Journal of Statistical Research *ISSN 0256 - 422 X*
2008, Vol. 42, No. 2, pp. 195-213
Bangladesh

SEARCHING ACROSS MARKOV EQUIVALENT DIRECTED ACYCLIC GRAPH MODELS

R. Ayesha Ali
Department of Mathematics & Statistics
University of Guelph, Guelph, ON N1G 2W1, Canada
Email: aali@uoguelph.ca

M. Angélique Massie
Department of Mathematics & Statistics
University of Guelph, Guelph, ON N1G 2W1, Canada
Email: mmassie@uoguelph.ca

SUMMARY

Learning the structure of a process that can be represented by a directed acyclic graph (DAG) based on data alone can be a challenging problem because many graphs may encode the same conditional independence relations. However, searching across equivalence classes can greatly reduce the search space, thereby making the search more efficient. This paper presents the DECS algorithm, which is an extension of Edwards and Havernack's EH-procedure (Edwards, 1995) for undirected graphs to DAG equivalence classes. We also provide necessary graphical criterion for the DAG submodel relation and prove its sufficiency in special cases. This criterion facilitates the moves made across equivalence classes in the search space. Finally, the DECS algorithm is demonstrated on real data sets.

Keywords and phrases: directed acyclic graphs, Markov equivalence, essential graph, submodel relation, model search

AMS Classification: 68T30, 05C75, 68T37

1 Introduction

Learning Bayesian networks from data has long been a focus of machine learning, as well as problems in the social sciences. For systems that can be represented by directed acyclic graphs (DAGs), it is well-known that the number of possible DAGs grows super-exponentially with the number of variables in the system. It is also well-known that for any DAG, there are often many other DAGs to which it is Markov equivalent. That is, many graphs may represent the same set of conditional independence relations. Hence, searching across equivalence classes of DAGs may prove to be more efficient than searching across individual DAGs. In this paper, we present a search-and-score procedure for

small systems, which extends Edwards and Havernack's EH-procedure (Edwards, 1995) for undirected graphs to DAGs.

Gillispie and Perlman (2001) showed that, for $N \leq 10$ variables, while the number of equivalence classes may be smaller than the number of DAGs over a given variable set, the number of equivalence classes still grows super-exponentially. For example, over $N = 4$ variables, there are 185 different equivalence classes. But over $N = 5$ variables, the number of equivalence classes jumps to $8,782$. Hence, an exhaustive search procedure is typically computationally infeasible. Steinsky (2003) established this super-exponential rate of growth of the number of DAG equivalence classes by providing a lower bound on the growth rate.

Chickering (2002b) presented a greedy equivalence class search algorithm (GES) which either adds or removes edges from the output graph, based on improvement of an associated search score. In phase I, edges are successively added to an empty graph until there is no improvement in the search score. In phase II, edges are successively removed until there is no improvement of the associated search score. The advantage of this procedure is that it vastly reduces the complexity of the search space because at each step only neighbours of a given state need be evaluated. Further, the number of states visited is, in general, sparse relative to the total number of states. GES is optimal in the sense that it will asymptotically identify a local maximum, provided it exists. However, it is not clear how close this local maximum is to the global maximum, and the procedure risks not considering other good candidate models.

For applications with only a few variables in the system, we believe it is feasible to explore more of the search space and seek out some of these other good candidate models. However, since the search space is rather complex, it is still necessary to reduce the complexity of the search space. We will exploit the *coherence principle* to quickly evaluate submodels and supermodels associated with visited states. Based on results from Chickering (2002b), we derive graphical relations that hold when one DAG is a submodel of another.

In Section 2 we set up our notation and provide relevant background. In Section 3 we discuss the DAG submodel relation and present the DAG equivalence class search (DECS) procedure, and in Section 4 we provide a simple demonstration of the algorithm.

2 Background

2.1 Notation and Terminology

Throughout the paper, we will let N represent the number of variables (vertices) or nodes present in a graph, n represent the maximum number of edges in a graph with N nodes, i.e. $n = \binom{N}{2}$, and let m represent the number of observations in the available data set.

A graph is given by $\mathcal{G} = \{V, E\}$, where $V = \{x_1, \ldots, x_N\}$ is the set of variables and E is the set of edges that connect the vertices. A graphical model is a set of distributions that can be represented by $(\mathcal{G}, \boldsymbol{\Theta})$ where $\mathcal{G}$ encodes the set of conditional independence relations compatible with the distributions it represents, and $\boldsymbol{\Theta}$ represents the set of parameter values

that specify the conditional distributions encoded by $\mathcal{G}$. We consider distributions that belong to the curved exponential family, such as the Gaussian and multinomial distributions.

If all edges in E are undirected, then the graph is called an *undirected graph*. If E consists of only *directed* edges and there is no path $x \to \ldots \to x$, then $\mathcal{G}$ is a directed acyclic graph. Nodes x and y are *adjacent* if there is an edge between them. If $x \to y$ then x is a *parent* of y, and y is a *child* of x. Similarly, if $x \to \cdots \to y$ then x is an *ancestor* of y and y is a *descendant* of x. We use $pa_{\mathcal{D}}(\cdot)$ to denote the parents of some node in the graph $\mathcal{D}$. The *skeleton* of a DAG is the undirected graph obtained by converting all arrows to undirected edges.

For DAGs, $\langle x, y, z\rangle$ forms a triple if x is adjacent to y and y is adjacent to z (x is possibly adjacent to z). If $x \to y \leftarrow z$, we call the triple a *collider*, otherwise the triple is a *non-collider* (i.e. $x \to y \to z$, $x \leftarrow y \leftarrow z$, $x \leftarrow y \to z$). Further, if x is adjacent to z, then the triple is *shielded*; otherwise, it is *unshielded*. If $\boldsymbol{\mu}(a, b)$ is a path between nodes a and b containing vertex y, then we may refer to y as a collider or non-collider on $\boldsymbol{\mu}$, thereby implicitly specifying the triple as the node before y on $\boldsymbol{\mu}(a, b)$, y, and the node after y on $\boldsymbol{\mu}(a, b)$.

2.2 d-connection

Verma and Pearl (1991) introduced *d-separation*, a set of graphical conditions by which conditional independence relations could be read from a DAG.

Definition 2.1. Let a and b be distinct vertices in a DAG $\mathcal{D}$ and let S be a subset of vertices with $a, b \notin S$. A path $\boldsymbol{\mu}$, between a and b, is said to be *d-connecting given* S if the following hold:

(i) no non-collider on $\boldsymbol{\mu}$ is in S; and

(ii) every collider on $\boldsymbol{\mu}$ is an ancestor of a vertex in S.

