Machine Learning, 50, 127–158, 2003

Improving Markov Chain Monte Carlo Model Search for Data Mining

PAOLO GIUDICI giudici@unipv.it
Department of Economics and Quantitative Methods, University of Pavia, Via San Felice n. 5, 27100 Pavia, Italy

ROBERT CASTELO roberto@cs.uu.nl
Institute of Information and Computing Sciences, University of Utrecht, P.O. Box 80089, 3508 Utrecht, The Netherlands

Editors: Nando de Freitas, Christophe Andrieu, Arnaud Doucet

Abstract. The motivation of this paper is the application of MCMC model scoring procedures to data mining problems, involving a large number of competing models and other relevant model choice aspects.

To achieve this aim we analyze one of the most popular Markov Chain Monte Carlo methods for structural learning in graphical models, namely, the MC^3 algorithm proposed by D. Madigan and J. York (*International Statistical Review, 63*, 215–232, 1995). Our aim is to improve their algorithm to make it an effective and reliable tool in the field of data mining. In such context, typically highly dimensional in the number of variables, little can be known a priori and, therefore, a good model search algorithm is crucial.

We present and describe in detail our implementation of the MC^3 algorithm, which provides an efficient general framework for computations with both Directed Acyclic Graphical (DAG) models and Undirected Decomposable Models (UDG). We believe that the possibility of commuting easily between the two classes of models constitutes an important asset in data mining, where an a priori knowledge of causal effects is usually difficult to establish.

Furthermore, in order to improve the MC^3 method we propose provide several graphical monitors which can help extracting results and assessing the goodness of the Markov chain Monte Carlo approximation to the posterior distribution of interest.

We apply our proposed methodology first to the well-known coronary heart disease dataset (D. Edwards & T. Havránek, *Biometrika, 72:2*, 339–351, 1985). We then introduce a novel data mining application which concerns market basket analysis.

Keywords: Bayesian structural learning, convergence diagnostics, Dirichlet distribution, market basket analysis, Markov chain Monte Carlo

1. Introduction

A graphical model (see, for instance Lauritzen, 1996), is a family of probability distributions incorporating the conditional independence assumptions represented by a graph. It is constructed by specifying *local dependencies* of each node of the graph in terms of its immediate neighbours.

For high-dimensional discrete random vectors the set of plausible models is large, and a full comparison of all the posterior probabilities associated to the competing models becomes infeasible. A solution to this problem is based on the Markov Chain Monte Carlo method. In the context of graphical models, Madigan and York (1995) adapted the

Metropolis-Hastings sampler for model selection among graphical structures, and call it *Markov Chain Monte Carlo Model Composition* (MC^3 hereafter).

We remark that, while MC^3 addresses the structural learning problem, the reversible jump algorithm, applied to graphical models in Giudici and Green (1999) and Dellaportas and Forster (1999), although certainly more difficult to implement and test, is indeed more suited for problems in which also quantitative learning requires the development of approximate inferences (as occurs, for instance, with incomplete data).

Our motivation here is that, in order to compare alternative MCMC methodologies on actual data mining problems, it is necessary to have good performance measures. We believe the literature on diagnostics for MCMC model search is at the moment not sufficient for this purpose. The aim of the present paper is to give a contribution in this direction.

While Section 2 gives some preliminary background on Bayesian graphical model scoring, Section 3 contains a description of our proposed methodology designed towards an efficient implementation of MCMC model search in the area of data mining.

A further important contribution of the paper lies in Section 4, where we introduce diagnostics to assess the MCMC model selection process. These diagnostics are described by means of two applications on real datasets. In Section 4.1 we analyze the coronary heart disease dataset, and compare our results with those in Madigan and York (1995). In Section 4.2 we introduce a novel data mining application, concerning the search for association structures in market basket analysis. The greater complexity and sparseness of this dataset well illustrates the potential of our methodology. Finally, Section 5 contains some concluding remarks and discussion.

2. Bayesian graphical model scoring

We now briefly review the methodology we employ to perform Bayesian structural learning.

Let a graph g be described by the pair (V, E), where V is a set of vertices, each corresponding to a random variable, and E is the set of edges between such vertices. We shall consider here both undirected edges, such that, if $(a, b) \in E$ also $(b, a) \in E$ and directed edges, for which only one of the previous holds.

For any triplet of disjoint subsets A, B, $S \subset V$ such that A, B are non-empty, one may say that A is conditional independent from B given S in a graph $g = (V, E)$, noted $A \perp\!\!\!\perp B \mid S[g]$ (Dawid, 1979). In order to check whether a conditional independence $A \perp\!\!\!\perp B \mid S[g]$ is represented in g, one follows a given graphical criterion.

In the case of undirected graphs this criterion corresponds to check whether all paths between vertices in A and B intersect S. In such case $A \perp\!\!\!\perp B \mid S[g]$ for an undirected graph g. For directed graphs the notion is less intuitive and we recommend the interested reader to consult Lauritzen et al. (1990) or Pearl and Verma (1987).

Using the previous graphical notion of conditional independence, one may say that a given probability distribution is *Markov with respect to a graph*, if all the conditional probabilities that can be read off the graph hold in the probability distribution. This notion corresponds to the definition of *independency map*, or I-map, from Pearl (1988).

A graphical model is a family of probability distributions P, which are Markov with respect to a graph g. A statement of conditional independence that holds under a given

family of probability distributions P is noted as $A \perp\!\!\!\perp B \mid S[P]$. It is usually clear from the context whether a statement of conditional independence holds under $[g]$ or $[P]$, thus it will be dropped from the notation in such cases. Here we restrict ourselves to two classes of possible graphical models: undirected decomposable graphical models (UDG) and directed acyclic graphical models (DAG).

UDG models are determined by undirected chordal graphs. An undirected chordal graph is an undirected graph without undirected chordless cycles of length greater than three. DAG models are determined by directed graphs that do not contain directed cycles.

Concerning the nature of the random variables considered, we shall concentrate on graphs representing only discrete random variables. Therefore, our object of inference will be a vector of discrete random variables, X_V, whose counts are arranged in a contingency table, $\mathcal{I}$, with cells i. Let $\theta_g = (\theta(i), i \in \mathcal{I})$ indicate the vector of unknown cell probabilities in such table.

Consider a complete sample from P, $X_V = x_V$. Our objective of interest is structural learning. Statistically this means to choose, on the basis of the observed sample, the graph whose independency model best represents the mechanism that generated the data. In data mining, there is typically little a priori knowledge, so one may want to compare *all* possible graphs for a given set of random variables.

Let $p(x \mid \theta_g, g)$ be the likelihood function of a graph g, having observed the evidence $X_V = x_V$ ($X = x$ for short). To carry out model selection, we need to attach a *score* to each considered model. The classical (frequentist) score is obtained as

$$S(g) = \max_{\theta_g} p(x \mid \theta_g, g);$$

models are then typically selected via a stepwise selection. Alternatively, one can use penalized likelihood methods, such as AIC or BIC.

The problem with employing classical scores for model selection in data mining is that, when a large number of variables is considered, stepwise procedures are often very unstable, namely, the chosen model is sensitive to the chosen path (e.g. backward vs. forward).

Furthermore, when there is little subject-matter knowledge on which models are substantially important, as it occurs in data mining, it is advisable to report conclusions from more than one model. A natural way to summarise such conclusions is to take a weighted average of the results, with weights that reflect the importance of each model. This can be easily accomplished in the Bayesian approach, averaging model-specific inferences with the Bayesian model scores as weights. It can be shown that the result is a marginal probability distribution, whose interpretation is straightforward. We remark that classical scores can also be inserted in a Bayesian model averaging procedure, but the result will not typically be a probability distribution.

The Bayesian model score is a (posterior) probability, obtained by applying Bayes' theorem to the marginal likelihood of a model, $p(x \mid g)$, which will be denoted for brevity by $L(g)$:

$$L(g) = p(x \mid g) = \int_{\theta_g} p(x \mid \theta_g, g)\pi(\theta_g)\, d\theta_g,$$

where $\pi(\theta_g)$ is the prior distribution of the parameter vector, conditionally on the graph g.

We now briefly describe how a Bayesian score can be assigned to both DAG and UDG models.

For DAG models, a typical assumption to assess a prior distribution on the unknown cell probabilities is to assign a Dirichlet distribution for each node, conditionally on the configurations of its parents, as described in Heckerman, Geiger, and Chickering (1995), to which we refer for further details. Here we only recall that, following their approach, the marginal likelihood of a discrete DAG model is

$$L(g) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N_{ij} + N'_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + N'_{ijk})}{\Gamma(N'_{ijk})}, \tag{1}$$

where the index i indicates a node in the graph, j a configuration of the parents of such node, and k a realization of the considered node under each of such configurations. Finally, for each combination of the indexes, N_{ijk} indicated the observed count and N'_{ijk} the prior hyperparameter.

Concerning the prior hyperparameters we have assumed complete model equivalence and taken a total prior precision of $N' = 1$ and, therefore, $N'_{ijk} = 1/(ri * qi)$, with q_i the number of configurations of the parents and r_i the number of realizations of the node i. Note that this prior gives equal marginal likelihoods to equivalent DAGs (Buntine, 1991).

On the other hand, the marginal likelihood estimator for UDG models was given by Dawid and Lauritzen (1993), and has the form:

$$L(g) = \frac{\prod_{C \in \mathcal{C}} H(C)}{\prod_{S \in \mathcal{S}} H(S)},$$

where $\mathcal{C}$ is the set of cliques and $\mathcal{S}$ is the set of separators of the chordal graph g. The set of cliques and the set of separators may be obtained using the *maximum cardinality search* algorithm from Tarjan and Yannakakis (1984) and its concrete modification for this particular purpose by Cowell et al. (1999).

For any given clique or separator Q, the computation of $H(Q)$ is as follows (Dawid & Lauritzen, 1993):

$$H(Q) = \frac{\Gamma(N'_Q)}{\Gamma(N + N'_Q)} \prod_{k=1}^{r_Q} \frac{\Gamma(N_k + N'_k)}{\Gamma(N'_k)},$$

where N is the total size of the dataset, r_Q is the number of configurations of the clique or separator Q, N_k are the observed counts for the kth configuration of Q, and finally N'_k and N'_Q are the prior hyperparameters. As in the case of DAG models we have taken an uninformative choice where $N'_k = 1/r_Q$ and $N'_Q = \sum N_k$.

Having assessed a prior distribution over the parameters, the Bayesian model score can be obtained by application of Bayes' theorem. Let $p(x) = \sum_{g \in \mathcal{G}} p(x \mid g) p(g)$ indicate the marginal likelihood of the data, with $\mathcal{G}$ the set of all considered models. Then, for any given graph g, Bayes' theorem gives the score:

$$p(g \mid x) = \frac{p(x \mid g) p(g)}{p(x)}.$$

From the previous expression it appears that, in order to obtain the Bayesian model score for both DAG and UDG models, we need to assign a prior distribution on the model space. A simple assumption, that we have considered here, is to take a uniform distribution over the considered graphs.

