Approximate Counting of Graphical Models Via MCMC

Jose M. Peña
Division of Computational Biology, Department of Physics, Chemistry and Biology
Linköping University, SE-58183 Linköping, Sweden
jmp@ifm.liu.se

Abstract

We apply MCMC to approximately calculate (i) the ratio of directed acyclic graph (DAG) models to DAGs for up to 20 nodes, and (ii) the fraction of chain graph (CG) models that are neither undirected graph (UG) models nor DAG models for up to 13 nodes. Our results suggest that, for the numbers of nodes considered, (i) the ratio of DAG models to DAGs is not very low, (ii) the ratio of DAG models to UG models is very high, (iii) the fraction of CG models that are neither UG models nor DAG models is rather high, and (iv) the ratio of CG models to CGs is rather low. Therefore, our results suggest that (i) when learning DAG/CG models, searching the space of DAG/CG models instead of the space of DAGs/CGs can result in a moderate/considerable gain in efficiency, and (ii) learning a CG model instead of an UG model or DAG model can result in a substantially better fit of the learning data.

1 INTRODUCTION

Probably the most common approach to learning directed acyclic graph (DAG) models, also known as Bayesian network models, is that of performing a search in the space of either DAGs or DAG models. In the latter case, DAG models are typically represented as essential graphs (EGs). Knowing the ratio of DAG models to DAGs is a valuable piece of information when deciding which space to search. For instance, if the ratio is low, then one may prefer to search the space of DAG models rather than the space of DAGs, though the latter is usually considered easier to traverse. Unfortunately, while the number of DAGs for a given number of nodes can be computed without enumerating them all (Robinson 1977), the only method for counting DAG models we are aware of is by enumeration. For instance, (Gillispie and Perlman 2002) counted the number of DAG models for up to 10 nodes by enumerating all the EGs by means of a computer program. Counting DAG models by enumerating EGs seems challenging for more than 10 nodes: To enumerate all the EGs over 10 nodes, the computer program in (Gillispie and Perlman 2002) needed 2253 hours in a "mid-1990s-era, midrange minicomputer".

We obviously prefer to know the exact ratio of DAG models to DAGs for a given number of nodes rather than an approximation to it. However, an approximate ratio may be easier to obtain and serve as well as the exact one to decide which space to search. In this paper, we propose a Markov chain Monte Carlo (MCMC) approach to approximately calculate the ratio while avoiding enumerating EGs. Our proposal consists of the following steps. First, we construct a Markov chain (MC) whose stationary distribution is uniform over the space of EGs for the given number of nodes. Then, we sample this stationary distribution and compute the ratio of essential DAGs (EDAGs) to EGs in the sample. Finally, we transform this approximate ratio into the desired approximate ratio of EGs to DAGs. We report the so-obtained approximate ratio for up to 20 nodes. The approximate ratios agree well with the exact ones available in the literature, i.e. for up to 10 nodes (Gillispie and Perlman 2002), and suggest that, for up to 20 nodes, the ratio is not very low. Furthermore, we transform the approximate ratios of EGs to DAGs into approximate ratios of EGs to undirected graphs (UGs) in order to show that, for up to 20 nodes, there seems to be many more DAG models than UG models.

In this paper, we also address the following question. Chain graphs (CGs) include both UGs and DAGs and, thus, they can represent at least as many independence models as UGs and DAGs. Knowing the fraction of CG models that are pure CG (PCG) models, i.e. that are neither UG models nor DAG models, is a valuable

piece of information when deciding which class of graphical models to use. For instance, if the fraction is high, then one may prefer to use CG models rather than UG models or DAG models. The only method for computing the fraction we are aware of is by enumerating CG models and PCG models. However, this seems a challenging approach for more than five nodes: (Volf and Studený 1999) report the ratio of PCG models to CG models for up to five nodes only, though they used a computer program to enumerate all the CG models and detect which are PCG models.

Again, we propose approximating the ratio of PCG models to CG models rather than calculating it exactly to avoid incurring prohibitive computational costs. Specifically, we present a MCMC approach to estimating the ratio while avoiding enumerating CG models. Our proposal is based on the following steps. First, we construct a MC whose stationary distribution is uniform over the space of CG models, represented as largest CGs (LCGs), for a given number of nodes. Then, we sample this stationary distribution and compute the ratio of pure LCGs (PLCGs) to LCGs in the sample. We report the so-obtained approximate ratio for up to 13 nodes. The approximate ratios agree well with the exact ones available in the literature, i.e. for up to five nodes (Volf and Studený 1999), and suggest that, for up to 13 nodes, the ratio of PCG models to CG models is considerably high. Furthermore, we transform the approximate ratios of PLCGs to LCGs into approximate ratios of LCGs to CGs in order to show that, for up to 13 nodes, the ratio of CG models to CGs is very low. Therefore, when learning CG models, one may prefer to search the former space rather than the latter.

