Towards an inclusion driven learning of Bayesian Networks

Robert Castelo
Institute of Information and Computing Sciences
University of Utrecht
P.O. Box 80089, 3508TB Utrecht
The Netherlands
`roberto@cs.uu.nl`

Tomáš Kočka
Decision Support Systems Group
Department of Computer Science
Aalborg University
Fredrik Bajers Vej 7E, DK-9220 Aalborg
Denmark
`kocka@cs.auc.dk`

Keywords: Bayesian Networks; graphical Markov model inclusion; structural learning.

Abstract

Two or more Bayesian Networks are Markov equivalent when their corresponding acyclic digraphs encode the same set of conditional independence ($\equiv$ CI) restrictions. Therefore, the search space of Bayesian Networks may be organized in classes of equivalence, where each of them consists of a particular set of CI restrictions. The collection of sets of CI restrictions obeys a partial order, the graphical Markov model inclusion partial order, or *inclusion order* for short.

This paper discusses in depth the role that inclusion order plays in learning the structure of Bayesian networks. We prove that under very special conditions the traditional hill-climber always recovers the right structure.

Moreover, we extend the recent experimental results presented in (Kočka and Castelo, 2001). We show how learning algorithms for Bayesian Networks, that take the inclusion order into account, perform better than those that do not, and we introduce two new ones in the context of heuristic search and the MCMC method.

1 Introduction

A probability distribution P is Markov over a graph G if and only if every CI restriction encoded in G is satisfied by P. A graphical Markov model ($\equiv$ GMM) $\mathbf{M}(G)$ is the family of probability distributions that are Markov over G. One also says that G determines the GMM $\mathbf{M}(G)$. Bayesian Networks form a class of GMMs where the graph G that determines the model is an acyclic digraph ($\equiv$ DAG). More concretely, a Bayesian Network is a GMM $\mathbf{D}(G)$ whose set of probability distributions satisfies the directed global Markov property ($\equiv$ DGMP) relative to G. The DGMP is the sharpest possible graphical criterion that allows to read off CI restrictions from a given DAG (Pearl and Verma, 1987; Lauritzen et al., 1990).

Let $G = (V, E)$ be a DAG whose vertex set V indexes a set of categorical random variables $\mathbf{X}_V$. Let $p(\mathbf{X}_V)$ be a joint probability distribution over $\mathbf{X}_V$. Lauritzen et al. (1990) show that by using the conditional probability distributions of every X_i given $\mathbf{X}_{pa(i)}$ (the parent set of vertex $i \in V$ in G), $p(\mathbf{X}_V)$ admits a *recursive factorization* as follows:

$$p(\mathbf{X}_V) = \prod_{i \in V} p(X_i | \mathbf{X}_{pa(i)}). \tag{1}$$

The fact that the vertices $\{i\} \cup pa(i)$ form a complete subgraph in the moralized version of G, i.e. in G^m, implies this factorization and the more interesting fact that $p(\mathbf{X}_V)$ obeys the global directed Markov property relative to G^m (Lauritzen et al., 1990).

The recursive factorization in (1) allows to obtain a closed formula for the likelihood of the data D given a Bayesian Network $M \equiv \mathbf{D}(G)$, $p(D|M)$, under a certain set of assumptions about D (Cooper and Herskovits, 1992; Buntine, 1991; Heckerman et al., 1995). The logarithm of the likelihood $\log p(D|M)$ is often used as a scoring metric for Bayesian Networks. One uses this score metric to efficiently rank many Bayesian Networks in a short time, allowing to devise learning algorithms that work in a systematic way. Throughout this paper we have used the BDeu score metric, which corresponds to the BDe metric from Heckerman et al. (1995) with uniform hyper-dirichlet priors (Buntine, 1991). We have also considered a uniform prior distribution $p(M)$ over the space of Bayesian Networks.
A learning algorithm for Bayesian Networks usually consists of three components:

1. A *scoring metric* that in the light of data ranks two or more alternative Bayesian Networks.

2. A *traversal operator* that by means of local transformations of the Bayesian network structure creates a set of *neighbors*.

3. A *search strategy* that repeatedly applies the traversal operator to a particular subset of Bayesian Networks. The members of this particular subset are considered in relation to the ranking that the scoring metric provides, and some policy that the search strategy follows.

The search space of Bayesian Networks corresponds, in principle, to the space of DAGs, which grows exponentially in the number of vertices. An equivalence class of Bayesian Networks comprises one or more DAGs that, as we already mentioned, encode the same set of CI restrictions. Each equivalence class has a canonical representation in form of an acyclic partially directed graph, where the edges may be directed and undirected and satisfy some characterizing conditions (Spirtes et al., 1993; Chickering, 1995; Andersson et al., 1997). This representation has been introduced independently by several authors under different names as, *completed PDAG* (Chickering, 1995), *pattern* (Spirtes et al., 1993) and *essential graph* (Andersson et al., 1997). We will adopt here the term essential graph. An essential graph equivalent to a DAG has the same underlying graph and each edge in the essential graph is directed iff this edge has the same orientation in all equivalent DAGs.

From a non-causal perspective one is interested in learning equivalence classes of Bayesian Networks from data, and for that purpose one uses a scoring metric that gives equal scores to equivalent networks. In such a situation it makes sense to use the space of essential graphs ($\equiv$ EG-space) instead of the space of DAGs ($\equiv$ DAG-space). This argument is supported further by several advantages (Heckerman et al., 1995; Madigan et al., 1996):

1. The cardinality of EG-space is smaller than in DAG-space.
2. The prior laws for the parameters are no longer constrained to yield a score equivalent metric.
3. Quantities computed from Bayesian model averaging outputs cannot be biased by the size of the equivalence classes of Bayesian Networks.

Algorithms using the EG-space have been already developed (Spirtes and Meek, 1995; Chickering, 1996; Madigan et al., 1996; Chickering, 2001) and some of them are able to learn much better Bayesian Networks than the standard algorithms on DAG space. However, the computational overhead, with respect to DAG space, is far from being satisfactory for the first three of them. This extra computational cost is produced mainly by either the need to switch between EG-space and DAG-space to compute the score (Spirtes and Meek, 1995; Chickering, 1996) or the need to consider transformations on two adjacencies at a time (Madigan et al., 1996) to fulfill irreducibility of the Markov chain in Markov Chain Monte Carlo ($\equiv$ MCMC) procedures.

In this paper we argue that the better results obtained by the use of EG-space are not merely a consequence of using essential graphs only but by the side-effect that essential graphs respect better the inclusion order. Therefore, a sensible approach is to try to devise a way to use DAG-space, which is computationally cheaper, and account for the inclusion order.

This paper provides three major contributions. First, a thorough study of the inclusion order and the reasons of its relevance to the learning problem. Second, the introduction of a new traversal operator that (partially) accounts for the inclusion order. Third, the implementation of the previous ideas within two of the main model-based learning paradigms in Bayesian Networks: heuristic learning and the MCMC method. This last contribution will be illustrated by a set of experiments that comprises and extends those already presented in (Kočka and Castelo, 2001).

In the next section we introduce the basic concepts regarding DAGs, their equivalence relation, their inclusion order relation and a discussion of the sizes of DAG-space and EG-space.

2 Preliminaries

2.1 Graphical Terminology

The terms and notation about graphs within the context of GMMs have been borrowed mainly from (Lauritzen, 1996). A directed graph G is a pair (V, E) where V is the set of vertices and E is the set of directed edges or arcs. Since in this paper we deal only with directed graphs we will often use the shorter form *edge* to denote *directed edge* or *arc*.

A sequence of vertices $a = v_0, v_1, \dots, v_n = b$ forms a *path* between vertices a and b if for every pair (v_i, v_{i+1}), there is either an arc $v_i \rightarrow v_{i+1}$ or $v_i \leftarrow v_{i+1}$. The path is *directed* if the arc is always $v_i \rightarrow v_{i+1}$. A directed cycle is a directed path, as the previous one, where $a = b$. An acyclic digraph, or DAG[1], is a directed graph where its set of arcs E does not induce directed cycles nor has loops (arcs where the two endpoints are the same vertex) nor has multiple edges.

Given a vertex v, the set of vertices reachable from v by directed paths is known as the set of *descendants* of v, and noted $de(v)$. Consequently, the set of *non-descendants* of v is defined as $nd(v) = V \backslash \{de(v) \cup \{v\}\}$. For a given vertex v, the set of vertices that can reach v by directed paths (including the vertex v itself) form the *ancestor set* of v, and noted $an(v)$. The set of vertices with arcs pointing to a common vertex v is the *parent set* of v, noted $pa(v)$.

[1]Sometimes also referred as ADG.

2.2 The directed global Markov property and the d-separation criterion

In the introduction we already mentioned that a Bayesian Network codes a set of CI restrictions through a particular graphical criterion, the directed global Markov property ($\equiv$ DGMP).

Definition 2.1. *Directed global Markov property (DGMP)*
Let $G = (V, E)$ be a DAG, a probability distribution P is said to satisfy the directed global Markov property *(DGMP) if, for any triple (A, B, S) of disjoint subsets of V, where A, B are non-empty, such that S separates A from B in the moralized version of the subgraph induced by the vertices in $An(A \cup B \cup S)$, i.e. in $G^m_{An(A\cup B\cup S)}$, P satisfies*

$$A \perp\!\!\!\perp B \,|\, S[P].$$

The DGMP means that two non-empty subsets of vertices A, B are conditionally independent given a third subset S if and only if S separates A and B in the moralized subgraph induced by the smallest ancestral set of $A \cup B \cup S$.

An alternative way of reading conditional independencies in a DAG is using the *d-separation* criterion of Pearl and Verma (1987), which we review now. A node v_i in a path $v_0, v_1, \ldots, v_n$ is *collider* if $v_{i-1} \rightarrow v_i \leftarrow v_{i+1}$. A node in the path which is not collider is *non-collider.* Given two vertices $u, v \in V$ and a subset $S \subseteq V$ where $u, v \notin S$, one says that a path between u and v is *active* with respect to S if

1. every non-collider in the path is not in S, and
2. every collider in the path is in S or has a descendant in S.

When a path between two vertices u, v *is not* active with respect to S, one says that the path is *blocked* by S. Given these notions of active and blocked path, the d-separation criterion is defined as follows.

Definition 2.2. *d-separation*
Let $G = (V, E)$ be a DAG. For any triple (A, B, S) of disjoint subsets of V, where A, B are non-empty, A and B are d-separated by S if every path between the vertices in A and B is blocked by S.

Lauritzen et al. (1990) prove that the *d-separation* criterion encodes in a DAG exactly the same CI restrictions as the DGMP.

2.3 Markov Equivalence

Let $\mathbf{D}(G)$ be a Bayesian Network that belongs to an equivalence class with two or more members. Then G will have at least one edge which can be reversed such that the resulting DAG remains in the same equivalence class. Chickering (1995) characterized such edges in the following way.

Definition 2.3. *Covered Edge (Chickering, 1995)*
Let $G = (V, E)$ be a DAG. An edge $a \to b \in E$ is covered *in G if $pa(b) = \{a\} \cup pa(a)$.*

This characterization allows to describe more formally the notion of Markov equivalence among Bayesian Networks.

Lemma 2.1. *Let $\mathbf{D}(G)$ and $\mathbf{D}(G')$ be two Bayesian Networks. The following three conditions are equivalent:*

1. *$\mathbf{D}(G) = \mathbf{D}(G')$ i.e. G and G' are equivalent.*
2. *G and G' have the same skeleton (underlying undirected graph) and contain the same immoralities.*
3. *There exists a sequence $L_1, \ldots, L_n$ of DAGs such that $L_1 = G$ and $L_n = G'$ and L_{i+1} is obtained from L_i by reversing a covered arc in L_i for $i = 1, \ldots, n-1$.*

The equivalence (1) $\Leftrightarrow$ (2) was proven in (Verma and Pearl, 1990), and in the more general framework of chain graphs in (Frydenberg, 1990). The equivalence (1) $\Leftrightarrow$ (3) was proven in (Chickering, 1995; Heckerman et al., 1995).

The relationship of Markov equivalence organizes the DAG-space into classes of equivalence which form what we called the EG-space. Each member of EG-space has a graphical representation as an essential graph. From the asymptotic number of DAGs given by Robinson (1973) it follows that DAG-space grows more than exponentially in the number of vertices. One may think that EG-space departs substantially from such rate but recent empirical investigation gives a quite different perspective.

Observation 2.1. *(Gillispie and Perlman, 2001)*
The average ratio of DAGs per equivalence class seems to converge to an asymptotic value smaller than 3.7. This was observed up to 10 vertices.

In Figure 1 we may see plotted the cardinalities of DAG-space and EG-space up to 10 vertices.

2.4 Concepts of Neighborhood

The transformations on a single adjacency of a DAG, that learning algorithms for Bayesian Networks perform, are usually addition, removal and reversal of the arc such that acyclicity in the directed graph is preserved. Using these transformations we may formalize the following concepts of neighborhood:

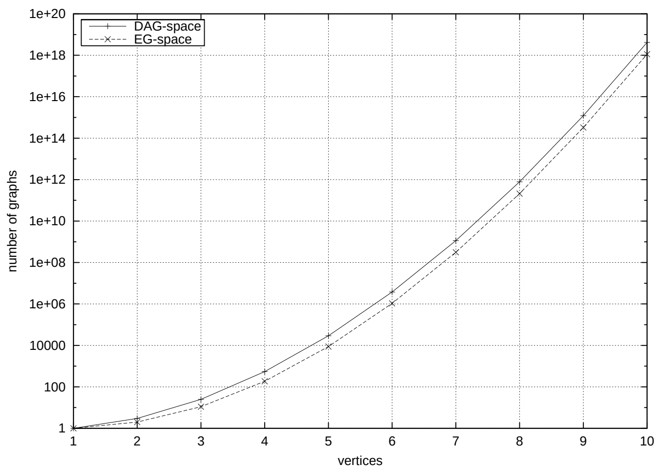


Figure 1: Cardinalities of DAG-space and EG-space.

- **NR** (No Reversals) All DAGs with one arc more or one arc less that do not introduce a directed cycle.
- **AR** (All Reversals) The NR neighborhood plus all DAGs with one arc reversed that does not introduce a directed cycle.
- **CR** (Covered Reversals) The NR neighborhood plus all DAGs with one covered arc reversed[2].
- **NCR** (Non-Covered Reversal) The NR neighborhood plus all DAGs with one non-covered arc reversed that does not introduce a directed cycle.

For any given DAG G upon which we build its corresponding neighborhood, the previously described neighborhoods will be noted as $\mathcal{N}_{\mathrm{NR}}(G)$, $\mathcal{N}_{\mathrm{AR}}(G)$, $\mathcal{N}_{\mathrm{CR}}(G)$ and $\mathcal{N}_{\mathrm{NCR}}(G)$, respectively. The NR neighborhood is used in MCMC search by the MC^3 algorithm from Madigan and York (1995). The NR neighborhood may lead easily to local maxima in heuristic search, or to an extremely small probability of reaching the most probable model given the data in MCMC search. This problem may be alleviated by using an AR neighborhood, which is quite common in many other learning algorithms. The CR and NCR neighborhoods are variations of the AR neighborhood that are not intended to enhance the AR neighborhood but are used here, as we shall see later, for comparison purposes.