We finally remark that alternative choices for the prior on the parameters are discussed, for example, in Dellaportas and Forster (1999). Furthermore, in order to alleviate possible sensitivity of the results to the prior hyperparameters, a hierarchical prior distribution may be considered, as in Giudici and Green (1999).

Alternative priors may also be considered for the model space, without substantially changing our methodology, as the results of the MCMC output can be simply reweighted, following an importance sampling procedure.

3. MCMC graphical model selection

A problem with the Bayesian approach, that has prevented for long time its applicability, is the need to solve the highly dimensional integrals involved in the derivation of the model scores. This has been made possible, at least approximately, by the upsurge of Markov Chain Monte Carlo methods (MCMC, for a review see e.g. Brooks (1998)).

Our aim is to construct a Markov chain that will eventually converge to the posterior distribution of the models given the data. Although the theoretical convergence of the chain will be achieved by construction, it is important to develop diagnostic tools to assess practical convergence.

The MC^3 method of Madigan and York (1995) is a Metropolis-Hastings method which builds a Markov chain that has the posterior distribution $p(g \mid x)$ as its target distribution. It consists in evaluating, at each step of the Markov chain, whether a candidate model g' can replace the current model g, with a specified acceptance probability, that we are now going to derive.

Given a graph g, let $nbd(g)$ be its neighbourhood, that is the collection of graphs that can be reached from g in one step, including itself. The transition probability $q(g, g')$ is equal to 0 for all $g' \notin nbd(g)$ and greater than 0 for all $g' \in nbd(g)$. The neighbourhood $nbd(g)$ must be built in such a way that the transition matrix q, and therefore the Markov chain, are irreducible, i.e. there is a positive probability of going from any point to any other point of the search space.

In both cases of UDG and DAG models, irreducibility can be guaranteed by addition and deletion of (undirected) edges in the corresponding graph g. However, a further test is needed in order to remain within the considered graph space. In the case of DAGs, acylicity of every directed path of g' must be checked. If this condition is satisfied, by adding and removing arcs with a positive probability the transition matrix q fulfills irreducibility, as argued in Madigan and York (1995). In the case of UDGs, it is necessary to test if g' is decomposable. Then Lemma 5 of Frydenberg and Lauritzen (1989) shows that addition and deletion of undirected edges suffices for this purpose.

Once irreducibility is tested, the proposed move is accepted with probability equal to:

$$\alpha = \min\{1, R_a\},$$

where

$$R_a = \frac{\#(nbd(g))p(g' \mid x)}{\#(nbd(g'))p(g \mid x)},$$

where, for a graph g, $\#(nbd(g))$ indicates the cardinality of its neighborhood. Since g and g' will differ in one single adjacency, it is reasonable to assume that the ratio $\#(nbd(g))/\#(nbd(g'))$, known as the proposal ratio, is equal to one for both, UDG and DAG models.

Note that, since the posterior probability of a model, $p(g \mid x)$, is such that $p(g \mid x) \propto p(x \mid g)p(g)$, the acceptance ratio involves the data only through the Bayes factor $p(x \mid g')/p(x \mid g)$. Such Bayes factor can be calculated by local computations, as shown for instance in Heckerman, Geiger, and Chickering (1995).

More precisely, in the case of DAG models the Bayes factor is computed using two terms of the likelihood $L(g)$ in (1):

$$\frac{L(g', i)}{L(g, i)},$$

where the likelihood $L(g, i)$ refers to the likelihood $L(g)$ in (1), but computing the term for the ith variable:

$$L(g, i) = \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N_{ij} + N'_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + N'_{ijk})}{\Gamma(N'_{ijk})},$$

where i refers to the random variable for which the parent set differs in g and g'.

In this paper we propose to use, in addition to the previous types of move, the reversal of arcs for DAGs, when such reversal does not introduce a directed cycle. We shall see later in this section why we propose to use such a move. When the proposed move is a reversal the two parent sets in each g and g' will be different. Therefore, when reversing an arc $j \to i$ in g, the Bayes factor is a function of four likelihood terms:

$$\frac{L(g', i)L(g', j)}{L(g, i)L(g, j)}.$$

In the case of UDG models, Dawid and Lauritzen (1993) showed how to compute the Bayes factor efficiently. Let u–v be the edge in g' that is not in g. Let C be the clique in g' that containing the edge u–v. Let $C_u = C\backslash\{v\}$, $C_v = C\backslash\{u\}$ and $C_0 = C\backslash\{u, v\}$. The Bayes factor of g' against g is computed as

$$\frac{H(C)H(C_0)}{H(C_u)H(C_v)}.$$

Note that, in the opposite case where the edge u–v is in g and not in g', the clique C that contains u–v belongs to g, and the previous ratio is computed by swapping the numerator with the denominator.

We have already remarked that, previously to the computation of the acceptance probability, it is necessary to test whether a local transformation of the current graph belongs to

the considered space of models. We now describe the tests we have implemented, for the DAG and the UDG model spaces.

3.1. Legal moves for DAG models

A directed acyclic graph model (DAG, also known as acyclic digraph ADG), is a directed graph without directed cycles. Thus any move that does not introduce a directed cycle within a given DAG is legal.

In the previous section we already mentioned that additions and deletions of arcs are sufficient to achieve irreducibility. However, as it follows from the next example, arc reversals determine a faster convergence.

Let us consider a space of DAGs over three random variables for which the configuration $a \rightarrow c \leftarrow b$ has a probability of 9/10 which is obviously the highest in such space. Provided that some DAGs are equivalent, when, as first move, we add an arrow to the independence model, this can be added in either direction since both directions are equally probable. If the Markov chain fails to add the arrow in the direction of c, it will be rather difficult to reach the model with the highest probability since we would need to remove the first arrow we added and this movement may be very unlikely because of a strong evidence against the marginal independence between b and c. A fast way to reach the model with the immorality in such a case is by *reversing* arcs.

Motivated by examples as the above, we consider here three types of moves in DAG-space: addition, removal and reversal. Now we will describe an efficient way of testing when these moves are actually *legal* (i.e. they do not introduce directed cycles).

Our intuition may lead us to the simple procedure of looking for directed cycles over the entire graph once we have performed one of these moves. This can be highly inefficient since we would have to follow all possible paths until we find this cycle. Of course, we can restrict the search to the paths involved by the pair of vertices over which we perform the move. However, we can be more efficient by using the ancestor matrix and splitting the computations necessary to move from g to g' in those necessary for *proposing* g' and *moving* to g'.

3.1.1. Proposing. Let $g = (V, E)$ be a DAG with a set of vertices V, where $|V| = n$, and set of edges (arcs) E. Let I be its *incidence matrix* with a row for every vertex, which has n columns. Given $v_i, v_j \in V$, an entry $I(i, j)$, where $1 \leq i, j \leq n$, of this matrix is set to 1 if the arrow $v_j \rightarrow v_i \in E$, and 0 otherwise.

The *ancestor matrix* A is also a $n \times n$ matrix where an entry $A(i, j)$ is set to 1 if E contains a directed path from the vertex v_j to the vertex v_i (i.e. v_j is ancestor of v_i), and 0 otherwise.

- *Addition*. When we consider the addition of an arc $v_j \rightarrow v_i$, this addition is legal if the entry $A(j, i)$ is 0. If the entry $A(j, i)$ was 1, it would mean that v_i is ancestor of v_j and therefore there would exist a directed path from v_i to v_j which becomes a directed cycle when we add $v_j \rightarrow v_i$.
- *Removal.* The removal of an arc cannot introduce any directed cycle, thus it is always legal.

– *Reversal.* The reversal of an arc is a two-step move, in which first we remove the arc, and then we add it in the opposite direction. The first step poses no problem since removals are always legal, but the second step can introduce a directed cycle since it is an addition.

  Let $v_i \rightarrow v_j$ be the arc we want to reverse. If this reversal is not legal, the addition of the arc $v_i \leftarrow v_j$, after the removal of the former one, will introduce a directed path from v_i to v_j. We cannot detect this by using the ancestor matrix A in the same way we did in the case of addition because for the v_j row, this matrix describes the ancestors inherited also by the vertex v_i which now are not part of its *parents* anymore.

  If this directed path from v_i to v_j exists, it means that v_i was ancestor of v_j by some directed path that goes through the parent set of vertices of v_j. Therefore it suffices to check the ancestorship of v_i through every vertex v_k that is parent of v_j, with $k \neq i$.

  The search space of DAGs is organized in equivalence classes, and that was the reason, in the previous section, to decide to use a model scoring function that gives equal scores to equivalent DAGs. Equivalent DAGs are characterized as those with the same undirected skeleton and the same immoralities (Verma and Pearl, 1990). They differ in certain edges that can be reversed without changing the independency model that they represent. Those edges that can be reversed at the same time in a given graph are known as *covered*. The term *covered edge* was defined in Chickering (1995) as an edge $a \rightarrow b$ for which $pa(b) = a \cup pa(a)$, where $pa(a)$ and $pa(b)$ stand for the sets of parent vertices of a and b, respectively.

  From the previous definitions it follows that by reversing arcs the Markov chain may directly move to a DAG within the same equivalence class. Medigan et al. (1996) point out that one of the drawbacks of using a Markov chain in the space of DAGs is that the chain may visit equivalence classes proportionally to their size (in terms of how many DAG members they may have). In order to alleviate such problem when using arc reversal we introduce here the *non-covered arc reversal* (NCR), which consists of restricting reversals to *non-covered* arcs. Thus, here we will consider legal only non-covered arc reversals.

To decide whether a move is legal using an ancestor matrix is of a cost $\mathcal{O}(1)$ for addition, and $\mathcal{O}(n)$ for reversal. This efficiency is achieved because the computational costs necessary to search for directed paths are turned into the costs necessary to update the ancestor matrix. However, the latter needs to be carried out only after the Markov chain decides (accepts) which move to perform. This is the key feature of our algorithm, in which the overheads for banning illegal moves are lowered to the minimum during move proposal.

The optimality of this acylicity testing overhead for move proposal follows from the fact that, for additions cost cannot be lower than $\mathcal{O}(1)$, i.e. constant. For reversals, a cost of $\mathcal{O}(n)$, i.e. linear in the number of vertices, is the lowest possible because the current ancestor information of the former sink vertex cannot be used. Furthermore, if the reversal introduces a cycle, the former source vertex must have, before the reversal, a directed path to the former sink vertex, through any of the other parents of this former sink; therefore, ancestorship of the former source should be checked for each of those parents.

3.1.2. Moving. Once the Markov chain accepts a move, this has to be performed, which actually means to update the incidence matrix I, as well as the ancestor matrix A. When we add an arc $v_i \rightarrow v_j$, we have to set $I(j, i) = 1$, when we remove this same arc, the

update is accomplished setting $I(j, i) = 0$ and, finally, when we reverse the arc $v_i \rightarrow v_j$ to $v_i \leftarrow v_j$, we have to update two entries, $I(j, i) = 0$ and $I(i, j) = 1$.

To update the ancestor matrix A, it may be required to modify one or more rows, as follows.

