Noname manuscript No.
(will be inserted by the editor)

Blankets Joint Posterior score for learning irregular Markov network structures

Federico Schlüter · Yanela Strappa · Facundo Bromberg · Diego H. Milone

Received: date / Accepted: date

Abstract Markov networks are extensively used to model complex sequential, spatial, and relational interactions in a wide range of fields. By learning the structure of independences of a domain, more accurate joint probability distributions can be obtained for inference tasks or, more directly, for interpreting the most significant relations among the variables. However, the performance of current available methods for learning the structure is heavily dependent on the choice of two factors: the structure representation, and the approach for learning such representation. This work follows the probabilistic maximum-a-posteriori approach for learning undirected graph structures, which has gained interest recently. Thus, the *Blankets Joint Posterior* score is designed for computing the posterior probability of structures given data. In particular, the score proposed can improve the learning process when the solution structure is irregular (that is, when there exists an imbalance in the number of edges over the nodes), which is a property present in many real-world networks. The approximation proposed computes the joint posterior distribution from the collection of Markov blankets of the structure. Essentially, a series of conditional distributions are calculated by using, information about other Markov blankets in the network as evidence. Our experimental results demonstrate that the proposed score has better sample complexity for learning irregular structures, when compared to state-of-the-art scores. By considering optimization with greedy hill-climbing search, we prove for several study cases that our score identifies structures with fewer errors than competitors.

Keywords Markov network · structure learning · scoring function · blankets posterior · irregular structures

F. Schlüter · Y. Strappa · F. Bromberg
DHARMa Lab, Dept of Information Systems.
Facultad Regional Mendoza,
Universidad Tecnológica Nacional,
Mendoza, Argentina. Tel.: +54-261-5240066
E-mail: federico.schluter@frm.utn.edu.ar

D. H. Milone
Research Institute for Signals, Systems and Computational Intelligence, sinc(i),
FICH-UNL/CONICET
Santa Fe, Argentina.

1 Introduction

A Markov network (MN) is a popular probabilistic graphical model that efficiently encodes the joint probability distribution for a set of random variables of a specific domain [Pearl, 1988, Lauritzen, 1982, Koller and Friedman, 2009]. MNs usually represent probability distributions by using two interdependent components: an independence structure, and a set of numerical parameters over the structure. The first is a qualitative component that represents structural information about a problem domain in the form of conditional independence relationships between variables. The numerical parameters are a quantitative component that represents the strength of the dependencies in the structure. There is a large list of applications of MNs in a wide range of fields, such as computer vision and image analysis [Li, 2001, Hwang and Kim, 2015, Peng et al., 2016], computational biology [Li et al., 2015], biomedicine [Schmidt et al., 2008, Wan et al., 2015], and evolutionary computation [Larrañaga and Lozano, 2002, Shakya et al., 2012], among many others. For some of these applications, the model can be constructed

manually by human experts, but in many other problems this can become unfeasible, mainly due to the dimensionality of the problem.

Learning the model from data consists of two interdependent problems: learning the structure; and given the structure, learning its parameters. This work focuses on the task of learning the structure. The structures learned may be used to construct accurate models for inference tasks (such as the estimation of marginal and conditional probabilities), and also may be interesting per se, since they can be used as interpretable models that show the most significant interactions of a domain [Lee et al., 2006, Van Haaren et al., 2013, Claeskens et al., 2015, Nyman et al., 2014, Pensar et al., In Press]. The first scenario is known in practice as the density estimation goal of learning, and the second one is known as the knowledge discovery goal of learning [Chapter 16 [Koller and Friedman, 2009]].

An interesting approach to MN structure learning is to use constraint-based (also known as independence-based) algorithms [Spirtes et al., 2000, Bromberg et al., 2009, Aliferis et al., 2010, Schlüter, 2012]. Such algorithms proceed by performing statistical independence tests on data, and discard all structures inconsistent with the tests. This is an efficient approach, and it is correct under the assumption that the distribution can be represented by a graph, and that the tests are reliable. However, the algorithms that follow this approach are quite sensitive to errors in the tests, which may be unreliable for large conditioning sets [Spirtes et al., 2000, Koller and Friedman, 2009]. A second approach to MN structure learning is to use score-based algorithms [Della Pietra et al., 1997, McCallum, 2003, Lee et al., 2006, Ganapathi et al., 2008]. Such algorithms formulate the problem as an optimization, combining a strategy for searching through the space of possible structures with a scoring function measuring the fitness of each structure to the data. The structure learned is the one that achieves the highest score.

It is important to mention that both constraint-based and score-based approaches have been originally motivated by distinct learning goals. According to the existing literature [Koller and Friedman, 2009], constraint-based methods are generally designed for the knowledge-discovery goal of learning [Aliferis et al., 2010, Bromberg et al., 2009], and their quality is often measured in terms of the correctness of the structure learned (structural errors). In contrast, most score-based approaches have been designed for the density estimation goal of learning [Lowd and Davis, 2014, Van Haaren and Davis, 2012, Davis and Domingos, 2010], and they are in general evaluated in terms of inference accuracy. For this reason, score-based algorithms often work by considering the whole MN at once during the search, interleaving the parameters learning step. This makes them more accurate for inference tasks. However, since learning the parameters is known to be NP-hard for MNs [Barahona, 1982], it has a negative effect on their scalability.