[2]The reversal of a covered edge cannot introduce a directed cycle (Kočka et al., 2001).

3 Graphical Markov Model Inclusion

By *graphical Markov model inclusion* we denote a particular relation of partial order among GMMs. A relation of partial order is irreflexive, asymmetric and transitive, and some pairs of elements may not be related, otherwise it would be a *total order*. The intuition behind the inclusion order is that one GMM $\mathbf{M}(G)$ precedes another GMM $\mathbf{M}(G')$ if and only if all the CI restrictions encoded in G are also encoded in G'.

The complete DAG G_c that encodes no CI restriction at all, determines a Bayesian Network $\mathbf{D}(G_c)$ that consists of all possible discrete probability distributions over the corresponding set of random variables, due to the fact that any of such distributions is always Markov over the complete DAG G_c.

On the opposite side, we find the empty DAG $G_\emptyset$, with no edges, under which all random variables are marginally independent, such that it encodes all possible CI restrictions among these random variables under the closure of the semi-graphoid axioms (Pearl, 1988). The DAG $G_\emptyset$ determines a Bayesian Network $\mathbf{D}(G_\emptyset)$ that consists of "only" those discrete probability distributions under which all the random variables are marginally independent. Clearly, the set of probability distributions that are Markov over G_c includes those that are Markov over $G_\emptyset$, therefore

$$\mathbf{D}(G_\emptyset) \subseteq \mathbf{D}(G_c). \tag{2}$$

However, the CI restrictions encoded by G_c (none!) are included into those encoded by $G_\emptyset$ (all!), and this latter notion is the one that determines the inclusion order. The notation used in (2) might be somewhat counterintuitive with the idea that $\mathbf{D}(G_c)$ precedes $\mathbf{D}(G_\emptyset)$ under the inclusion order. Therefore, we will explicitly express the set of the CI restrictions encoded by a graph G (validity of each CI in the graph can be decided using d-separation criterion (Pearl, 1988) or moralization criterion (Lauritzen et al., 1990)) that determines the GMM $\mathbf{M}(G)$ as:

$$\mathbf{M}^I(G) = \{(A, C, B) \,:\, A, B \neq \emptyset \wedge A \perp\!\!\!\perp B | C[G]\}.$$

Now, in order to note the inclusion relationship between the fully restricted Bayesian Network $\mathbf{D}(G_\emptyset)$ and the unrestricted Bayesian Network $\mathbf{D}(G_c)$ we may simply write it as:

$$\mathbf{D}^I(G_c) \subseteq \mathbf{D}^I(G_\emptyset).$$

Unfortunately, to characterize the inclusion order between two arbitrary Bayesian network using some graphical or transformational rules (like the characterization of equivalence) is still an open problem[3]. The first attempt to provide necessary and sufficient conditions to characterize the inclusion

[3]To the best knowledge of the authors at the moment.

order among DAG Markov models was done in (Verma and Pearl, 1988), and in (Kočka, 2001) it is shown that these conditions are necessary but not sufficient. Later, Meek (1997) conjectured the following operational criterion to decide Bayesian Network inclusion.

Conjecture 3.1. *Meek's conjecture Meek (1997)*
Let $\mathbf{D}(G)$ and $\mathbf{D}(G')$ be two Bayesian Networks determined by two DAGs G and G'. The conditional independence model induced by $\mathbf{D}(G)$ is included in the one induced by $\mathbf{D}(G')$, i.e. $\mathbf{D}^I(G) \subseteq \mathbf{D}^I(G')$, if and only if there exists a sequence of DAGs $L_1, \ldots, L_n$ such that $G = L_1$, $G' = L_n$ and the DAG L_{i+1} is obtained from L_i by applying either the operation of covered arc reversal or the operation of arc removal for $i = 1, \ldots, n$.

Very recently, Chickering (2002) has proved Meek's conjecture, and prior to that work, Kočka et al. (2001) proved Meek's conjecture for the particular case in which G and G' differ in at most one adjacency, providing the following operational criterion.

Lemma 3.1. *Kočka et al. (2001)*
Let $\mathbf{D}(G)$ and $\mathbf{D}(G')$ be two Bayesian Networks determined by two DAGs G and G' such that their skeletons differ in at most one edge. $\mathbf{D}^I(G) \subset \mathbf{D}^I(G')$ iff there exists a sequence of DAGs $L_1, \ldots, L_n$ such that $G = L_1$, $G' = L_n$ where the graph L_{i+1} is obtained from L_i by applying the operation of covered arc reversal for $i = 1, \ldots, j-1$, the operation of arc removal for $i = j$ and the operation of covered arc reversal for $i = j+1, \ldots, n-1$ where $j \in \{1, \ldots, n-1\}$.

It is easy to realize that the collection of sets of CI restrictions $\mathbf{M}^I(G) \in \mathcal{M}$ where $\mathcal{M}$ is any given class of GMMs, forms a poset that we can represent by means of a Hasse diagram. In Figure 2 we may see this representation for the EG-space over three variables.

From the perspective of the search space that the Hasse diagram in Figure 2 provides, the concept of *inclusion boundary* follows. This concept applies to every type of GMM. As we shall see throughout the paper, this concept is the key to understanding the relevance of the inclusion order in the learning task.

Definition 3.1. *Inclusion Boundary (Kočka, 2001)*
Let $\mathbf{M}(H), \mathbf{M}(L)$ be two GMMs determined by the graphs H and L. Let $\mathbf{M}^I(H) \prec \mathbf{M}^I(L)$ denote $\mathbf{M}^I(H) \subset \mathbf{M}^I(L)$ and for no $\mathbf{M}^I(K)$, $\mathbf{M}^I(H) \subset \mathbf{M}^I(K) \subset \mathbf{M}^I(L)$.
The inclusion boundary *of the GMM $\mathbf{M}(G)$, noted $\mathcal{IB}(G)$, is*

$$\mathcal{IB}(G) = \{\mathbf{M}(H) \,:\, \mathbf{M}^I(H) \prec \mathbf{M}^I(G)\} \cup \{\mathbf{M}(L) \,:\, \mathbf{M}^I(G) \prec \mathbf{M}^I(L)\}$$

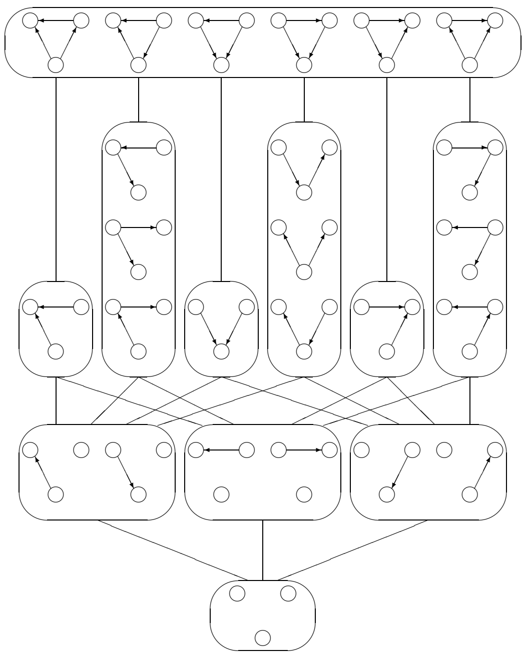


Figure 2: Hasse diagram of the EG-space for Bayesian Networks over three variables.

Intuitively, the inclusion boundary of a given GMM $\mathbf{M}(G)$ consists of those GMMs $\mathbf{M}(G_i)$ that induce a set of CI restrictions $\mathbf{M}^I(G_i)$ which *immediately* follow or precede $\mathbf{M}^I(G)$ under the inclusion order. In the Hasse diagram of Figure 2, the inclusion boundary for a given node is formed by those nodes adjacent to it.

We will say that a sequence of GMMs $\mathbf{M}(G_1), \ldots, \mathbf{M}(G_n)$ forms an *inclusion path*, or are *in inclusion*, if $\mathbf{M}^I(G_1) \supset \ldots \supset \mathbf{M}^I(G_n)$. We will say that inclusion path is a *longest inclusion path* between $\mathbf{M}(G_1)$ and $\mathbf{M}(G_n)$ if each GMM from the sequence $\mathbf{M}^I(G_i)$ has its neighbors in its inclusion boundary i.e. $\mathbf{M}^I(G_{i+1}), \mathbf{M}^I(G_{i-1}) \in \mathcal{IB}(G_i)$. Obviously for each pair of GMMs in inclusion $\mathbf{M}^I(G) \supset \mathbf{M}^I(H)$ there is at least one longest inclusion path which starts with $\mathbf{M}^I(G)$ and ends with $\mathbf{M}^I(H)$. The existence follows from the definition of inclusion boundary.

The following definition of *inclusion boundary condition* establishes a necessary condition that a traversal operator (used for local search) must satisfy in order to avoid local maxima. Later we will show that under some additional assumptions this condition is sufficient.

Definition 3.2. *Inclusion Boundary Condition (Kočka, 2001)*
A learning algorithm for GMMs satisfies the Inclusion Boundary Condition *if for every GMM determined by a graph G, the traversal operator creates*

neighborhood $\mathcal{N}(G)$ *such that* $\mathcal{N}(G) \supseteq \mathcal{IB}(G)$.

Note that some computational performance may be gained if the traversal operator creates neighborhood $\mathcal{N}(G) = \mathcal{IB}(G)$, since it is the smallest $\mathcal{N}(G)$ satisfying the inclusion boundary condition.

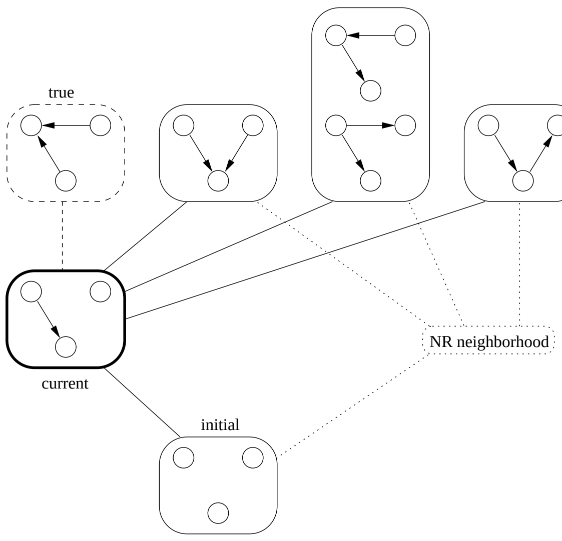


Figure 3: Example of getting stucked in a local maxima because the concept of neighborhood employed (NR) does not contain the inclusion boundary.

In Figure 3 we find a situation, in the context of DAG models, in which the model highlighted with a thick line is our current model. The model highlighted with a dashed line is the one that reflects the data (the *true* model).

The rest of the models are the NR neighborhood of the current one. All models around the current one form its inclusion boundary. As we may appreciate, because the current neighborhood does not entirely include the inclusion boundary, it is not possible to reach the *true* model from the current one in a single step. In this situation, it may become very difficult for the learning algorithm to reach the *true* model. Note that starting from the empty model it may easily happen that we select the current model, provided that a score equivalent metric would score equally in either direction the addition of a single isolated edge.

From the previous example, it is clear that the concepts of neighborhoods introduced for Bayesian Networks so far (NR, AR, CR and NCR) do not retain the property $\mathcal{N}(G) \supseteq \mathcal{IB}(G)$. In Figure 4 we may find this problem more clearly illustrated. In the center we have a circle that represents an equivalence class of DAG models with 16 members determined by the graphs G_0 to G_{15}. Surrounding this circle we have other equivalence classes, where those that are shaded represent its inclusion boundary. Given any of the

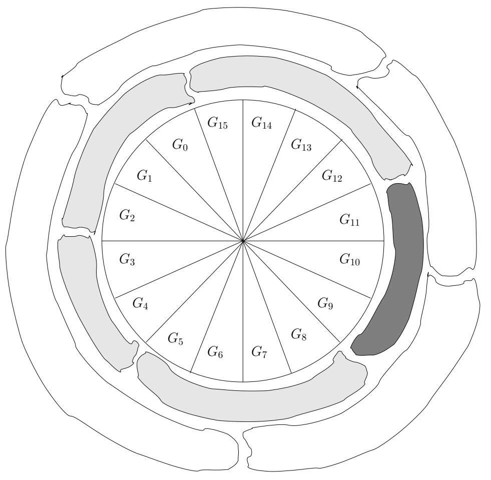


Figure 4: Every member $G_0, \ldots, G_{15}$ of an equivalence class cannot reach the entire inclusion boundary (shaded classes) by a transformation of a single adjacency.

sketched classes, its immediate neighbors are reachable by a transformation in a single adjacency of the graph.

Consider the equivalence class in the inclusion boundary that is darker shaded. As it is clear from the figure, only G_9, G_{10} and G_{11} can reach this neighboring equivalence class by a transformation in a single adjacency. Therefore, if our learning algorithm does not consider any of G_9, G_{10} or G_{11}, it will not be able to reach that neighbor class in a single move.

We are going to define a neighborhood that that coincides with the inclusion boundary. Later, by using the partial characterization in Lemma 3.1 we will provide an efficient implementation of this neighborhood.

Let $\mathbf{D}(G)$ be any given DAG Markov model that belongs to some equivalence class $\mathcal{C} = \{\mathbf{D}(G_1), \ldots, \mathbf{D}(G_n)\}$, i.e. $\mathbf{D}(G_1) = \mathbf{D}(G_2) = \ldots = \mathbf{D}(G_n)$. Consider the following two neighborhoods for $\mathbf{D}(G)$:

- **ENR** (Equivalence class No Reversals) All DAGs with one arc more or one arc less than any DAG G_i where $\mathbf{D}(G_i) \in \mathcal{C}$.
- **ENCR** (Equivalence class Non-Covered Reversals) All DAGs with one arc more or one arc less or one non-covered arc reversed than any DAG G_i where $\mathbf{D}(G_i) \in \mathcal{C}$.

Now, we will investigate the relationships among the ENR and ENCR neighborhoods, and the other neighborhoods previously defined. We begin by recalling the directed pairwise Markov property (Lauritzen et al., 1990; Lauritzen, 1996):

Definition 3.3. *Directed pairwise Markov property (DPMP)*
Let $G = (V, E)$ be a DAG. A probability distribution P is said to satisfy the directed pairwise Markov property *(DPMP) if, for any pair $u, v \in V$ of non-adjacent vertices such that $v \in nd(u)$, P satisfies*

$$u \perp\!\!\!\perp v \,|\, nd(u) \backslash \{v\} [P].$$

Lauritzen et al. (1990) prove that all CI restrictions implied by the DPMP are implied by the DGMP and the d-separation criterion as well.

