v-rooted sub-classes and v-rooted essential graphs, so $\mathcal{U}^{(v)}$ can be used to represent uniquely the v-rooted sub-class of $\mathcal{U}$.

From Theorem 7, we see that the number of DAGs in a v-rooted essential graph $\mathcal{U}^{(v)}$ can be calculated by Equation (1) which holds for any essential graph. To use Equation (1), we have to generate all chain components of $\mathcal{U}^{(v)}$. Below we introduce an algorithm called *ChainCom*($\mathcal{U}$, v) in Algorithm 1 to generate $\mathcal{U}^{(v)}$ and all of its chain components.

We show that Algorithm 1 can generate rooted essential graph and the chain components of this essential graph correctly in the following theorem.

Theorem 8 *Let $\mathcal{U}$ be a UCCG and let v be a vertex in $\mathcal{U}$. Let $\mathcal{O}$ and $\mathcal{U}^{(v)}$ be the outputs of Algorithm 1 given $\mathcal{U}$ and v. Then $\mathcal{U}^{(v)}$ is the v-rooted essential graph of $\mathcal{U}$ and $\mathcal{O}$ is the set of all chain components of $\mathcal{U}^{(v)}$.*

The following example displays rooted essential graphs of a UCCG and illustrates how to implement Algorithm 1 to construct a rooted essential graph and how to generate all DAGS in the corresponding rooted sub-classes.

Example 2. Figure 2 displays an undirected chordal graph $\mathcal{U}$ and its rooted essential graphs. There are five rooted essential graphs $\{\mathcal{U}^{(v_i)}\}_{i=1,\cdots,5}$. We need to construct only $\mathcal{U}^{(v_1)}$, $\mathcal{U}^{(v_2)}$ and $\mathcal{U}^{(v_3)}$ since $\mathcal{U}^{(v_4)}$ and $\mathcal{U}^{(v_5)}$ are symmetrical to $\mathcal{U}^{(v_1)}$ and $\mathcal{U}^{(v_3)}$ respectively.

Algorithm 1: *ChainCom*$(\mathcal{U}, v)$

```
   Input: $\mathcal{U}$, a UCCG; $v$, a vertex of $\mathcal{U}$.
   Output: $\mathcal{U}^{(v)}$ and all of its chain components.
1  Set $A = \{v\}$, $B = \tau \setminus v$, $\mathcal{G} = \mathcal{U}$ and $\mathcal{O} = \emptyset$
2  while $B$ is not empty do
3      Set $T = \{w : w$ in $B$ and adjacent to $A\}$ ;
4      Orient all edges between $A$ and $T$ as $c \to t$ in $\mathcal{G}$, where $c \in A, t \in T$;
5      repeat
6          for each edge $y - z$ in the vertex-induced subgraph $\mathcal{G}_T$ do
7              if $x \to y - z$ in $\mathcal{G}$ and $x$ and $z$ are not adjacent in $\mathcal{G}$ then
8                  Orient $y - z$ to $y \to z$ in $\mathcal{G}$
9      until no more undirected edges in the vertex-induced subgraph $\mathcal{G}_T$ can be
       oriented;
10     Set $A = T$ and $B = B \setminus T$;
11     Append all isolated undirected graphs in $\mathcal{G}_T$ to $\mathcal{O}$;
12 Let $\mathcal{U}^{(v)} = \mathcal{G}$;
13 return $\mathcal{U}^{(v)}$ and $\mathcal{O}$
```

Clearly, they satisfy the conditions shown in Theorem 7. Given $\mathcal{U}$ in Figure 2, $\mathcal{U}^{(v_1)}$ is constructed according to Algorithm 1 as follows: (1) set $T = \{v_2, v_3\}$ in which vertices are adjacent to v_1, orient $v_1 - v_2, v_1 - v_3$ to $v_1 \to v_2, v_1 \to v_3$ respectively; (2) set $T = \{v_4, v_5\}$ in which vertices are adjacent to $\{v_2, v_3\}$, orient $v_2 - v_4, v_2 - v_5, v_3 - v_5$ to $v_2 \to v_4, v_2 \to v_5, v_3 \to v_5$ respectively; (3) orient $v_5 - v_4$ to $v_5 \to v_4$ because $v_3 \to v_5 - v_4$ occurs but v_3 and v_4 are not adjacent. By orientating the undirected edges of the chain components of a rooted essential graph with the constraint that no new v-structures and directed cycle occur, we can generate all DAGS in the corresponding sub-class (He and Geng, 2008; Meek, 1995; Verma, 1992). For example, consider $\mathcal{U}^{(v_1)}$ in Figure 2, we get two v_1-rooted DAGs by orienting $v_2 - v_3$ to $v_2 \to v_3$ or $v_2 \leftarrow v_3$.

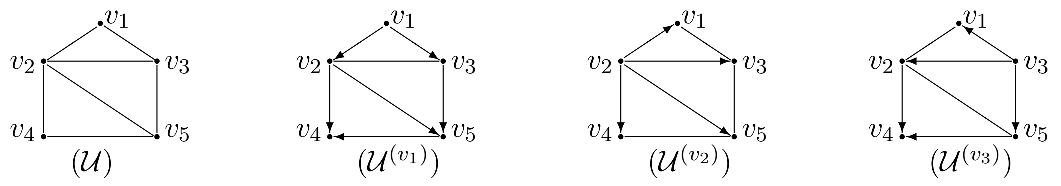

Figure 2: An undirected chordal graph $\mathcal{U}$ and its rooted essential graphs: $\mathcal{U}^{(v_1)}$, $\mathcal{U}^{(v_2)}$, and $\mathcal{U}^{(v_3)}$.

Now we can partition a Markov equivalence class represented by a UCCG into disjoint sub-classes, each of which can be represented by a rooted essential graph. In the next section, we will show how to recursively implement these partitions until the sizes of the subclasses or their essential graphs can be calculated with the five formulas in Theorem 3.

3.2.2 Calculating the Size of a Markov Equivalence Class

Let $\mathcal{U}$ be an undirected and connected chordal graph (UCCG) over τ. For any $v \in \tau$, $\mathrm{SizeMEC}(\mathcal{U}^{(v)})$ denotes the number of DAGs in v-rooted sub-class of $\mathcal{U}$. According to Theorem 5, the size of $\mathcal{U}$ can be calculated via the following corollary.

Corollary 9 *Let $\mathcal{U}$ be a UCCG over $\tau = \{v_i\}_{i=1,\cdots,p}$. We have $\mathrm{SizeMEC}(\mathcal{U}^{(v_i)}) > 1$ for $i = 1, \cdots, p$ and*

$$\mathrm{SizeMEC}(\mathcal{U}) = \sum_{i=1}^{p} \mathrm{SizeMEC}(\mathcal{U}^{(v_i)}). \tag{2}$$

This corollary shows that the size of Markov equivalence class represented by $\mathcal{U}$ can be calculated via the sizes of smaller sub-classes represented by $\{\mathcal{U}^{(v_i)}\}_{i=1,\cdots,p}$. The following example illustrates how to calculate the size of $\mathcal{U}$ in Figure 2.