Two vertices a and b are said to be *d-separated given* S in $\mathcal{D}$ if there is no path d-connecting a and b given S in $\mathcal{D}$. Likewise, sets A and B are d-separated given S in $\mathcal{D}$ if for every pair $a \in A$, and $b \in B$, a and b are d-separated given S.

Definition 2.2. In a DAG, a d-connecting path $\boldsymbol{\pi}(a, b)$ given S, between a and b, is said to be *minimal* if no order preserving (proper) subsequence of the vertices on $\boldsymbol{\pi}$ forms a d-connecting path between a and b given S.

It is simple to see that if there is some path d-connecting a and b given S then there is a minimal path which d-connects a and b given S. If $\boldsymbol{\pi} = \langle v_1, \ldots, v_p\rangle$ is a path, then we will refer to any pair of vertices (v_i, v_j) for which $|i - j| > 1$ as *non-consecutive vertices on* $\boldsymbol{\pi}$. Ali et al. (2008) proved the next Lemma which states that on a minimal d-connecting path only certain non-consecutive vertices may be adjacent. (This result was actually proved for minimal m-connecting paths in ancestral graphs, but is still valid in the present context since DAGs are a subclass of ancestral graphs, and m-connection is equivalent to d-connection when applied to DAGs).

Lemma 2.1. *Let $\boldsymbol{\pi}$ be a minimal d-connecting path between a and b given S in the DAG $\mathcal{D}$. If i and j are two non-consecutive vertices on $\boldsymbol{\pi}$ that are adjacent in $\mathcal{D}$ ($a = i$ or $j = b$ are possible) then exactly one of i and j is:* (i) *a collider on $\boldsymbol{\pi}$;* (ii) *in S; and* (iii) *a parent of the other vertex.*

Note that the existence of non-consecutive vertices on a minimal d-connecting path implies that there are at least four vertices on the path.

Corollary 2.1. Let $\mathcal{D}$ be a DAG with no unshielded colliders and $\boldsymbol{\pi}(a, b)$ be a minimal d-connecting path in $\mathcal{D}$ between nodes a and b. Then every triple along $\boldsymbol{\pi}(a, b)$ is an unshielded non-collider.

Proof. For a contradiction, suppose there are shielded triples along $\boldsymbol{\pi}$. Let $\langle x, y, z\rangle$ be the triple closest to a on $\boldsymbol{\pi}$ that is shielded. $\langle x, y, z\rangle$ is not a collider: $\mathcal{D}$ contains only singly directed edges, and by Lemma 2.1, at least one of x or z is a collider on $\boldsymbol{\pi}$ which contradicts y being a collider on $\boldsymbol{\pi}$. Hence, $\langle x, y, z\rangle$ is a non-collider.

By definition of $\langle x, y, z\rangle$, and by Lemma 2.1, z is a collider on $\boldsymbol{\pi}$ and $z \rightarrow x$ in $\mathcal{D}$. Let z_1 be the node after z on $\boldsymbol{\pi}$. Since $\mathcal{D}$ does not contain any unshielded colliders, $\langle y, z, z_1\rangle$ is shielded. Further, since y is a non-collider on $\boldsymbol{\pi}$, by Lemma 2.1 z_1 is a collider on $\boldsymbol{\pi}$, which contradicts z being a collider on $\boldsymbol{\pi}$. Hence, every triple on $\boldsymbol{\pi}$ is unshielded, and by definition of $\mathcal{D}$, is a non-collider. □

Corollary 2.1 is used to prove Theorem 3 in Section 3.1.

2.3 Markov Equivalence

Associated with every DAG $\mathcal{D}$ is a *global Markov property* which states that if disjoint sets A and B are d-separated given S (S may be the null set), then A is independent of B given S in the set of distributions represented by $\mathcal{D}$. If two graphs $\mathcal{D}$ and $\mathcal{D}^*$ encode the same set of independence relations, then we say the graphs are *Markov equivalent* to each other. Let $\mathfrak{I}(\mathcal{D})$ be the independence relations associated with graph $\mathcal{D}$. Then for a Markov equivalent graph $\mathcal{D}^*$, $\mathfrak{I}(\mathcal{D}) = \mathfrak{I}(\mathcal{D}^*)$. Verma and Pearl (1991) provided the following graphical characterization of Markov equivalence for DAGs.

Theorem 1. *Let $\mathcal{D}$ and $\mathcal{D}^*$ be two DAGs over vertex set V. $\mathcal{D}$ is* Markov equivalent *to $\mathcal{D}^*$ if and only if $\mathcal{D}$ and $\mathcal{D}^*$ have the same skeleton and same unshielded colliders.*

Chickering (2002a) also provided a transformational characterization of Markov equivalence for DAG models. The set of all DAGs that are Markov equivalent to each other forms an *equivalence class*, and we will use $\mathcal{E}$ to denote equivalence classes of DAGs. If DAG $\mathcal{D}$ is in the class $\mathcal{E}$, then $\mathfrak{I}(\mathcal{E}) = \mathfrak{I}(\mathcal{D})$.

Consider an edge that has the same orientation in every member (DAG) of an equivalence class to be *compelled*. Then an easy way to construct $\mathcal{E}$ from a DAG $\mathcal{D}$ is to let $\mathcal{E}$ have the same skeleton and same compelled edges as $\mathcal{D}$. In other words, for each directed edge in $\mathcal{E}$, the edge has the same orientation in every DAG in the equivalence class. For every

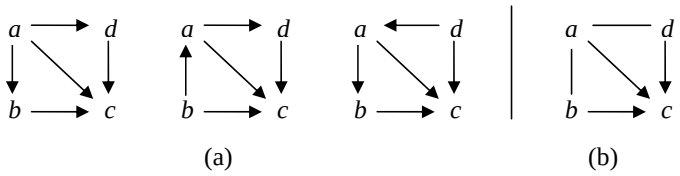

Figure 1: (a) members of a Markov equivalence class; (b) the associated essential graph.

undirected edge $x - y$ in $\mathcal{E}$, there is at least one member with $x \rightarrow y$ and another member with $x \leftarrow y$. The representative $\mathcal{E}$ is called an essential graph by Andersson et al. (1997), a pattern by Spirtes and Richardson (1997), a maximally oriented graph by Meek (1995) and a completed PDAG by Chickering (2002b). Figure 1(a) shows a set of Markov equivalent DAGs and Figure 1(b) shows the associated essential graph. Andersson et al. (1997) showed that the essential graph is a chain graph and that, indeed, $\mathfrak{I}(\mathcal{E}) = \mathfrak{I}(\mathcal{D})$.