- *Addition.* Let $v_i \rightarrow v_j$ be the added arc. In the first place we have to set v_i as ancestor in v_j. In the second place we have to add all the the ancestors of v_i as ancestors of v_j. And, finally, the ancestors of v_j must be added to the ancestors of the descendants of v_j. The descendants of v_j are all those vertices that have v_j as ancestor (i.e. $A(k, j)$ is $1, k = 1, \ldots, n$). Since we keep an ancestor matrix the latter step will be performed at most $n - 1$ times, therefore the cost of the whole operation is of an order linear in the number of vertices, i.e. $\mathcal{O}(n)$.
- *Removal.* Let $v_i \rightarrow v_j$ be the removed arc. Ancestors cannot be propagated as in the previous case, but they have to be rebuilt for the sink vertex v_j and all the descendants of v_j in topological order. In the first place, the ancestors of v_j are set to the parents of v_j, i.e. is the updated row of v_j in the incidence matrix, $I(j)$. In the second place the ancestors of every parent vertex of v_j are added to the ancestors of v_j. Now, by inspecting the incidence matrix I we go through all the descendants of v_j in topological order, and for each of them we rebuild the corresponding row in the ancestor matrix in the same way we did with v_j but with respect to the corresponding parent set. Therefore the cost is, at most, of order $\mathcal{O}(n^2)$.
- *Reversal.* Let $v_i \rightarrow v_j$ be the arc to be reversed to $v_i \leftarrow v_j$. In the first place we update a removal $v_i \rightarrow v_j$, and in the second place we update an addition $v_i \leftarrow v_j$.

It appears that, in our algorithm, the procedures that require to search through the paths of the DAG are to update ancestors from a removal move and from a reversal move. However, they can be efficiently implemented with the use of bitwise operations. Furthermore, they are only performed when it is strictly necessary, namely, after having accepted the move.

3.2. Legal moves for UDG models

Legal additions of edges for chordal graphs that determine UDG models were characterized by Giudici and Green (1999), while legal removals were characterized by Frydenberg and Lauritzen (1989). A chordal graph has a representation as a junction forest. A junction forest is a collection of junction trees. A junction tree, in this context, is the organization of the cliques of a chordal graph into a tree. In this tree there are two types of nodes, those representing cliques, and those representing separators. Clique nodes are never adjacent in the tree and always have a separator node in between which corresponds to the non-empty intersection of vertices between the two cliques. For more insight into junction forests of UDG models the reader may consult Lauritzen (1996) or Cowell et al. (1999).

3.2.1. Proposing. Let $g = (V, E)$ be an UDG with a set of vertices V, where $|V| = n$, and a set of undirected edges E such that g is chordal. A *path* in an undirected graph between two vertices $a, b \in V$, is a sequence of vertices $a = v_0, \ldots, b = v_k$ such that $(v_i, v_{i+1}) \in E$.

A *connected component* in g is an induced subgraph $g_S = (V_S, E_S)$ for which there are paths between all pair of vertices in V_S.

– Addition (Giudici & Green, 1999). Let $v_i, v_j \in V$ be two non-adjacent vertices. The addition of an edge between v_i and v_j is legal if and only if either:
  - v_i and v_j belong to different connected components.
  - v_i and v_j belong to the same connected component, and there exists a path in the associated junction forest, between a clique C_i, that contains v_i, and a clique C_j, that contains v_j, such that $C_i \cap C_j$ is a separator in that path.

– Removal (Frydenberg & Lauritzen, 1989). Let $v_i, v_j \in V$ be two adjacent vertices. The removal of the edge between v_i and v_j is legal if and only if this edge belongs to a single clique.

3.2.2. Moving. A *move* in the context of UDG models means to update the associated junction forest according to the new organization of the cliques. For this purpose, we have used the maximum cardinality search from Tarjan and Yannakakis (1984) in the particular form given by Cowell et al. (1999). Although the complexity of this algorithm is of order $\mathcal{O}(|V|+|E|)$ we remark again that moving is a much less frequent operation than proposing.

4. Evaluation of the performance of the method

In order to evaluate the performance of the proposed method, we shall first consider a dataset, already analyzed in the literature: the *Coronary heart disease* (CHD) dataset, which will be used to test and compare our MCMC method over both the DAG and the UDG model space, with analytic methods, such as the Occam's razor (Madigan & Raftery, 1994). We then consider a novel application in data mining, concerning market basket analysis.

4.1. Coronary heart disease dataset

This set of data, well known in the statistical literature, was introduced by Edwards and Havránek (1985). It concerns 1, 841 cross-classified men, according to six binary coronary heart disease factors. For a description of the problem and the dataset we also refer to Madigan and Raftery (1994) who analyzed it using Bayesian discrete graphical models and a model scoring algorithm based on Occam's razor. The random variables of interest are:

A smoking
B strenuous mental work
C strenuous physical work
D systolic blood pressure
E ratio of β and α proteins
F family anamnesis of coronary heart disease

The aim of the analysis is to investigate the association structure among such risk factors. Madigan & Raftery (1994) consider model search over the spaces of UDG and DAG

models. Madigan et al. (1996) extend the analysis considering model search over Markov equivalence classes of DAG models. Dellaportas and Forster (1999) consider a reversible jump MCMC algorithm over the broader class of hierarchical loglinear models.

Recall that, in the decomposable case, the number of possible UDG models for six variables corresponds to the number of chordal graphs on six vertices, which is 18,154 (Wormald, 1985). In the directed case (DAG), the number of graphical structures, in this case acyclic digraphs, amounts to 3,781,503 (Robinson, 1973), but since their Markov interpretation organizes them in equivalence classes, represented by essential graphs, the real number of different statistical models for six variables is smaller, precisely 1,067,825 (Gillispie & Perlman 2001).

In all of our experiments on this dataset we have considered a run length of $n = 100{,}000$ iterations, without burn-in, and starting from the independence model.

4.1.1. UDG model selection. We first consider model selection over UDG models. Figure 1 presents three performance monitors we propose, to assess the degree of convergence of the Markov chain. Such diagnostics, which we call dynamic, as they are calculated cumulatively along the Markov chain, are (a) the ratio between the number of accepted and rejected candidate models (b) the approximated marginal likelihood of the data and (c) the average number of edges present.

In order to calculate an approximation of the marginal likelihood of the data note that, Bayes' theorem can be used as follows:

$$p(x) = \frac{p(x \mid g)p(g)}{p(g \mid x)}.$$

Note that the previous equality holds for any given graph g. Obviously, when the posterior $p(g \mid x)$ is approximated by MCMC, only an approximate marginal likelihood $\hat{p}(x)$ can be calculated. Such an approximation is better for graphs which have a high posterior probability. Furthermore, as Kass and Raftery (1995) point out, small likelihoods may have large effects on the final approximation and make the resulting estimator $\hat{p}(x)$ very unstable. This suggest to compute the approximate marginal likelihood as an average of the approximations from the best graphs, as follows.

Let $\hat{p}(g \mid x)$ be the current estimated posterior for model g given data x. Let $p(x \mid g)$ be the current likelihood of the model g. The marginal likelihood $\hat{p}(x)$ can be estimated as:

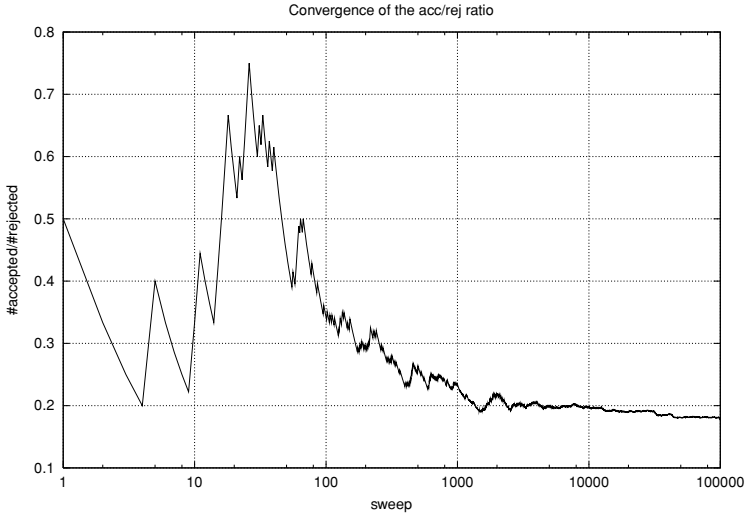
$$\hat{p}(x) = \frac{1}{|\mathcal{B}|} \sum_{\mathcal{B}} \frac{p(x \mid g)p(g)}{\hat{p}(g \mid x)} \quad g \in \mathcal{B},$$

where $\mathcal{B}$ is a set formed by the models g with highest posterior at each stage of the MCMC procedure. In our applications, the chosen number of best graphs in $\mathcal{B}$ has been taken equal to five.

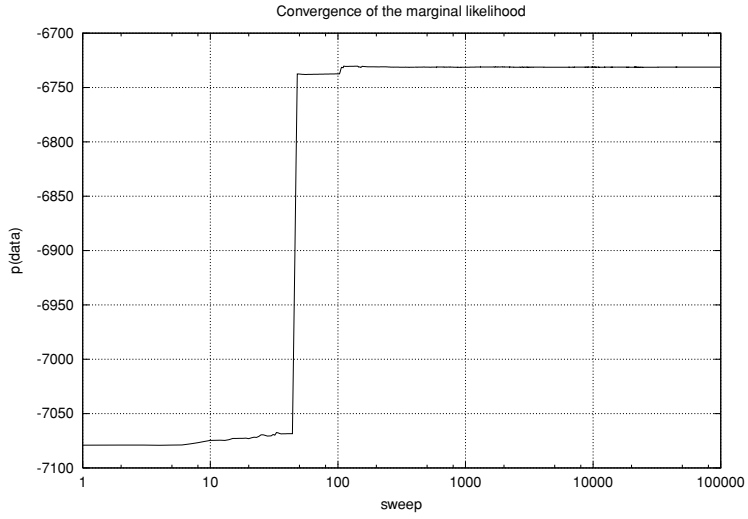
From figure 1 note that all graphs exhibit a long run stability of the monitored quantities and, therefore, give an indication of clear convergence of the proposed MCMC algorithm, well before the end of the simulation.

On the other hand, figures 2 and 3 present what we define static performance monitors of the output, as they are calculated only once, at the end of simulation. Such measures convey the main summary information on structural learning that can be extracted from the MCMC output.