Example 3. Consider again the undirected chordal graph $\mathcal{U}$ in Figure 2, $\mathrm{SizeMEC}(\mathcal{U})$ can be calculated as $\sum_{i=1}^{5} \mathrm{SizeMEC}(\mathcal{U}^{(v_i)})$ according to Corollary 9. The sizes of the five subclasses represented by $\mathcal{U}^{(v_1)}, \cdots, \mathcal{U}^{(v_5)}$ are $2, 4, 3, 2, 3$ respectively. Therefore, we can get that $\mathrm{SizeMEC}(\mathcal{U}) = 2 + 4 + 3 + 2 + 3 = 14$.

According to Theorem 7, for any $i \in \{1, \cdots, p\}$, the v_i-rooted essential graph $\mathcal{U}^{(v_i)}$ is a chain graph. If $\mathcal{U}^{(v_i)}$ is not directed, each of their isolated undirected subgraphs is a UCCG. Recall that we can calculate the size of a Markov equivalence class through its chain components using Equation (1), similarly, we can calculate the size of v_i-rooted sub-class of $\mathcal{U}$ with its isolated UCCGs as follows.

Corollary 10 *Let $\mathcal{U}^{(v_i)}$ be a v_i-rooted equivalent sub-class of $\mathcal{U}$ defined in Definition 6 and $\{\mathcal{U}^{(v_i)}_{\tau_j}\}_{j=1,\cdots,l}$ be the isolated undirected chordal sub-graphs of $\mathcal{U}^{(v_i)}$ over the vertex set τ_j for $j = 1, \cdots, l$. We have*

$$\mathrm{SizeMEC}(\mathcal{U}^{(v_i)}) = \prod_{j=1}^{l} \mathrm{SizeMEC}(\mathcal{U}^{(v_i)}_{\tau_j}). \tag{3}$$

Since $\{\mathcal{U}^{(v_i)}_{\tau_j}\}_{j=1,\cdots,l}$ are UCCGs according to Theorem 7, $\mathrm{SizeMEC}(\mathcal{U}^{(v_i)}_{\tau_j})$ can be calculated again via Equation (2) in Corollary 9 recursively. In this iterative approach, Equation (2) and Equation (3) are used alternately to calculate the sizes of equivalence classes represented by an undirected essential graph and a rooted essential graph.

Now in Algorithm 2 we present an enumeration to give *SizeMEC*$(\mathcal{U})$. Corollary 11 shows that the enumeration returns the size correctly. For any essential graph $\mathcal{C}$, we can calculate the size of Markov equivalence class represented by $\mathcal{C}$ according to Equation (1) and Algorithm 2.

Corollary 11 *Let $\mathcal{U}$ be a UCCG and SizeMEC$(\cdot)$ be the function defined in Algorithm 2. The function SizeMEC$(\mathcal{U})$ returns the size of Markov equivalence class represented by $\mathcal{U}$.*

The complexity of calculating $\mathrm{SizeMEC}(\mathcal{U})$ via Algorithm 2 depends on the number of times this recursive function is called. Our experiments in the next section show that when the number of vertices in $\mathcal{U}$ is small, or when the number is large but $\mathcal{U}$ is sparse, our

Algorithm 2: *SizeMEC*($\mathcal{U}$)

```
Input: $\mathcal{U}$: a UCCG.
Output: the size of Markov equivalence classes represented by $\mathcal{U}$
1  Let $p$ and $n$ be the numbers of vertices and edges in $\mathcal{U}$;
2  switch $n$ do
3      case $p-1$ return $p$;
4      case $p$ return $2p$;
5      case $p(p-1)/2-2$ return $(p^2-p-4)(p-3)!$;
6      case $p(p-1)/2-1$ return $2(p-1)!-(p-2)!$;
7      case $p(p-1)/2$ return $p!$;
8  for $j \leftarrow 1$ to $p$ do
9      $\{\mathcal{U}_1, \cdots, \mathcal{U}_{l_j}\} \leftarrow ChainCom(\mathcal{U}, v_j)$;
10     $s_j \leftarrow \prod_{i=1}^{l_j} SizeMEC(\mathcal{U}_i)$
11 return $\sum_{i=1}^{p} s_i$
```

proposed approach is efficient. However, when $\mathcal{U}$ is large and dense, the proposed approach may be computational infeasible since calculating SizeMEC($\mathcal{U}$) via Algorithm 2 may require a very deep recursion. In the worst case, the time complexity of Algorithm 2 might be $O(p!)$. For example, it might be extremely time-consuming to count SizeMEC($\mathcal{U}$) via Algorithm 2 when U is a UCCG with large p vertices and $p(p-1)/2-3$ edges. Fortunately, according to the experimental results in He et al. (2013), the undirected and connected chordal subgraphs in sparse essential graphs with hundreds of vertices are mostly small. This implies that our approach may be valuable for size counting in most situations of causal learning based on sparse graphical models.

In the next section, we demonstrate our approach experimentally and explore the size and edge distributions of Markov equivalence classes in sparse graphical models.

4. Experimental results

We conduct experiments to evaluate the proposed size counting algorithms in Section 4.1, and then to study sizes of Markov equivalence classes in Section 4.2. The main points obtained from these experiments are as follows.

1. Our proposed approach can calculate the size of classes represented by a UCCG with a few vertices ($p < 15$) in seconds on a laptop of 2.7GHz and 8G RAM. When the number of vertices is large, our approach is also efficient for the graphs with sparsity constraints.

2. For the essential graphs with sparsity constraint, the sizes of the corresponding Markov equivalence classes are nearly exponential in p. This explains the result in Chickering (2002) that causal structure searching in the space of Markov equivalence class models could be substantially more efficient than the searching in the space of DAG models for learning the sparse graphical models .

3. In the set of all Markov equivalence classes of DAGs with p vertices, most graphs are half-completed (nearly $p^2/4$ edges exist); and the Markov equivalent classes represented by these graphs have small sizes. This is a reason why the Markov equivalence classes have a small average size (approximately 3.7 reported by Gillispie and Perlman (2002)) even sparse Markov equivalence class is huge.

4.1 Calculating the Size of Classes Represented by UCCGs

In this section, we experimentally study the time complexity of our proposed counting algorithms for the UCCGs with a small p or with a large p but sparsity constraint. All experiments are run on a laptop with Intel 2.7GHz and 8G RAM. Note that the chain components are mostly small from sparse Markov equivalence classes with hundreds of vertices (He et al., 2013). The experimental results show that the proposed method is efficient to count the sizes of sparse Markov equivalence classes with hundreds of vertices.