2.4 Non-colliders in Markov equivalent DAGs

Although one need not worry about non-colliders when determining Markov equivalence between two DAGs, this does not mean that Markov equivalent DAGs do not have common non-colliders. For instance, trivially, if two DAGS share the same unshielded colliders, then they also share the same unshielded non-colliders. In fact, some Markov equivalent graphs also share some shielded non-colliders.

Consider the graph $\mathcal{D}_2$ shown in Figure 2(a). Every edge in this graph is compelled since any edge reversal would either create or destroy an unshielded collider, or would violate the acyclic condition. Hence, b is a shielded (tail-to-tail) non-collider along the path $\langle x, q, b, y\rangle$ in every graph Markov equivalent to $\mathcal{D}_2$. In fact, $\mathcal{D}_2$ is the only member of the equivalence class, and is its own essential graph.

Definition 2.3. A path $\boldsymbol{\pi} = \langle x, q, b, y\rangle$ forms a *compelled shielded non-collider* in a DAG $\mathcal{D}$ if all the edges among these four vertices are oriented as in $\mathcal{D}_2$ of Figure 2(a).

This type of path was first introduced in the more general setting of ancestral graphs by Spirtes et al. (1993) as *discriminating paths*, and by Ali and Richardson (2002) as *discriminating paths with order*. No mention of such paths arise in the graphical criterion for determining Markov equivalence of DAGs because identifying $\langle x, q, b\rangle$ as an unshielded collider (and implicitly $\langle x, q, y\rangle$ as an unshielded non-collider) in every DAG within the equivalence class is sufficient for determining that $\langle q, b, y\rangle$ is a non-collider in every DAG within the equivalence class. Further, since all edges are singly directed, and DAGs are acyclic, the non-collider is always of the form $q \leftarrow b \rightarrow y$. Hence, $\{q, b\}$ must be in any set that d-separates x and y.

However, identifying such paths becomes important for identifying submodels and supermodels of DAGs. The DAG submodel relation is briefly discussed in the next section, and the importance of compelled shielded non-colliders is highlighted in Section 3.

2.5 Submodel Relations

Let $\mathcal{G}$ and $\mathcal{G}^*$ be two graphs. If $\mathfrak{I}(\mathcal{G}^*) \subseteq \mathfrak{I}(\mathcal{G})$, then $\mathcal{G}$ is a submodel of $\mathcal{G}^*$, and likewise, $\mathcal{G}^*$ is a supermodel of $\mathcal{G}$. For undirected graphs, subgraphs are submodels, but this is not the case for DAGs or for essential graphs. In particular, essential graph $\mathcal{E}_1$ is a submodel of $\mathcal{E}_2$ if $\mathcal{D}_1 \in \mathcal{E}_1$ and $\mathcal{D}_2 \in \mathcal{E}_2$ such that $\mathfrak{I}(\mathcal{D}_2) \subseteq \mathfrak{I}(\mathcal{D}_1)$. This definition is sound since $\mathfrak{I}(\mathcal{E}_2) = \mathfrak{I}(\mathcal{D}_2) \subseteq \mathfrak{I}(\mathcal{D}_1) = \mathfrak{I}(\mathcal{E}_1)$.

If the edge $\langle x, y \rangle$ is in a graph $\mathcal{D}$ then define the edge to be *covered* if $\{pa_{\mathcal{D}}(x) \setminus y\} = \{pa_{\mathcal{D}}(y) \setminus x\}$. Meek (1997) conjectured that a DAG $\mathcal{D}_1$ is a submodel of DAG $\mathcal{D}_2$ if and only if there exists a finite sequence of covered edge additions and reversals that transform $\mathcal{D}_1$ to $\mathcal{D}_2$. Chickering (2002b) proved this conjecture, stated more formally in the following theorem.

Theorem 2 (Chickering, 2002b). *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two different DAGs such that $\mathcal{D}_1$ is a submodel of $\mathcal{D}_2$. Let r be the number of edges in $\mathcal{D}_2$ that are in opposite orientation in $\mathcal{D}_1$, and m be the number of edges in $\mathcal{D}_2$ that do not exist in $\mathcal{D}_1$. Then there exists a sequence of at most $r + 2m$ edge reversals and edge additions in $\mathcal{D}_1$ such that:*

1. *Each edge reversed is a covered edge.*
2. *After each edge reversal/addition, $\mathcal{D}_1$ is still a DAG, and submodel of $\mathcal{D}_2$.*
3. *After all edge reversals/additions $\mathcal{D}_1 = \mathcal{D}_2$.*

Note that if covered edges are reversed in DAG $\mathcal{D}$ such that no unshielded collider is created nor destroyed, and the resulting graph is still a DAG, then the resulting graph is in the same equivalence class as $\mathcal{D}$. (Note that the set of "legal" edge reversals correspond to the set of undirected edges in the essential graph associated with $\mathcal{D}$). Further, adding any edge to $\mathcal{D}$ such that the resulting graph is still a DAG would produce a supermodel of $\mathcal{D}$, and removing any edge from $\mathcal{D}$ would produce a submodel of $\mathcal{D}$. Hence, Theorem 2 does in fact provide graphical transformations that prove Meek's Conjecture. Theorem 2 is useful in that it provides a basis for determining graphical criterion for the DAG submodel relation and we will make use of it in proving Lemma 3.2.

Kočka et al. (2001a) and Kočka et al. (2001b) also proved Meek's Conjecture in the special case that $\mathcal{D}_1$ and $\mathcal{D}_2$ differ by exactly one edge. Further, they provided the following necessary conditions for determining whether one DAG is a submodel of another in the general case; however, they were unable to prove sufficiency of these conditions.

Lemma 2.2 (Kočka et al., 2001b). *Suppose that $\mathcal{D}_1$ and $\mathcal{D}_2$ are DAGs over vertex set V such that $\mathfrak{I}(\mathcal{D}_2) \subseteq \mathfrak{I}(\mathcal{D}_1)$. Then the following conditions hold:*

(a) Every adjacency in $\mathcal{D}_1$ is an adjacency in $\mathcal{D}_2$.

(b) If triple $\langle x, y, z \rangle$ forms an unshielded collider in $\mathcal{D}_1$, then in $\mathcal{D}_2$ either x and z are adjacent, or $\langle x, y, z \rangle$ forms an unshielded collider.

(c) If triple $\langle x, y, z\rangle$ forms an unshielded collider in $\mathcal{D}_2$, then in $\mathcal{D}_1$ x is independent of z given $pa_{\mathcal{D}_2}(x) \cup pa_{\mathcal{D}_2}(z)$.

Condition (c) simply guarantees that nodes that are independent given S in the supermodel are also independent given S in the submodel. However, this condition is not local, which makes it difficult to check in general. Kočka et al. (2001b) also proved the following result, which will be used to prove Lemma 3.1. Define a *non-collider path* to be a path on which every triple is a non-collider.