The following two lemmas provide insight into the relationship between the inclusion order and the graphical structure of Bayesian Networks.

Lemma 3.2.
Let $\mathbf{D}(G)$ and $\mathbf{D}(G')$ be two Bayesian Networks. If G and G' have the same number of edges then either the two Bayesian networks are equivalent, i.e. $\mathbf{D}(G) = \mathbf{D}(G')$, or the two Bayesian networks are not in inclusion, i.e. $\mathbf{D}(G) \not\subset \mathbf{D}(G')$ and $\mathbf{D}(G) \not\supset \mathbf{D}(G')$.

Proof. We will distinguish three cases. First when the two networks have different skeletons, second when the two networks have the same skeletons but different immoralities and third when the two networks have the same skeletons and the same immoralities.

In the first case, the two skeletons of G and G' are different but have the same number of edges. This implies that there are at least two distinct vertices u and v such that they are adjacent in G and are non-adjacent in G'. Either $v \in nd(u)$ or $u \in nd(v)$ holds in G' as otherwise there would be a cycle. By the DPMP (see Definition 3.3) there exists a CI restriction between u and v in G'. This CI restriction cannot hold in G according to the d-separation criterion because u and v are adjacent in G. Thus $\mathbf{D}(G) \not\supset \mathbf{D}(G')$. The same argument applies for $\mathbf{D}(G) \not\subset \mathbf{D}(G')$.

In the second case, G and G' have the same skeletons but different immoralities. Let $u \rightarrow w \leftarrow v$ be an immorality formed by three distinct vertices u, v and w which, without loss of generality, is in G but not in G'. This implies that the subgraph induced in G' by u, v and w will be either $u \leftarrow w \leftarrow v$ or $u \rightarrow w \rightarrow v$ or $u \leftarrow w \rightarrow v$ where u and v are non-adjacent as in G.

Either $v \in nd(u)$ or $u \in nd(v)$ holds in G' as otherwise there would be a cycle. By means of the DPMP (see Definition 3.3), there is a CI restriction $u \perp\!\!\!\perp v | C$ in G' for some C where $w \in C$. If $w \notin C$, $u \perp\!\!\!\perp v | C$ would not hold in G' according to the d-separation criterion. The same CI restriction $u \perp\!\!\!\perp v | C$, where $w \in C$, cannot hold in G because $w \in C$ creates an active path between u and v, so they are not d-separated. This leads to $\mathbf{D}(G) \not\supset \mathbf{D}(G')$. Analogously, for some subset C' where $w \notin C'$, the CI restriction $u \perp\!\!\!\perp v | C$ holds in G while it does not hold in G', therefore $\mathbf{D}(G) \not\subset \mathbf{D}(G')$.

The third case follows from Lemma 2.1, which leads to $\mathbf{D}(G) = \mathbf{D}(G')$. □

Lemma 3.3.
Let $\mathbf{D}(G)$ and $\mathbf{D}(G')$ be two Bayesian Networks. If $\mathbf{D}(G) \subset \mathbf{D}(G')$ then G' has less edges than G.

Proof. We will prove this lemma by contradiction. Assume $\mathbf{D}(G) \subset \mathbf{D}(G')$ and the two possibilities: (a) the two DAGs G and G' have the same number of edges or (b) G' has at least one more edge than G.

In (a) the contradiction follows from Lemma 3.2 which forces $\mathbf{D}(G)$ and $\mathbf{D}(G')$ to be either equivalent or not in inclusion.

In (b) the contradiction follows from the fact that there is a pair of vertices u, v which are non-adjacent in G and adjacent in G'. By the DPMP (see Definition 3.3) there is a CI restriction $u \perp\!\!\!\perp v | nd(u) \backslash v$ in G which does not hold in G', i.e. $\mathbf{D}(G) \not\subset \mathbf{D}(G')$. □

The following result discusses some of the relationships among the different concepts of neighborhoods as well as with respect to the inclusion boundary.

Theorem 3.1. *Let $\mathbf{D}(G)$ denote a Bayesian Network. Let $\mathcal{N}_{\mathrm{NR}}(G)$, $\mathcal{N}_{\mathrm{CR}}(G)$, $\mathcal{N}_{\mathrm{NCR}}(G)$, $\mathcal{N}_{\mathrm{AR}}(G)$, $\mathcal{N}_{\mathrm{ENR}}(G)$ and $\mathcal{N}_{\mathrm{ENCR}}$ be the sets of Bayesian Networks that form, respectively, the NR, CR, NCR, AR, ENR and ENCR neighborhoods for $\mathbf{D}(G)$. The following statements hold:*

1. *For all G: $\mathcal{N}_{\mathrm{NR}}(G) \subseteq \mathcal{N}_{\mathrm{ENR}}(G) \subseteq \mathcal{IB}(G)$; $\mathcal{N}_{\mathrm{NR}}(G) \subseteq \mathcal{N}_{\mathrm{NCR}}(G) \subseteq \mathcal{N}_{\mathrm{ENCR}}(G)$; $\mathcal{N}_{\mathrm{NR}}(G) \subseteq \mathcal{N}_{\mathrm{CR}}(G) \subseteq \mathcal{N}_{\mathrm{AR}}(G)$; $\mathcal{N}_{\mathrm{NR}}(G) \subseteq \mathcal{N}_{\mathrm{NCR}}(G) \subseteq \mathcal{N}_{\mathrm{AR}}(G)$; $\mathcal{N}_{\mathrm{ENR}}(G) \subseteq \mathcal{N}_{\mathrm{ENCR}}(G)$.*
2. *For all G: $\{\mathcal{N}_{\mathrm{NCR}}(G) \backslash \mathcal{N}_{\mathrm{NR}}(G)\} \cap \mathcal{IB}(G) = \{\mathcal{N}_{\mathrm{AR}}(G) \backslash \mathcal{N}_{\mathrm{CR}}(G)\} \cap \mathcal{IB}(G) =$
$\{\mathcal{N}_{\mathrm{ENCR}}(G) \backslash \mathcal{N}_{\mathrm{ENR}}(G)\} \cap \mathcal{IB}(G) = \{\mathcal{N}_{\mathrm{CR}}(G) \backslash \mathcal{N}_{\mathrm{NR}}(G)\} \cap \mathcal{IB}(G) = \emptyset$.*
3. *There exists G such that: $\mathcal{N}_{\mathrm{AR}}(G) \not\supseteq \mathcal{IB}(G)$.*

Proof. Statement 1. The part $\mathcal{N}_{\mathrm{NR}}(G) \subseteq \mathcal{N}_{\mathrm{ENR}}(G)$ follows directly from the fact that ENR performs all operations that NR does. The same argument applies to the $\mathcal{N}_{\mathrm{NR}}(G) \subseteq \mathcal{N}_{\mathrm{NCR}}(G)$, $\mathcal{N}_{\mathrm{NCR}}(G) \subseteq \mathcal{N}_{\mathrm{ENCR}}(G)$, $\mathcal{N}_{\mathrm{NR}}(G) \subseteq \mathcal{N}_{\mathrm{CR}}(G)$, $\mathcal{N}_{\mathrm{CR}}(G) \subseteq \mathcal{N}_{\mathrm{AR}}(G)$, $\mathcal{N}_{\mathrm{NCR}}(G) \subseteq \mathcal{N}_{\mathrm{AR}}(G)$ and $\mathcal{N}_{\mathrm{ENR}}(G) \subseteq \mathcal{N}_{\mathrm{ENCR}}(G)$.

The relationship $\mathcal{N}_{\mathrm{ENR}} \subseteq \mathcal{IB}(G)$ follows from Lemmas 2.1, 3.1 and 3.3. The first lemma guarantees us that the ENR neighborhood is created by transforming every member of an equivalence class. The second lemma proves that the Bayesian Network, from which the ENR neighborhood is created, precedes all Bayesian Networks in the ENR neigborhood under the inclusion order. The third lemma shows that, under the inclusion order, there are no Bayesian Networks *in between* the one from which the ENR neighborhood is created, and those from the ENR neighborhood, i.e. $\mathcal{N}_{\mathrm{ENR}} \subseteq \mathcal{IB}(G)$.

Statement 2. The Bayesian Networks in the difference sets $\{\mathcal{N}_{\mathrm{NCR}}(G) \backslash \mathcal{N}_{\mathrm{NR}}(G)\}$, $\{\mathcal{N}_{\mathrm{AR}}(G) \backslash \mathcal{N}_{\mathrm{CR}}(G)\}$ and $\{\mathcal{N}_{\mathrm{ENCR}}(G) \backslash \mathcal{N}_{\mathrm{ENR}}(G)\}$ are created by the reversal of a non-covered arc in G. This statement says that, for any given Bayesian Network $\mathbf{D}(G)$, if we reverse a non-covered arc of G obtaining a new DAG G', then $\mathbf{D}(G') \notin \mathcal{IB}(G)$. We prove this as follows. If an arc is not covered in G, its reversal either introduces or destroys an immorality in G' (see Definition 2.3) that yields a non-equivalent model (see Lemma 2.1). Since the number of edges remains the same, by Lemma 3.2, $\mathbf{D}^I(G) \not\subset \mathbf{D}^I(G')$, and therefore $\mathbf{D}^I(G) \nsubseteq \mathbf{D}^I(G')$.

The difference set $\{\mathcal{N}_{\mathrm{CR}}(G) \backslash \mathcal{N}_{\mathrm{NR}}(G)\}$ contains only DAGs equivalent to G and thus its intersection with the $\mathcal{IB}(G)$ is empty set.

Statement 3. Figure 4 illustrates this statement. □

The intuition behind the previous theorem is that the ENR neighborhood covers a larger part of the inclusion boundary than any of the previously introduced neighborhoods. In particular, the ENR will coincide with the inclusion boundary under the next circumstance.

Theorem 3.2. *Let $\mathbf{D}(G)$ be a DAG Markov model. If Meek's conjecture holds, then it follows that*

$$\mathcal{N}_{\mathrm{ENR}}(G) = \mathcal{IB}(G).$$

Proof. From the characterization of the inclusion order provided by Meek's conjecture it follows that the inclusion boundary are all DAGs with one arc more and one arc less than every member of the equivalence class of a given DAG, which coincides with the definition of ENR neighborhood. □

As it follows from its definition (and formally stated in Theorem 3.1), the ENCR neighborhood contains the ENR neighborhood plus other models that are not part of the inclusion boundary. Assuming that Meek's conjecture holds, the ENR neighborhood retains already the desired property of containing the inclusion boundary. Nevertheless, as we shall see later on the experiments, a variation of the ENCR neighborhood seems to be actually very useful, hence we formalize this neighborhood.

Finally, we are going to emphasize further the relevance of the inclusion order in the learning process by proving that the inclusion boundary condition (see Definition 3.2) is sufficient under the assumptions of data faithfulness and unbounded data length.

Before we formally introduce these assumptions let's recall the definition of penalized score function. Let $\mathrm{PL}(\mathbf{D}(G); D)$ denote the *penalized score function* of the dataset D against the Bayesian network $\mathbf{D}(G)$:

$$\mathrm{PL}(\mathbf{D}(G); D) = -H(\mathbf{D}(G), D) - \frac{f(|D|)}{|D|} * k(\mathbf{D}(G)),$$

where $H(\mathbf{D}(G), D)$ is the *entropy* (Kullback, 1959) of the Bayesian Network $\mathbf{D}(G)$ and the data D, i.e. the approximated distribution from $\mathbf{D}(G)$ and the distribution of D. The function $k(\mathbf{D}(G))$ provides the number of free (independent) parameters of the Bayesian Network $\mathbf{D}(G)$, and $f(|D|)$ is a positive function driving the penalization such that $\lim_{|D|\to\infty} \frac{f(|D|)}{|D|} = 0$. A typical example of a penalized score function is the Bayesian Information Criterion (Schwarz, 1978), or BIC, where $f(|D|) = \frac{1}{2}\log(|D|)$. The BIC is a large-sample approximation of the likelihood of a dataset given a model (Chickering and Heckerman, 1997).

A dataset D, sampled from some probability distribution P, is *faithful* (Spirtes et al., 1993) to some Bayesian Network $\mathbf{D}(G)$ if all and only the CI restrictions in P, make P Markov over G. One also says that G is a perfect map of D (Pearl, 1988).

If two models are in inclusion, i.e. $\mathbf{D}^I(X) \supset \mathbf{D}^I(X^*)$, then obviously the entropy of the data and the more complex model is smaller or equal than the entropy of the data and the less complex model, i.e. $H(\mathbf{D}(X^*), D) \leq H(\mathbf{D}(X), D)$. Similarly the number of free parameters of the more complex model is always higher than the number of free parameters of the less complex model, i.e. $k(\mathbf{D}(X^*)) > k(\mathbf{D}(X))$.

If two models are in inclusion, i.e. $\mathbf{D}^I(X) \supset \mathbf{D}^I(X^*)$, and we have data D faithful to the model $\mathbf{D}(G)$ then the entropy $H(\mathbf{D}(X^*), D) = H(\mathbf{D}(X), D)$ if and only if all the CI restrictions in $\mathbf{D}^I(X)$ and not in $\mathbf{D}^I(X^*)$ are valid in the data D, i.e. $\mathbf{D}^I(X) \backslash \mathbf{D}^I(X^*) \backslash \mathbf{D}^I(G) = \emptyset$. This is the situation when it does not make sense to go to the more complex model $\mathbf{D}(X^*)$ because it does not enable us to include any information more from the data.

We will use these basic properties to derive the following two useful lemmas. They characterize relations between PL scores of pairs of model in inclusion with respect to faithful data of unbounded length.

Lemma 3.4.
Let D^∞ be a set of data of unbounded length faithful to the model $\mathbf{D}(G)$. Let $\mathbf{D}(X)$ and $\mathbf{D}(X^*)$ be two models in inclusion, i.e. $\mathbf{D}^I(X) \supset \mathbf{D}^I(X^*)$, where $\mathbf{D}^I(X) \backslash \mathbf{D}^I(X^*) \backslash \mathbf{D}^I(G) \neq \emptyset$. Then $\mathrm{PL}(\mathbf{D}(X^*); D^\infty) > \mathrm{PL}(\mathbf{D}(X); D^\infty)$.