Let $\mathbb{U}_p^{n*}$ be the set of Markov equivalence classes with p vertices and n edges. The graphs in $\mathbb{U}_p^{n*}$ are sparse if n is small multiply of p. We generate random choral graphs in $\mathbb{U}_p^{n*}$ as follows. First, we construct a tree by connecting two vertices (one is sampled from the connected vertices and the other from the isolated vertices) sequentially until all p vertices are connected. Then we randomly insert an edge such that the resulting graph is chordal, repeatedly until the number of edges reaches n. Repeating this procedure N times, we obtain N samples from $\mathbb{U}_p^{i*}$ for each $i \in [p-1, n]$.

We first consider the undirected chordal graphs with 5 to 13 vertices. Our experiments on $\mathbb{U}_p^{n*}$ for any $n < p(p-1)/2-3$ show that it is most time-consuming to calculate the size of UCCGs when $n = p(p-1)/2-3$. Based on the samples from $\mathbb{U}_p^{n*}$ where $n = p(p-1)/2-3$, we report in Table 1 the the maximum, the minimum and the average of the sizes of Markov equivalence classes and the time to count them. We see that the size is increasing exponentially in p and the proposed size-counting algorithm is computationally efficient for undirected chordal graphs with a few vertices.

Table 1: The size of Markov equivalence class and the time to calculate it via Algorithm 2 based on 10^5 samples from $\mathbb{U}_p^{n*}$, where p ranges from 5 to 13 and $n = p(p-1)/2-3$ (the worst case for classes with p vertices).

p		5	6	7	8	9	10	11	12	13
Size	Min	14	60	312	1920	1.36e4	1.11e5	1.00e6	1.02e7	1.12e8
	Mean	22	104	658	4508	3.27e4	2.90e5	2.96e6	2.92e7	3.57e8
	Max	30	144	828	5616	4.39e4	3.89e5	3.84e6	4.19e7	4.99e8
Time (sec.)	Min	0	0	1.0e-3	5.0e-3	2.8e-2	1.7e-1	1.3	10.6	95
	Mean	1.3e-4	4.3e-4	1.5e-3	6.8e-3	3.6e-2	2.2e-1	1.6	13.6	140
	Max	1.0e-3	1.0e-3	4.0e-3	1.3e-2	9.6e-2	6.4e-1	5.1	53.5	476

We also study the sets $\mathbb{U}_p^{n*}$ that contain UCCGs with tens of vertices. The number of vertices p is set to $15, 20, \cdots, 100$ and the edge constraint m is set to rp where r is the ratio of m to p. For each p, we consider four ratios: 2, 3, 4 and 5. The undirected chordal graphs in $\mathbb{U}_p^{rp*}$ are sparse since $r \leq 5$. Based on 10^5 samples, we report the averages of size and time in Table 2. We can see that when $r \leq 4$, the calculation just take a few seconds even if the sizes are very huge, when the chordal graphs become denser ($r > 4$), the calculation take more time.

Table 2: The average size of Markov equivalence class and its average counting time via Algorithm 2 are reported based on 10^5 samples from $\mathbb{U}_p^{pr*}$, where p ranges from 15 to 100.

r	p	15	20	30	40	50	60	70	80	90	100
2	Size	7363	6.98e4	4.74e6	6.94e8	1.9e10	1.2e12	1.2e14	1.5e15	1.8e17	2.6e19
3		3.0e5	3.3e6	1.1e10	7.1e12	4.4e15	8.6e18	1.3e21	6.1e23	1.4e27	9.1e27
4		2.7e6	5.4e8	6.7e12	2.8e16	3.5e19	5.9e22	5.8e25	1.3e29	1.3e38	1.5e34
5		4.9e7	6.7e9	8.3e14	5.4e18	1.1e24	2.8e26	2.3e30	4.8e33	5.6e40	3.8e40
2	Time (sec.)	3.2e-3	5.7e-3	1.2e-2	2.3e-2	0.028	0.037	0.059	0.074	0.090	0.15
3		1.7e-2	3.8e-2	8.8e-2	0.15	0.17	0.27	0.42	0.53	0.75	0.86
4		0.19	0.43	0.72	1.37	1.51	2.16	3.35	3.64	6.14	9.03
5		2.89	7.07	7.91	17.49	50.43	82.99	90.37	95.54	127.25	213

4.2 Size and Edge Distributions of Markov Equivalence Classes

In this section, we focus on the size and edge distributions of Markov equivalence classes of directed acyclic graphs by using an approach as follows. First, we generate a Markov chain on Markov equivalence classes of interest and simultaneously obtain the stationary distribution of the chain according to the methods in He et al. (2013). Then, based on the stationary distribution of the chain, we reweigh the samples from the chain and further use them to calculate the distribution of Markov equivalence classes of interest. In Section 4.2.1, we study the size and edge distributions of Markov equivalence classes with tens of vertices, and in Section 4.2.2, we provide the size distributions of Markov equivalence classes with hundred of vertices under sparsity constraints.

4.2.1 Size and Edge Distribution of Markov Equivalence Classes

In this section, we discuss the distributions of Markov equivalence classes on their sizes and number of edges. We use “size distribution” for the distribution on sizes of Markov equivalence classes, and “edge distribution” for the distribution on the number of edges. First, we consider the number of edges of Markov equivalence classes with p vertices for $10 \leq p < 20$. Then, we focus on the size and edge distribution of Markov equivalence classes with 20 vertices. Finally, we explore the size distributions of Markov equivalence classes with different numbers of edges to show how size distributions change with increasing numbers of edges.

The numbers of edges in the Markov equivalence classes with p vertices range from 0 to $p(p-1)/2$. Based on a Markov chain with length of 10^6 for each p, we display in Table 3 the modes and 99% intervals of edge distributions of Markov equivalence classes with p vertices for $10 \leq p < 20$. The mode is the number that appears with the maximum probability, 99% interval is the shortest interval that covers more than 99% of Markov equivalence classes. The ratios that measure the fraction of 99% interval to $p(p-1)/2+1$ are also given. For example, consider the edge distribution of Markov equivalence classes with 10 vertices; we see that 99% of Markov equivalence classes have between 17 and 32 edges. The ratio is $16/46 \approx 0.348$, where the number 16 is the length of the 99% interval $[17, 32]$ and 46 is the length of the edge distribution's support $[0, 45]$. From the 99% intervals and the corresponding ratios, we see that the numbers of edges of Markov equivalence classes are sharply distributed around $p^2/4$, and these distributions become sharper with increasing of p. This result is reasonable since the number of skeletons of essential graphs with k edges is $\binom{p(p-1)/2}{k}$, and the k-combination reaches maximum around $k = p^2/4$.