The rest of the paper is organized as follows. We review some key concepts in Section 2. Then, we describe our MCMC approach to estimate the ratio of DAG models to DAGs and the ratio of PCG models to CG models in Sections 3 and 4, respectively. Finally, we close with some discussion in Section 5.

2 PRELIMINARIES

The definitions and results in this section are borrowed from (Andersson et al. 1997a, Lauritzen 1996, Volf and Studený 1999). For the sake of readability, we leave undefined some concepts that are not central to this paper and, instead, refer the reader to the original works for their definitions.

A path from X_1 to X_n in a graph G is a sequence of distinct nodes $X_1, \ldots, X_n$ such that there exists an edge in G between every two consecutive nodes in the sequence. A cycle in G is a sequence of nodes $X_1, \ldots, X_n$ such that there exists an edge in G between every two consecutive nodes in the sequence, $X_1, \ldots, X_{n-1}$ are distinct, and $X_n = X_1$. A path from X_1 to X_n in G is called descending if either the undirected edge $X_i - X_{i+1}$ or the directed edge $X_i \to X_{i+1}$ is in G for $i = 1, \ldots, n-1$. A descending path is called directed if it contains at least one directed edge. A path $X_1, \ldots, X_n$ in G is called a complex if the subgraph of G induced by the path looks like $X_1 \to X_2 - X_3 - \ldots - X_{n-2} - X_{n-1} \leftarrow X_n$. A complex is called an immorality if $n = 3$. A chain graph (CG) is a graph containing (possibly) both undirected and directed edges and no directed cycles. An undirected graph (UG) is a CG containing only undirected edges. A directed acyclic graph (DAG) is a CG containing only directed edges.

An independence model is a set of independencies between sets of random variables. The independence model represented by a CG G whose nodes are random variables is the set of independencies of the form $\mathbf{X}$ is independent of $\mathbf{Y}$ given $\mathbf{Z}$ such that $\mathbf{Z}$ blocks all the paths from $\mathbf{X}$ to $\mathbf{Y}$ in the moral graph of the subgraph of G induced by the smallest ancestral set containing $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z}$ (Lauritzen 1996). An independence model is called a CG/UG/DAG model if it can be represented by a CG/UG/DAG. A CG model that is neither an UG model nor a DAG model is called a pure CG (PCG) model.

Two CGs are called equivalent if they represent the same independence model. It has been proven that two CGs are equivalent iff they have the same adjacencies and complexes. Note that this implies that there are not equivalent UGs. Every class of equivalent CGs/DAGs and, thus, every CG/DAG model can be uniquely represented by a CG, known as the largest CG (LCG)/essential graph (EG), such that it contains the directed edge $X \to Y$ iff all the CGs/DAGs in the class contain $X \to Y$, while it contains the undirected edge $X - Y$ iff some but not all the CGs/DAGs in the class contain $X \to Y$. A LCG is called a pure LCG (PLCG) if it represents a PCG model. An EG is called an essential DAG (EDAG) if it is a DAG, i.e. if it represents a class of equivalent DAGs with a single member.

3 APPROXIMATE RATIO OF DAG MODELS TO DAGS

We estimate the ratio of DAG models to DAGs for a given number of nodes by estimating the ratio of EGs to DAGs. For this purpose, we construct a MC over the space of EGs for the given number of nodes with the transition matrix defined as follows:

- Copy the current EG G into H, and choose uni-

formly and perform one of the following seven modifications of H:

- No change: Keep H as it is.
- Add line: Choose two nodes X and Y of H uniformly and with replacement. If X and Y are distinct and non-adjacent in H then add the undirected edge $X - Y$ to H, otherwise keep H as it is.
- Remove line: Choose two nodes X and Y of H uniformly and with replacement. If X and Y are distinct and the undirected edge $X - Y$ is in H then remove it from H, otherwise keep H as it is.
- Add arrow: Choose two nodes X and Y of H uniformly and with replacement. If X and Y are distinct and non-adjacent in H then add the directed edge $X \rightarrow Y$ to H, otherwise keep H as it is.
- Remove arrow: Choose two nodes X and Y of H uniformly and with replacement. If X and Y are distinct and the directed edge $X \rightarrow Y$ is in H then remove it from H, otherwise keep H as it is.
- Add immorality: Choose three nodes X, Y and Z of H uniformly and with replacement. If X, Y and Z are distinct and pairwisely non-adjacent in H then add the immorality $X \rightarrow Y \leftarrow Z$ to H, otherwise keep H as it is.
- Remove immorality: Choose three nodes X, Y and Z of H uniformly and with replacement. If X and Y are distinct and the immorality $X \rightarrow Y \leftarrow Z$ is in H then remove the directed edges $X \rightarrow Y$ and $Z \rightarrow Y$ from H, otherwise keep H as it is.

- If H is an EG then move to H, otherwise stay in G.

To check whether H is an EG, we make use of (Andersson et al. 1997a) which proves that H is an EG iff (i) H is a CG, (ii) all the chain components of H are chordal, (iii) H has no flags, and (iv) all the directed edges in H are strongly protected.

We now show that the MC above is aperiodic, i.e. there exists a strictly positive probability of remaining in the current state, and irreducible, i.e. there exists a strictly positive probability of reaching any state from any other state. These two properties guarantee that the MC has a unique stationary distribution to which it converges when the number of transitions goes to infinity (Häggström 2002). We also show that the uniform distribution is reversible for the MC, i.e. the probability of transition from an EG G to another EG H is the same as the probability of transition from H to G. This implies that the uniform distribution is the stationary distribution of the MC (Häggström 2002). The MC is clearly aperiodic owing to the first modification proposed above. Moreover, for any two EGs G and H there exists a sequence of EGs starting with G and ending with H such that every two consecutive EGs in the sequence differ in exactly one undirected edge, one directed edge, or two directed edges forming an immorality (Andersson et al. 1997a, Perlman 2000). Therefore, any EG can be reached from any other EG by applying the last six modifications proposed above and, thus, the MC is irreducible. If G and H are two EGs such that G has one undirected edge more or less than H, then the probability of transition from G to H equals the probability of choosing the appropriate modification, i.e. $\frac{1}{7}$, times the probability of choosing the appropriate nodes, i.e. $\frac{2}{n^2}$ for n nodes. Clearly, this is the same as the probability of transition from H to G. The same reasoning applies if G has one directed edge or two directed edges forming an immorality more or less than H. Consequently, the uniform distribution is reversible for the MC.

Our MCMC approach to estimating the ratio of EGs to DAGs for a given number of nodes consists in sampling EGs uniformly by sampling the stationary distribution of the MC above and, then, compute the ratio R of EDAGs to EGs in the sample. Since the desired ratio $\frac{\#EGs}{\#DAGs}$ can be expressed as $\frac{\#EDAGs}{\#DAGs}\frac{\#EGs}{\#EDAGs}$, then we can approximate it by $\frac{\#EDAGs}{\#DAGs}\frac{1}{R}$ where $\#DAGs$ and $\#EDAGs$ are computed via the recursive formulae in (Robinson 1977, Steinsky 2003), respectively. Finally, note that, strictly speaking, the EGs in the sample do not really come from the stationary distribution of the MC but from one, hopefully, close to it. Specifically, each of these EGs is obtained as the state of the MC after performing a large number of transitions. The larger the number of transitions performed the better because, as shown above, the MC converges to the stationary distribution when the number of transitions performed goes to infinity.

It is tempting to think that we do not need to construct a MC with the properties above in order to sample an EG uniformly and that, instead, it suffices to choose any EG and randomly perturb it a random number of times. However, the fact that it is impossible to predict the so-obtained EG does not imply that we are sampling an EG uniformly.

3.1 RESULTS

We report in Table 1 the approximate ratios of DAG models to DAGs for up to 20 nodes that we have obtained via the MCMC approach described in the previous section. Each approximate ratio reported is based on

Table 1: Exact and approximate ratios of DAG models to DAGs. Each entry of the table contains, in this order, the exact/approximate ratio of DAG models to DAGs, the exact/approximate ratio of EDAGs to DAG models, and the time required to compute them (in hours).