Proof. $\mathbf{D}^I(X) \supset \mathbf{D}^I(X^*)$ thus $H(\mathbf{D}(X^*), D) \leq H(\mathbf{D}(X), D)$ and $k(\mathbf{D}(X^*)) > k(\mathbf{D}(X))$. Because $\mathbf{D}^I(X) \backslash \mathbf{D}^I(X^*) \backslash \mathbf{D}^I(G) \neq \emptyset$ we know that $H(\mathbf{D}(X^*), D) \neq H(\mathbf{D}(X), D)$. Thus $H(\mathbf{D}(X^*), D) < H(\mathbf{D}(X), D)$. Knowing from the definition of the PL score that $\lim_{|D|\to\infty} \frac{f(|D|)}{|D|} = 0$ we conclude $\mathrm{PL}(\mathbf{D}(X^*); D^\infty) > \mathrm{PL}(\mathbf{D}(X); D^\infty)$. □

Lemma 3.5.
Let D^∞ be a set of data of unbounded length faithful to the model $\mathbf{D}(G)$. Let $\mathbf{D}(X^*)$ and $\mathbf{D}(X)$ be two models in inclusion, i.e. $\mathbf{D}^I(X) \subset \mathbf{D}^I(X^*)$, where $\mathbf{D}^I(X^*) \subseteq \mathbf{D}^I(G)$. Then $\mathrm{PL}(\mathbf{D}(X^*); D^\infty) > \mathrm{PL}(\mathbf{D}(X); D^\infty)$.

Proof. $\mathbf{D}^I(X) \subset \mathbf{D}^I(X^*)$ thus $H(\mathbf{D}(X^*), D) \geq H(\mathbf{D}(X), D)$ and $k(\mathbf{D}(X^*)) < k(\mathbf{D}(X))$. From $\mathbf{D}^I(X) \subset \mathbf{D}^I(X^*) \subseteq \mathbf{D}^I(G)$ follows that $\mathbf{D}^I(X^*) \backslash \mathbf{D}^I(X) \backslash \mathbf{D}^I(G) = \emptyset$. Thus $H(\mathbf{D}(X^*), D) = H(\mathbf{D}(X), D)$. From $H(\mathbf{D}(X^*), D) = H(\mathbf{D}(X), D)$ and $k(\mathbf{D}(X^*)) < k(\mathbf{D}(X))$ follows the desired by definition of the PL score. □

As we shall see now, there is a fundamental relationship between learning Bayesian Networks from data and the inclusion order. The next theorem shows that following the right inclusion path, the PL score always increases.

Theorem 3.3. *Let $\mathbf{D}(G^*)$ be a Bayesian Network faithful to a dataset D^∞ of unbounded length. Let $\mathrm{PL}(\mathbf{D}(G); D^\infty)$ be any PL score of the dataset D^∞ against the Bayesian Network $\mathbf{D}(G)$.*

The PL score increases monotonically through any inclusion path $\mathbf{D}^I(G_1) \supset \mathbf{D}^I(G_2) \supset \ldots \supset \mathbf{D}^I(G_n)$ ending with the faithful model, i.e. with $G_n = G^*$:

$$\mathrm{PL}(\mathbf{D}(G_1); D^\infty) < \mathrm{PL}(\mathbf{D}(G_2); D^\infty) < \ldots < \mathrm{PL}(\mathbf{D}(G_n); D^\infty) \quad \text{where } \mathrm{G_n} = \mathrm{G}^*.$$

Proof. We will show that if $\mathbf{D}^I(G_i) \supset \mathbf{D}^I(G_j) \supseteq \mathbf{D}^I(G^*)$ then $\mathrm{PL}(\mathbf{D}(G_i); D^\infty) < \mathrm{PL}(\mathbf{D}(G_j); D^\infty)$. From $\mathbf{D}^I(G_i) \supset \mathbf{D}^I(G_j) \supseteq \mathbf{D}^I(G^*)$ it follows that $\mathbf{D}^I(G_i) \backslash \mathbf{D}^I(G_j) \backslash \mathbf{D}^I(G^*) \neq \emptyset$. By Lemma 3.4 we get $\mathrm{PL}(\mathbf{D}(G_i); D^\infty) < \mathrm{PL}(\mathbf{D}(G_j); D^\infty)$. □

Note that a learning algorithm which satisfies the inclusion boundary condition can possibly follow the longest inclusion path to the faithful model. There always exists a longest inclusion path from the empty graph model to any model.

The following theorem proves correctness of the hill-climbing algorithm under the faithfulness and unbounded data assumptions.

Theorem 3.4. *Let $\mathbf{D}(G^*)$ be a Bayesian Network faithful to a dataset D^∞ of unbounded length. Let $\mathrm{PL}(\mathbf{D}(G); D^\infty)$ be any PL score of the dataset D^∞ against the Bayesian Network $\mathbf{D}(G)$.*

The hill-climbing algorithm using the PL score and a traversal operator satisfying the inclusion boundary condition always finds the faithful model $\mathbf{D}(G^*)$ which has the highest score.

Proof. First we will shortly review how the hill-climber works. It starts with some model, usually the empty graph model. It computes the score of all neighbours of current model and goes to the model with highest score. It stops when no model in the neighborhood of the current model has higher score than the current model.

Second we will prove the theorem. We will show that the hill-climber always stops and we will show that the hill-climber returns the faithful model $\mathbf{D}(G^*)$. Moreover we will show that the faithful model $\mathbf{D}(G^*)$ has the highest score among all models.

The hill-climber produces a sequence of models with increasing score. Because the number of all models is finite it will stop.

The model where the hill-climber stops has in its neighbourhood no model with higher score. We show that for each model $\mathbf{D}(X) \neq \mathbf{D}(G^*)$ there is a model $\mathbf{D}(X^*)$ where $\mathbf{D}(X^*) \in \mathcal{N}(X)$ and $\mathrm{PL}(\mathbf{D}(X^*); D^\infty) > \mathrm{PL}(\mathbf{D}(X); D^\infty)$. Thus the hill-climber cannot stop in $\mathbf{D}(X)$, i.e. it has to stop in $\mathbf{D}(G^*)$. Each model $\mathbf{D}(X) \neq \mathbf{D}(G^*)$ satisfies either $\mathbf{D}^{\boldsymbol{I}}(X) \subset \mathbf{D}^{\boldsymbol{I}}(G^*)$ or $\mathbf{D}^{\boldsymbol{I}}(X) \not\subseteq \mathbf{D}^{\boldsymbol{I}}(G^*)$. The proof of existence of $\mathbf{D}(X^*)$ follows separately for these two cases.

Suppose a model $\mathbf{D}(K)$ where $\mathbf{D}^{\boldsymbol{I}}(K) \subset \mathbf{D}^{\boldsymbol{I}}(G^*)$. Then there exists a longest inclusion path $\mathbf{D}^{\boldsymbol{I}}(G_1) \supset \mathbf{D}^{\boldsymbol{I}}(G_2) \supset \ldots \supset \mathbf{D}^{\boldsymbol{I}}(G_n)$ where $G_1 = G^*$ and $G_n = K$. Take the model $\mathbf{D}(G_{n-1})$ from this path and denote it by $\mathbf{D}(K^*)$. Thus $\mathbf{D}^{\boldsymbol{I}}(G^*) \supseteq \mathbf{D}^{\boldsymbol{I}}(K^*) \supset \mathbf{D}^{\boldsymbol{I}}(K)$. By Lemma 3.5 we get $\mathrm{PL}(\mathbf{D}(K^*); D^\infty) > \mathrm{PL}(\mathbf{D}(K); D^\infty)$. The model $\mathbf{D}(K^*)$ is neighbor of the model $\mathbf{D}(K)$ in the longest inclusion path, thus $\mathbf{D}(K^*) \in \mathcal{IB}(K)$. Because the traversal operator satisfies the inclusion boundary condition, $\mathbf{D}(K^*) \in \mathcal{N}(K)$.

Suppose a model $\mathbf{D}(L)$ where $\mathbf{D}^{\boldsymbol{I}}(L) \not\subseteq \mathbf{D}^{\boldsymbol{I}}(G^*)$. Then $\mathbf{D}^{\boldsymbol{I}}(L) \backslash \mathbf{D}^{\boldsymbol{I}}(G^*) \neq \emptyset$. Thus there is some CI restriction of the type $a \perp\!\!\!\perp b | C$ in $\mathbf{D}^{\boldsymbol{I}}(L) \backslash \mathbf{D}^{\boldsymbol{I}}(G^*)$. Thus in the DAG L the nodes a and b are not adjacent (otherwise there would be an active path in L and the CI would not hold). Denote by L^* the DAG obtained from the DAG L by adding an arc between the nodes a and b (at least one orientation of the arc which does not create any cycle is always possible, otherwise the graph L would be cyclic). Then $\mathbf{D}(L^*) \in \mathcal{N}_{\mathrm{NR}}(L)$. By Theorem 3.1 which says that $\mathcal{N}_{\mathrm{NR}}(L) \subseteq \mathcal{IB}(L)$ we get $\mathbf{D}(L^*) \in \mathcal{IB}(L)$ and $\mathbf{D}^{\boldsymbol{I}}(L^*) \subset \mathbf{D}^{\boldsymbol{I}}(L)$. $\{a \perp\!\!\!\perp b | C\} \in \mathbf{D}^{\boldsymbol{I}}(L) \backslash \mathbf{D}^{\boldsymbol{I}}(L^*) \backslash \mathbf{D}^{\boldsymbol{I}}(G^*)$ thus $\mathbf{D}^{\boldsymbol{I}}(L) \backslash \mathbf{D}^{\boldsymbol{I}}(L^*) \backslash \mathbf{D}^{\boldsymbol{I}}(G^*) \neq \emptyset$. By Lemma 3.4 we get $\mathrm{PL}(\mathbf{D}(L^*); D^\infty) > \mathrm{PL}(\mathbf{D}(L); D^\infty)$. From $\mathbf{D}(L^*) \in \mathcal{IB}(L)$ and the fact that the traversal operator satisfies the inclusion boundary condition we conclude that $\mathbf{D}(L^*) \in \mathcal{N}(L)$.

We have shown that the hill-climber always finds the faithful model $\mathbf{D}(G^*)$ and we did not assume any special model where the hill-climber starts, i.e. the hill-climber can start in any model.

Because the hill-climber always produces a sequence of models with increasing score, the faithful model $\mathbf{D}(G^*)$ has the highest score among all models. This finishes the proof. □

The idea of the proof above is that from any model there is always an inclusion path to some bigger model (in the worst case it is the complete one) from which there is another inclusion path to the faithful model. Theorem 3.3 says that when starting from the empty graph model the bigger model can always (possibly) be the faithful model itself. Because there is no guarantee that the hill-climber will follow the inclusion path to the faithful model it may use some other bigger model (more complex than the faithful one). It is clear that enabling the smallest possible bigger model (i.e. the faithful model, i.e. what Theorem 3.3 does) is crucial when working with finite size data. Meek (1997) in his PhD thesis proved a similar result to which, unfortunately, the authors have not had access to provide comparison.

4 The RCARNR and RCARR Neighborhoods

A straightforward realization is that both the ENR and ENCR neighborhoods are not computationally efficient to handle. More concretely, the effort to enumerate the members of an equivalence class is prohibitive since there is no *cheap* graphical characterization of those members.

However, we conjecture that because the average ratio of DAGs per equivalence class seems to be bounded by some constant, it might suffice to *simulate* somehow the ENR neighborhood. In order to do that, we introduce the *repeated covered arc reversal* algorithm, or RCAR algorithm, that allows to reach any member of the equivalence class with certain probability. We may see the RCAR algorithm in Figure 5.

```
algorithm g.rcar(int $r$) is
01  int $rr = rnd(0, r)$
02  for $i = 0$ to $rr$ do
03      vector $ce = g.covered\_edges()$
04      int $j = rnd(0, ce.size() - 1)$
05      edge $e = ce[j]$
06      $g.reverse\_edge(e)$
07  endfor
endalgorithm
```

Figure 5: RCAR algorithm implemented as a method for an object g that embodies a DAG and implements a method that returns a vector of the covered edges and an another method that reverses a given edge.

The algorithm in Figure 5 takes a constant r as parameter and iterates some random number of times between 0 (no iteration) and r. At each iteration, it picks at random a covered arc and reverses it. Lemma 2.1 guarantees us that the RCAR algorithm reaches any member of the equivalence class with a positive probability for a *sufficiently large* maximum number r of iterations. The bounded ratio of DAGs per equivalence class suggests that a small number between 4 and 10 should be *sufficiently large*.

Note that when the number of undirected edges in the corresponding essential graph is lower or equal to the number of iterations of RCAR, then RCAR is able to reach any Bayesian Network in its equivalence class. Gillispie and Perlman (2001) show that the distribution of the sizes of the equivalence classes represented by essential graphs, follows a very particular pattern. In concrete, they make the following observation (Gillispie and Perlman, 2001):

> The pattern of the distribution shows that certain sizes appear more frequently than others. In particular, larger compound numbers occur more often than larger prime numbers. This is probably due to separate sets of undirected edges in the essential graph acting independently to produce class sizes that are products of the sizes of their independent components.

In a nutshell, essential graphs with many disconnected components represent equivalence classes with a large number of members. However, the fact that these classes have many disconnected components allows RCAR to reach a substantial amount of the members of those equivalence classes.

Using the RCAR algorithm, we may define the following two new concepts of neighborhood for Bayesian Networks:

- **RCARNR** (RCAR+NR) Perform the RCAR algorithm and then create a NR neighborhood, noted $\mathcal{N}_{\mathrm{RCARNR}}(G)$.
- **RCARR** (RCAR+NCR) Perform the RCAR algorithm and then create a NCR neighborhood, noted $\mathcal{N}_{\mathrm{RCARR}}(G)$.

The RCARNR neighborhood may be seen as a simulation, or an approximation, of the ENR neighborhood, and analogously between the RCARR and ENCR neighborhoods. We can establish the following property for the RCARNR and RCARR neighborhoods.

Theorem 4.1. *Let* $\mathbf{D}(G)$ *be DAG Markov model. For a* sufficiently large *maximum number of iterations* r *of the RCAR algorithm,* $\mathbf{D}(G') \in \mathcal{N}_{\mathrm{ENR}}(G) \Rightarrow \mathbf{D}(G') \in \mathcal{N}_{\mathrm{RCARNR}}(G)$ *and* $\mathbf{D}(G') \in \mathcal{N}_{\mathrm{ENCR}}(G) \Rightarrow \mathbf{D}(G') \in \mathcal{N}_{\mathrm{RCARR}}(G)$*, with probability* $p > 0$.

Proof. It follows directly from Lemma 2.1 and the definitions of ENR, ENCR, RCARNR and RCARR neighborhoods. □

5 Heuristic Search

The usual learning algorithm used in heuristic search consists of a hill-climber that iterates until the score metric does not improve. The score metric is evaluated typically throughout a NR or an AR neighborhood at each iteration. The NR or AR neighborhoods for DAG models have been traditionally restricted by a maximum number of parents and a causal ordering because otherwise the hill-climber would not find a model of a reasonable *quality* or simply the algorithm would not scale up in a reasonable time to a larger number of variables. Therefore, some of the existing algorithms assume that the causal ordering is known (Cooper and Herskovits, 1992), or they search for a *good* causal ordering that can be used later (Bouckaert, 1992; Singh and Valtorta, 1993; Larrañaga et al., 1996; Friedman and Koller, 2000).