Recently, there has been a recent surge of interest towards efficient methods based on a Bayesian approach. This strategy follows a score-based approach, but with the knowledge discovery goal in mind. Basically, an undirected graph structure is learned by obtaining the probabilistic maximum-a-posteriori structure [Schlüter et al., 2014, Pensar et al., In Press]. Such contributions consist in the design of efficient scoring functions for MN structures, expressing the problem formally as follows: given a complete training data set D, find an undirected graph $G^\star$ such that

$$G^\star = \arg\max_{G \in \mathcal{G}} \Pr(G|D), \tag{1}$$

where $\Pr(G|D)$ is the posterior probability of a structure, and $\mathcal{G}$ is the familiy of all the possible undirected graphs for the domain size. This class of algorithms has been shown to outperform constraint-based algorithms in the quality of the learned structures. The contribution of this paper follows this hybrid approach.

The method proposed in this work can improve the quality of structure learning by examining the *irregularity* of each structure. According to [Albertson, 1997], the irregularity of an undirected graph can be computed by summing the imbalance of its edges:

$$irr(G) = \sum_{(i,j) \in E(G)} |d_G(i) - d_G(j)|, \tag{2}$$

where $d_G(i)$ is the degree of the node i in that graph. Clearly $irr(G) = 0$ if and only if G is regular. For non-regular graphs $irr(G)$ is a measure of the lack of regularity. Although there are more complex measures of irregularity for undirected graphs [Rautenbach and Schiermeyer, 2006, Dimitrov et al., 2014], this naïve definition will suffice for the purposes of this work. In this work, we present the *Blankets Joint Posterior* (BJP) as a score that computes the posterior probability of MN structures by taking advantage of the irregularities of the evaluated structure. This allow us to improve the learning process for domains with complex networks, where the topologies exhibit irregularities, which is a common property in many real-world networks [Silva and Zhao, 2016].

After providing some preliminaries, notations and definitions in Section 2, we introduce the BJP scoring

function in Section 3. Section 4 shows our experiments for several study cases. Finally, Section 5 summarizes this work, and poses several possible directions of future work.

2 Background

We begin by introducing the notation used for MNs. Then we provide some additional background about these models and the problem of learning their independence structure, and also discuss the state-of-the-art of MN structure learning.

2.1 Markov networks

Have V as a finite set of indexes, lowercase subscripts for denoting particular indexes, e.g., $i, j \in V$, and uppercase subscripts for subsets of indexes, e.g., $W \subseteq V$. Let X_V be the set of random variables of a domain, denoting single variables as single indexes in V, e.g., $X_i, X_j \in X_V$ when $i, j \in V$. For a MN representing a probability distribution $P(X_V)$ its two components are denoted as follows: G, and θ. G is the structure, an undirected graph $G = (V, E)$ where the nodes $V = \{0, ..., n-1\}$ are the indices of each random variable X_i of the domain, and $E \subseteq \{V \times V\}$ is the edge set of the graph. A node i is a neighbor of j when the pair $(i, j) \in E$. The edges encode direct probabilistic influence between the variables. Instead, the absence of an edge manifests that the dependence could be mediated by some other subset of variables, corresponding to conditional independences between these variables.

A variable X_i is conditionally independent of another non-adjacent variable X_j given a set of variables X_Z if $\Pr(X_i \mid X_j, X_Z) = \Pr(X_i \mid X_Z)$. This is denoted by $\langle X_i \perp X_j | X_Z \rangle$ (or $\langle X_i \not\perp X_j | X_Z \rangle$ for the dependence assertion). As proven by [Hammersley and Clifford, 1971], the independences encoded by G allow the decomposition of the joint distribution into simpler lower-dimensional functions called factors, or potential functions. The distribution can be factorized as the product of the potential functions $\phi_c(V_c)$ over each clique V_c (i.e., each completely connected sub-graph) of G, that is

$$P(V) = \frac{1}{Z} \prod_{c \in \text{cliques}(G)} \phi_c(V_c), \tag{3}$$

where Z is a constant that normalizes the product of potentials. Such potential functions are parameterized by the set of numerical parameters θ.

For each variable X_i of a MN, its Markov blanket (MB) is composed by the set of all its neighbor nodes in the graph. Hereon we denote the MB of a variable X_i as B^{X_i}. An important concept that is satisfied by MNs is the Local Markov property, formally described as:

> **Local Markov property**. A variable is conditionally independent of all its non-neighbor variables given its MB. That is
>
> $$\langle X_i \perp \{X_V \setminus B^i\} | B^{X_i} \rangle. \tag{4}$$

By using such property, the conditional independences of $P(X_V)$ can be read from the structure G. This is done by considering the concept of separability. Each pair of non-adjacent variables (X_i, X_j) are said to be separated by a set of variables $X_Z \subseteq X_V \setminus \{X_i, X_j\}$ when every path between X_i and X_j in G contains some node in X_Z [Pearl, 1988].