NODES	EXACT	APPROXIMATE
2	0.66667, 0.50000, 0.0 h	0.66007, 0.50500, 3.5 h
3	0.44000, 0.36364, 0.0 h	0.43704, 0.36610, 5.2 h
4	0.34070, 0.31892, 0.0 h	0.33913, 0.32040, 6.8 h
5	0.29992, 0.29788, 0.0 h	0.30132, 0.29650, 8.0 h
6	0.28238, 0.28667, 0.0 h	0.28118, 0.28790, 9.4 h
7	0.27443, 0.28068, 0.0 h	0.27228, 0.28290, 12.4 h
8	0.27068, 0.27754, 0.0 h	0.26984, 0.27840, 13.8 h
9	0.26888, 0.27590, 7.0 h	0.27124, 0.27350, 16.5 h
10	0.26799, 0.27507, 2253.0 h	0.26690, 0.27620, 18.8 h
11		0.26179, 0.28070, 20.4 h
12		0.26737, 0.27440, 21.9 h
13		0.26098, 0.28090, 23.3 h
14		0.26560, 0.27590, 25.3 h
15		0.27125, 0.27010, 25.6 h
16		0.25777, 0.28420, 27.3 h
17		0.26667, 0.27470, 29.9 h
18		0.25893, 0.28290, 37.4 h
19		0.26901, 0.27230, 38.1 h
20		0.27120, 0.27010, 40.3 h

a sample of 10^4 EGs, each obtained as the state of the MC after performing 10^6 transitions with the empty EG as initial state. For the sake of completeness, we also report the approximate ratios of EDAGs to DAG models obtained. Finally, we also report the time required to obtain each approximate ratio. The time is given in hours and corresponds to a C++ implementation[1] of the MCMC approach described above run on a Pentium 2.4 GHz, 512 MB RAM and Windows 2000. We also report in the table the exact ratios that are available in the literature, i.e. for up to 10 nodes (Gillispie and Perlman 2002). The times reported for the exact ratios are borrowed from that work and correspond to a computer program run on a "mid-1990s-era, midrange minicomputer". Therefore, a direct comparison to our times seems unadvisable.

The first conclusion that we draw from Table 1 is that the approximate ratios are very close to the exact ones. This makes us confident on the accuracy of the approximate ratios for 11 to 20 nodes, where no exact ratios are available in the literature due to the high computational cost involved in calculating them. Another conclusion that we draw from the table is that the ratio for 11 to 20 nodes seems to be between 0.258 and 0.271 and, thus, that it is not very low. This agrees well with (Gillispie and Perlman 2002) where a ratio around 0.267 is conjectured for more than 10 nodes by extrapolating the exact ratios for up to 10 nodes. Therefore, our results suggest that, for up to 20 nodes, the size of the space of DAG models is not smaller than one-forth of the size of the space of DAGs. It is also worth mentioning that the exact and approximate ratios of EDAGs to DAG models in the table indicate that, for up to 20 nodes, the percentage of DAG models that can be represented by a unique DAG is above 27 %, a substantial percentage.

Note that we can obtain approximate numbers of DAG models for up to 20 nodes by just multiplying the approximate ratios of DAG models to DAGs in Table 1 by the numbers of DAGs, which can be computed via the recursive formula in (Robinson 1977). This is equivalent to multiplying the inverse of the approximate ratios of EDAGs to DAG models in the table by the numbers of EDAGs, which can be computed via the recursive formula in (Steinsky 2003). If we now divide these approximate numbers of DAG models by the corresponding number of UG models, which is $2^{\frac{n(n-1)}{2}}$ for n nodes, we obtain approximate ratios of DAG models to UG models for up to 20 nodes. These approximate ratios are compiled in Table 2, together with the exact ratios for up to 10 nodes. From the exact ratios, we can see that DAGs are more expressive than UGs for up to 10 nodes, as they allow representing more independence models. Moreover, for up to 10 nodes, the number of DAG models increases faster with the number

[1] Available at http://www.ifm.liu.se/∼jmp.

Table 2: Exact and approximate ratios of DAG models to UG models.

NODES	EXACT	APPROXIMATE
2	1.00000	0.99010
3	1.37500	1.36575
4	2.89063	2.87726
5	8.57617	8.61615
6	32.5874	32.4486
7	149.017	147.849
8	790.259	787.804
9	4747.80	4789.54
10	31801.1	31670.9
11		229632
12		1.89129×10^{6}
13		1.61273×10^{7}
14		1.54416×10^{8}
15		1.58964×10^{9}
16		1.62424×10^{10}
17		1.91962×10^{11}
18		2.25464×10^{12}
19		2.99079×10^{13}
20		4.05238×10^{14}

of nodes than the number of UG models. According to the approximate ratios obtained, these observations seem to apply for up to 20 nodes. Note, however, that not all the UG models are DAG models: An UG model with UG G is a DAG model iff G is decomposable (Andersson et al. 1997b).