However, the causal ordering reduces the already small part of the inclusion boundary that was reachable from a NR or AR neighborhood. Therefore, errors in the ordering may easily lead to very bad local maxima, as shown in Chickering et al. (1995). The same reasoning holds for the number of parents. Heuristic algorithms that do not use any form of causal ordering at all correspond in fact to those that use EG-space (Spirtes and Meek, 1995; Chickering, 1996, 2001).

We will introduce now a new heuristic algorithm which works in DAG-space, and partially accounts for the inclusion order. This will be achieved by the use of the RCAR algorithm (see Figure 5) that allows to create the RCARNR and RCARR neighborhoods.

The algorithm we propose is in Figure 6. It consists of a usual hill-climber that iterates through lines 4 to 17. It contains two modifications. One, on line 5, is to perform the RCAR algorithm from Figure 5 on the current DAG. The other is to perform again the RCAR algorithm when a local maxima is reached, as in line 13. This last step will be performed a maximum number of times (MAXTRIALS). If within this maximum number of times, it has not been possible to escape from that local maxima, then the RCAR algorithm is not called again and the hill-climber will stop iterating.

The first of the two modifications, on line 5, is followed by the creation of a NR or NCR neighborhood, depending on the truth value of the parameter *ncr*. In this way, a RCARNR or RCARR neighborhood will be employed. Afterwards, on line 7, the method $g.score\_and\_pick\_best(nh)$ scores all members of the neighborhood and returns the one that provides the highest score, which is assigned to g'.

The second of the two modifications, on lines 12 to 16, resembles in a way the *iterative hill-climber* introduced in Chickering et al. (1995), which consists of perturbing randomly the current model once a local maxima is reached. This is done in line 13 as well but the perturbation is constrained to a move within the same equivalence class of the current model. Due to

```
algorithm hcmc(int $r$, bool $ncr$) returns dag
dag $g = emptydag$
bool $local\_maxima = false$
int $trials = 0$
while ($\neg local\_maxima$) do
  $g$.rcar($r$)
  set $nh = g$.neighborhood($ncr$)
  dag $g' = g$.score_and_pick_best($nh$)
  $local\_maxima = (g'.score() < g.score())$
  if ($\neg local\_maxima$) then
    $g = g'$
    $trials = 0$
  else if ($trials <$ MAXTRIALS) then
    $g$.rcar($r$)
    $local\_maxima = false$
    $trials = trials + 1$
  endif
endwhile
return $g$
endalgorithm
```

Figure 6: Hill-Climber Monte Carlo algorithm

the random nature of the new steps introduced in the hill-climber, we call this algorithm the Hill-Climber Monte Carlo, or HCMC for short.

6 The Markov Chain Monte Carlo Method

The need to account for the uncertainty of models (Draper, 1995) has led to the development of computational methods that implement the full Bayesian approach to modelling. Recall Bayes' theorem:

$$p(M|D) = \frac{p(D|M)p(M)}{p(D)}, \tag{3}$$

where $p(D)$ is known as the normalizing constant which is computed as follows

$$p(D) = \sum_{M \in \mathcal{M}} p(D|M)p(M). \tag{4}$$

Once we account for the uncertainty of the models, it is possible to compute the posterior distribution of some quantity of interest Δ, by averaging over all the models in the following way

$$p(\Delta|D) = \sum_{M \in \mathcal{M}} p(\Delta|M, D)p(M|D). \tag{5}$$

As we saw in the first section, DAG-space has a prohibitive size as to go exhaustively through all the models. Therefore, it is not computationally feasible to carry out the sums in (4) and (5).

The method of Markov Chain Monte Carlo (MCMC hereafter) solves this problem by sampling directly from the posterior distributions $p(M|D)$ and $p(\Delta|D)$, thus performing the sumations implicitly. The MCMC method had its origins in a sampling method introduced by Metropolis et al. (1953) within the context of statistical physics. However, it was in the work of Hastings (1970), that this sampling method was generalized for statistical problems, by using the theory of Markov chains, introducing the well known *Metropolis-Hastings* algorithm.

The Metropolis-Hastings algorithm was adapted for structural learning of GMMs by Madigan and York (1995), who called it the *Markov Chain Monte Carlo Model Composition*, or MC^3 algorithm for short.

For each Bayesian Network $M \equiv \mathbf{D}(G)$, let $\mathcal{N}(G)$ be the set of neighbors of M. Let q be a transition matrix such that for some other Bayesian Network $M' \equiv \mathbf{D}(G')$, $q(M \rightarrow M') = 0$ if $M' \notin \mathcal{N}(G)$ and $q(M \rightarrow M') > 0$ if $M' \in \mathcal{N}(G)$. Using the transition matrix q we build a Markov chain $M(t, q)$, $t = 1, 2, \ldots, n$, with state space $\mathcal{M}$.

Let the chain $M(t, q)$ be in state M and let's draw a candidate model M' from $q(M \rightarrow M')$. The proposed model M' is accepted with probability

$$\alpha(M', M) = min\left\{1, \frac{|\mathcal{N}(G)|p(M'|D)}{|\mathcal{N}(G')|p(M|D)}\right\}, \tag{6}$$

where $|\mathcal{N}(G)|$ refers to the cardinality of the set of neighbors of the model M. If M' is not accepted, $M(t, q)$ remains in state M. The idea behind is that a Markov chain $M(t, q)$ built in such way, has $p(M|D)$ as its equilibrium distribution. This means that, after the chain has ran for *enough* time, the draws can be regarded as a sample from the target density $p(M|D)$, and one says that the chain has *converged.*

The *convergence* of the Markov chain to the target density $p(M|D)$ is guaranteed under two mild regularity conditions: irreducibility and aperiodicity. The reader may find a discussion in depth of these conditions in (Smith and Roberts, 1993).

Given output $M(t, q) = \{M_{t=1}, M_{t=2}, \ldots, M_{t=n}\}$ of the Markov chain, the regularity conditions allow to derive the following asymptotic results (Chung, 1967; Hastings, 1970; Smith and Roberts, 1993; Madigan and York, 1995):

$$M_{t=n} \overset{n\rightarrow\infty}{\longrightarrow} M \sim p(M|D) \tag{7}$$

$$\frac{1}{n}\sum_{t=1}^{n} f(M(t,q)) \overset{n\to\infty}{\longrightarrow} E(f(M)) \tag{8}$$

Which imply that, when the Markov chain $M(t,q)$ converges:

- the draws from the Markov chain mimic a random sample from $p(M|D)$. Therefore, in order to get an estimate of $p(M|D)$, it suffices to account for the frequency of visits of each model M and divide it by the number iterations n,

and

- the average of the realizations of any function of the model, $f(M)$, is an estimator of the expectation of $f(M)$. Therefore, by setting $f(M) = p(\Delta|M, D)$ we will approximate the sum in (5) (Madigan and York, 1995),

respectively. In this setting, the transition matrix q choses randomly the Bayesian Network typically from a NR or AR neighborhoods. The enhancement we introduce here consists of simply modifying q such that it uses RCARNR or RCARR neighborhoods. In Figure 7 we may see the pseudocode of the modified MC^3 algorithm, which we will call the *enhanced* MC^3 or *e*MC^3 for short. For later comparison in the experiments, we have tuned the algorithm also to work with NR, CR and NCR neighborhoods.

```
algorithm eMC^3(dag $init$, int $n$, int(dag) $f$, int $r$, bool $ncr$,
                int $burn\_in$) returns {dbl[],dbl[]}
dag $g = init$
int $i = 0$
dbl $p[]$
dbl $d[]$
while ($i < n$) do
  $g$.rcar($r$)
  set $nh = g$.neighborhood($ncr$)
  dag $g'$ =pick_at_random($nh$)
  dbl $x = \mathcal{U}(0,1)$
  if ($x \leq \alpha(g, g')$) then $g = g'$ endif
  if ($i > burn\_in$) then
    $p[g] = p[g] + 1$
    $d[f(g)] = d[f(g)] + 1$
  endif
  $i = i + 1$
endwhile
for $g = p$.fst() to $p$.lst() do $p[g] = p[g]/n$ enddo
for $i = 1$ to $d$.size() do $d[i] = d[i]/n$ enddo
return $\{p, d\}$
endalgorithm
```

Figure 7: The *e*MC^3 algorithm.

The algorithm in Figure 7 needs the specification of the following six parameters:

- *init*: some arbitrary Bayesian Network from which the Markov chain starts.
- n: number of iterations that the Markov chain will perform.
- f: function that takes a Bayesian Network as input and gives an integer number as output, which indexes some quantity of interest of the model.
- r maximum number of iterations for the RCAR algorithm.
- *ncr* flag set to true when the algorithm should use a RCARR neighborhood, and false for a RCARNR neigborhood.
- *burn_in*: number of iterations we want to discard before the algorithm starts accounting for the frequency of visits to the models.

The algorithm uses two vectors p and d to store the estimated posteriors $p(M|D)$ and $p(\Delta|D)$, which are the result that the algorithm returns.

The Markov chain iterates through lines 5 to 16. On line 6, it performs the RCAR operation (see Figure 5) on the current Bayesian Network g. Then, from this new Bayesian Network a NR or NCR neighborhood is created, depending on the truth value of the parmeter *ncr*. In line 8 a Bayesian Network is picked randomly from this neighborhood. These three lines of code implement the transition matrix q.

In line 9 a random number x is generated from a uniform distribution between 0 and 1. This random number x is used to accept the candidate Bayesian Network g' with probability as specified in (6). If the current iteration i is beyond those to be discarded (line 11), then on lines 12 and 13 the current state of the Markov chain is stored.

Finally, on lines 17 and 18, the averages of the stored quantites p and d are computed and they are returned on line 19.

7 Experimental Comparison

Throughout all the experimentation we have used the Alarm dataset (Beinlich et al., 1989), which is a synthetic dataset that has become a standard benchmark for the assessment of learning algorithms for DAG models on discrete data. The Alarm dataset was sampled from the Bayesian Network in Figure 8, which was designed for a monitoring system in the context of intensive care unit ventilator management.

The Alarm dataset, originally employed by Herskovits (1991), contains 20000 records. From this dataset the first 10000 records where used by Cooper and Herskovits (1992) to assess the K2 learning algorithm. We will use here this latter dataset of 10000 records. From this 10000 records we have sampled six different datasets of two different sizes: three of 1000 records and

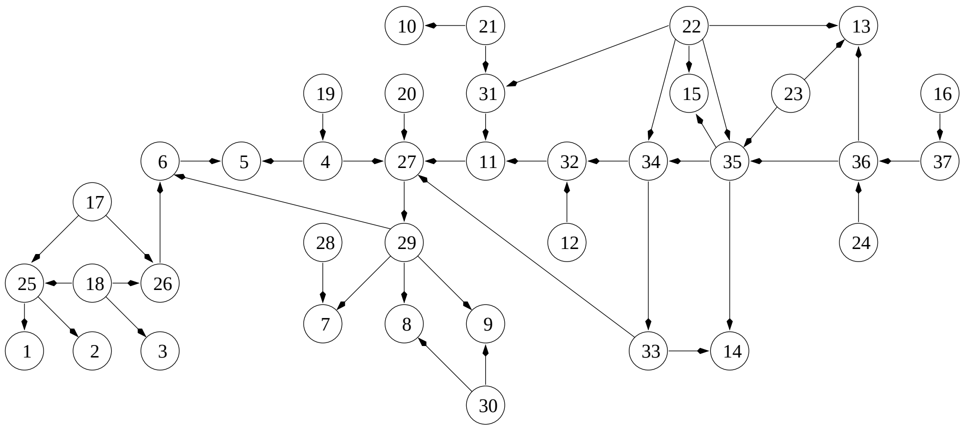


Figure 8: The Alarm Network (Beinlich et al., 1989): 37 vertices and 46 edges.

three of 5000 records. The experiments reported on 10000 records regard only the single dataset of the first 10000 records. As reported by Cooper and Herskovits (1992), this dataset of 10000 records does not support the arc between vertices 12 and 32 (see Figure 8).

7.1 Heuristic Learning

The assessment has been done as follows. On each of the seven datasets, the HCMC was run ten times for four different cardinalities of RCAR (2, 4, 7 and 10) and three different neighborhoods (AR, RCARR and RCARNR). The traditional hill-climber that uses an AR neighborhood will be referred here as RCAR 0.

The reason for running the HCMC several times is obvious since this new hill-climber performs random moves that may lead to different results in different runs. The maximum number of trials for escaping local maxima (MAXTRIALS) is set to 50. The results have been averaged over ten runs and confidence intervals, at a level of 95% for the means of score and structural difference, have been included. We may see these results in Table 1.

The information in Table 1 regards two aspects, the performance of the algorithm and the accuracy of the learned models with respect of the true one (Figure 8). The performance is provided in terms of number of steps, i.e. local transformations of a DAG, that the HCMC performs before stopping and the average time per step. This is meant to be a timeless way of assessing performance since we would only need to scale the average time per iteration as the hardware performance improves.