Lemma 2.3. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two DAGs over vertex set V that satisfy conditions (a) and (c) of Lemma 2.2. Further, let $\boldsymbol{\mu}(a, b)$ be a non-collider path in $\mathcal{D}_1$. Then there is an order-preserving subsequence of the vertices along $\boldsymbol{\mu}(a, b)$ in $\mathcal{D}_2$ that form a non-collider path, say $\boldsymbol{\mu}^*$ ($\boldsymbol{\mu}^* = \boldsymbol{\mu}$ is possible).*

2.6 EH-Procedure

Edwards (1995) presented the EH-procedure for undirected graphs, which seeks the simplest undirected graph consistent with some given data. The EH-procedure makes extensive use of the coherence principle (Gabriel, 1969) which states that submodels of rejected models are also rejected and supermodels of accepted models are also accepted.

Briefly, the EH-procedure proceeds as follows and is outlined in Table 1. All possible undirected graph models are categorized as "undetermined". The *maximal set* is defined as the subset of undetermined models with maximum number of edges, and the *minimal set* is defined as the subset of undetermined models with minimum number of edges. The algorithm alternates between scoring and categorizing elements of the minimal and maximal sets. Each graph is evaluated as either "accepted" or "rejected", based on a χ^2-deviance test and a pre-specified α-level. Models consistent with the data (associated p-value $> \alpha$) are accepted; otherwise the models are rejected. The coherence principle is then applied to reject submodels of rejected models, and accept supermodels of accepted models. The algorithm stops when the undetermined set is empty, and the accepted models with fewest edges are returned.

Although the EH-procedure uses the χ^2-deviance test to evaluate test models, another score, such as the Bayesian Information Criterion (BIC), could be used instead. The BIC of a DAG model, given data $\mathbf{D}$ is computed as:

$$BIC(\mathcal{D}|\mathbf{D}) = \log p(\mathbf{D}|\hat{\theta}, \mathcal{D}) - (d/2)\log m$$

where $\hat{\theta}$ is the maximum likelihood estimate for the model parameters, d is the dimension of $\mathcal{D}$ (number free parameters), m is the number of observations in $\mathbf{D}$, and $p(\mathbf{D}|\hat{\theta}, \mathcal{D})$ is the likelihood of the data given $\hat{\theta}$ and the graph.

Haughton (1988) showed that the BIC is consistent for curved exponential models in the sense that as m grows, (i) models compatible with the distribution $p(\cdot)$ that generated the observed data get higher BIC scores than models not compatible with $p(\cdot)$, and (ii) among

Table 1: EH-Procedure

Input: **V** variable set, **D** data, α threshold
Output: $\mathcal{U}$, the simplest undirected graph consistent with **D**

Initialize: undetermined set = all possible undirected models over **V**

- while (undetermined set not empty) do {
  1. minimal set = graphs in undetermined set with fewest number edges
  2. maximal set = graphs in undetermined set with most number edges
  3. if (|minimal set| < |maximal set|) { test set = minimal set }
     else {test set = maximal set}
  4. for (all models in test set) {
     compute p-value for χ^2-deviance test of test model given data **D**
     if (p-value $\leq \alpha$) { reject model and all its submodels }
     else { accept model and all its supermodels }
     }
- $\mathcal{U}$ = accepted model with fewest edges
- return $\mathcal{U}$

models compatible with $p(\cdot)$, those with fewer parameters (i.e. simpler models) get higher scores. In general, the distributions associated with a particular DAG $\mathcal{D}$ may impose non-independence constraints (Chickering, 2002b) that are not encoded by $\mathcal{D}$. However, for the Gaussian and multinomial distributions this is not the case; hence, Markov equivalent DAGs which encode the same set of conditional independence statements, are *distributionally equivalent* (compatible with same set of distributions) as well. Consequently, in the absence of background knowledge, the likelihood of observing some given data is the same for all models within an equivalence class, as is the BIC score. Hence, to score an equivalence class, one need only score a single member of the class. In other words, BIC($\mathcal{E}(\mathcal{D})$) = BIC($\mathcal{D}$). Chickering (2002b) uses the BIC score for the GES algorithm, and provides a detailed discussion about scoring functions.

Note that for undirected graphs, each graph forms its own unique equivalence class. Hence, the EH-procedure can be thought of as an equivalence class search for undirected graphs.

3 DAG Equivalence Class Search

3.1 DAG submodel relations

Understanding the graphical criterion that specifies when one DAG is a submodel of another will facilitate applying the coherence principle in the DECS algorithm. We first present two

necessary conditions.

Lemma 3.1. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be DAGs over the variable set V such that $\mathcal{D}_1$ is a submodel of $\mathcal{D}_2$. Then the following conditions hold:*

(C1) The skeleton of $\mathcal{D}_1$ is a subgraph of the skeleton of $\mathcal{D}_2$, and

(C2) Every unshielded triple in $\mathcal{D}_1$ that is an unshielded triple in $\mathcal{D}_2$ is of the same type (collider/non-collider).

Proof. The proof is immediate by Lemma 2.2 since conditions (C1) and (C2) are equivalent to conditions (a) and (b) respectively. □

Unfortunately, conditions (C1) and (C2) are not sufficient in general. However, they are sufficient for DAGs containing no unshielded colliders.

Theorem 3. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be DAGs over the variable set V such that neither graph contains any unshielded colliders. Then $\mathcal{D}_1$ is a submodel of $\mathcal{D}_2$ if and only if condition (C1) is satisfied.*

Proof. ($\Rightarrow$) By Lemma 3.1 the result is immediate.
($\Leftarrow$) Let a and b be two non-adjacent nodes in V, and let $\boldsymbol{\pi}$ be a minimal path between a and b in $\mathcal{D}_1$. For a contradiction, suppose a and b are d-connected in $\mathcal{D}_1$ given S but d-separated in $\mathcal{D}_2$ given S. By Corollary 2.1 $\boldsymbol{\pi}$ is a non-collider path. By Lemma 2.3, there exists a non-collider path $\boldsymbol{\pi}^*$ in $\mathcal{D}_2$ between a and b formed by an order-preserving subsequence of the vertices along $\boldsymbol{\pi}$. Since $\boldsymbol{\pi}^*$ is a non-collider path, it is d-connecting given S, which is a contradiction. □

In the general case, in which either $\mathcal{D}_1$ or $\mathcal{D}_2$ may contain unshielded colliders, conditions (C1) and (C2) from Lemma 3.1, though necessary, are not sufficient. Consider the counter-examples shown in Figure 2, provided by (Chickering (2002b) (Figures 5, 7, and 10 in their paper, but reproduced here with nodes re-labelled).