4 APPROXIMATE RATIO OF PCG MODELS TO CG MODELS

We estimate the ratio of PCG models to CG models for a given number of nodes by estimating the ratio of PLCGs to LCGs. For this purpose, we construct a MC over the space of LCGs for the given number of nodes with the transition matrix defined as follows:

- Copy the current LCG G into H, and choose uniformly and perform one of the following seven modifications of H:
  - No change: Keep H as it is.
  - Add line: Choose two nodes X and Y of H uniformly and with replacement. If X and Y are distinct and non-adjacent in H then add the undirected edge $X - Y$ to H, otherwise keep H as it is.
  - Remove line: Choose two nodes X and Y of H uniformly and with replacement. If X and Y are distinct and the undirected edge $X - Y$ is in H then remove it from H, otherwise keep H as it is.
  - Add arrow: Choose two nodes X and Y of H uniformly and with replacement. If X and Y are distinct and non-adjacent in H then add the directed edge $X \rightarrow Y$ to H, otherwise keep H as it is.
  - Remove arrow: Choose two nodes X and Y of H uniformly and with replacement. If X and Y are distinct and the directed edge $X \rightarrow Y$ is in H then remove it from H, otherwise keep H as it is.
  - Add complex: Choose four nodes X, Y, Z and W of H uniformly and with replacement. If X, Y, Z and W are (i) distinct with the only exception that Y and W may be equal, and (ii) neither X and Y nor X and Z nor Z and W are adjacent in H, then add the directed edges $X \rightarrow Y$ and $Z \rightarrow W$ to H, otherwise keep H as it is.
  - Remove complex: Choose four nodes X, Y, Z and W of H uniformly and with replacement. If X, Y, Z and W are (i) distinct with the only exception that Y and W may be equal, (ii) X and Z are non-adjacent in H, and (iii) the directed edges $X \rightarrow Y$ and $Z \rightarrow W$ are in H, then remove these two directed edges from H, otherwise keep H as it is.
- If H is a LCG then move to H, otherwise stay in G.

To check whether H is a LCG, we make use of (Volf and Studený 1999) which proves that H is a LCG iff it is a CG and all its directed edges are protected.

We now show that the MC above is aperiodic and irreducible, and that the uniform distribution is reversible for it. As discussed in Section 3, these three properties guarantee that the MC converges to the uniform distribution when the number of transitions goes to infinity. To see that the MC satisfies the first and third properties, we can apply the same reasoning as in Section 3. To see that the MC satisfies the second property, we resort to Theorem 1 in the appendix which proves that for any two LCGs G and H there exists a sequence of LCGs starting with G and ending with H such that every two consecutive LCGs in the sequence differ in exactly one undirected edge, one directed edge, or two directed edges forming a complex. Therefore, any LCG can be reached from any other LCG by applying the last six modifications proposed above and, thus, the MC is irreducible. Note that the last two modifications described above are more general than adding and removing a complex, as they also

Table 3: Exact and approximate ratios of PCG models to CG models. Each entry of the table contains the exact/approximate fractions of CG models that are, in this order, UG models, DAG models, both UG models and DAG models, PCG models, and the time required to compute them (in hours).

NODES	EXACT	APPROXIMATE
2	1.00000, 1.00000, 1.00000, 0.00000	1.00000, 1.00000, 1.00000, 0.00000, 1.5 h
3	0.72727, 1.00000, 0.72727, 0.00000	0.73600, 1.00000, 0.73600, 0.00000, 1.9 h
4	0.32000, 0.92500, 0.30500, 0.06000	0.32200, 0.92700, 0.31100, 0.06200, 2.3 h
5	0.08890, 0.76239, 0.07136, 0.22007	0.08200, 0.76500, 0.07300, 0.22600, 2.8 h
6		0.02100, 0.56900, 0.01100, 0.42100, 3.4 h
7		0.00500, 0.40200, 0.00100, 0.59400, 4.2 h
8		0.00000, 0.30200, 0.00000, 0.69800, 5.1 h
9		0.00000, 0.19800, 0.00000, 0.80200, 6.4 h
10		0.00000, 0.13700, 0.00000, 0.86300, 8.2 h
11		0.00000, 0.06400, 0.00000, 0.93600, 12.5 h
12		0.00000, 0.05100, 0.00000, 0.94900, 12.9 h
13		0.00000, 0.04100, 0.00000, 0.95900, 19.2 h

allow adding and removing directed edges that neither create nor destroy any complex.