The accuracy of the learned models is measured in terms of the score metric, where the higher is the better and the structural difference between

Table 1: Results (1st part) of the HCMC learning algorithm over the Alarm dataset. Performance is averaged over RCARR and RCARNR.

smpl	rcar	performance		score		struct diff	
		steps	sec/st	RCARNR	RCARR	RCARNR	RCARR
1ka	0	55	0.27	-11480.47	-11480.47	29	29
	2	58	0.40	-11491.52±15.12	-11470.46±15.79	18.90±3.06	16.50±2.00
	4	55	0.44	-11484.69±19.29	-11473.41±14.18	18.00±3.28	16.30±2.18
	7	54	0.43	-11469.06±07.88	-11470.75±14.67	15.50±1.55	15.60±1.88
	10	53	0.43	-11470.43±15.94	-11464.03±11.00	14.80±2.15	15.20±1.78
1kb	0	60	0.28	-11115.13	-11115.13	28	28
	2	58	0.40	-11113.50±19.07	-11105.10±20.62	18.10±2.77	14.70±3.87
	4	56	0.42	-11121.49±24.64	-11090.15±08.51	17.90±4.41	11.10±2.35
	7	53	0.42	-11095.19±12.38	-11083.13±05.07	13.40±2.55	10.00±1.47
	10	53	0.43	-11095.87±11.25	-11094.17±18.72	12.40±2.05	11.50±2.11
1kc	0	62	0.61	-11530.80	-11530.80	37	37
	2	60	0.41	-11451.58±14.23	-11453.70±15.75	18.20±2.23	15.90±3.10
	4	59	0.43	-11438.31±08.01	-11436.65±07.46	14.90±2.95	13.40±1.94
	7	56	0.67	-11431.02±06.23	-11427.84±03.62	11.80±1.61	10.80±0.81
	10	53	0.86	-11440.88±10.78	-11428.89±08.95	13.70±2.13	11.00±1.17
5ka	0	69	1.54	-55249.43	-55249.43	46	46
	2	66	2.50	-55072.99±67.61	-54993.41±08.70	11.60±6.70	7.20±2.23
	4	57	2.09	-55051.93±40.93	-54992.40±10.92	7.90±2.12	5.20±1.38
	7	56	2.14	-55024.53±48.12	-54989.70±10.12	7.10±2.17	4.90±1.37
	10	56	2.08	-55025.19±43.49	-54985.99±06.68	6.10±1.95	5.10±2.49
5kb	0	57	0.92	-54732.19	-54732	33	33
	2	57	2.02	-54679.46±34.06	-54641.27±60.60	12.50±6.25	6.10±4.08
	4	56	1.35	-54610.82±15.24	-54607.60±14.34	5.20±3.93	3.80±1.38
	7	53	1.29	-54611.85±25.63	-54602.77±10.93	4.50±1.52	3.70±1.31
	10	52	1.28	-54602.98±10.85	-54606.47±12.48	4.00±1.26	4.10±1.32
5kc	0	59	0.88	-54454.16	-54454.16	36	36
	2	63	1.19	-54340.02±16.48	-54335.27±32.25	10.20±3.97	8.00±2.18
	4	59	1.20	-54335.49±19.99	-54326.25±11.15	8.60±1.91	8.40±2.05
	7	55	1.25	-54331.19±12.09	-54315.06±07.33	8.50±1.55	7.70±1.35
	10	55	1.28	-54363.17±52.63	-54329.40±11.13	10.00±2.18	8.50±1.85
10k	0	56	1.86	-108697.78	-108697.78	21	21
	2	56	2.23	-108495.65±68.33	-108463.65±46.17	4.90±2.20	5.40±4.10
	4	54	2.28	-108549.53±63.63	-108437.83±35.72	6.80±2.25	**1.60±0.90**
	7	50	2.29	-108477.50±52.06	-108485.55±58.14	5.50±3.22	**2.80±1.11**
	10	50	2.41	-108468.56±53.07	-108477.98±51.65	4.20±1.34	**3.30±1.17**

the essential graph of the learned DAG and the essential graph of the true model of Figure 8. In summary, these are the measures taken:

- *steps*: number of steps ($g = g'$ in the algorithm) of the HCMC.
- *sec/st*: speed of the HCMC in seconds per step.
- *score*: confidence interval of the mean of the score.
- *struct diff*: confidence interval of the mean of the structural difference.

As we may appreciate, there is a substantial difference in using RCAR within the hill-climber. The structural differences, throughout all the seven samples, for the standard hill-climber (RCAR 0) fall far away outside the confidence intervals for any of the different cardinalities of RCAR, which are nicely centered at a much lower values than the values for RCAR 0.

Regarding the score, on samples of 1000 records, RCAR 7 and 10 show a significantly higher score than RCAR 0, since the latter does not fall into their confidence intervals. At 5000 and 1000 records, the score for RCAR 0 falls outside the intervals for any of the cardinalities of RCAR.

The highest accuracy of the learned models is achieved when a RCARR neighborhood is used. The most striking evidence lies in the case of 10000 records and RCARR. There, an average of just only 1.6 structural differences is achieved in about 54 steps. As its confidence interval already suggests, let's remark the fact that on *eight* out of ten of those runs, there was only just one structural difference, corresponding to the missing arc not supported by the data. In some of those eight times, the result was reached in 49 steps and the same result was reached for RCAR 7 and 10 even in 48 steps, which is extremely close to the optimal path of the right result (46 additions).

In order to gain further insight into the HCMC algorithm and discuss its performance, we have taken further measures that we may see in Table 2 and we describe as follows.

- *considered models*: accumulated number of models that have improve the score at each step of the learning algorithm.
- *escapes/trials*: number of escapes of local maxima performed by doing RCAR and average number of trials per escape.

As we did with the scores, confidence intervals at a level of 95% are provided for the number of considered models. Although all of them are quite wide, they show that the HCMC algorithm considers significantly less amount of models to achieve a better result. This means that the HCMC algorithm makes better choices during the search process which, provide its greedy nature, implies that the way HCMC traverses the search space is definitely better.

The number of escapes and trials, provide an idea of how effective is the RCAR algorithm in avoiding local maxima. We may see that a higher cardinality of RCAR implies a lower number of times that the HCMC escapes from a local maxima achieving a similar result. We may appreciate that with the RCARR neighborhood the number of times HCMC is able to escape from local maxima is slightly higher. In preliminary experiments not reported here, we noted that this difference is more significant if the maximum number of trials (MAXTRIALS) is smaller. This fact may be one of the reasons why RCARR seems to perform slightly better than RCARNR.

Finally, let's highlight the computational trade-off that this approach affords. We want to know what is the overhead of using the RCAR operation, as in the HCMC algorithm, in front of the usual hill-climber. We can see

Table 2: Results (2nd part) of the HCMC learning algorithm over the Alarm dataset. Performance is averaged over RCARR and RCARNR.

smpl	rcar	performance		considered models		escapes/trials		struct diff	
		steps	sec/st	RCARNR	RCARR	RCARR	RCARNR	RCARNR	RCARR
1ka	0	55	0.27	6505	6505	0	0	29	29
	2	58	0.40	6037±137.1	6061±131.4	2.40/7.72	2.30/6.82	18.90±3.06	16.50±2.00
	4	55	0.44	5856±117.2	5936±145.9	1.60/6.55	2.00/3.80	18.00±3.28	16.30±2.18
	7	54	0.43	5810±71.4	5862±113.5	1.60/4.47	1.70/2.92	15.50±1.55	15.60±1.88
	10	53	0.43	5723±63.6	5816±111.6	1.20/3.00	1.70/3.45	14.80±2.15	15.20±1.78
1kb	0	60	0.28	6460	6460	0	0	28	28
	2	58	0.40	5940±79.7	5956±131.7	1.70/5.07	2.20/11.36	18.10±2.77	14.70±3.87
	4	56	0.42	5933±97.3	5861±120.1	2.00/10.53	1.60/6.17	17.90±4.41	11.10±2.35
	7	53	0.42	5800±89.7	5810±109.9	0.20/4.00	0.70/6.00	13.40±2.55	10.00±1.47
	10	53	0.43	5787±59.0	5794±67.0	0.40/1.33	0.60/16.33	12.40±2.05	11.50±2.11
1kc	0	62	0.61	6760	6760	0	0	37	37
	2	60	0.41	5943±101.6	6123±141.5	5.40/8.68	2.50/2.02	18.20±2.23	15.90±3.10
	4	59	0.43	5903±127.5	5928±126.8	3.70/4.66	3.50/6.10	14.90±2.95	13.40±1.94
	7	56	0.67	5904±94.7	5782±80.0	1.80/3.02	1.40/3.10	11.80±1.61	10.80±0.81
	10	53	0.86	5823±53.0	5711±43.6	1.70/3.52	1.90/1.96	13.70±2.13	11.00±1.17
5ka	0	69	1.54	8307	8307	0	0	46	46
	2	66	2.50	6956±153.5	7346±236.5	4.70/9.08	6.10/6.80	11.60±6.70	7.20±2.23
	4	57	2.09	6849±82.7	6921±188.8	1.30/7.94	1.80/7.99	7.90±2.12	5.20±1.38
	7	56	2.14	6784±97.8	6733±160.6	1.60/10.39	1.50/5.87	7.10±2.17	4.90±1.37
	10	56	2.08	6760±197.5	6678±88.3	0.90/7.21	1.00/3.97	6.10±1.95	5.10±2.49
5kb	0	57	0.92	7418	7418	0	0	33	33
	2	57	2.02	6881±171.4	6704±217.4	4.20/7.11	2.10/15.12	12.50±6.25	6.10±4.08
	4	56	1.35	6552±158.9	6565±159.8	2.70/6.18	2.50/7.68	5.20±3.93	3.80±1.38
	7	53	1.29	6414±116.3	6464±160.6	1.60/7.64	1.50/7.28	4.50±1.52	3.70±1.31
	10	52	1.28	6305±114.7	6417±129.5	0.80/7.71	1.30/5.96	4.00±1.26	4.10±1.32
5kc	0	59	0.88	7560	7560	0	0	36	36
	2	63	1.19	7165±149.5	7081±149.5	4.70/6.45	3.70/7.65	10.20±3.97	8.00±2.18
	4	59	1.20	6931±159.9	6905±178.2	1.70/5.10	2.40/3.42	8.60±1.91	8.40±2.05
	7	55	1.25	6873±122.4	6803±114.8	1.30/7.94	0.70/7.22	8.50±1.55	7.70±1.35
	10	55	1.28	6846±44.8	6783±111.8	1.20/3.60	0.60/2.33	10.00±2.18	8.50±1.85
10k	0	56	1.86	7774	7774	0	0	21	21
	2	56	2.23	6915±184.6	6908±185.0	2.80/14.25	2.20/8.93	4.90±2.20	5.40±4.10
	4	54	2.28	6865±199.1	6874±163.5	1.30/5.57	2.60/12.57	6.80±2.25	**1.60±0.90**
	7	50	2.29	6695±111.4	6790±147.5	0.80/4.83	1.30/7.91	5.50±3.22	**2.80±1.11**
	10	50	2.41	6682±140.0	6784±78.1	1.00/6.56	1.20/7.22	4.20±1.34	**3.30±1.17**

that by comparing the seconds taken per step in RCAR 0 with respect to any other cardinality of RCAR. This information is in both Tables 1 and 2, and we may see that the HCMC algorithm is, at most, *two times* slower than the usual hill-climber.

In the three works[4] that develop an heuristic algorithm that works directly in the search space of essential graphs, one of them (Chickering, 1996) reports that the algorithm is about 20 times slower than the usual hill-climber in the space of DAGs for a 10000 record dataset sampled from the Alarm network. In (Spirtes and Meek, 1995) only absolute times in a specific platform are provided, thus comparison is not possible. However, both algorithms rely in the transformation of DAG to essential graph, and essential graph to DAG, in order to score the models. Therefore we conjecture that the trade-off of the algorithm in (Spirtes and Meek, 1995) will be similar to the one in (Chickering, 1996). This fact allows us to conclude that our

[4]To the best knowledge of the authors.

algorithm is *an order of magnitude* faster than these two works, yielding the same result.

In the third work (Chickering, 2001), the author provides a set of traversal operators in EG-space that always yield local changes in the score, such that they are as efficient as the usual traversal operators in DAG-space. No experiments in the Alarm dataset are reported thus direct comparison with our approach is not possible as in (Spirtes and Meek, 1995; Chickering, 1996). Nevertheless, the approach in (Chickering, 2001) does not account for the inclusion order and may easily suffer of bad local maxima as in the situation illustrated in Figure 3. Therefore we may assert that our approach is also fully competitive with this latter EG-space based learning algorithm.

7.2 MCMC Learning

When we reviewed the MC^3 algorithm, we saw that the acceptance ratio (6) is in fact the product of two ratios. One is the ratio of the posteriors $p(M'|D)/p(M|D)$, which nicely cancels the normalizing constant (4) and therefore the data D is involved only through the Bayes' factor $p(D|M')/p(D|M)$. The other is the ratio of the cardinalities of the neighborhoods $|\mathcal{N}(G)|/|\mathcal{N}(G')|$ which is known as the candidate-generating ratio. In our experimentation we have assumed a *symmetric* candidate-generating density (Chib and Greenberg, 1995), where $|\mathcal{N}(G)| = |\mathcal{N}(G')|$. This is reasonable in our context since G and G' will differ in one single adjacency.

The eMC^3 algorithm of Figure 7 needs the specification of some Bayesian Network as a starting point (the *init* parameter). It is of general agreement within the MCMC literature that the run of the Markov chain is sometimes sensible to the starting point. Therefore it makes sense to try several runs from several starting points chosen at random.

However, within the context of random generation of acyclic digraphs, Melançon et al. (2000) point out that starting the process on the empty graph gives an effective way of achieving a good mixing rate of the chain. They explain such effect as follows:

> Observe that the maximal distance between any two acyclic digraphs is bounded by $n(n-1)$, since an obvious (but far from optimal) path connecting them goes through the empty graph (by first deleting all edges from the first graph and then adding the edges of the second graph).

We consider that in our context, where the distribution of the DAGs that determine the Bayesian Networks, is not uniform, it still makes sense to follow that advice because one may reason analogously in terms of graphical Markov model inclusion.

In addition, we will consider as good starting points those Bayesian Networks with a high likelihood, as they will be close to the mode of the distribution and therefore it may accelerate convergence of the chain. In concrete, we will use the output of the HCMC algorithm which was the original Alarm network structure of Figure 8 with one arc missed (the one not supported by the data), and to which we will refer as the *almost true* Alarm network. When such starting point is used it will be noted with an asterisk in the legends, next to the name of the neighborhood.

We have ran the $e\mathrm{MC}^3$ algorithm for 10^5 iterations over each of the seven samples of the Alarm dataset, and for the 10000 records sample we ran the chain starting from the almost true Alarm network. We do not provide all the results for every sample, since for some combinations the conclusions are the same.

A first aspect we are going to look at, is the *mobility* of the Markov chain. The mobility is a relevant aspect because the higher the mobility, the lower the chance that the approximated posterior distribution does not reflect an important area of the search space.

Table 3: Mobility of the Markov chain and Kullback-Leibler distance per essential graph

	size	AR	CR	NCR	RCARR	RCARNR
number essential graphs	1k	1764	1622	1632	1898	2017
	5k	830	780	660	991	955
	10k	561	470	526	727	654
	10k*	553	485	612	577	626
K-L per e.g.	1k	4.36525	4.46228	4.37798	4.33639	4.38295
	5k	3.78184	3.78113	3.95602	3.66680	3.75692
	10k	3.73537	4.05203	3.97652	3.53715	3.75776
	10k*	2.70025	2.70035	2.70018	2.70029	2.70046

In Table 3 we may see the averages across the samples and across the RCAR cardinalities of 2,4 and 10 of the different essential graphs visited during the process. We may see as well the average Kullback-Leibler distance (Kullback and Leibler, 1951) per essential graph. It is clear that RCAR yields a higher mobility of the Markov chain since more essential graphs were visited when RCAR was used.

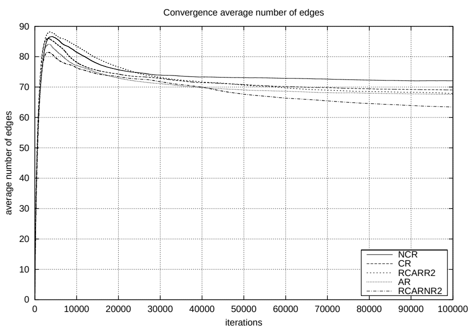
This higher mobility results in a better choice of the DAG Markov models during the process, as we may see from the lower Kullback-Leibler ratios for the cases where RCAR was used. The asterisk in Table 3 denotes the case where we used the output of the HCMC algorithm as starting point. In this latter situation, the Kullback-Leibler ratio does not show a gain while using RCAR. This is because the Markov chain starts from a good point and all the neighboring models to where the chain jumps still provide a good Kullback-Leibler distance.