More precisely, we have in figure 2(a) the posterior distribution of the models, arranged in the order in which the models were visited the first time by the Markov chain, and



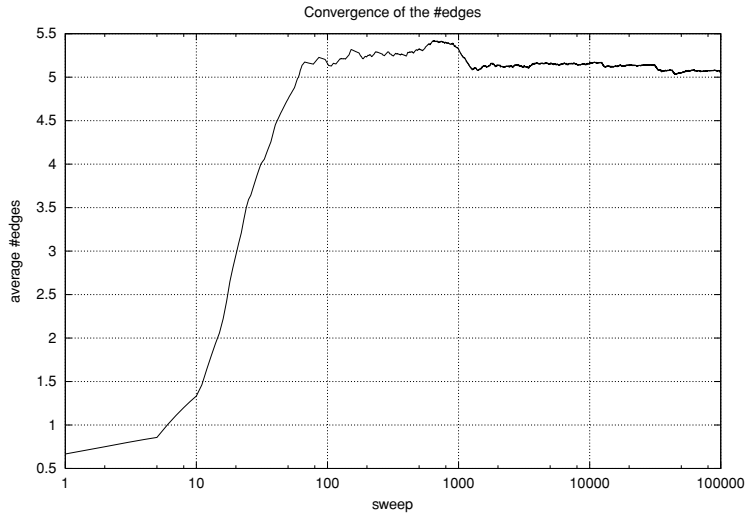
(a)



(b)

Figure 1. Dynamic diagnostics for UDG models in the CHD dataset.

(*Continued on next page*.)



(c)

Figure 1. (*Continued*).

in figure 2(b) the cumulative distribution of the models, proceeding from the most to the least probable. Both graphs give, from a different perspective, a description of the posterior distribution.

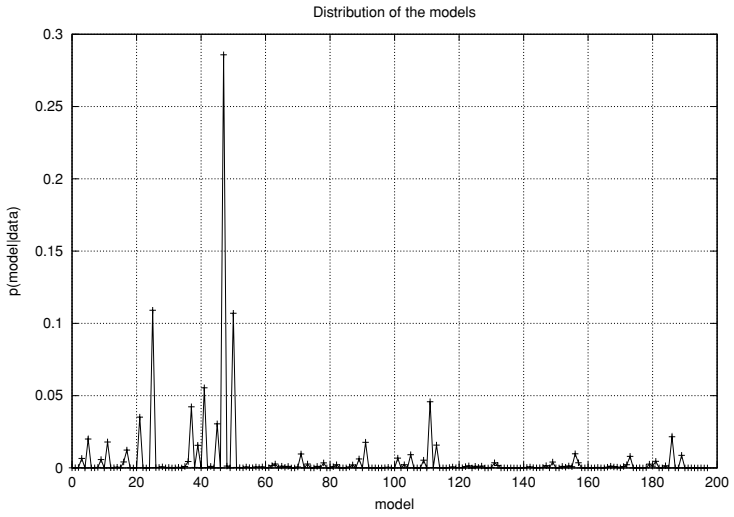
The plot 3b reports the most important information, from an interpretational viewpoint, namely, the posterior distribution of presence of each edge. Such a plot gives clear indication on the data strength of each arc, as it reports the posterior probability of presence for each arc given the data. The x-axis contains the specified edges, and the y-axis the probabilities.

Finally, the plot 3a reports the posterior distribution of the total number of edges present, which can be taken as a measure of model complexity. Note that the posterior distribution is not far from having a normal shape around the value of 5. This can be explained observing that the random variable describing the absence/presence of any additional edge can be seen as a bernoulli distribution, with success probability, say p, equal to the mean posterior probability of edge presence. Therefore, the number of edges, say n, present can be approximately thought as a binomial distribution with parameters (n, p) which, for n large can be well approximated by a Gaussian distribution.

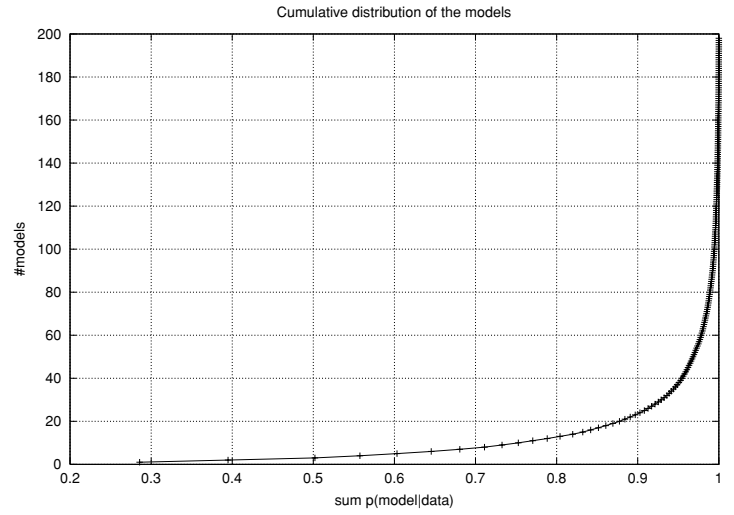
From the plots of figure 2, notice that there is a strong concentration of the posterior distribution. The number of graphs that emerge as being most supported by the data appear to be at most 3.

From the analysis of plot 3b it turns out that the highest probability of presence (0.99) is associated to the edges (B, C) and (A, C), followed by (A, E) (0.83), (C, E) (0.77), (D, E) (0.71) and, finally, by (A, D) (0.37). The remaining estimated posterior probabilities of arc presence are very low. There is a clear marginal independence of variable F.

To conclude with the information we can extract from UDG models, let us examine the posterior distributions of the conditional independencies. The posterior probability of a



(a)



(b)

Figure 2. Static diagnostics for UDG models in the CHD dataset (part I).

given conditional independence statement is computed by summing over those UDG models that contain such statement represented in the graph. The fact that the graphs that determine UDG models are undirected, makes this summation easy to carry out. It is only necessary to take into account the different subsets of vertices (conditioning sets) that intersect the paths from a given vertex to the rest of non-adjacent vertices. We may see the posteriors that account for more than 95% of the probability distribution in Table 1.

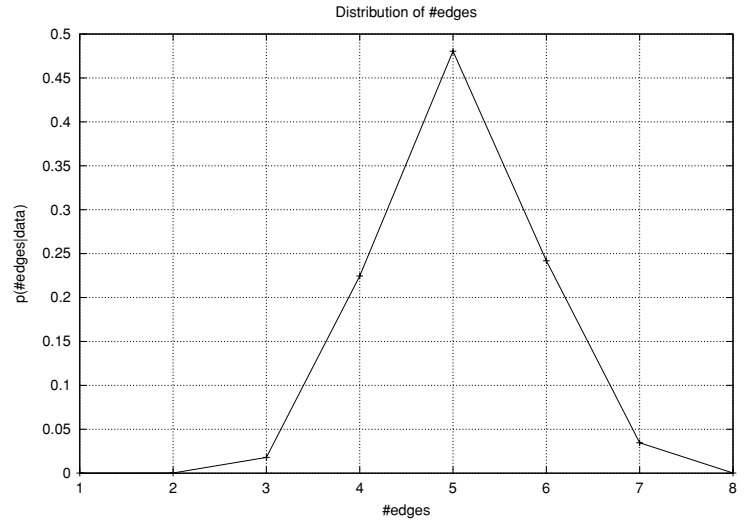
Table 1. Posterior distributions of conditional independencies.

A	$p(\perp\!\!\!\perp \mid \text{data})$	B	$p(\perp\!\!\!\perp \mid \text{data})$
$\perp\!\!\!\perp BDF \mid CE$	0.48	$\perp\!\!\!\perp ADEF \mid C$	0.78
$\perp\!\!\!\perp BF \mid CDE$	0.32	$\perp\!\!\!\perp ADE \mid CF$	0.11
$\perp\!\!\!\perp BDEF \mid C$	0.12	$\perp\!\!\!\perp ADF \mid CE$	0.09
$\perp\!\!\!\perp BEF \mid CD$	0.04	$\perp\!\!\!\perp AD \mid CEF$	0.01
C	$p(\perp\!\!\!\perp \mid \text{data})$	D	$p(\perp\!\!\!\perp \mid \text{data})$
$\perp\!\!\!\perp DF \mid ABE$	0.76	$\perp\!\!\!\perp ABCF \mid E$	0.54
$\perp\!\!\!\perp DEF \mid AB$	0.22	$\perp\!\!\!\perp BCEF \mid A$	0.20
		$\perp\!\!\!\perp ABCEF \mid AE$	0.16
		$\perp\!\!\!\perp ABC \mid \emptyset$	0.07
E	$p(\perp\!\!\!\perp \mid \text{data})$	F	$p(\perp\!\!\!\perp \mid \text{data})$
$\perp\!\!\!\perp BF \mid ACD$	0.49	$\perp\!\!\!\perp ABCDE \mid \emptyset$	0.76
$\perp\!\!\!\perp BDF \mid AC$	0.18	$\perp\!\!\!\perp ACDE \mid B$	0.12
$\perp\!\!\!\perp BCF \mid AD$	0.07	$\perp\!\!\!\perp ABCD \mid E$	0.05
$\perp\!\!\!\perp ACF \mid BD$	0.07	$\perp\!\!\!\perp ABCE \mid D$	0.02
$\perp\!\!\!\perp ABF \mid CD$	0.04	$\perp\!\!\!\perp BCDE \mid A$	0.02
$\perp\!\!\!\perp BCDF \mid A$	0.03	$\perp\!\!\!\perp ABDE \mid C$	0.01
$\perp\!\!\!\perp B \mid ACDF$	0.03		
$\perp\!\!\!\perp ACDF \mid B$	0.03		
$\perp\!\!\!\perp ABDF \mid C$	0.02		

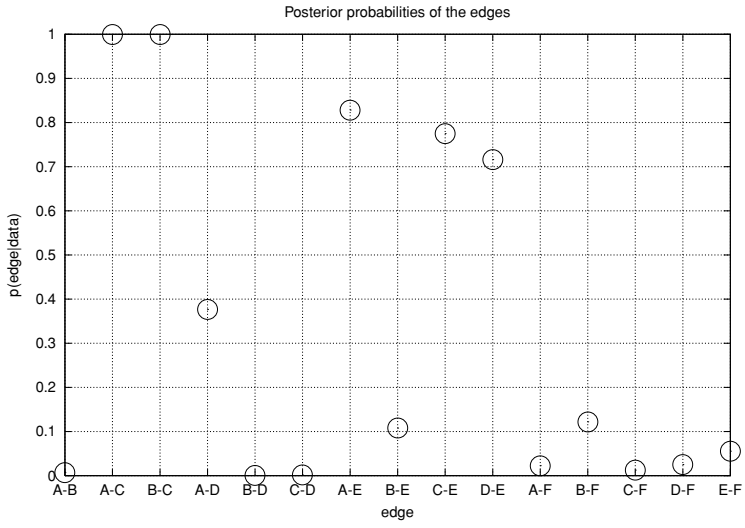
Madigan et al. (1996) show some of these posteriors as well, computed using the posterior distribution of Markov equivalence classes of DAG models. Specifically, their method assigns probability 0.77 to $F \perp\!\!\!\perp ABCDE \mid \emptyset$, 0.60 to $D \perp\!\!\!\perp ABC \mid E$ and 0.35 to $B \perp\!\!\!\perp E \mid C$. From the properties of conditional independence (Pearl, 1988), it follows that $D \perp\!\!\!\perp ABCF \mid E \Rightarrow D \perp\!\!\!\perp ABC \mid E$ and that $B \perp\!\!\!\perp ADEF \mid C \Rightarrow B \perp\!\!\!\perp E \mid C$. As we see from Table 1, these statements have here the probabilities 0.76, 0.54 and 0.78 respectively. Note that the first two have very similar values to those from Madigan et al. (1996), although they have been computed from quite a different type of graphical model.