Our MCMC approach to estimating the ratio of PLCGs to LCGs for a given number of nodes consists in sampling LCGs uniformly by sampling the stationary distribution of the MC above and, then, compute the ratio of PLCGs to LCGs in the sample. To check whether a LCG is a PLCG, we make use of (Andersson et al. 1997b) which proves (i) that a LCG represents an UG model iff it has no complexes, and (ii) that a LCG represents a DAG model iff the moral graph of the subgraph of the LCG induced by the closure of each of its chain components is decomposable.

4.1 RESULTS

We report in Table 3 the approximate ratio of PCG models to CG models for up to 13 nodes that we have obtained via the MCMC approach described in the previous section. Each ratio reported is based on a sample of 10^3 LCGs, each obtained as the state of the MC after performing 10^7 transitions with the empty LCG as initial state. For the sake of completeness, we also report the fractions of CG models that are UG models, DAG models, and both UG models and DAG models. Finally, we also report the time required to obtain each approximate ratio. The time is given in hours and corresponds to a C++ implementation[2] of the MCMC approach described above run on a Pentium 2.4 GHz, 512 MB RAM and Windows 2000. We also report in the table the exact ratios that are available in the literature, i.e. for up to five nodes (Volf and Studený 1999). No times are reported in that work.

[2]Available at http://www.ifm.liu.se/∼jmp.

The first conclusion that we draw from Table 3 is that the approximate ratios are very close to the exact ones. This makes us confident on the accuracy of the approximate ratios for six to 13 nodes, where no exact ratios are available in the literature. Another conclusion that we draw from the table is that, for up to 13 nodes, the ratio of UG models to CG models seems to decrease very fast with the number of nodes. For instance, the ratio seems to fall below 0.01 already for seven nodes. On the other hand, the ratio of DAG models to CG models for up to 13 nodes seems to decrease slower with the number of nodes. This was expected, since the results in Section 3.1 indicate that the ratio of DAG models to UG models for up to 13 nodes grows fast with the number of nodes. In any case, the ratio of DAG models to CG models for up to 13 nodes also seems to decrease rather rapidly with the number of nodes. For instance, the ratio seems to be below 0.1 for 11 nodes. In other words, our results suggest that the ratio of PCG models to CG models increases rather fast with the number of nodes and, thus, that CGs are considerably more expressive than UGs and DAGs. For instance, the ratio seems to be above 0.5 already for seven nodes and above 0.9 for 11 nodes. This is a strong argument in favor of using CGs for representing independence models.

Note that we can obtain approximate numbers of CG models for up to 13 nodes by just multiplying the inverse of the approximate ratios of UG models to CG models in Table 3 by the corresponding number of UG models, which is $2^{\frac{n(n-1)}{2}}$ for n nodes. Alternatively, we can multiply the inverse of the approximate ratios of DAG models to CG models in the table by the numbers of DAG models, which are known for up to 10 nodes (Gillispie and Perlman 2002) or can be estima-

Table 4: Exact and approximate ratios of CG models to CGs.

NODES	EXACT	APPROXIMATE
2	0.50000	0.50000
3	0.22000	0.22000
4	0.11848	0.11823
5	0.08077	0.08049
6	0.06414	0.06493
7		0.05690
8		0.04758
9		0.04592
10		0.04213
11		0.05610
12		0.04579
13		0.03541

ted for up to 20 nodes with the help of Table 1 as we have described in Section 3.1. Furthermore, if we now divide the so-obtained approximate numbers of CG models by the corresponding number of CGs, which can be computed via the recursive formula in (Steinsky 2003), we obtain approximate ratios of CG models to CGs for up to 13 nodes. These approximate ratios are compiled in Table 4, together with the exact ratios for up to six nodes (Volf and Studený 1999). Clearly, the ratio decreases with the number of nodes and becomes lower than 0.1 already for five nodes. According to the approximate ratios obtained, the ratio seems to continue decreasing for seven to 10 nodes. For 11 to 13 nodes, the ratio seems to be constant, though this is unclear. Compare these results with those in Table 1, which indicate that the ratio of DAG models to DAGs is higher than 0.25 at least for up to 20 nodes. In summary, our results suggest that, for up to 13 nodes, the size of the space of CG models is much smaller than the size of the space of CGs and thus, when learning CG models, it may pay off to search the former space rather than the latter.