Recall that by Theorem 2 there always exists a sequence of covered edge reversals and edge additions that can transform a DAG to one of its supermodels. Conversely, there is a sequence of covered edge reversals and edge removals that can transform a DAG to one of its submodels. Fact 1 will be needed to prove Lemma 3.2.

FACT 1: Let $\boldsymbol{\pi}$ be a compelled shielded non-collider in a DAG $\mathcal{D}$. Since every edge on $\boldsymbol{\pi}$ is compelled, any sequence of covered edge reversals and edge additions (removals) to $\boldsymbol{\pi}$, that transforms $\mathcal{D}$ to a supermodel (submodel), starts with an edge addition (removal).

Lemma 3.2 provides additional conditions needed in the presence of compelled shielded non-colliders, using insight provided by the legal edge transformations specified by Theorem 2.

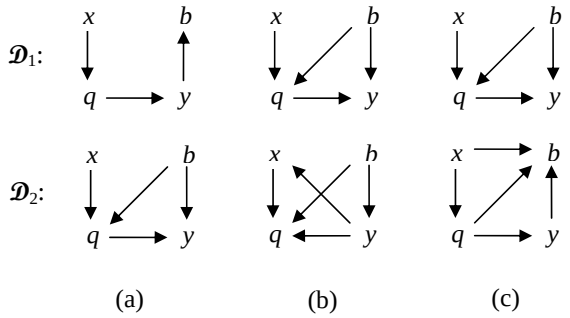

Figure 2: $\mathcal{D}_1$ and $\mathcal{D}_2$ satisfy conditions (C1) and (C2) of Theorem 3.1, but $\mathcal{D}_1$ is not a submodel of $\mathcal{D}_2$. $\langle x, q, b, y\rangle$ forms a compelled shielded non-collider in $\mathcal{D}_2$ of (a) and in $\mathcal{D}_1$ of (b) and (c).

Lemma 3.2. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be DAGs over vertex set V such that $\mathcal{D}_1$ is a submodel of $\mathcal{D}_2$. Then conditions (C1) and (C2) hold, as well as the following:*

(C3) If $\langle x, q, b, y\rangle$ forms a compelled shielded non-collider in $\mathcal{D}_1$, then in $\mathcal{D}_2$ either: $\langle x, y, b\rangle$ is a collider and $\langle x, b, y\rangle$ is a non-collider, or $\{x, q, b, y\}$ forms a clique.

(C4) If $\langle x, q, b, y\rangle$ forms a compelled shielded non-collider in $\mathcal{D}_2$, then in $\mathcal{D}_1$ either: x is not adjacent to q, or every triple is of the same type (collider/non-collider) as in $\mathcal{D}_2$.

Condition (C3) guarantees that if $\{q, b\}$ is in every set S that d-separates x and y in $\mathcal{D}_1$, then $\{q, b\}$ is in every set S that d-separates x and y in $\mathcal{D}_2$. Define the *induced subgraph* over subset A of a DAG $\mathcal{D}$ as the graph formed by the vertices in A and all edges among the variables in A that are in $\mathcal{D}$.

Proof. By Lemma 3.1 (C1) and (C2) hold. By Theorem 2 there is a sequence of edge reversals and edge additions (removals) that can transform $\mathcal{D}_1$ ($\mathcal{D}_2$) to $\mathcal{D}_2$ ($\mathcal{D}_1$). Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be the induced subgraphs over $\{x, q, b, y\}$ in $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively.

We now prove that (C3) holds. Suppose $\mathcal{T}_1$ forms a compelled shielded non-collider. Let $\mathcal{T}^*$ be the induced subgraph over $\{x, q, b, y\}$ in the most recent transformed version of $\mathcal{D}_1$. By Fact 1, the first move to transform $\mathcal{D}_1$ to $\mathcal{D}_2$ that involves edges that are in $\mathcal{T}_1$ is an edge addition. Note that there are only two missing edges in $\mathcal{T}_1$: $\langle x, y\rangle$ and $\langle x, b\rangle$. $x \leftarrow y$ is not a legal edge addition because it creates the cycle $x \rightarrow q \rightarrow y \rightarrow x$.

If $x \rightarrow y$ is added, then by Theorem 2 the only legal edge reversal in $\mathcal{T}^*$ is on edge $\langle q, y\rangle$; hence, $\langle x, y, b\rangle$ is a collider. If subsequently the $\langle x, b\rangle$ edge is added, then $\{x, q, b, y\}$ forms a clique in $\mathcal{D}_2$. If instead edge $\langle x, b\rangle$ is added first, then $\langle x, b, y\rangle$ is an unshielded non-collider (recall that $b \rightarrow y$ is in $\mathcal{D}_1$). By Theorem 2, no edge reversal in $\mathcal{T}^*$ can change $\langle x, b, y\rangle$ to

an unshielded collider. If a subsequent edge addition is $\langle x, y \rangle$, then clearly $\{x, q, b, y\}$ forms a clique in $\mathcal{D}_2$. Hence (C3) holds.

We now prove that (C4) holds. Suppose $\mathcal{T}_2$ forms a compelled shielded non-collider. Let $\mathcal{T}^*$ be the induced subgraph over $\{x, q, b, y\}$ in the most recent transformed version of $\mathcal{D}_2$. By Fact 1, the first move to transform $\mathcal{D}_2$ to $\mathcal{D}_1$ that involves edges that are in $\mathcal{T}_2$ is an edge removal. If edge $x \rightarrow q$ is removed, then x is a singleton in $\mathcal{T}^*$, and thus in $\mathcal{T}_1$ as well. Note that $\{q, b, y\}$ forms a clique in $\mathcal{T}_2$. If instead any edge in $q \leftarrow b \rightarrow y \leftarrow q$ is removed first, then every triple in $\mathcal{T}^*$ is unshielded and of the same type as in $\mathcal{T}_2$. Hence, by (C2), each of these triples that exist in $\mathcal{T}_1$ are of the same type as in $\mathcal{T}_2$, and (C4) holds. □

Table 2: DAG Equivalence Class Search (DECS)

Input: **V** variable set, **D** data, $\mathbf{BIC}_0$ score threshold
Output: $\mathcal{E}$, equivalence class with largest BIC score consistent with **D**

Initialize: test set = $(n-1)$-edge models, stop = 0

- while (test set not null) do {
  for (every model in test set) {
  1. score model
     if ($\text{BIC}_{model} < \text{BIC}_0$) { reject model and all its submodels }
     else { accept model and all its supermodels }}
  2. A = sum of adjacency matrices of all accepted models in test set
     R = sum of adjacency matrices of all rejected models in test set
  3. if (test set is maximal set) {
     if (A = 0) { break } else {
     test set = submodels of accepted models with fewest edges that are not submodels of rejected models and have not been evaluated }
     } else if (test set = minimal set) {
     if (R = 0) { break } else {
     test set = supermodels of rejected models with the most edges that are not submodels of rejected models and have not been evaluated }
     }
  }
- $\mathcal{E}$ = accepted model with highest BIC score
- return $\mathcal{E}$

3.2 DECS algorithm

The DAG equivalence class search is an extension of the EH-procedure and is outlined in Table 2. Briefly, the procedure takes a set of data and a BIC-threshold as input. A sequence of equivalence classes is scored via BIC and deemed as either "accepted" or "rejected". The coherence principle is applied so that submodels of rejected models are also rejected, and supermodels of accepted models are also accepted.