The second, and most important aspect to assess, is how the RCARNR and RCARR neighborhoods afford a faster rate of convergence to the equilibrium distribution. In order to do so, we will use several convergence diagnostics originally introduced by Giudici and Castelo (2001).

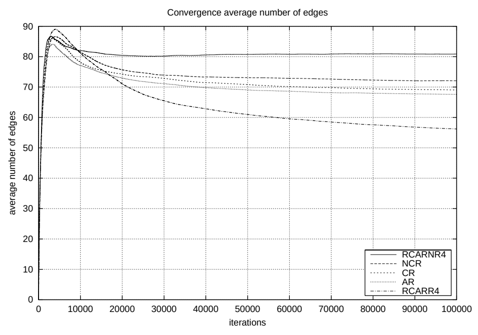
The first convergence diagnostic will be the behavior of the running average number of edges during the run. The rationale behind monitoring this average is that those Bayesian Networks with higher posterior will be sampled more often, and therefore the average number of edges of these networks should be approached by the running average number of edges. If this running average shows some slope at some point of assessment, it is most likely that the Markov chain has not converged at that point.

We have monitored this diagnostic starting from the empty Bayesian Network and using the standard neighborhoods AR, CR and NCR, and the newly introduced RCARNR and RCARR. In the case of these two latter ones, we have ran the experiments with three different cardinalities on the RCAR algorithm (see Figure 5), namely 2,4 and 10.

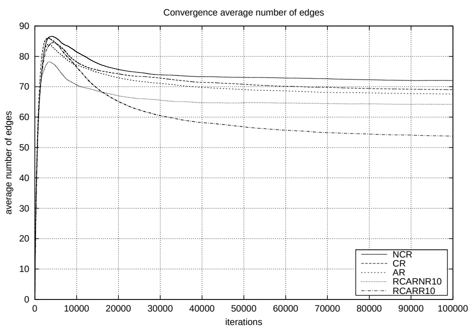
We have plotted the AR, CR and NCR neighborhoods against RCARNR and RCARR, for each of the three cardinalities, 2, 4 and 10. We may see these plots in Figure 9, plots (a),(b) and (c). They correspond to the runs against the 10000 records dataset.



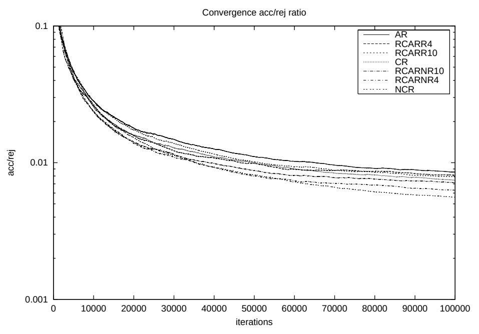
(a)



(b)



(c)



(d)

Figure 9: Convergence average number of edges (a,b,c) and accepts/rejects ratio (d) for the 10k dataset. Legends are ordered with lines.

Recall that the true Alarm network has 46 edges (see Figure 8). Therefore we should expect that the average number of edges converges towards that number. We have ran the Markov chain for 10^5 iterations and we may see in all these three plots that all lines still have some slope downwards at the last iteration. According to this convergence diagnostic, it implies that the chain has not converged. However, it is already possible to observe which approach affords a faster convergence.

In all three plots, the faster convergence is carried out by either a RCARR or a RCARNR neighborhood. We may appreciate that the larger the cardinality of RCAR is, the larger the difference between the use of RCAR and any of the other *non-RCAR* neighborhoods. In the particular case of cardinality 4, although RCARR4 clearly outperforms the rest, RCARNR4 shows the slowest convergence. This may be due to a sequence of jumps that led the Markov chain into an area of local maxima from which is very improbable to escape.

We may see this effect examining the next convergence diagnostic, the ratio of accepts over rejects, in Figure 9d. The RCARNR4 has one of the lowest ratios, only larger than NCR. We did not include RCARR2 and RCARNR2 but they are larger than RCARNR4. Note from this convergence diagnostic that the lines still have some slope, so we can also conclude from this diagnostic that the chain did not converge.

The third convergence diagnostic is to monitor the approximated marginal log-likelihood of the data $\log p(D)$. After swapping terms in Bayes' theorem (3), we may obtain an explicit expression of this marginal:

$$p(D) = \frac{p(D|M)p(M)}{p(M|D)}.$$

Note that this equality holds for any given Bayesian Network M. Obviously, when the posterior $p(M|D)$ is approximated by MCMC, only an approximate marginal likelihood $\hat{p}(D)$ can be obtained. Such an approximation will be better for models within an area of high probability in the posterior distribution. Furthermore, as Kass and Raftery (1995) point out, small likelihoods may have large effects on the final approximation and make the resulting estimator $\hat{p}(D)$ very unstable. This suggests us to compute the approximate marginal likelihood as an average of the approximations from the models with highest posteriors, as follows.

Let $\hat{p}(M|D)$ be the current estimated posterior for model M give data D. Let $p(D|M)$ be the current likelihood of the model M. The marginal likelihood $\hat{p}(D)$ can be estimated as:

$$\hat{p}(D) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} \frac{p(D|M)p(M)}{\hat{p}(M|D)} \quad M \in \mathcal{B}, \tag{9}$$

where $\mathcal{B}$ is a set formed by the Bayesian Networks M with highest posterior at each iteration of the $e\text{MC}^3$ algorithm. In our experiments below, we have chosen a number of five Bayesian Networks to form $\mathcal{B}$.

We may see this convergence diagnostic in Figure 10 for the 10^4 records sample starting from the empty Bayesian Network as well. Here a larger $\log p(D)$ indicates that the Markov chain moves in an area of a higher posterior and therefore affords a faster convergence. Again RCARR and RCARNR outperform CR, NCR and AR neighborhoods. We may observe for the cardinality 4 of RCAR that the slow convergence shown in the previous plot 9b is in agreement with a low $\log p(D)$.

Another interesting fact is in the slope of the curve for the AR neigborhood when we look at the entire length of the Markov chain (left plots). We may appreciate that using the AR neighborhood, $p(D)$ increases faster than any of the others. This means that the AR neighborhood allows to perform larger steps in the search space, but then later seems to get stucked in worse local maxima than using RCARR or RCARNR neighborhoods.

Finally, the last convergence diagnostic corresponds to the posterior distribution of the total number of edges present. More formally, let n be the number of vertices of the DAG that determines the Bayesian Network. Let W be the random variable that takes as value, the number of edges of the DAG at each iteration of the Markov chain. The integer random variable W will take values in the range $[0, 1, 2, \dots, n(n-1)/2)]$, which are all possible cardinalities of the set of edges.

The quantity to monitor is $p(W|D)$, which is computed as in the general case of any quantity of interest Δ (see expression (5)). This posterior distribution will have a normal shape, and it will be centered close to the cardinality of the model for which the Markov chain gives the highest posterior. If the center of the normal shape shifts through longer runs, it means that the Markov chain has not converged.

The fact that $p(W|D)$ takes a normal shape can be explained observing that for each adjacency we may attach a random variable describing the absence or presence of an edge in that adjacency. This random variable can be seen as a Bernouilli distribution with success probability equal to the mean posterior probability of edge presence. If we see W, defined above, as the sum of the $n(n-1)/2$ Bernouilli variables, i.e. following a Binomial distribution, we realize that we are in a particular case of the Central Limit Theorem, where the variable W tends to a normal distribution.

In Figure 11 we have the posterior distribution of the different numbers of edges. In this case, we have started the Markov chain from both the empty DAG model and the almost true Alarm network, which has been noted with an asterisk next to the name of the corresponding neighborhood. In plot 11a we may see the runs for the AR, CR and NCR neigborhoods. Clearly, those that started at the almost true Alarm network show a faster rate of convergence than those that did not.

In plot 11b we may see the runs for cardinalities 4 and 10 of the RCARNR and RCARR neigborhoods. In contrast with the previous case, here two of them, namely RCARR4 and RCARR10, are able to provide distributions quite similar to those of the ones that started in the almost true alarm network.

So far, from all these convergence diagnostics, we can conclude that the RCAR operation improves substantially the convergence rate of the traditional MC^3 algorithm.

A further interesting outcome of our experimentation arises from looking at number of members of each equivalence class of DAG Markov models, visited by the Markov chain during the 10^5 iterations. For comparison, we have computed a lower bound on the size of each equivalence class, as follows.

Let $\mathbf{M}(G)$ be a Bayesian Network. Let $\mathbf{E}(G^*)$ be its equivalent Essential Graph Markov model, i.e. $\mathbf{M}(G) = \mathbf{E}(G^*)$, where G^* is the essential graph representation of G. Let G^* have m connected components, where each of them has ρ_i reversible edges. The lower bound on the number of members of the equivalence class represented by G^* is

$$\prod_{i=1}^{m}(\rho_i + 1).$$

In Figure 12 we may see the plots of these numbers for the runs that started on the almost true Alarm network over the 10000 records sample. In plots (a) and (b) we may see CR and NCR, where in this latter case there are no more than two members visited due to the nature of the non-covered reversal operation.

In plots (c) and (d) we may see that AR visits a similar amount of members per class as the CR neigborhood and RCARR4 can estimate much better the number of members of each equivalence class. This is the empirical evidence of Theorem 3.1, which explains why the RCAR operation enhances both heuristic search and the MCMC method. Note that we visited up to 100 equivalent DAGs using the RCAR4.

The length of Markov chain we have used (10^5 iterations) is long enough as to clearly distinguish among different rates of convergence. However, all the diagnostics show lack of convergence of the chain. We have ran a longer chain for 10^6 iterations starting from the almost true Alarm network. We may see the convergence diagnostics in Figure 13.

In this case we do not include the CR and NCR neighborhoods for comparison. In plot 13a we find the average number of edges during the run. In plot 13b we find the convergence of the marginal of the data. In plot 13c we find the posterior distribution of the cardinalities of the corresponding sets of edges. In all these three diagnostics we may distinguish two groups that converge at a different rate. One formed by AR, RCARR10, RCARR4 and RCARR2, and another by RCARNR10, RCARNR2 and RCARNR4, being the latter the one that shows better convergence.

In a way, this is surprising because, as we had seen so far, RCARR was outperforming RCARNR. Now, it is clear that starting from the almost true Alarm network and running that long, RCARNR converges faster. Nevertheless, this behavior should not be suprising to us, if we would assume that Meek's conjecture (see Conjecture 3.1) holds. As we saw in Theorem 3.1, the ENR neighborhood provides the largest portion of the inclusion boundary among all concepts of neighborhood presented here. Recall that the RCARNR is an approximation of the ENR neighborhood. Thus, in theory we had no reason to believe that any of the other neighborhoods would improve the RCARNR neighborhood. Only we could see empirically that when starting from the empty Alarm network, the RCARR neighborhood converges faster.

We may conclude that when the Markov chain is close to the model with largest likelihood, the non-covered reversal operation in the RCARR neighborhood leads the chain to local maxima that slows down the convergence. We may see in plot 13d that the RCARR neigborhoods have a lower ratio of accepts over rejects, meaning precisely this effect, that there is a very low probability of escaping from where the chain is. This might be happening because the non-covered reversal makes the chain always to jump to a model out of the inclusion boundary (see Theorem 3.1).

We conjecture that probably a better strategy would combine, during the run of the Markov chain, both the RCARR and RCARNR neighborhoods.

Finally, we will take a look at the computational overhead produced by the use of the RCAR operation within the MC^3 algorithm. In Figure 14 we have plotted the average number of iterations (lines 5 through 16 in Figure 7) per second. This plot corresponds to the runs of length 10^5 iterations. In addition to the legend, we have labeled the lines with their corresponding neighborhood and also including the total average ratio of accepts and rejects. A lower ratio speeds up the chain as it is making, on average, less moves.

For clarity we only report comparison between AR, RCARR4 and RCARR10, as the differences are similar for other combinations. Analyzing separately those runs that start from the empty DAG model from those that start from the almost true Alarm network, we may see that in both cases, using the RCAR operation is between two and three times slower than no using it.

This cost is similar to the one we observed for heuristic search, and we consider it to be a very good trade off. Madigan et al. (1996) extended the MC^3 algorithm to work directly in EG-space. They do not show the computational overhead with respect to DAG-space. However, the fact that their Markov chain modifies two adjacencies at a time, to fulfill irreducibility, allows us to conjecture that their method is computationally more expensive than ours.

8 Conclusion

The standard AR neighbourhood for a given DAG is not invariant to other DAGs within the same equivalence class (see Figure 4). In order to solve this problem, we have introduced new concepts of traversal operators for DAGs, based upon the idea to make more moves within the equivalence classes by applying the covered arc reversal operation.

The experiments with RCAR within the MCMC method show a higher mobility of the chain and a faster convergence rate according to the diagnostics, without compromising the already heavy computational cost of the method. Madigan et al. (1996) point out a problem for Markov chains that do not move in EG-space. The approximated posterior may be proportional to the size of the equivalence classes. Obviously RCAR does not solve this problem but there is no guarantee that the moves in the space of essential graphs approximate the posterior well. We believe that the quality of the approximation depends more on the concept of neighbourhood than on the search space used.

We have shown usage of RCAR within the hill climbing heuristic where it brings very impressive results in reasonable time as, e.g. recovering 45 out of the 46 edges of the Alarm network in 48 steps (transformations of single adjacencies). Our work included a few user defined constants (number of covered arc reversals in RCAR, number of repetitions of RCAR in the local maximum of the hill-climber) which could be assigned in a more clever way. We expect that it could bring another improvement of obtained results.

The study of the inclusion order sheds light on the reasons why the approach presented here works well. We proved that a whole class of hill-climber methods, which use any penalized score and any traversal operator satisfying the inclusion boundary condition, always recovers the true underlying model for large data sets. This in fact highlights the extreme relevance, in learning Bayesian Networks, of the (open) problems related to graphical Markov model inclusion, namely:

- Given two DAGs or EGs that differ in more than one adjacency, decide graphically if one of them is included into the other.
- Efficiently generate all models from the ENR neighborhood of one given model.

Acknowledgements

This paper has benefited from discussions with Max Chickering, Steven Gillispie, Finn V. Jensen and Michael Perlman.

References

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

Beinlich, I., Suermondt, H., Chavez, R., and Cooper, G. (1989). The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proc. of the Second European Conference on Artificial Intelligence in Medicine*, pages 247–256. Springer-Verlag.

Bouckaert, R. (1992). Optimizing causal orderings for generating dags from data. In Dubois, D., Wellman, M., D'Ambrosio, B., and Smets, P., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 9–16. Morgan Kaufmann.