Table 3: The edge distributions of Markov equivalence classes with p vertices for $10 \leq p < 20$. The mode is the number that appears with the maximum probability, the 99% interval covers more than 99% of Markov equivalence classes, ratio is the fraction of the length of the 99% interval to the length of the support of edge distribution.

p	mode	99% interval	ratio	p	mode	99% interval	ratio
10	25	[17,32]	0.348	15	56	[44,68]	0.236
11	30	[22,39]	0.321	16	64	[51,77]	0.223
12	36	[26,45]	0.299	17	73	[59,87]	0.216
13	42	[32,53]	0.278	18	81	[66,96]	0.201
14	49	[38,60]	0.25	19	91	[75,106]	0.180

In Figure 3, we display the proportions of Markov equivalence classes with 20 vertices according to their sizes and the number of edges. Two scaled marginal distributions in the planes are also shown. The black dash line is the size distribution and the black solid line is the edge distribution of Markov equivalence classes. According to the marginal size distribution, we see that most of the Markov equivalence classes with 20 vertices have small sizes. For example, 26.89% of Markov equivalence classes are of size one; the proportion of Markov equivalence classes with size ≤ 10 is greater than 95%. We also see that the marginal edge distribution of Markov equivalences is concentrated around $100(= 20^2/4)$. The proportion of Markov equivalence classes with 20 vertices and 100 edges is nearly 6%.

To study how the size distribution changes with the number of edges, we consider Markov equivalence classes with 100 vertices and n edges for different n.

Figure 4 displays the size distribution of Markov equivalence classes with 100 vertices and n edges for $n = 10, 50, 100, 200, 400, 600, 1000, 1500, 2000$ and 2500, respectively. We see that the sizes of Markov equivalence classes are very small when the number of edges is close to $p^2/4 = 2500$. For example, when $n \in (1000, 2500)$, the median of the sizes is no more than 4. These results shed light on why the Markov equivalence classes with p vertices have a small average size.

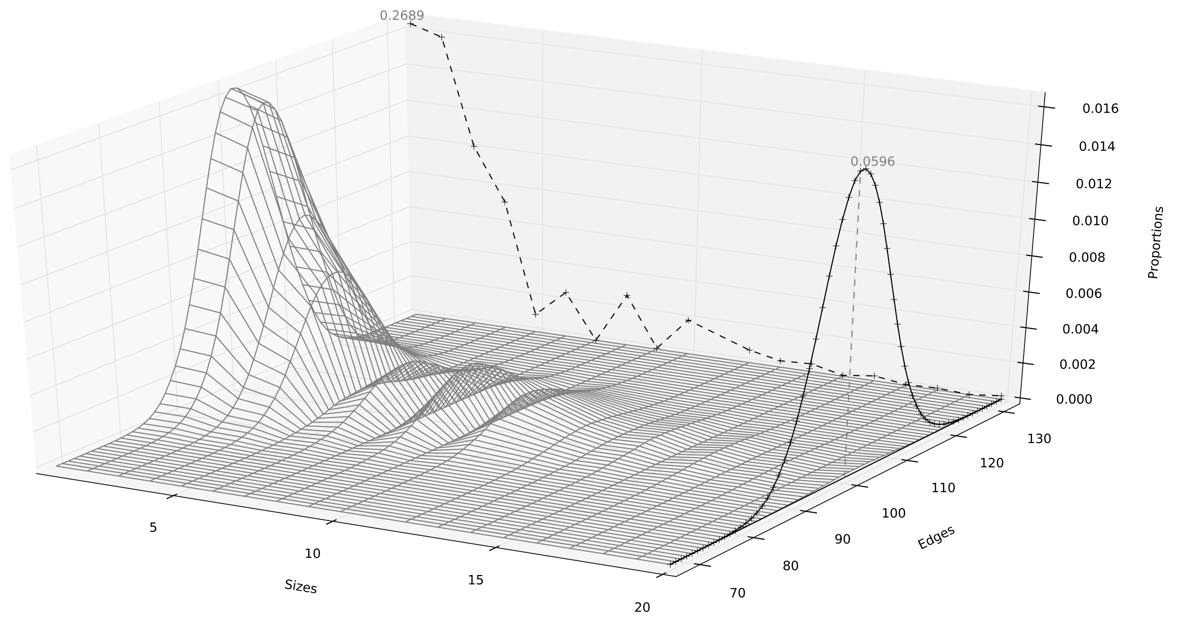

Figure 3: The surface displays the distribution of the Markov equivalence classes with 20 vertices. Two rescaled marginal distributions are shown in the planes. The black dash line is the size distribution and the black solid line is the edge distribution of Markov equivalence classes.

4.2.2 Size Distributions of Markov Equivalence Classes with Sparsity Constraints

We study Markov equivalence classes with p vertices and n vertices. The number of vertices p is set to $100, 200, 500$ or 1000 and the maximum number of edges n is set to rp where r is the ratio of n to p. For each p, we consider four ratios: 1.2, 1.5, 3 and 5. The essential graphs with p vertices and rp edges are sparse since $r \leq 5$. In each simulation, given p and r, a Markov chain with length of 10^6 Markov equivalence classes is generated.

There are sixteen distributions, each of which is calculated with 10^6 essential graphs. We plot the four distributions for $r = 1.2$ in the main window, and the other 12 distributions for $r = 1.5, 3, 5$ in three sub-windows, respectively. In each distribution, the 95% quantiles and 99% quantiles are marked with diamonds and circles, respectively. We see that the sizes of equivalence classes are extremely large. The medians of size distributions are connected by a dash line in Figure 5. It seems that there is a linear relationship between the logarithm of size and the number of vertices p. These results suggest that, to learn directed graphical models, a searching among Markov equivalence classes might be more efficient than that among DAGs.