4.1.2. DAG model selection. We now consider the analysis of the same dataset in the space of DAG models. Figures 4 and 5 present the performance indicators we have discussed previously, respectively dynamic and static. Instead of the probabilities of edge presence, as those in figure 3(b), we report in Table 2 the most probable parent configurations for each vertex.

In Section 2 we pointed out that we are using prior hyperparameters that make model scores equal between DAG models belonging to the same equivalence class. Under such constraint it is sensible to consider the posterior distribution of essential graphs instead of the posterior distribution of DAGs. In order to obtain the posterior distribution of essential



(a)



(b)

Figure 3. Static diagnostics for UDG models in the CHD dataset (part II).

graphs we have transformed each DAG into its corresponding essential graph by using the algorithm of Chickering (1995) and we have summed up probabilities of DAGs having the same essential graph representation, that is, belonging to the same equivalence class.

From the dynamic measures in figure 4 we can conclude that there is indication of good convergence of the MCMC output. From the static measures in figure 5 note that the

Table 2. Posterior distributions of parent sets.

A	$p(\text{pa}(A) \mid \text{data})$	B	$p(\text{pa}(B) \mid \text{data})$
C E	0.62	C	0.45
C	0.20	Ø	0.28
Ø	0.11	E	0.20
E	0.03	F	0.05
C	$p(\text{pa}(C) \mid \text{data})$	D	$p(\text{pa}(D) \mid \text{data})$
B	0.51	E	0.40
Ø	0.21	A E	0.30
A	0.10	A	0.17
E	0.08	Ø	0.12
E	$p(\text{pa}(E) \mid \text{data})$	F	$p(\text{pa}(F) \mid \text{data})$
Ø	0.22	Ø	0.64
C	0.21	B	0.20
B	0.19	E	0.08
A C	0.13	A	0.04
A B	0.09		
D	0.08		
A	0.03		

posterior distribution is now less concentrated than in the UDG case, again as a consequence of the higher number of models. Three models seem relatively more supported than the others, but their posterior probability is about a half of the two corresponding best UDG models.

4.1.3. Discussion on model selection. Five of the six variables on the heart disease dataset can be retained to occur simultaneously, with variable F preceding. This means that, leaving alone variable F, which indeed appears to be disconnected from the others, there are no substantial reasons why the relationships among variables should be considered causal. Therefore, an undirected model would be more suited for this dataset.

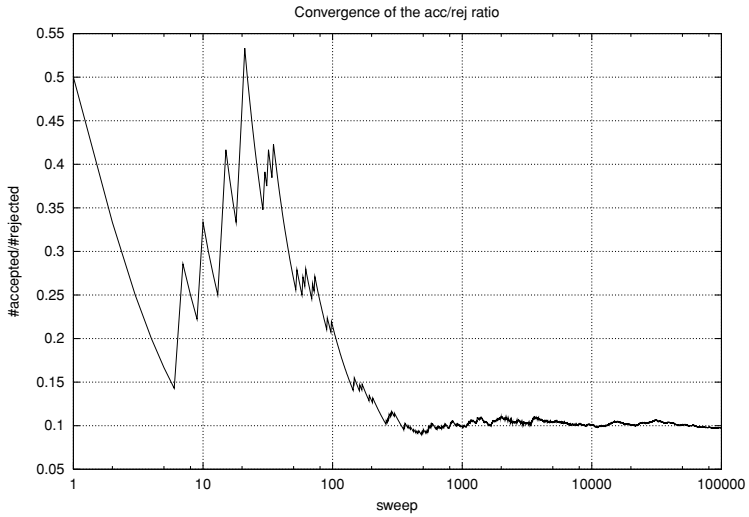
In figure 6 we show the four most probable UDG models found by the MC^3 selection process we have detailed. They receive a posterior probability of respectively, (0.29, 0.11, 0.11, 0.06).

In order to discuss this result, let us take a look first at the models selected in previous works that have already analyzed this data. In figure 7 we may see the two models selected by Edwards and Havránek (1985) and Madigan and Raftery (1994).

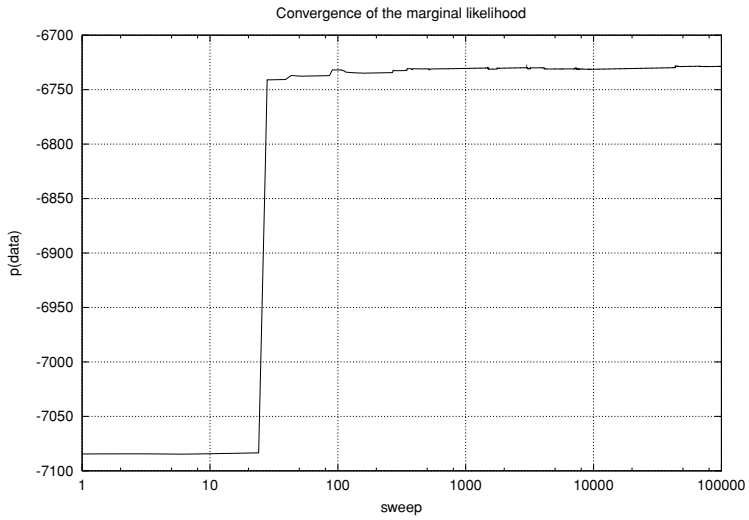
The classical stepwise procedure of Edwards and Havránek (1985) selected two graphical log-linear models. This space of models does not correspond to decomposable or directed models, that are the ones reported here and were reported by Madigan and Raftery (1994) as well. Nevertheless it is possible to see that most of the two-way interaction terms coincide.

Madigan and Raftery (1994), following a stepwise Bayesian procedure using an Occam's window, also select two graphical models.

We depart from these two approaches to model selection by sampling a substantial part of the probability distribution of the models given the data. This leads to the result we see on figure 6 on which the four most probable models account for 57% of the distribution, while Madigan and Raftery (1994) restrict the distribution only to those models that fall



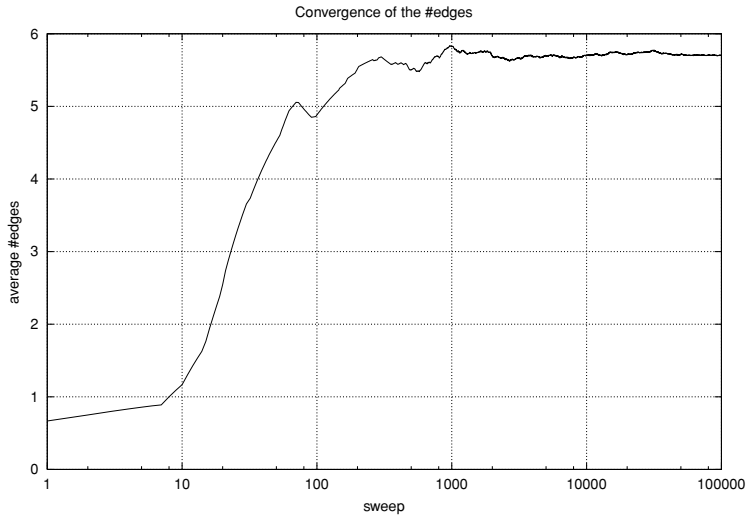
(a)



(b)

Figure 4. Dynamic diagnostics for DAG models in the CHD dataset.

(*Continued on next page*.)



(c)

Figure 4. (*Continued*).

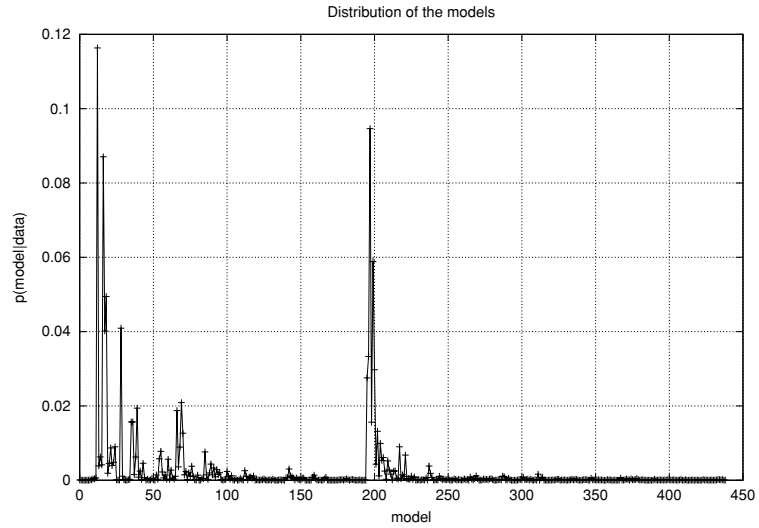
in the Occam's window and, therefore, their best model alone accounts for 92% of the distribution.

The three most probable models of figure 6 differ in the way the systolic blood pressure (D) interacts with the ratio of α and β proteins (E) and smoking (A). They have a similar support with a preference for the ratio of α and β proteins (E) as the factor that better predicts systolic blood pressure (D). This feature remains in the fourth model, that accounts for a 6% of the probability, but this latter model differs from the previous three in the presence of the edge B–E and the absence of the edge C–E, i.e. the way the ratio of α and β proteins (E) are directly related to strenuous mental or physical work (B or C). This disagreement coincides with the one between the two models selected by Edwards and Havránek (1985) and, as they point out, it may be caused by the high negative association between strenuous mental and physical work, i.e. strenuous work is either physical or mental but not both at the same time. A feature common to all models selected in the previous and our current work, is the marginal independence of the family anamnesis of coronary heart disease (F).

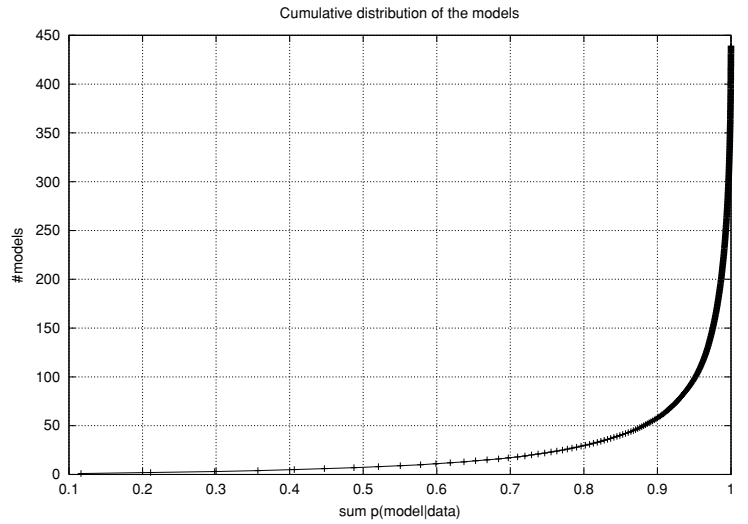
A feature that holds throughout the four models in figure 6, is the conditional independence of smoking (A) from strenuous mental work (B) given strenuous physical work (C). This conditional independence statement is also supported by the posterior distributions we have in Table 1. We believe this may follow from the fact that smoking may be more common to physical work than to mental work. This conditional independence statement is in one of the models of Edwards and Havránek (1985) and in the model of Madigan and Raftery (1994) that accounts for the 92% of the probability distribution.