The DECS algorithm begins by scoring the $(n-1)$-edge DAG equivalence classes (i.e. the maximal set). To determine the new test set when the test set is a maximal set, information from the set of accepted models is collated into matrix A, which is the sum of all adjacency matrices of the accepted models in the test set. Then the largest sum of the cross-diagonals of A, $a_{ij} + a_{ji}$ for $i, j = 1, \ldots, N$, identifies which edges are most common across the set of accepted models. If the sum of a pair of cross-diagonals equals the number of accepted models in the test set, then the $\langle i, j \rangle$ edge is common to all the accepted models of the test set. However, the next set of equivalence classes to be tested must consist of either: (a) submodels of accepted models that are not submodels of rejected models; or (b) supermodels of rejected models that are neither supermodels of accepted models nor submodels of rejected models; or a combination thereof.

The new test set is the minimal set. If none of the DAG equivalence classes in the minimal set are accepted, then the new test set is determined based on the R matrix, which is the sum of all adjacency matrices of the rejected models in the test set. Note that in all examples we have looked at, the algorithm terminated before this step was required. As with the EH-procedure, the DECS procedure terminates when all classes have been evaluated, i.e. when a minimal consistent model is found.

4 Real Data Examples

The DECS algorithm was programmed in Matlab, using functions from the Bayes Net Toolbox for Matlab (Murphy, 2007), and tested on the data sets described below.

4.1 Description of Data Sets

The following data sets were provided by The Data and Story Library on Statlib at the Department of Statistics, Carnegie Mellon University (http:\\lib.stat.cmu. edu\DASL).

1. Intensive Care Unit Data Set (ICU)
   This data set is comprised of 200 subjects from a larger study on patient survival after admission to an adult intensive care unit. Four of the twenty variables were used in this example.

   - S: vital status (0 = lived, 1 = died)
   - G: patient's gender (0 = male, 1 = female)

- C: was cancer part of the problem (0 = no, 1 = yes)
- T: type of admission (0 = elective, 1 = emergency)

2. Popular Kids Data Set (PK)
   The popular kids data contains 478 observations from students in grades 4-6 from three different school districts in Ingham and Clinton Counties, Michigan. Students were asked what their goals were and to rank the importance of certain factors on popularity. Five of the eleven variables were used.

   - S: gender (0 = male, 1 = female)
   - Gr: grade (1 = grade 4, 2 = grade 5, 3 = grade 6)
   - R: race (0 = white, 1 = other)
   - U: urban/rural status of school (1 = Rural, 2 = Suburban, 3 = Urban)
   - G: goals (1 = good grades, 2 = popularity, 3 = good in sports)

3. Montana Outlook Poll Data Set (MOP)
   A random sample of Montana residents were asked about their financial status as compared to the previous year, and whether they thought the state economic outlook was better over the next year. Of the 209 cases, only cases with no missing responses were used for a total of 135 observations. Five of the seven variables were used in the model selection.

   - A: age (1 = <35, 2 = 36-54, 3 = >55)
   - G: gender (0 = male, 1 = female)
   - I: yearly income (1= <20k, 2 = 20-35k, 3 = >35k)
   - F: financial status (1 = worse, 2 = same, 3 = better)
   - S: state economic outlook (0 = better, 1 = not better)

In the next section we provide a detailed description of the results for the ICU data set, and summarize the results for the PK and MOP data sets.

4.2 Results

For the ICU data, $N = 4$, $n = 6$ and the total number of 5-edge equivalence classes was 24. The BIC-threshold was set at -425.057, which was the BIC of the saturated model. Hence, every equivalence class with a score larger than -425.057 was deemed accepted; otherwise the class was rejected. Figure 3 shows the structure of all 5-edge equivalence classes, categorized by accepted and rejected models.

To determine the next test set, the adjacency matrices of all 16 accepted 5-edge models were added together, giving:

$$A = \begin{bmatrix} 0 & 4 & 6 & 8 \\ 7 & 0 & 5 & 8 \\ 5 & 7 & 0 & 8 \\ 8 & 4 & 8 & 0 \end{bmatrix}$$

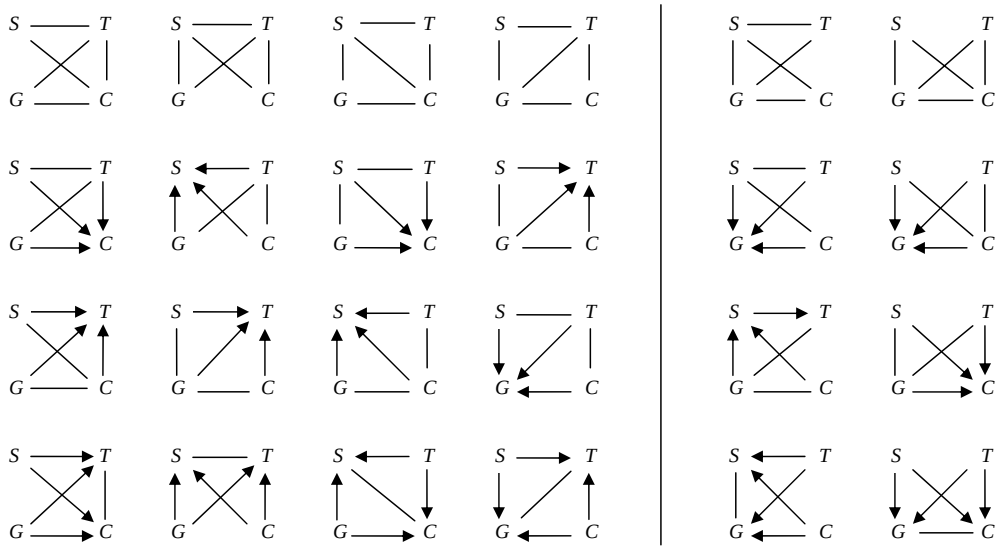

Figure 3: All 5-edge equivalence classes fit for the ICU data. (a) The set of *accepted* models. Every graph in the 16 accepted equivalence classes has edges $\langle S, T\rangle$ and $\langle C, T\rangle$, though not always in the same orientation. (b) The set of *rejected* models.