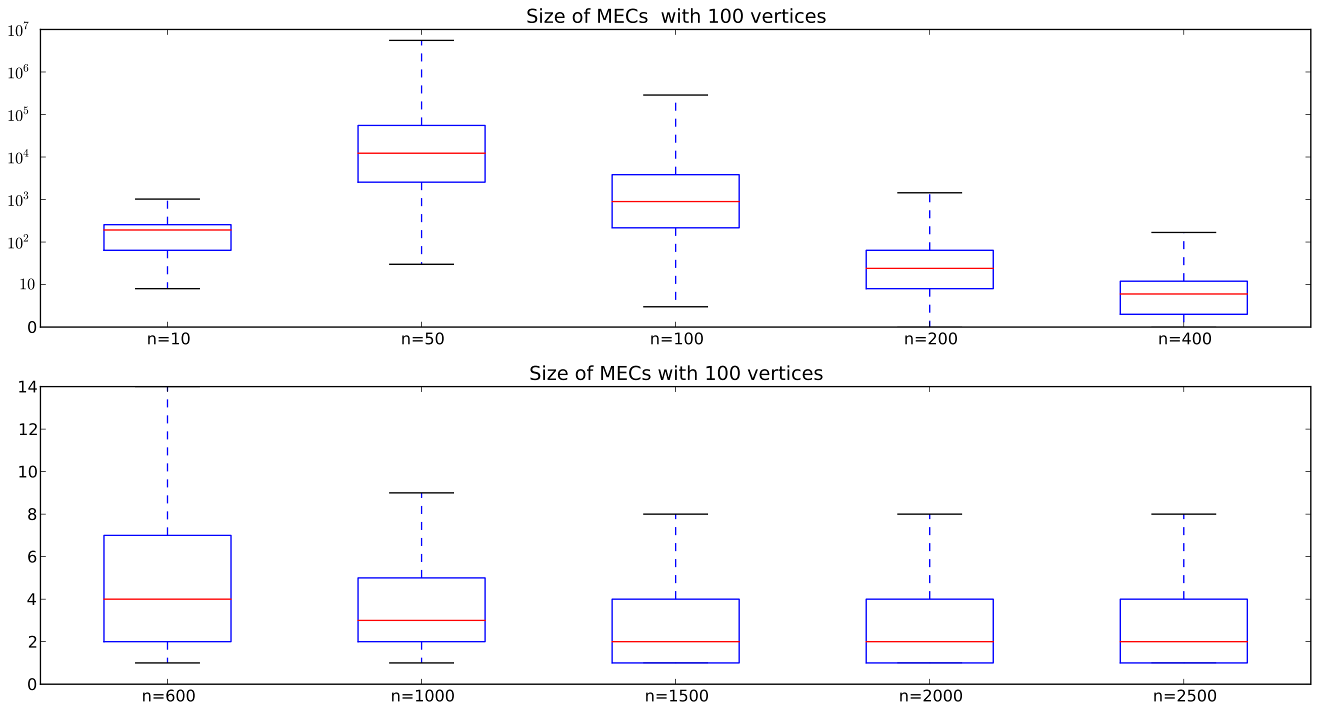

Figure 4: The size distributions of Markov equivalence classes with p vertices and n edges, where $n = 10, 50, 100,$ 200, 400, 600, 1000, 1500, 2000 and 2500, respectively.

5. Conclusions

In this paper, we propose a method to calculate the sizes of Markov equivalence classes. A rooted sub-class of a Markov equivalence class is introduced and the graph representation of this sub-class, called rooted essential graph, is proposed. We can partition a Markov equivalence class into smaller rooted sub-classes recursively until the sizes of all sub-classes can be obtained via five closed-form formulaes. Then we explore the size and edge distributions of Markov equivalence classes. We study experimentally how size distribution changes with the number of edges and report the size distributions of Markov equivalence classes with hundreds of vertices under sparsity constraints. We find that the essential graphs with around $p^2/4$ edges dominate in the set of all essential graphs with p vertices and the corresponding Markov equivalence classes have small sizes. This results in a small average size of all Markov equivalence classes with p vertices. For the sparse essential graphs with p vertices, we find that the sizes of the corresponding Markov equivalence classes are super-exponential in p.

Acknowledgments

This work was supported partially by NSFC (11101008,11101005, 71271211), DPHEC-20110001120113, US NSF grants DMS-1107000, CDS&E-MSS 1228246, ARO grant W911NF-

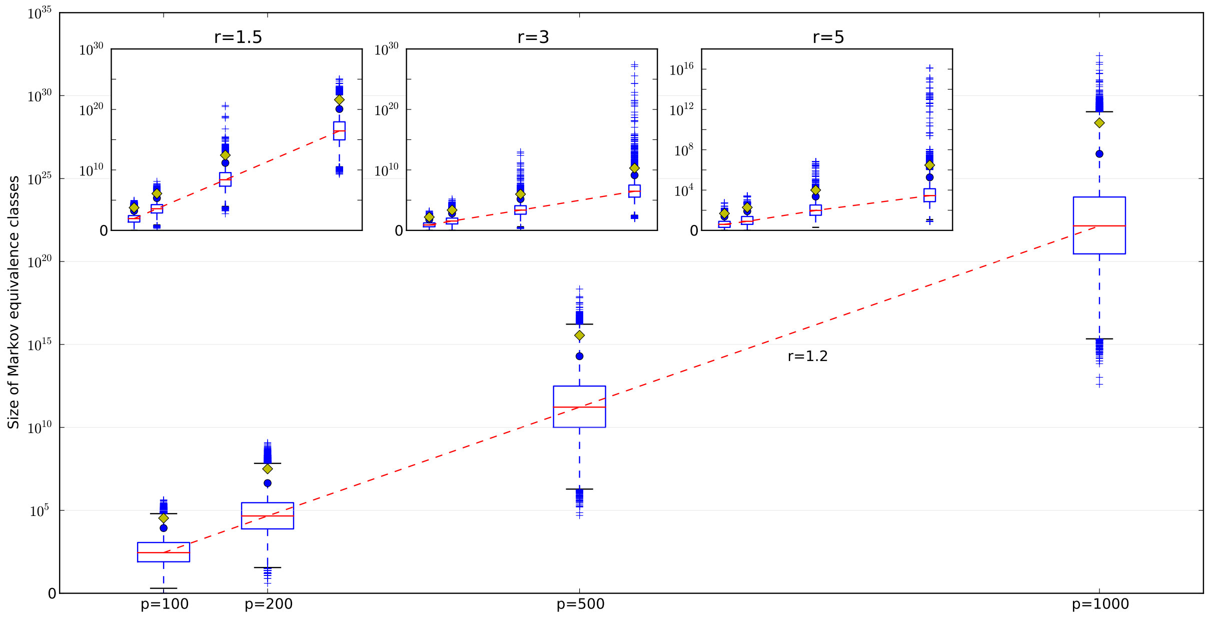

Figure 5: Size distributions of Markov equivalence classes with p vertices and at most rp edges. The lines in the boxes and the two circles above the boxes indicate the medians, the 95th, and the 99th percentiles respectively.

11-1-0114, and the Center for Science of Information (CSoI, a US NSF Science and Technology Center) under grant agreement CCF-0939370, and LMEQF.

References

S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

R. Castelo and M. D. Perlman. Learning essential graph Markov models from data. *Studies in Fuzziness and Soft Computing*, 146:255–270, 2004.

D. M. Chickering. Learning equivalence classes of Bayesian-network structures. *The Journal of Machine Learning Research*, 2:445–498, 2002.

M. Finegold and M. Drton. Robust graphical modeling of gene networks using classical and alternative t-distributions. *The Annals of Applied Statistics*, 5(2A):1057–1080, 2011.

N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science Signaling*, 303(5659):799, 2004.

S.B. Gillispie and M.D. Perlman. The size distribution for Markov equivalence classes of acyclic digraph models. *Artificial Intelligence*, 141(1-2):137–155, 2002.

Yangbo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9:2523–2547, 2008.