A last further observation common to all the analysis commented here is the unability of smoking (A), as a single factor, to render strenuous physical work (C) conditionally independent from the ratio of α and β proteins (E). Table 1 shows that this conditional

independence statement is not supported by the data. It does not fall under the 95% of the distribution for statements separating C, and among those separating E it just has a 3% of support. This may be interpreted as if, in a situation of physical work, smoking does not suffice to predict the ratio of α and β proteins. It is also necessary to know whether this physical work is carried out under strenuous conditions or not.



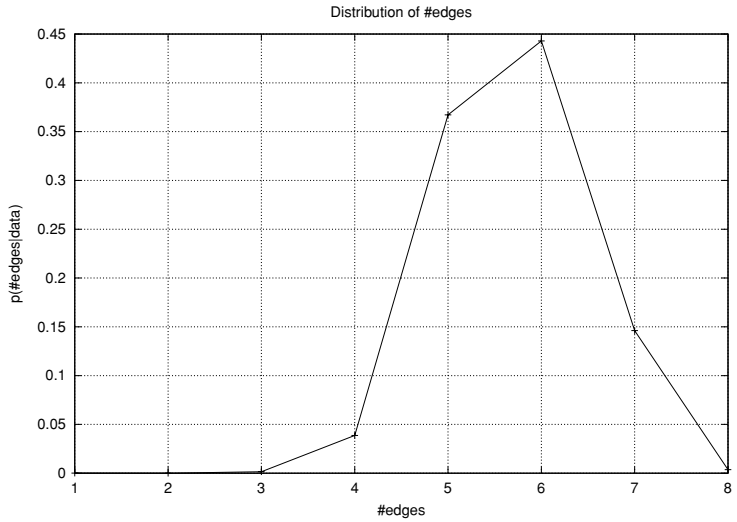
(a)



(b)

Figure 5. Static diagnostics for DAG models in the CHD dataset.

(*Continued on next page*.)



(c)

Figure 5. (*Continued*).

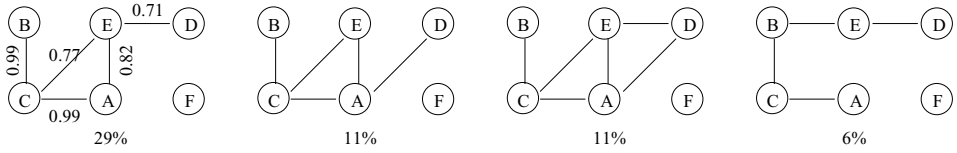


Figure 6. Four most probable UDG models for the CHD dataset.

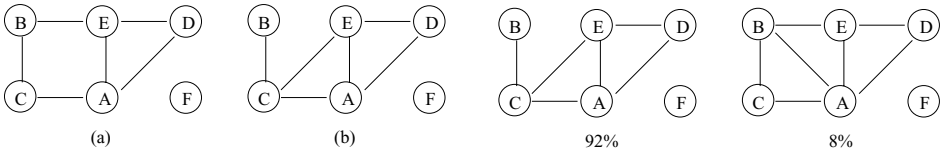


Figure 7. Undirected models selected by Edwards and Havránek (1985) and Madigan and Raftery (1994).

Some authors (e.g. Dellaportas & Forster, 1999), point out that even in this apparently small model selection exercise, one should look not only at the most probable graphs, as a considerable number of models may be quite alike in terms of support, but also at the posterior probabilities of edge presence.

We have seen already this posterior plotted in figure 3(b). A way of improving the interpretability of this posterior is by taking those edges above a certain threshold and drawing them as a graph. This graph shows marginal relationships of a certain strength, therefore no independencies may be read off from such graph.

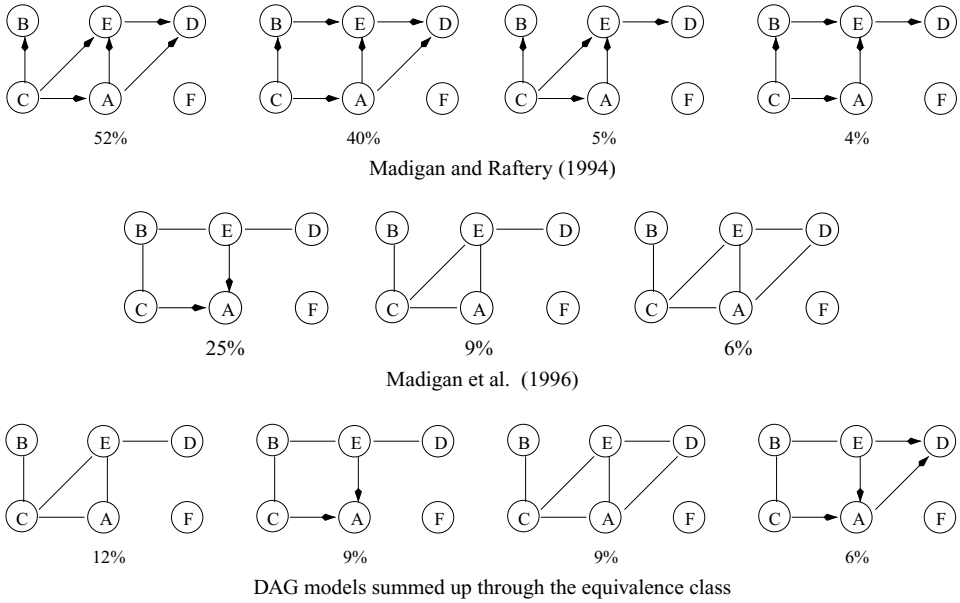


Figure 8. DAG models selected by Madigan and Raftery (1994). Essential graphs selected by Madigan et al. (1996). Essential graphs selected here by summing over equivalent DAG models.

In the plot in figure 3(b) we may see that there are five edges above 0.7 and the rest lie below 0.4. We have taken 0.7 as a threshold, and this set of edges coincides with the set of edges of the most probable UDG model we report in figure 6. Since the graph is the same in both cases we have written in figure 6 the probabilities of these five edges. Of course, one can expect that that the most probable edges appear in those UDG models that account for the largest part of the posterior distribution. The probabilities on the edges may help to confirm some of the conclusions one draws from the models, as for instance the very low probability of the edge A–B supports the assertion of conditional independence of A and B given C.

Finally, concerning structural learning over the space of DAG models, Madigan and Raftery (1994) suggest that a natural partial ordering of the variables could be F, (B, C), A, (E, D). The models they select are in figure 8. At the bottom of the same figure we may see the models selected in this article. As previously mentioned, we have summed over those DAGs belonging to the same equivalence class, and in figure 8 we report the essential graph representation of the four most probable equivalence classes of DAG models.

Our data-driven results are similar to those of Madigan and Raftery (1994) but they have some disagreement regarding the order among the variables, established by these authors. The models we selected match more the results Madigan et al. (1996) obtained by adapting MC^3 to select models directly in the space of essential graphs.

The ordering of the variables given by Madigan and Raftery (1994) establishes that smoking (A) preceeds the ratio of α and β proteins (E). As we may see in their models, two of them contain the immorality $A \rightarrow E \leftarrow B$ which does not appear at all in our analysis as

well as in the essential graphs from Madigan et al. (1996). We saw before in the case of UDG models, that smoking as a single factor was unable to separate strenuous physical work (C) from the ratio of α and β proteins (E). This fact agrees with the immorality $C \rightarrow A \leftarrow E$ that we may see in the most probable model of Madigan et al. (1996) as well as in our second and fourth most probable models.

To confirm our conclusions over the selected models we computed the posterior of the different parent sets of variables for each variable. We may see that information in Table 2. From that table it follows that in the case of the variable A, the parent set formed by C and E have the largest posterior, while A and B as parents of E have very little support. In fact, Madigan et al. (1996) also assert that from their analysis, the data does not support the precedence of smoking (A) over the α and β proteins (E).

When we formalized the legal moves for DAG models in Section 3.1, it was introduced the non-covered reversal (NCR) as a mechanism to alleviate the problem of obtaining a posterior distribution that may be proportional to the sizes of the equivalence classes of DAG models. Madigan et al. (1996) solve this problem by adapting the MC^3 method to the space of essential graphs (equivalence classes of DAG models), but this increases the computational cost of the method since the chain must be able to change two edges at the same time in order to fulfill irreducibility. We believe that the use of NCR affords an interesting trade-off that we are going to assess now, as a final remark in this section.

We have already seen that our results, in fact, do not differ much from the ones of Madigan et al. (1996) and the main conclusions may be drawn from either analysis. In order to provide a more accurate estimate of the difference between the two methods we have computed and compared their predictive performances. Following the same setup of Madigan et al. (1996), we have divided the CHD dataset in two subsets containing, each of them, 50% of the data. One of the subsets is used to select the models (the training set), and the other to compute the predictive performance (the test set).

We have done the split of the dataset and the computations five times in order to average across the five outputs. Let D_L be the training set and D_T the test set. The predictive performance of a single model M is measured as

$$\frac{1}{|D_T|} \sum_{d \in D_T} p(d \mid M, D_L),$$

while the predictive performance of a set of models for Bayesian model averaging is computed as

$$\frac{1}{|D_T|} \sum_{d \in D_T} \sum_{M} p(d \mid M, D_L) p(M \mid D_L).$$

A better model selection, or learning, method should produce models that assign higher probabilities to test cases. We have not used all models from the posterior distribution, but just those that together account for the 50% of this distribution, and then we have renormalize it for this set of models. In Table 3 we may see the results. The first column contains the number of models that account for the 50% of the posterior distribution, the second column contains the predictive performance for Bayesian model averaging, the third column contains the predictive performance for the single best model and the last column

Table 3. Predictive performance on the CHD dataset.

$n \leq 50\%$	pp bma	pp best	Gain (%)
7	0.0278	0.0277	0.4
10	0.0263	0.0263	0.0
6	0.0258	0.0257	0.6
22	0.0267	0.0265	0.6
13	0.0269	0.0267	0.8
12	0.0267	0.0266	0.5

shows the proportional gain in probability by doing Bayesian model averaging. The last line of the table contains the average of the previous five computations.

If we assigned an uniform distribution to the single test case, it would have a probability of 0.0156 which is substantially smaller (42%) than the assignment computed by averaging across the models (0.0267). Madigan et al. (1996) reported that averaging across the models selected using their MC^3 for essential graphs, gives a probability 0.0267, which is the same we have here, but they report a higher gain (1.5%) with respect to using the single best model. In this case we may see in Table 3 that this gain, on average, is not larger than 0.5%.

4.2. Market basket analysis dataset

Market Basket Analysis is an applied statistical methodology aimed at identifying associations between consumers buying choices of different products, usually inside a specific unit, such as a supermarket, which we shall keep as a reference example.