5 DISCUSSION

We have developed a MCMC approach to approximately compute the ratio of DAG models to DAGs and the ratio of PCG models to CG models. We have reported results for up to 20 nodes in the former case and for up to 13 nodes in the latter case. Note that computing the exact ratios is computationally costly and that, as a consequence, they have never been computed for more than 10 nodes in the former case (Gillispie and Perlman 2002) and five in the latter case (Volf and Studený 1999). Therefore, our approach allows us to trade off accuracy for speed. The conclusions that we have drawn from the ratios reported are as follows. For up to 20 nodes, (i) the size of the space of DAG models is not smaller than one-forth of the size of the space of DAGs, and (ii) DAGs can represent many more independence models than UGs. For up to 13 nodes, (i) the size of the space of CG models is much smaller than the size of the space of CGs, and (ii) CGs are much more expressive than UGs and DAGs. In more practical terms, our results suggest that (i) when learning DAG/CG models, searching the space of DAG/CG models instead of the space of DAGs/CGs can result in a moderate/considerable gain in efficiency, and (ii) learning a CG model instead of an UG model or DAG model can result in a significantly better fit of the learning data, given the superior expressiveness of CG models. We plan to develop and evaluate a MCMC algorithm for learning CG models from data by searching the space of LCGs. Specifically, we will consider the modifications presented in Section 4 which, as proven in Theorem 1 in the appendix, describe an aperiodic and irreducible MC. This algorithm will be similar to that proposed in (Andersson et al. 1997a, Madigan et al. 1996) for learning DAG models by searching the space of EGs via the modifications described in Section 3. Another topic for further research consists in improving these two sets of modifications, because they rather often produce a graph that does not belong to the search space. This problem has been previously pointed out by (Perlman 2000) for the modifications in Section 3. Furthermore, (Perlman 2000) presents an alternative set of modifications and claims that they are more efficient. We have not implemented them however, because no results supporting this claim are reported in that work. Moreover, they are more difficult to implement than those in Section 3.

Finally, it is worth mentioning that there exist in the literature some other MCMC approaches to approximate counting that may have been used in this work, e.g. (Häggström 2002, Sinclair 1993). We have not considered them however, because they seem more costly and less accurate than ours.

APPENDIX

The following definitions and results are borrowed from (Volf and Studený 1999). We use the terms lines and arrows as synonyms of undirected and directed edges, respectively. A route from X_1 to X_n in a graph G is a sequence of nodes $X_1, \ldots, X_n$ such that there exists an edge in G between every two consecutive nodes in the sequence. A path is a route involving distinct nodes. A path $X_1, \ldots, X_n$ in G is called a complex if the subgraph of G induced by the path looks like $X_1 \rightarrow X_2 - X_3 - \ldots - X_{n-2} - X_{n-1} \leftarrow X_n$. The arrows $X_1 \rightarrow X_2$ and $X_n \rightarrow X_{n-1}$ are called complex arrows. A descending route from X_1 to X_n in G,

$X_1 \rightsquigarrow X_n$, is a route from X_1 to X_n in G such that either the line $X_i - X_{i+1}$ or the arrow $X_i \rightarrow X_{i+1}$ is in G for $i = 1, \ldots, n-1$. A node X is an ancestor of another node Y in G if there exists a descending route $X \rightsquigarrow Y$ in G. An arrow $X_1 \rightarrow Y_1$ in G covers another arrow $X_2 \rightarrow Y_2$ in G if X_1 is an ancestor of X_2 in G and Y_2 is an ancestor of Y_1 in G. If G is a CG, then an arrow $X_1 \rightarrow Y_1$ in G covers another arrow $X_2 \rightarrow Y_2$ in G iff there exists a descending route $X_1 \rightsquigarrow X_2 \rightarrow Y_2 \rightsquigarrow Y_1$ in G, i.e. a descending route from X_1 to Y_1 containing the arrow $X_2 \rightarrow Y_2$. An arrow is protected in G if it covers a complex arrow in G. Note that all the complex arrows in G are protected in G. A CG is a LCG iff all its arrows are protected. We now prove the following new result, which is similar to Propositions 4.5 and 3.1 in (Andersson et al. 1997a, Perlman 2000), respectively, but for LCGs instead of EGs.