Yangbo He, Jinzhu Jia, and Bin Yu. Reversible mcmc on markov equivalence classes of sparse directed acyclic graphs. *The Annals of Statistics*, 41(1):1742–1779, 2013.

D. Heckerman, C. Meek, and G. Cooper. A Bayesian approach to causal discovery. *Computation, causation, and discovery*, pages 143–67, 1999.

R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N.J. Krogan, S. Chung, A. Emili, M. Snyder, J.F. Greenblatt, and M. Gerstein. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449, 2003.

M. H. Maathuis, M. Kalisch, and P. Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 2009. ISSN 0090-5364.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403–410, 1995.

Stijn Meganck, Philippe Leray, and Bernard Manderick. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In *Modeling Decisions for Artificial Intelligence*, pages 58–69. Springer, 2006.

J. Pearl. *Causality: Models, reasoning, and inference*. Cambridge Univ Pr, 2000.

M.D. Perlman. Graphical model search via essential graphs. *Contemporary Mathematics*, 287:255–266, 2001.

P. Spirtes, C.N. Glymour, and R. Scheines. *Causation, prediction, and search*. The MIT Press, 2001.

R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13:566, 1984.

T. Verma. A linear-time algorithm for finding a consistent expansion of a partially oriented graph,". Technical report, Technical Report R-180, UCLA Cognitive Systems Laboratory, 1992.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, page 270. Elsevier Science Inc., 1990.

Appendix A. Proofs

We place the proof of Theorem 3 in the end of Appendix because this proof will use the results in Algorithm 1 and Corollary 9.

Proof of Theorem 5:

We first show that τ_i-rooted sub-class is not empty. For any vertex $\tau_i \in \tau$, we just need to construct a DAG $\mathcal{D}$ in which no v-structures occurs and all edges adjacent to v are oriented out of v. The maximum cardinality search algorithm introduced by Tarjan and Yannakakis (1984) can be used to construct $\mathcal{D}$. Let p be the number of vertices in $\mathcal{U}$, the algorithm labels the vertices from p to 1 in decreasing order. We first label τ_i with p. As the next vertex to label, select an unlabeled vertex adjacent to the largest number of previously labeled vertices. We can obtain a directed acyclic graph $\mathcal{D}$ by orienting the undirected edges of $\mathcal{U}$ from higher number to lower number. Tarjan and Yannakakis (1984) show that no v-structures occur in $\mathcal{D}$ if $\mathcal{U}$ is chordal. Hence in $\mathcal{D}$, there is no v-structure and all edges adjacent to v are oriented out of v. We have that $\mathcal{D}$ is a τ_i-rooted equivalent DAG of $\mathcal{U}$, thus τ_i-rooted sub-class is not empty.

To prove that the p sub-classes, τ_i-rooted sub-classes for $i = 1, \cdots, p$, form a disjoint partition of Markov equivalence class represented by $\mathcal{U}$, we just need to show that every equivalent DAG of $\mathcal{U}$ is in only one of p sub-classes.

First, for any equivalent DAG of $\mathcal{U}$, denoted by $\mathcal{D}$, since $\mathcal{D}$ is a directed acyclic graph, there exists an order of its vertices such that all edges are oriented from the preceding vertices to their succeeding ones. Denoted by τ_i the first vertex of this order, we have that all edges adjacent to τ_i are oriented out of τ_i. Clearly, $\mathcal{D}$ is in the τ_i-rooted sub-class.

Suppose that $\mathcal{D}$ is also in another τ_j-rooted sub-class ($i \neq j$). Clearly, τ_i and τ_j are not adjacent. Since $\mathcal{U}$ is connected, we can find a shortest path $\mathcal{L} = \{\tau_i - \tau_k - \cdots - \tau_l - \tau_j\}$ from τ_i to τ_j with a length greater than two. We have that all vertices in $\mathcal{L}$ are not adjacent, and $v_i \to v_k$ and $v_j \to v_l$ are in $\mathcal{D}$. There must have a head to head structure like $\cdot \to \cdot \leftarrow \cdot$ in $\mathcal{L}$. Notice that $\mathcal{U}$ is an undirected chordal graph and $\mathcal{D}$ must be a DAG without v-structures. This is a contraction. We have that there is only one vertex that has no parents in $\mathcal{D}$ and so $\mathcal{D}$ is not in any other rooted sub-class. ■

Proof of Theorem 7:

Consider the proof of Theorem 6 in He and Geng (2008), we set the intervention variable to be v. If v is a root, Theorem 7 becomes a special case of Theorem 6 in He and Geng (2008). ■

Proof of Corollary 9:

Theorem 5 shows that for any $i \in \{1, 2, \cdots, p\}$, the τ_i-rooted sub-class of $\mathcal{U}$ is not empty and these p sub-classes form a disjoint partition of Markov equivalence class represented by $\mathcal{U}$. This deduces Corollary 9 directly. ■

Proof of Corollary 10:

Since $\{\mathcal{U}^{(v_i)}_{\tau_j}\}_{j=1,\cdots,l}$ are l isolated undirected chordal sub-graphs of $\mathcal{U}^{(v_i)}$, the orientations of the undirected edges in a component is irrelevant to the other undirected components. This results in Equation (3) follow directly. ■

Proof of *Theorem* 8:

The following Algorithm 3 and Lemma 12 are used in the proof of Theorem 8.

Algorithm 3: Find the v-rooted essential graph of $\mathcal{U}$

```
    Input: $\mathcal{U}$: an undirected and connected chordal graph; $v$: a vertex of $\mathcal{U}$.
    Output: the v-rooted essential graph of $\mathcal{U}$
1   Set $H = \mathcal{U}$;
2   for each edge $\cdot - v$ in $U$ do
3       Orient $\cdot - v$ to $\cdot \leftarrow v$ in $H$.
4   repeat
5       for each edge $y - z$ in $H$ do
            Rule 1: if there exists $\cdot \rightarrow y - z$ in $H$, and $\cdot$ and $z$ are not adjacent then
                Orient $y - z$ to $y \rightarrow z$ in $H$
            Rule 2: if there exists $y \rightarrow \cdot \rightarrow z$ in $H$ then
                Orient $y - z$ to $y \rightarrow z$ in $H$
6   until no more undirected edges in $H$ can be oriented;
7   return $H$
```

Lemma 12 *Let $\mathcal{U}$ be an undirected and connected chordal graph, v be a vertex of it. Let H be the output of Algorithm 3 given $\mathcal{U}$ and v. Then H is the v-rooted essential graph of $\mathcal{U}$.*

Proof This lemma follows from the proof of Theorem 6 in He and Geng (2008), in which they show Algorithm 3 can give the right essential graph. ∎

With the following lemma, Theorem 8 follows directly.