Within such a unit, data in market basket analysis usually consists of all buying transactions done by the clients in a certain unit of time. The aim of the analysis is to understand the association structure between the sales of the different products available. Once the associations are found, it is possible to better plan marketing policies. For instance, if two products turn out to be heavily associated, it will be sufficient to put only one of them on promotion (e.g. on price discount).

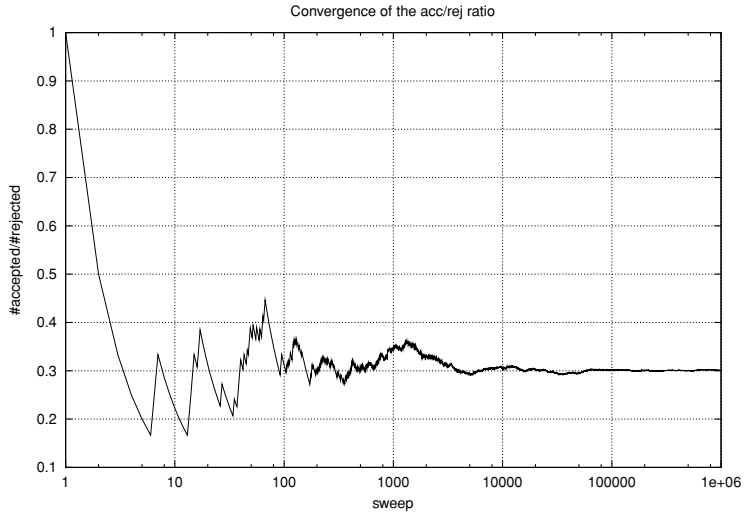
Our available data consists of the sales of a sample of 26 products (grouping different brands for the same good) in 52 periods of one week each, for the year 1997, in one large Italian supermarket (about 12,000 square meters), localized in the region of Piedmont. All products are of food type and have been chosen among those most sensitive to promotional effects. For each week, data are summarized in 26 binary variables indicating whether a certain product was sold above or below the median of the year.

The classical analysis of this very sparse dataset, using a discrete graphical model over the 26 binary variables available, has led to a rather complex model, containing 44 significant edges, 9 of which describe negative associations. More details on this analysis are contained in Giudici and Passerone (2001) to which we refer for further details.

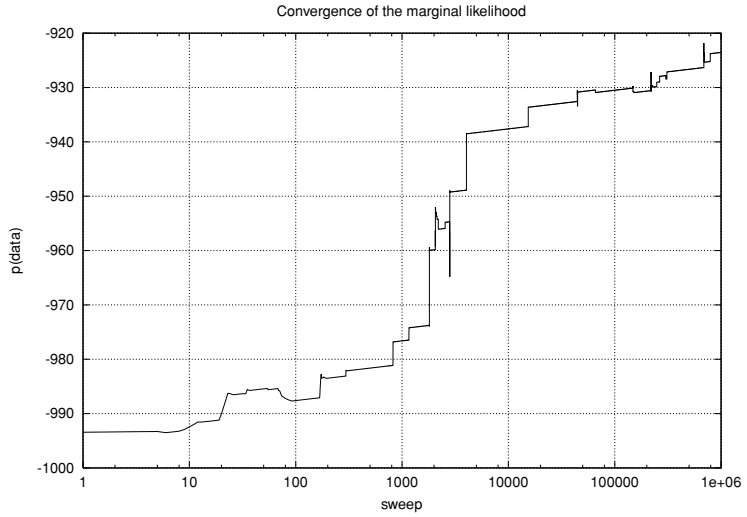
As it is difficult to assess what variables occur on different footing, we assume that the data generating mechanism sets all variables on equal footing, and therefore we consider here only model search on the class of UDG models.

We use now the method illustrated previously, for UDG models on this dataset, which we will name the MBA dataset (figure 9). Given the very large number of considered models, in all of our experiments on this dataset we have now considered a run length of $n = 1{,}000{,}000$ iterations, without burn-in, and starting from the independence model.

We remark that the diagnostics indicate a good degree of convergence of the output, although 1,000,000 iterations seem to be necessary. The ratio between accepted and rejected



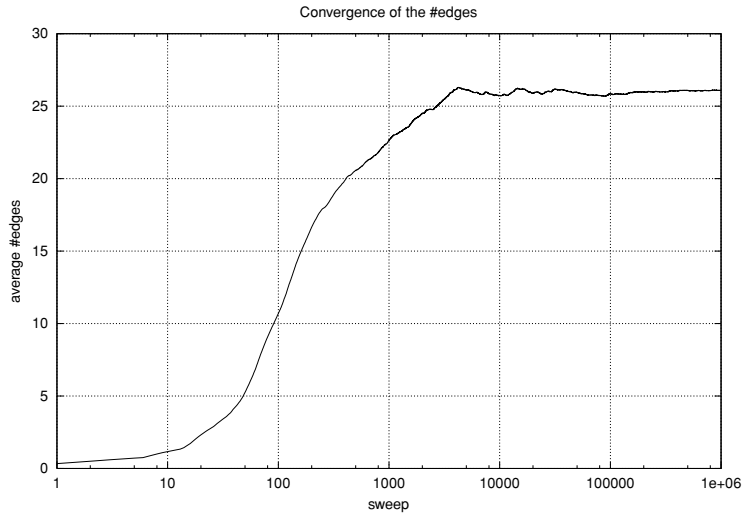
(a)



(b)

Figure 9. Dynamic diagnostics for UDG models in the MBA dataset.

(*Continued on next page*.)



(c)

Figure 9. (*Continued*).

models converges to 0.3. The number of edges present seems to converge around the value of 26, less than in the classical case.

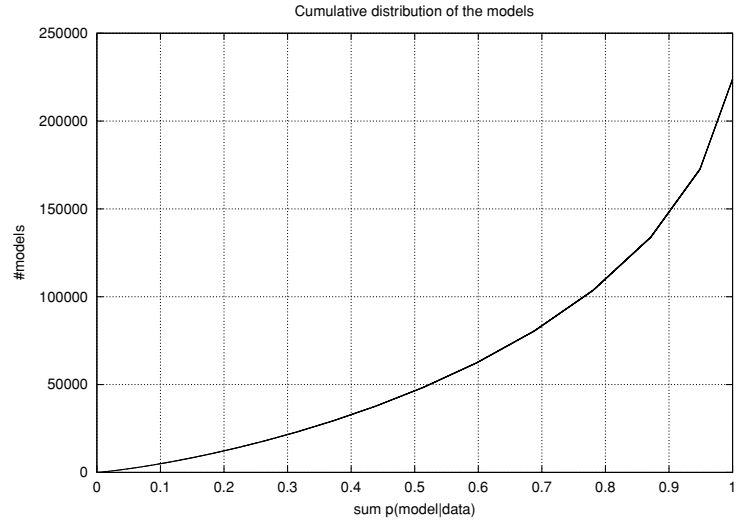
Figure 10 reports static monitors on the dataset, built as in the previous subsection. In this figure, the plot (a) clearly shows the low concentration of the posterior probabilities, due to the very large number of possible models, as well as the extreme sparseness of the data. However, the plot (c) indicates that, differently from what appears in (a), sensible results may be extracted from the analysis, in terms of posterior probabilities of edge presence. There is a limited number of edges, out of the 325 possible ones, which have a high probability of being present. For instance, if we set a threshold at 0.7, only 10 edges have a higher probability.

Notice how the posterior distribution on the number of edges is symmetrically distributed around a value of about 26 in plot 10b. This is a consequence of the large number of possible edges, which makes the normal approximation to the distribution of the number of edges rather accurate.

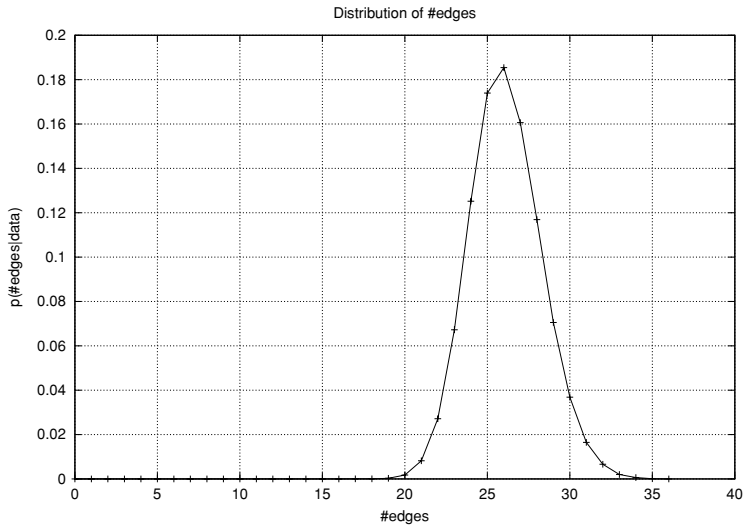
In contrast with the analysis based primarily on graphs, that was done for the CHD dataset, the high number of graphs with similar support makes unfeasible to discriminate between them on the basis of their posterior probabilities. Here it is clearly necessary to focus attention, rather than on the most supported graphs, on the two-way interaction representative graph. Here we have decided to include all edges with a probability of being present larger than 70%. Figure 11 reports such a graph.

From figure 11 it appears that the graph breaks up in 5 disconnected components, corresponding to different consumer behaviours. All associations are meant to be positive (e.g. with an odds ratio greater than unity) apart from three of them, plotted with a dashed line, that describe negative associations.

The first cluster to the left identifies milk, cookies and rice. The association between milk and cookies is clear, less interpretable is that with rice products. A possible reason is that these products have quite stable sales across the year. We remark that our dataset was built by simultaneously considering whether, in each week of the given year, products were sold below or above the median. This introduces a latent explanatory variable in the model, which may induce the observed associations.



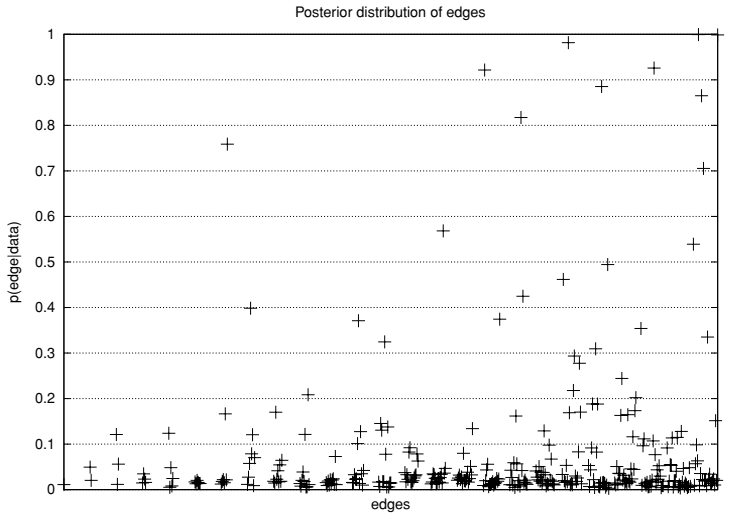
(a)



(b)

Figure 10. Static diagnostics for UDG models in the MBA dataset.