Buntine, W. (1991). Theory refinement on bayesian networks. In B. D'Ambrosio, P. S. and Bonissone, P., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann.

Chib, S. and Greenberg, E. (1995). Understanding the metropolis-hastings algorithm. *American Statistician*, 49(4):327–335.

Chickering, D. (1995). A transformational characterization of equivalent bayesian networks. In Besnard, P. and Hanks, S., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 87–98. Morgan Kaufmann.

Chickering, D. (1996). Learning equivalence classes of bayesian network structures. In Horvitz, E. and Jensen, F., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 150–157. Morgan Kaufmann.

Chickering, D. (2001). Learning equivalence classes of bayesian-network structures. Technical report, Microsoft Research.

Chickering, D. (2002). *Personal Communication.*

Chickering, D., Geiger, D., and Heckerman, D. (1995). Learning bayesian networks: Search methods and experimental results. In *Proc. of the Int'l. Workshop on Artificial Intelligence and Statistics*, pages 112–128.

Chickering, D. and Heckerman, D. (1997). Efficient approximations for the marginal likelihood of bayesian networks with hidden variables. *Machine Learning*, 29:181–212.

Chung, K. (1967). *Markov Chains with Stationary Transition Probabilities (2nd ed).* Springer-Verlag.

Cooper, G. and Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–405.

Draper, D. (1995). Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society B*, 57:45–97.

Friedman, N. and Koller, D. (2000). Being bayesian about network structure. In Boutilier, C. and Goldszmidt, M., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*. Morgan Kaufmann.

Frydenberg, M. (1990). The chain graph markov property. *Scandinavian Journal of Statistics*, 17:333–353.

Gillispie, S. and Perlman, M. (2001). Enumerating markov equivalence classes of acyclic digraph models. In Breese, J. and Koller, D., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 171–177. Morgan Kaufmann.

Giudici, P. and Castelo, R. (2001). Improving markov chain monte carlo model search for data mining. *To appear in Machine Learning*.

Hastings, W. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:194–243.

Herskovits, E. (1991). *Computer-Based Probabilistic Network Construction*. PhD thesis, Medical Information Sciences, Stanford University.

Kass, R. and Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association*, 90(430):773–795.

Kočka, T. (2001). *Graphical models - learning and applications*. PhD thesis, University of Prague.

Kočka, T., Bouckaert, R., and Studený, M. (2001). On characterizing inclusion of bayesian networks. In Breese, J. and Koller, D., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 261–268. Morgan Kaufmann.

Kočka, T. and Castelo, R. (2001). Improved learning of bayesian networks. In Breese, J. and Koller, D., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 269–276. Morgan Kaufmann.

Kullback, S. (1959). *Information Theory and Statistics*. Wiley, New York.

Kullback, S. and Leibler, R. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86.

Larrañaga, P., Kuijpers, C., Murga, R., and Yurramendi, Y. (1996). Learning bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE Transactions on System, Man and Cybernetics*, 26(4):487–493.

Lauritzen, S. (1996). *Graphical Models.* Oxford University Press, Oxford.

Lauritzen, S., Dawid, A., Larsen, B., and Leimer, H. (1990). Independence properties of directed markov fields. *Networks*, 20:491–505.

Madigan, D., Andersson, S., Perlman, M., and Volinsky, C. (1996). Bayesian model averaging and model selection for markov equivalence classes of acyclic digraphs. *Communications in Statistics (theory and methods)*, 25(11):2493–2512.

Madigan, D. and York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review*, 63:215–232.

Meek, C. (1997). *Graphical models, selecting causal and statistical models.* PhD thesis, Carnegie Mellon University.

Melançon, G., Dutour, I., and Bousquet-Melou, M. (2000). Random generation of dags for graphs drawing. Technical report, Centrum voor Wiskunde en Informatica.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, California.

Pearl, J. and Verma, T. (1987). The logic of representing dependencies by directed graphs. In *Proc. of the Conf. of the American Association of Artificial Intelligence*, pages 374–379.

Robinson, R. (1973). Counting labeled acyclic digraphs. In Harary, F., editor, *New Directions in the Theory of Graphs*, pages 239–273. Academic Press, New York.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 6:461–464.

Singh, M. and Valtorta, M. (1993). An algorithm for the construction of bayesian network structures from data. In Heckerman, D. and Mamdani, A., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 259–265. Morgan Kaufmann.

Smith, A. and Roberts, G. (1993). Bayesian computation via the gibbs sampler and related markov chain monte carlo methods. *Journal of the Royal Statistical Society B*, 55(1):3–23.

Spirtes, P., Glymour, C., and Scheimes, R. (1993). *Causation, Prediction and Search.* Springer-Verlag, New York.

Spirtes, P. and Meek, C. (1995). Learning bayesian networks with discrete variables from data. In Fayyad, U. and Uthurusamy, R., editors, *Proc. of the Int'l. Conf. on Knowledge Discovery and Data Mining*, pages 294–299, Montreal, Quebec. AAAI Press.

Verma, T. and Pearl, J. (1988). Influence diagrams and d-separation. Technical report, Cognitive Systems Laboratory, UCLA.

Verma, T. and Pearl, J. (1990). Equivalence and synthesis of causal models. In Bonissone, P., Henrion, M., Kanal, L., and Lemmer, J., editors, *Proc. of the Conf. on Uncertainty in Artificial Intelligence*, pages 255–268. Morgan Kaufmann.

entire marginal range | zoomed marginal range

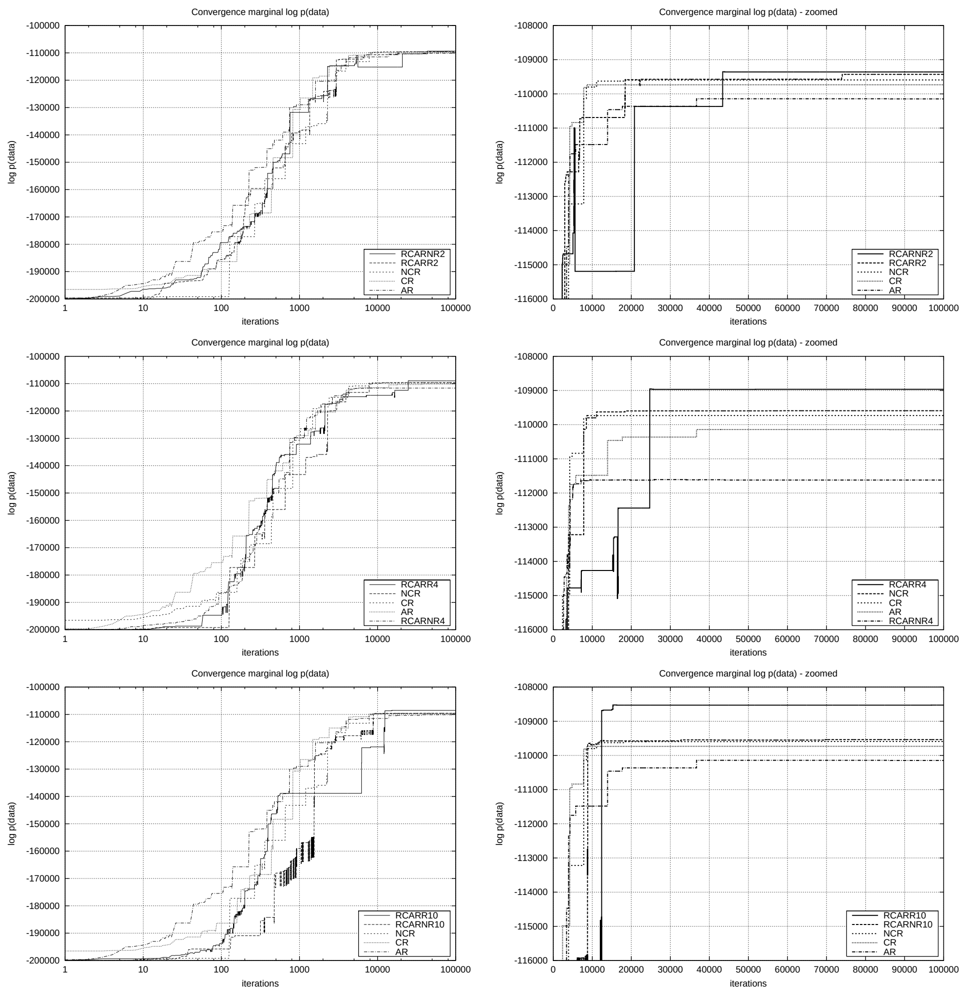
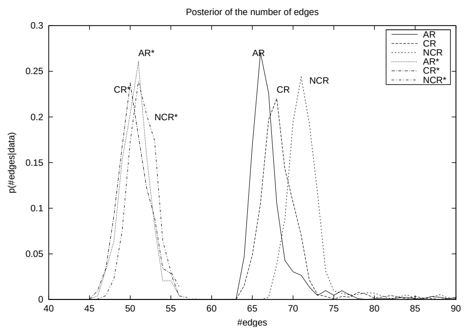
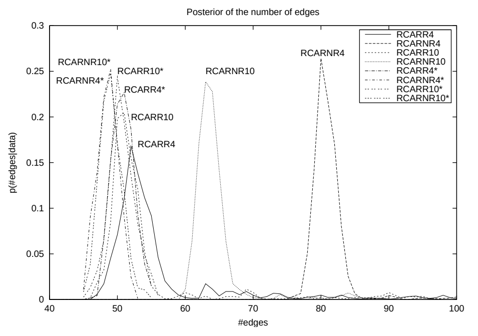


Figure 10: Convergence of the marginal of the data. Comparison between AR, CR, NCR, RCARNR and RCARR. Legend is ordered with lines.

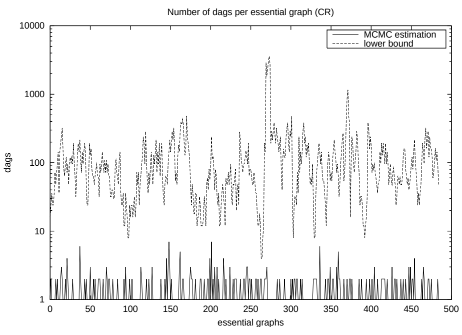


(a)

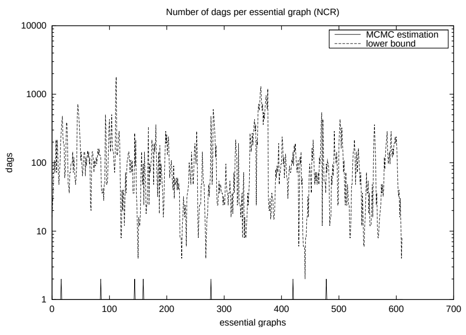


(b)

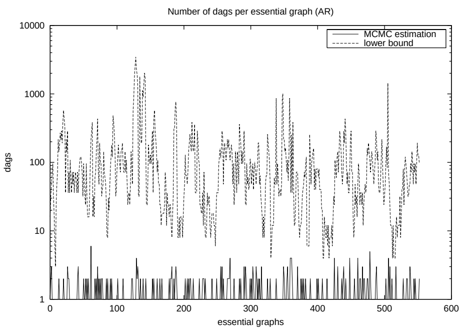
Figure 11: Comparison convergence ability.



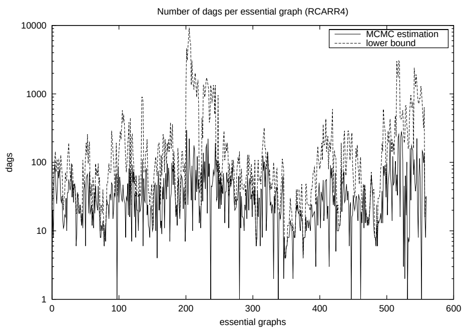
(a)



(b)

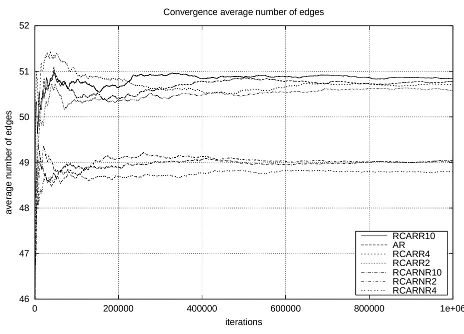


(c)

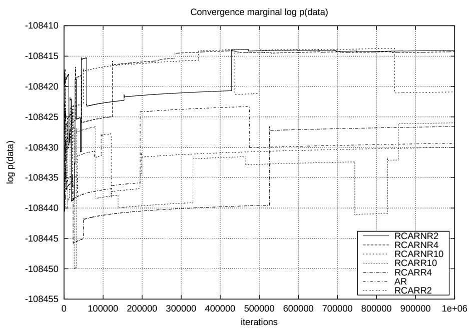


(d)

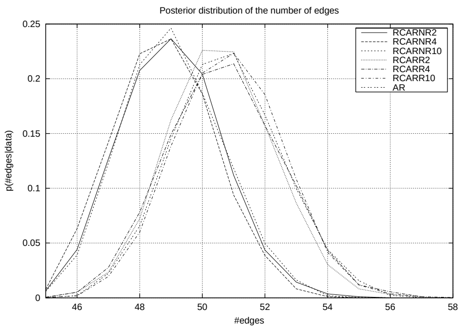
Figure 12: DAGs per essential graph.



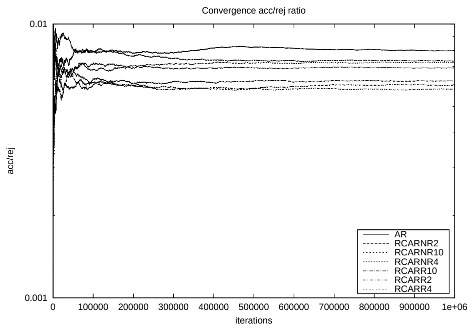
(a)



(b)



(c)



(d)

Figure 13: Convergence diagnostics for a Markov chain of length 10^6 iterations, starting from the almost true Alarm network. Legends are ordered with lines.

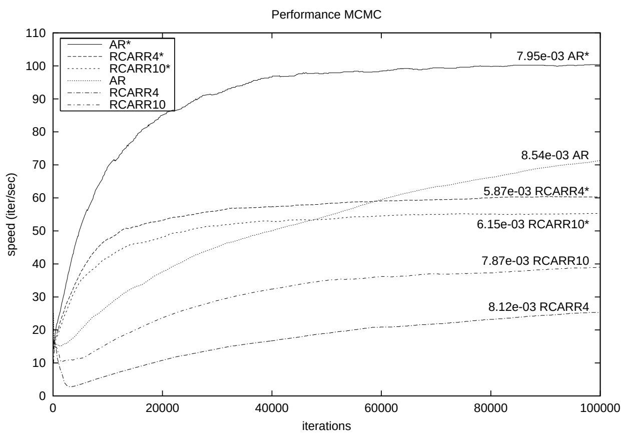


Figure 14: Performance.