Lemma 13 *The output H of Algorithm 3 is the same as the output $\mathcal{U}^{(v)}$ of Algorithm 1 given the same $\mathcal{U}$ and v.*

Proof By comparing Algorithm 1 to Algorithm 3, we find that in Algorithm 1, Rule 1 that is shown in Algorithm 3 is used repeatedly and at the end of Algorithm 1, undirected edge in H can no longer be oriented by the Rule 1. If we further apply Rule 2 in Algorithm 3 to orient undirected edges in H until no undirected edges satisfy the condition in Rule 2. Denote the output as H'. Clearly, the output H' is the same as $\mathcal{U}^{(v)}$ obtained from Algorithm 1. Therefore, to show H is the same as $\mathcal{U}^{(v)}$, we just need to show the condition in Rule 2 does not hold for any undirected edge in H.

In Algorithm 1, we generate a set T in each loop of "while" and the sequence is denoted by $\{T_1, \cdots, T_n\}$. Setting $T_0 = \{v\}$, we have three facts as following

Fact 1 All undirected edges in H occur in the subgraphs over T_i for $i = 1, \cdots, n$.

Fact 2 All edges in H between T_i and T_{i+1} are oriented from T_i to T_{i+1} for $i = 0, cdots, n-1$.

Fact 3 There is no edge between T_i and T_j if the difference between i and j is greater than one.

Fact 4 There are no v-structures in H.

Suppose there exist three vertices x, y and z such that both $y \to x \to z$ and $y - z$ occur in H. Then a contradiction is implied.

Clearly, from **Fact 1**, there exists a set, denoted as T_i containing both y and z. Since $y \to x \to z$ occurs, from **Fact 2** and **Fact 3**, we have that $x \in T_i$.

Next we show that x, y and z have the same parents in T_{i-1}. First, y and z have the same parents in T_{i-1}; otherwise $y - z$ will be oriented to a directed edge. Denote by P_1 the same parents of y and z in T_{i-1} and by P_2 the parents of x in T_{i-1}. Second, for any $u \in P_1$, if u is not a parent of x, then $z - x$ in $\mathcal{U}$ will be oriented to $z \to x$ in H according to Algorithm 1. We have that u is also a parent of x and consequently, $P_1 \subseteq P_2$. Third, For any $u \in P_2$, u must be a parent of y according to **Fact 4**.

We have that $P_2 \subseteq P_1$, and finally $P_2 = P_1$. We get that neither $y \to x$ nor $x \to z$ is oriented by any directed edge $u \to y$ or $u \to x$ with $u \in T_{i-1}$ since $P_2 = P_1$.

Let $u_1 \in T_i$ and $u_1 \to y$ be the directed edge that orients $y - x$ in $\mathcal{U}$ to $y \to x$ in H. Clearly, $u_1 \to y$ occurs in H, and u_1 and x are not adjacent. Since $y - z$ is not directed in H, $u_1 - z$ must occur in $\mathcal{U}$. Similarly, there exists a vertex $u_2 \in T_i$ such that $u_2 \to x$ in H, and u_2 and z are not adjacent. According to Algorithm 1, there exists a path $u_1 - \cdots - u_2$ through the vertices in $T_{i-1}, T_{i-2}, \cdots, T_0 = \{v\}$. We can get the shortest sub-path of $u_1 - \cdots - u_2$, such that no chord contained in this path. Denote the shortest sub-path as $u_1 - * - u_2$. We have $u_1 - z - x - u_2 - * - u_1$ is a cycle without chord in $\mathcal{U}$ since (1)neither z nor x is in the path $u_1 - * - u_2$, (2) u_1 and x are not adjacent, and (3) u_2 and z are not adjacent. Clearly, the length of this cycle is at least 4. This gives a contradiction since $\mathcal{U}$ is a chordal graph. ■

Proof of Corollary 11:

If all outputs of the function ChainCom$(\cdot, \cdot)$ in Algorithm 2 contain at least one chain component, the Algorithm 2 outputs a number when all recursions stop. According to Corollary 9, Corollary 10, and Theorem 8, the output is the size of Markov equivalence class represented by $\mathcal{U}$.

Let $\mathcal{U}'$ be a connected induced subgraph of $\mathcal{U}$ over τ'. We have that $\mathcal{U}'$ is an undirected chordal graph. Notice that in Algorithm 2, we use the function ChainCom$(\mathcal{U}', v)$ for any vertex $v \in \tau'$ only if $\mathcal{U}'$ is not any case of Theorem 3. Clearly, $\mathcal{U}'$ is not a tree. Next, we show that $\mathcal{U}'^{(v)}$ contains at least one chain component for any vertex $v \in \tau'$.

According to the proof of Theorem 8, we partition τ' into subsets as $\{T'_0, T'_1, \cdots, T'_{n'}\}$, where $T'_0 = \{v\}$. Clearly, $\mathcal{U}'$ is a tree if and only if there are no edges in the subgraphs induced by T'_i (for any $i = 1, \cdots, n'$). Since $\mathcal{U}'$ is not a tree, there exist at least one integer i such that the subgraph induced by T'_i is not empty. Let i_1 be the smallest integer such that the subgraph $\mathcal{U}'_{T'_{i_1}}$ is not empty. We have that $x, y \in T'_{i_1}$ and $x - y$ occurs in $\mathcal{U}'_{T'_{i_1}}$. Consider a connected subgraph of $\mathcal{U}'_{T'_{i_1}}$ that contains $x - y$, denoted as G. We have that all vertices in G are adjacent to a vertex in T'_{i_1-1} and there are no other edges between G and T'_{i_1-1}. In Algorithm 1, we have that all vertices in G have the same parents (also only one) in T'_{i_1-1}. According to Algorithm 1, all edges in G must be undirected in $\mathcal{U}'^{(v)}$. ■

Proof of Theorem 3:

Proof of (1):

For a UCCG $\mathcal{U}_{p,n}$, if $n = p - 1$, then the graph $\mathcal{U}_{p,n}$ is a tree. For any vertex in $\mathcal{U}_{p,n}$, we have that $\mathcal{U}_{p,n}^{(v)}$ is a DAG according to Algorithm 1. Thus SizeMEC($\mathcal{U}_{p,n}^{(v)}$) = 1. Then, according to Corollary 9, SizeMEC($\mathcal{U}_{p,n}$) = p.

Proof of (2):

For a UCCG $\mathcal{U}_{p,n}$, if $n = p$, then the graph $\mathcal{U}_{p,n}$ has one more edge than tree. Because $\mathcal{U}_{p,n}$ is chordal, a triangle occurs in $\mathcal{U}_{p,n}$. For any vertex v in $\mathcal{U}_{p,n}$, we have that SizeMEC($\mathcal{U}_{p,n}^{(v)}$) = 2. Consequently, we have that SizeMEC($\mathcal{U}$) = $2p$ according to Corollary 9.