(*Continued on next page*.)



(c)

Figure 10. (*Continued*).

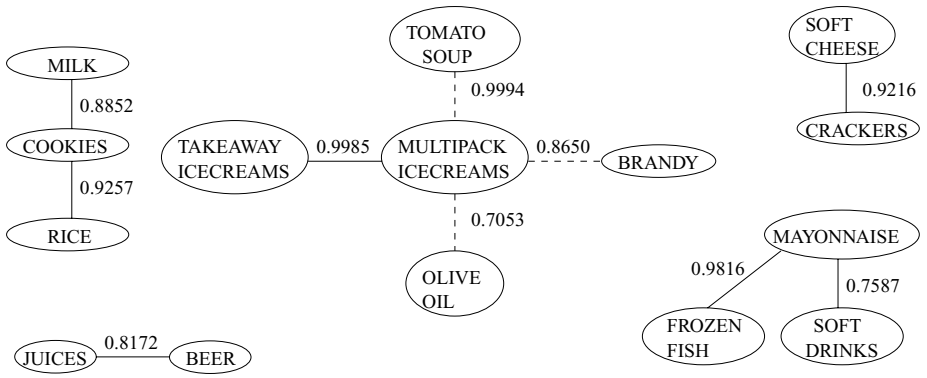


Figure 11. Representative graph for the MBA dataset.

The second cluster to the bottom left identifies fruit juices and beer. This association reflects a common use of these drinks in the household. Both these products have very variable sales along the year.

The third cluster, at the right bottom of the figure, associates soft drinks, mayonnaise and frozen fish. This connected component, contains products which are quite occasionally bought, for instance to organize parties (soft drinks) or to have quick meals (this is the case of the frozen fish and the mayonnaise).

The fourth cluster to the top right also identifies a category of fast-food customers, associating soft cheese and crackers.

Finally, the biggest cluster contains 5 nodes, describing seasonal sales, where four out of the five product interact only with multipack icecreams. The other nodes are icecreams

for take away, tomato soup, brandy and olive oil. Tomato soup, brandy and olive oil are negatively associated with multipack icecreams, reflecting different seasons of peak buy.

As we did in the analysis of the CHD dataset, we are going to examine now the posterior probabilities of conditional independence statements. Table 4 shows this information, for reasons of space, in a slightly different way than for the CHD dataset. One of the ways of using such information in the context of market basket analysis is, by focusing on products with a positive relationship, to find out for which combinations of products an increase of

Table 4. Forty-five most probable conditional independencies for the MBA dataset.

$U \perp\!\!\!\perp V \setminus C \mid C$	Variable (U)	Conditioning set (C)
4.647010e-01	BEER	JUICES
4.199370e-01	TA-ICECREAM	MP-ICECREAM
3.826470e-01	TOMATO SOUP	MP-ICECREAM
3.765420e-01	OLIVE OIL	MP-ICECREAM
3.333900e-01	SOFT-CHEESE	CRACKERS
3.302110e-01	TUNA-FISH	SODAS
3.103860e-01	SOFT-CHEESE	CRACKERS F-FISH
3.000540e-01	F-VEGETABLES	MILK
2.989510e-01	CRACKERS	SOFT-CHEESE
2.926120e-01	RICE	COOKIES
2.491750e-01	COFFEE	BRANDY
2.046430e-01	SOFT-DRINKS	MAYONNAISE
2.000360e-01	YOGURT	CRACKERS
1.912180e-01	BRANDY	MP-ICECREAM
1.839820e-01	MOZZARELLA	$\emptyset$
1.710120e-01	TIN-MEAT	MP-ICECREAM
1.672330e-01	JUICES	BEER COOKIES
1.654170e-01	MAYONNAISE	SOFT-DRINKS F-FISH
1.648010e-01	TIN-MEAT	RICE
1.612000e-01	MINERAL-WATER	$\emptyset$
1.599890e-01	RICE	TIN-MEAT COOKIES
1.534770e-01	CRACKERS	SOFT-CHEESE YOGURT
1.477690e-01	PASTA	$\emptyset$
1.439260e-01	SODAS	TUNA-FISH
1.365680e-01	MINERAL-WATER	MP-ICECREAM
1.361750e-01	MILK	COOKIES
1.327520e-01	JUICES	BEER
1.300640e-01	PARTY SAUSAGES	$\emptyset$
1.299300e-01	BRANDY	COFFEE MP-ICECREAM
1.280260e-01	PASTA	MILK

(*Continued on next page*.)

Table 4. (*Continued*).

$U \perp\!\!\!\perp V \setminus C \mid C$	Variable (U)	Conditioning set (C)
1.272230e-01	MILK	COOKIES F-VEGETABLES
1.178720e-01	YOGURT	Ø
1.152760e-01	PARTY SAUSAGES	SOFT-DRINKS
1.135050e-01	PARTY SAUSAGES	MILK
1.007980e-01	MP-ICECREAM	TOMATO-SOUP BRANDY OLIVE-OIL TA-ICECREAM
1.005020e-01	MAYONNAISE	SOFT-DRINKS SODAS F-FISH
9.947400e-02	MP-ICECREAM	TIN-MEAT TOMATO-SOUP BRANDY OLIVE-OIL TA-ICECREAM
9.736600e-02	F-FISH	SOFT-CHEESE MAYONNAISE
9.607600e-02	CRACKERS	SOFT-CHEESE F-FISH
9.380700e-02	SODAS	MAYONNAISE F-FISH
9.262100e-02	TUNA-FISH	Ø
8.995700e-02	MOZZARELLA	PASTA
8.856300e-02	COOKIES	MILK RICE
8.790000e-02	SODAS	MAYONNAISE TUNA-FISH F-FISH
8.742100e-02	BEER	JUICES F-FISH

The set V refers to the whole set of products.

sales, obtained for instance by setting a price discount, may potentially increase the sales of another given product. For instance, we knew from the edge probabilities that soft cheese and crackers were strongly related, but by looking at posteriors on conditional independencies, we may see that by considering crackers and frozen fish together we actually (almost) double the chance that we can increase the sales of soft cheese in an indirect way.

The sparseness of this data makes impossible to apply a multidimensional hypothesis testing for independence. Nevertheless, as we have seen, the Bayesian graphical approach still allows to extract information on indirect relationships.

To conclude the analysis, we stress that most of these associations were not known a priori, and indeed may not seem very obvious. However, the problem of interest of the providers of this dataset is exactly to find such unknown associations, and this is what makes it a difficult data mining problem.

5. Concluding remarks

In this paper we have concentrated on the problem of Bayesian model choice for discrete UDG and DAG graphical models, showing that Markov Chain Monte Carlo techniques can be a useful tool in this field.

The major contribution of this paper is the introduction of diagnostics for MCMC model selection in order to assess the results of such method in a systematic way. As we have seen throughout the experiments, the MC^3 method of Madigan and York (1995) benefits

from such contribution as we have been able to extend their analysis and conclusions over the CHD dataset. The analysis on the MBA dataset also shows the potential use of the methodology in front of real-world data from which we had no a priori knowledge.

If the aim of the approximate inference is heavily focused on quantitative learning aspects we suggest considering, as a more powerful alternative, the reversible jump approach suggested in Giudici and Green (1999) and Dellaportas and Forster (1999) which does MCMC model determination over both the model and the parameter space.

We finally remark that our proposed methodology is quite general, and can be extended to other families of graphical models.

Acknowledgments

This work has received support from the Network for Highly Structured Stochastic Systems, of the European Science Foundation. Furthermore, both Authors have benefited from a visiting period in November 1999 at the Fields Institute, Toronto. We thank Stefano Paggi and Gianluca Passerone, former students at the University of Pavia, for the market basket analysis data and its classical analysis. The authors are also grateful to the editor, two anonymous reviewers and Tomáš Kočka, for useful comments and suggestions that have improved the paper substantially.

References

Brooks, S. (1998). Markov chain Monte Carlo method and its application. *The Statistician, 47*, 69–100.

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

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

Cowell, R., Dawid, A., Lauritzen, S., & Spiegelhalter, D. (1999). *Probabilistic networks and expert systems*. New York: Springer-Verlag.

Dawid, A. (1979). Conditional independence in statistical theory (with discussion). *Journal of the Royal Statistical Society* B, *41:1*, 1–31.

Dawid, A., & Lauritzen, S. (1993). Hyper-Markov laws in the statistical analysis of decomposable graphical models. *Annals of Statistics, 21:3*, 1272–1317.

Dellaportas, P., & Forster, J. (1999). Markov chain Monte Carlo model determination for hierarchical and graphical log-linear models. *Biometrika, 86:3*, 615–633.

Edwards, D., & Havránek, T. (1985). A fast procedure for model search in multidimensional contingency tables. *Biometrika, 72:2*, 339–351.

Frydenberg, M., & Lauritzen, S. (1989). Decomposition of maximum likelihood in mixed interaction models. *Biometrika, 76:3*, 539–555.

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

Giudici, P., & Green, P. (1999). Decomposable graphical gaussian model determination. *Biometrika, 86:4*, 785–801.

Giudici, P., & Passerone, G. (2001). Data mining of association structures to model consumer behaviour. *Journal of Computational Statistics and Data Analysis*, to appear.

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

Kass, R., & Raftery, A. (1995). Bayes factors. *Journal of the American Statistical Association, 90:430*, 773–795.
Lauritzen, S. (1996). *Graphical models*. Oxford: Oxford University Press.
Lauritzen, S., Dawid, A., Larsen, B., & Leimer, H. (1990). Independence properties of directed Markov fields. *Networks, 20*, 491–505.
Madigan, D., Andersson, S., Perlman, M., & 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., & Raftery, A. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association, 89:428*, 1535–1546.
Madigan, D., & York, J. (1995). Bayesian graphical models for discrete data. *International Statistical Review, 63*, 215–232.
Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Mateo, California: Morgan Kaufmann.
Pearl, J., & Verma, T. (1987). The logic of representing dependencies by directed graphs. In *Proc. of the Conf. of the American Association of Artificial Intelligence* (pp. 374–379).
Robinson, R. (1973). Counting labeled acyclic digraphs. In F. Harary (Ed.), *New directions in the theory of graphs* (pp. 239–273). Academic Press: New York.
Tarjan, R., & Yannakakis, M. (1984). Simple linear time algorithms to test chordality of graphs, test acyclicity of hypergraphs and selectively reduc acyclic hypergraphs. *SIAM Journal of Computing, 13*, 566–579.
Verma, T., & Pearl, J. (1990). Equivalence and synthesis of causal models. In P. Bonissone, M. Henrion, L. Kanal, & J. Lemmer (Eds.), *Proc. of the Conf. on Uncertainty in Artificial Intelligence* (pp. 255–268). Morgan Kaufmann.
Wormald, N. (1985). Counting labeled chordal graphs. *Graphs and Combinatorics, 1*, 193–200.

Received July 6, 2000
Revised April 11, 2001
Accepted October 26, 2001
Final manuscript October 26, 2001