Theorem 1 *For any two LCGs G and H there exists a sequence of LCGs starting with G and ending with H such that every two consecutive LCGs in the sequence differ in exactly one undirected edge, one directed edge, or two directed edges forming a complex.*

Proof: Choose any non-complex arrow $X_1 \rightarrow Y_1$ in G and remove it from G. Call the resulting graph G_1. Clearly, G_1 is a CG. Moreover, all the arrows in G_1 are protected. To see it, consider any non-complex arrow $X_2 \rightarrow Y_2$ in G_1 and note that all the complex arrows in G are complex arrows in G_1. If $X_2 \rightarrow Y_2$ does not cover $X_1 \rightarrow Y_1$ in G, then there exists a complex arrow $X_3 \rightarrow Y_3$ in G such that there exists a descending path $X_2 \rightsquigarrow X_3 \rightarrow Y_3 \rightsquigarrow X_2$ in G that does not include $X_1 \rightarrow Y_1$, because otherwise $X_2 \rightarrow Y_2$ would cover it in G. Therefore, such a descending path exists in G_1 and, thus, $X_2 \rightarrow Y_2$ is protected in G_1. On the other hand, if $X_2 \rightarrow Y_2$ covers $X_1 \rightarrow Y_1$ in G, then there exists a descending path $X_2 \rightsquigarrow X_1 \rightarrow Y_1 \rightsquigarrow Y_2$ in G. Moreover, since $X_1 \rightarrow Y_1$ covers some complex arrow $X_3 \rightarrow Y_3$ in G, the descending path above can be modified into a descending route $X_2 \rightsquigarrow X_1 \rightsquigarrow X_3 \rightarrow Y_3 \rightsquigarrow Y_1 \rightsquigarrow Y_2$ in G that does not contain $X_1 \rightarrow Y_1$. Therefore, such a descending route exists in G_1 and, thus, $X_2 \rightarrow Y_2$ is protected in G_1. Consequently, G_1 is a CG and all its arrows are protected in G_1, which implies that G_1 is a LCG.

Repeat the step above while possible and call the resulting LCG G_2. Note that all the arrows in G_2 are complex arrows. Take the empty graph, which is a LCG, add to it all the lines in G_2, and call the resulting graph G_3. Each line addition produces a LCG because the resulting graph is a CG with no arrows. Take any complex in G_2 that it is not already in G_3 and add it to G_3. Note that this can be accomplished by simply adding to G_3 either the two arrows forming the complex or only one of the arrows if the other is already in G_3. Moreover, the complex addition produces a LCG because the resulting graph is a CG and all its arrows are complex arrows, since no complex previously added to G_3 is destroyed because otherwise it would not be a complex in G_2. Continue adding complexes to G_3 while possible, i.e. until G_3 coincides with G_2. So, we have proven that there exists a sequence of LCGs starting with G and ending with the empty graph such that every two consecutive LCGs in the sequence differ in exactly one undirected edge, one directed edge, or two directed edges forming a complex. Repeating the steps above for H instead of G proves the theorem. □

Acknowledgements

This work is funded by the Swedish Research Council (ref. VR-621-2005-4202). We thank the anonymous referees for their comments.

References

Andersson, S. A., Madigan, D. and Perlman, M. D.: A Characterization of Markov Equivalence Classes for Acyclic Digraphs. The Annals of Statistics 25 (1997a) 505-541.

Andersson, S. A., Madigan, D. and Perlman, M. D.: On the Markov Equivalence of Chain Graphs, Undirected Graphs, and Acyclic Digraphs. Scandinavian Journal of Statistics 24 (1997b) 81-102.

Gillispie, S. B. and Perlman, M. D.: The Size Distribution for Markov Equivalence Classes of Acyclic Digraph Models. Artificial Intelligence 141 (2002) 137-155.

Häggström, O.: Finite Markov Chains and Algorithmic Applications. Cambridge University Press (2002).

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

Madigan, D., Andersson, S. A., Perlman, M. D. and Volinsky, C. T.: Bayesian Model Averaging and Model Selection for Markov Equivalence Classes of Acyclic Digraphs. Communications in Statistics: Theory and Methods 25 (1996) 2493-2520.

Perlman, M. D.: Graphical Model Search Via Essential Graphs. Technical Report 367, University of Washington (2000).

Robinson, R. W.: Counting Unlabeled Acyclic Digraphs. In Proceedings of the Fifth Australian Conference on Combinatorial Mathematics (1977) 28-43.

Sinclair, A.: Algorithms for Random Generation and Counting: A Markov Chain Approach. Birkhäuser (1993).

Steinsky, B.: Enumeration of Labelled Chain Graphs and Labelled Essential Directed Acyclic Graphs. Discrete Mathematics 270 (2003) 266-277.

Volf, M. and Studený, M.: A Graphical Characterization of the Largest Chain Graphs. International Journal of Approximate Reasoning 20 (1999) 209-236.