Each column/row represents variables S, G, C and T, respectively. From A above, there were two common edges: $\langle S, T\rangle$ and $\langle C, T\rangle$, but no common directed edges. Hence, the next test set contained graphs with edges $\langle S, T\rangle$ and $\langle C, T\rangle$, that were not subgraphs of any rejected model, which gave rise to two 2-edge graphs: one in which $\langle S, T, C\rangle$ forms an unshielded non-collider, and one in which $\langle S, T, C\rangle$ forms an unshielded collider, as shown in Figure 4.

Both 2-edge models were accepted, though the model with an unshielded non-collider (BIC = -410.944) had a larger BIC score than the model with an unshielded collider (BIC = -411.624). Hence, the equivalence class most consistent with the data was the 2-edge model shown in Figure 4(a). It is interesting to note that gender was not adjacent to any nodes, which suggests that gender was not associated with patient survival. Since all DAGs within an equivalence class have the same BIC score, background knowledge is needed to identify which DAGs within the output equivalence class are most plausible.

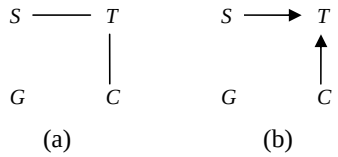

Figure 4: Minimal models determined after collating all accepted 5-edge models. (a) $\langle S, T, C\rangle$ is an unshielded non-collider. (b) $\langle S, T, C\rangle$ is an unshielded collider.

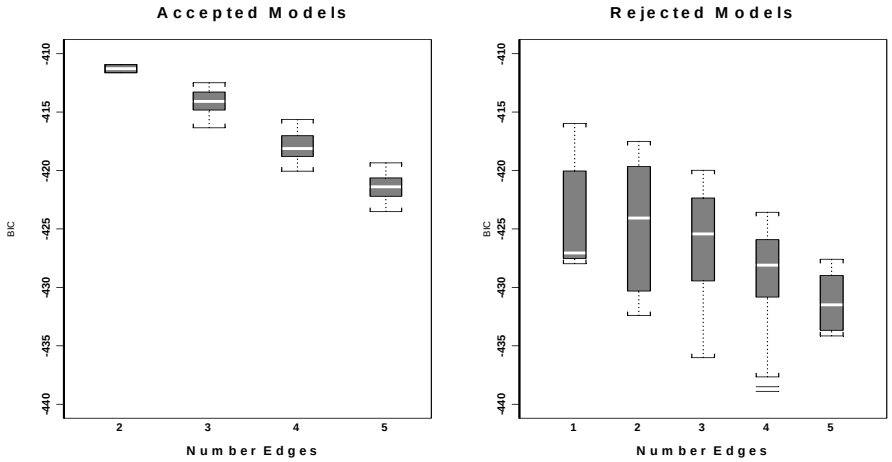

Figure 5: Boxplots of BIC scores of 185 equivalence classes for ICU data. (a) Accepted models (b) Rejected models.

We fit and scored all 185 possible equivalence classes for the ICU data set to examine how the BIC for the final model from the DECS algorithm compared to that of the other models. Reassuringly, the 2-edge non-collider model (see Figure 4(a)) had the highest BIC score among all 185 classes and hence, is a global maximum. Figure 5 shows boxplots of the BIC scores for all 185 models, categorized by the number of edges in the graph, and by whether the model was accepted or rejected. Figure 5(a) shows that, in general, as the number of edges in the equivalence class increases, the BIC score decreases, which validates the finding that among the models compatible with data, BIC gives higher scores to simpler models (Haughton, 1988).

Table 3 and Figure 6 summarize the results from all three data sets. The final model for the Popular Kids data set had an edge between school grade and urbanity of the school and an edge between gender and children's goals. Note that race is not adjacent to the other variables in the data set, which suggests that race and goals are independent of one another. Instead of fitting all 8,782 equivalence classes over the PK variables, equivalence classes in the neighbourhood of the final model were scored. Locally, the 2-edge equivalence

Table 3: Summary of Results. BIC_0 is the BIC threshold.

				BIC			
Data Set	m	N	n	BIC_0	0-edge model	n-edge model	Final model
ICU	200	4	6	-425	-425	-425	-411
PK	478	5	10	-2237	-2011	-2237	-2007
MOP	135	5	10	-990	-773	-990	-773

class output by DECS had the largest BIC score.

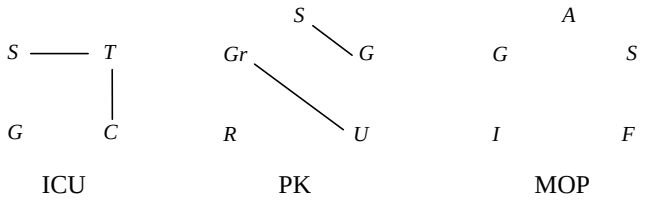

Figure 6: The equivalence classes with highest BIC score for each data set analyzed.

The final model output for the Montana Outlook Poll data contained no edges. Analogous to the PK data set, a local search around the final model showed that the null model output by DECS was also a local maximum.

5 Discussion

The final model for the ICU data set was a global maximum, and for the PK and MOP data sets the final models were local maxima. These observations are encouraging and show that the DECS algorithm can find the global maximum or at least a local maximum, if it exists, but a rigorous proof is not yet available.

For all three data sets, after scoring the $n-1$ edge models, the algorithm terminated quickly and only two test sets were scored. Collating the information from the accepted models in the A matrix facilitated finding an efficient path through the search space of equivalence classes. The computational burden of the DECS algorithm lies in (i) generating the equivalence classes for each test set, and (ii) applying the coherence principle after scoring the test set models. Hence, identifying submodels and supermodels of a given DAG quickly would make the algorithm more efficient.

Lemma 3.2 provides elegant necessary conditions for the submodel relation which facilitate verification that models in the test set are neither submodels of rejected models,

nor supermodels of accepted models. In other words, this lemma helps one to apply the coherence principle. Conditions (C1) and (C2) are basic ones that are essentially equivalent to conditions given by Kočka et al. (2001b) and Chickering (2002b). However, sufficiency of (C1) and (C2) only holds in a special case. Kočka et al. (2001b) provide a detailed discussion of why these conditions are not sufficient in general, and propose condition (c) of Lemma 2.2 to resolve the problem. Unfortunately, condition (c) is not a local one. We present conditions (C3) and (C4), which highlight the significance of compelled shielded non-colliders, as additional necessary conditions. They were mainly motivated by the counter-examples provided by Kočka et al. (2001b), and are local conditions that can be checked efficiently.