Proof of (3):

Let $v_1, \cdots, v_p$ be the p vertices of $\mathcal{U}_{p,n}$. There are only two pairs of vertices that are nonadjacent since $p(p-1)/2 - 2$ edges appear in $\mathcal{U}_{p,n}$. We first prove that these two pairs have a common vertex. Suppose $v_i - v_j$ and $v_k - v_l$ do not occur in $\mathcal{U}_{p,n}$ and v_i, v_j, v_k, v_l are distinct vertices. Consider the subgraph induced by v_i, v_j, v_k, v_l of $\mathcal{U}_{p,n}$. Clearly, the cycle $v_i - v_k - v_j - v_l - v_i$ occurs in the induced graph and $\mathcal{U}_{p,n}$ is not a chordal graph. We have that the missing two edges in $\mathcal{U}_{p,n}$ are like $v_1 - v_j - v_3$.

According to Corollary 9, we have that

$$\mathit{SizeMEC}(\mathcal{U}_{p,n}) = \sum_{i=1}^{p} \mathit{SizeMEC}(\mathcal{U}_{p,n}^{(v_i)}).$$

We first consider $\mathcal{U}_{p,n}^{(v_1)}$. All edges adjacent to v_2 in $\mathcal{U}_{p,n}^{(v_1)}$ are oriented to directed edges whose arrow is v_2 according to Algorithm 1 since v_2 is a neighbor of all neighbors of v_1, and v_1, v_2 are not adjacent in $\mathcal{U}_{p,n}^{(v_1)}$. Removing v_2 from $\mathcal{U}_{p,n}^{(v_1)}$, we have that the induced graph over $v_1, v_3, \cdots, v_p$ is a completed graph. This implies that the induced graph over $v_3, \cdots, v_p$ is an undirected completed graph with $p-2$ vertices. Therefore, we have $\mathit{SizeMEC}(\mathcal{U}_{p,n}^{(v_1)}) = (p-2)!$.

Similarly, we can get that $\mathit{SizeMEC}(\mathcal{U}_{p,n}^{(v_3)}) = (p-2)!$ since v_1 and v_3 are symmetric in $\mathcal{U}_{p,n}$.

We now consider $\mathcal{U}_{p,n}^{(v_4)}$. According to Algorithm 1, to construct $\mathcal{U}_{p,n}^{(v_4)}$, we first orient the undirected edges adjacent to v_4 in $\mathcal{U}_{p,n}$ to directed edges out of v_4. Since v_4 is adjacent to all other vertices in $\mathcal{U}_{p,n}$, there are no subgraphs like $v_4 \rightarrow v_i - v_j$ with v_4 and v_j nonadjacent. This results in that the chain component of $\mathcal{U}_{p,n}^{(v_4)}$ is a graph with $p-1$ vertices and $(p-1)(p-2)/2 - 2$ edges (only $v_1 - v_2 - v_3$ missing). We have that $\mathit{SizeMEC}(\mathcal{U}_{p,n}^{(v_4)}) = \mathit{SizeMEC}(\mathcal{U}_{p-1,(p-1)(p-2)-2})$.

Similarly, we can get that $\mathit{SizeMEC}(\mathcal{U}^{(v_i)}) = \mathit{SizeMEC}(\mathcal{U}_{p-1,(p-1)(p-2)-2})$ for any $i \leq 4$ since exchange the labels of these vertices will not change $\mathcal{U}$.

Therefore, we have proved the following formula

$$\mathit{SizeMEC}(\mathcal{U}_{p,n}) = (p-3)\mathit{SizeMEC}(\mathcal{U}_{p-1,(p-1)(p-2)-2}) + 2(p-2)! + 2(p-3)!,$$

Finally, we show that

$$\mathit{SizeMEC}(\mathcal{U}_{p,n}) = (p^2 - p - 4)(p-3)!$$

satisfies the formula and initial condition. First, we have $\mathit{SizeMEC}(\mathcal{U}_{4,4}) = (16-8) * 1 = 8$. Suppose $\mathit{SizeMEC}(\mathcal{U}_{p,n}) = (p^2 - p - 4)(p-3)!$ holds for $p = j - 1$,

$$\begin{aligned}
&\mathit{SizeMEC}(\mathcal{U}_{j,j(j-1)/2-2})\\
&= (j-3)\mathit{SizeMEC}(\mathcal{U}_{j-1,(j-1)(j-2)/2-2}) + 2(j-2)! + 2(j-3)!\\
&= (j-3)\{[(j-1)^2-(j-1)-4][(j-1)-3]!\} + 2(j-2)! + 2(j-3)!\\
&= [(j-1)^2-(j-1)-4+2(j-2)+2](j-3)!\\
&= (j^2-j-4)!(j-3)!
\end{aligned}$$

As a result, $\mathit{SizeMEC}(\mathcal{U}_{p,p(p-1)/2-2}) = (p^2-p-4)(p-3)!$ holds for $p=j$.

Proof of (4):

From the condition, only one pair of vertices, denoted by v and u, is not adjacent in $\mathcal{U}_{p,n}$. Consider a v-rooted equivalence sub-class, all undirected edges adjacent to u are oriented to directed edges with arrows pointing to u, and all other edges can be orientated as a completed undirected graph. We have that SizeMEC($\mathcal{U}^{(v)}_{p,n}$) $= (p-2)!$. Similarly, we have that SizeMEC($\mathcal{U}^{(u)}_{p,n}$) $= (p-2)!$. For any vertex w other than v and u, consider any DAG in the w-rooted equivalent sub-class, all edges adjacent to w are oriented away from w, and all other edges form a new chain component with $p-1$ vertices and $(p-1)(p-2)/2-1$ edges. Consider SizeMEC($\mathcal{U}_{p,p(p-1)/2-1}$) as a function of p, denoted by $f(p)$. When $p=3$, we have $f(3)=3$. Hence we have following formula:

$$f(p) = (p-2)f(p-1) + 2((p-2)!).$$

Now, we show that

$$f(p) = 2(p-1)! - (p-2)!$$

satisfies the formula and initial condition. First, we have $f(3) = 2*2-1 = 3$. Suppose $f(p) = 2(p-1)! - (p-2)!$ holds for $p=j-1$,

$$\begin{aligned}
f(j) &= (j-2)f(j-1) + 2(j-2)!\\
&= (j-2)(2(j-2)! - (j-3)!) + 2(j-2)!\\
&= 2(j-2)(j-2)! - (j-2)! + 2(j-2)!\\
&= (j-2)!(2j-3)\\
&= 2(j-1)! - (j-2)!
\end{aligned}$$

As a result, $f(p) = 2(p-1)! - (p-2)!$ holds for $p=j$.

Proof of (5):

If $\mathcal{U}$ is an undirected and connected graph with p vertices, and $p(p-1)/2$ edges, then the graph is a complete graph. There are $p!$ DAGs in the Markov equivalence class. ■