We conjecture that (C3) and (C4) hold the key to finding general graphical criterion for the DAG submodel relation. We further conjecture that (C1) to (C4) are necessary and sufficient graphical conditions for the DAG submodel relation when $N \leq 4$. Since the DECS algorithm is across essential graphs, finding necessary and sufficient graphical conditions for model inclusion in terms of essential graphs would be ideal. We conjecture that (C1) to (C4) are also necessary and sufficient for the essential graph submodel relation when $N \leq 4$.

Since the saturated model imposes no restrictions on the joint distribution of the variables in a graph, it cannot be rejected. This observation explains why our default BIC-threshold was the BIC score of the saturated model. However, a general approach to choosing an appropriate BIC-threshold is still an open problem. One idea, motivated by Figure 5, is to decrease the threshold at each step, according to the number of edges in the equivalence classes in the test set.

Just as with the GES algorithm, the BIC is not the only score that could be used to evaluate the equivalence classes. Other score functions that are the same for all models within an equivalence class may be more appropriate than the BIC. However, unlike the GES algorithm, DECS searches a broader set of equivalence classes than GES, and one can easily request that the best k models be output, particularly when one is using the algorithm for hypothesis-generating purposes.

Other model searches across DAG equivalence classes include the MC^3 method (Madigan et al., 1996), which uses Bayesian model averaging; and the SIN procedure (Drton and Perlman, 2007; Drton and Perlman, 2008), which uses tests for zero-partial correlations between variables. It would be interesting to see how these procedures compare to the DECS algorithm in a systematic simulation study. However, note that both of these methods require background knowledge in the form of either prior distributions on model parameters (MC^3), or a total ordering on the variables in the DAG being recovered (SIN). The DECS algorithm, on the other hand, requires no additional background knowledge, though a priori knowledge of variable ordering or of edge orientations can easily be incorporated into the model search.

Note that the ratio of the number of observations to the number of variables ($m : N$) is lowest for the MOP data. When applying the DECS algorithm to the ICU and PK data sets, the accepted equivalence classes with $n-1$ edges provided a nice partition of the search space and the algorithm quickly found the final models. With the MOP data set, the A

matrix, which highlighted common edges in the set of accepted models did not indicate such a nice partition of the search space. This phenomenon may be due to the lack of data (only 135 observations for 5 nodes in the MOP data compared to 478 observations for 5 nodes in the PK data). The authors are in the process of performing simulations to study what ratio of $(m : N)$ is suitable to reach reliable conclusions, as well as other asymptotic properties of the algorithm.

6 Acknowledgements

We would like to thank an anonymous referee whose helpful comments and suggestions improved the presentation of this paper. This research was supported by the Natural Sciences and Engineering Research Council of Canada grant RG 326951-06.

References

[1] Ali, R.A. and Richardson, T. (2002). Markov equivalence classes for maximal ancestral graphs. In A. Darwiche and N. Friedman (Eds.), *UAI–18*, San Francisco, pp. 1-8. Morgan Kaufmann.

[2] Ali, R.A., Richardson, T., and Spirtes, P. (2008). Markov equivalence for ancestral graphs. *To appear in Annals of Statistics*.

[3] Andersson, S.A., Madigan, D., and Perlman, M.D. (1997). A characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics*, **25**, 505-541.

[4] Chickering, D. (2002a). Learning equivalence classes of Bayesian network structures. *Journal of Machine Learning Research*, **2**, 445-498.

[5] Chickering, D. (2002b). Optimal structure identification with greedy search. *Journal of Machine Learning Research*, **3**, 507-554.

[6] Drton, M. and Perlman, M.D. (2007). Multiple testing and error control in Gaussian graphical model selection. *Statistical Science*, **22**, 430-449.

[7] Drton, M. and Perlman, M.D. (2008). A sinful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference* **138**, 1179-1200.

[8] Edwards, D.M. (1995). *Introduction to Graphical Modelling*. New York: Springer-Verlag.

[9] Gabriel, K. (1969). Simultaneous test procedures-some theory of multiple comparisons. *The Annals of Mathematical Statistics*, **40**, 224-250.

[10] Gillispie, S. and Perlman, M.D. (2001). Enumerating Markov equivalence classes of acyclic digraph models. In J. Breese and D. Koller (Eds.), *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pp 171-177. San Francisco: Morgan Kaufmann.

[11] Haughton, D. (1988). On the choice of a model to fit the data from an exponential family. *Annals of Statistics*, **16**, 342-355.

[12] Kočka, T., Bouckaert, R.R., and Studený, M. (2001a). On characterizing inclusion of bayesian networks. In UAI 01: *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, San Francisco, CA, USA, pp. 261-268. Morgan Kaufmann Publishers Inc.

[13] Kočka, T., Bouckaert, R.R., and Studený, M. (2001b). On the inclusion problem. Technical Report 2010, Academy of Sciences of the Czech Republic, Institute of Information Theory and Automation.

[14] Madigan, D., Andersson, S.A, Perlman, M.D., and Volinsky, C.M. (1996). Bayesian model averaging and model selection for Markov equivalence classes of acyclic digraphs. *Communications in Statistics - Theory and Methods*, **25**, 2493-2519.

[15] Meek, C. (1995). Causal inference and causal explanation with background knowledge. In P. Besnard and S. Hanks (Eds.), *Uncertainty in Artificial Intelligence: Proceedings of the 11th Conference*, San Francisco, pp. 403-410. Morgan Kaufmann.

[16] Meek, C. (1997). *Graphical Models: Selecting causal and statistical models*. Ph. D. thesis, Carnegie Mellon University.

[17] Murphy, K.P. (19 October 2007). Bayes net toolbox for Matlab. Kevin Patrick Murphy. University of British Columbia, Department of Computer Science. 15 June 2008. http:\\ www.cs.ubc.ca\murphyk\Software\BNT\bnt.html.

[18] Spirtes, P., Glymour, C., and Scheines, R. (1993). *Causation, Prediction and Search*. Number 81 in Lecture Notes in Statistics. Springer-Verlag.

[19] Spirtes, P. and Richardson, T.S. (1997). A polynomial-time algorithm or determining DAG equivalence in the presence of latent variables and selection bias. In D. Madigan and P. Smyth (Eds.), *Preliminary papers of the Sixth International Workshop on AI and Statistics, January 4-7, Fort Lauderdale, Florida*, pp. 489-501.

[20] Steinsky, B. (2003). Enumeration of labelled chain graphs and labelled essential directed acyclic graphs. *Discrete Mathematics*, **270**, 267-278.

[21] Verma, T. and Pearl, J. (1991). Equivalence and synthesis of causal models. Technical Report R-150, Cognitive Systems Laboratory, UCLA.