In machine learning, statistical independence tests are a well-known tool to decide whether a conditional independence is supported by the data. Examples of independence tests used in practice are Mutual Information [Cover and Thomas, 1991], Pearson's χ^2 and G^2 [Agresti, 2002], the Bayesian statistical test of independence [Margaritis, 2005], and the partial correlation test for continuous Gaussian data [Spirtes et al., 2000]. Such tests require the construction of a contingency table of counts for each complete configuration of the variables involved; as a result, they would have an exponential cost in the number of variables [Cochran, 1954]. For this reason, the use of the local Markov property has a positive effect for learning independence structures, allowing the use of smaller tests. Accordingly, the scoring function proposed in this work takes advantage of this property to avoid the computation of potentially expensive and unreliable tests. This is achieved by examining the irregularities present in a structure.

2.2 MN structure learning

The structure of a MN can be learned from a training dataset $D = \{D_1, ..., D_d\}$, assumed to be a representative sample of the underlying distribution $P(X_V)$. Commonly, D has a tabular format, with a column for each variable of the domain X_V, and one row per data point. This work assumes that each variable is discrete, with a finite number of possible values, and that no data point in D has missing values.

As mentioned in Section 1, this work focuses on the Bayesian approach for MN structure learning of (1). For this reason, in this subsection we discuss two recently proposed scoring functions that follow such approach: the Marginal Pseudo Likelihood (MPL) score

[Pensar et al., In Press], and the Independence-based score (IB-score) [Schlüter et al., 2014].

In MPL, each graph is scored by using an efficient approximation to the posterior probability of structures given the data. This score approximates the posterior by considering $P(G \mid D) \propto P(D \mid G) \times P(G)$. Since the data likelihood of the graph $P(D \mid G)$ is in general extremely hard to evaluate, MPL utilizes the well-known approximation called the pseudo-likelihood [Besag, 1972]. This score was proved to be consistent, that is, in the limit of infinite data the solution structure has the maximum score. For finding the MPL-optimal structure, two algorithms were presented: an exact algorithm using pseudo-boolean optimization, and a fast alternative to the exact method, which uses greedy hill-climbing with near-optimal performance. This algorithm learns the MB for each variable, locally optimizing the MPL for each node, independently of the solutions of the other nodes. For this, it uses an approximate deterministic hill-climbing procedure similar to the well-known IAMB algorithm [Tsamardinos et al., 2003]. Finally, a global graph discovery method is applied by using a greedy hill-climbing algorithm, searching for the structure with maximum MPL score, but only restricting the search space to the conflicting edges.

The independence-based score (IB-score) [Schlüter et al., 2014], is also based on the computation of the posterior, but using the statistics of a set of conditional independence tests. In this score the posterior $\Pr(G \mid D)$ is computed by combining the outcomes of a set of conditional independence assertions that completely determine G. Such set was called the *closure* of the structure, denoted $\mathcal{C}(G)$. Thus, when using IB-score the problem of structure learning is posed as the maximization of the posterior of the closure for each structure. Formally,

$$G^{\star} = \arg\max_{G} \Pr(\mathcal{C}(G) \mid D). \tag{5}$$

Applying the chain rule over the posterior of the closure,

$$\Pr(\mathcal{C}(G) \mid D) = \prod_{c_i \in \mathcal{C}(G)} \Pr(c_i | c_1, \ldots, c_{i-1}, D), \tag{6}$$

the IB-score approximates such probability by assuming that all the independence assertions c_i in the closure $\mathcal{C}(G)$ are mutually independent. The resulting scoring funtion is computed as:

$$\text{IB-score}(G) = \prod_{c_i \in \mathcal{C}(G)} \log \Pr(c_i \mid D), \tag{7}$$

where each term $\log \Pr(c_i \mid D)$ is computed by using the Bayesian statistical test of conditional independence [Margaritis, 2005, Margaritis and Bromberg, 2009]. Together with the IB-score, an efficient algorithm called IBMAP-HC is presented to learn the structure by using a heuristic local search over the space of possible structures.

3 Blankets Joint Posterior scoring function

This section proposes the Blankets Joint Posterior (BJP), a scoring function to compute the posterior probability of the independence structure of a MN. In particular, BJP has been designed in order to accurately approximate the posterior of structures for cases where the underlying structure contains irregularities. The correctness of BJP is discussed in the Appendix A.

Consider some graph G representing the independence structure of a positive MN. It is a well-known fact that, by exploiting the graphical properties of such models, the independence structure can be decomposed as the unique collection of the MBs of the variables [Koller and Friedman, 2009, Theorem 4.6 on p. 121]. Thus, the computation of the posterior probability of G given a dataset D is equivalent to the joint posterior of the collection of MBs of G, that is,

$$\Pr(G \mid D) = \Pr(B^{X_0}, B^{X_1}, \ldots, B^{X_{n-1}} \mid D). \tag{8}$$

In contrast with previous works, where the MB posteriors are simply assumed to be independent [Pensar et al., In Press, Schlüter et al., 2014], the chain rule is applied to (8), obtaining

$$\Pr(B^{X_0}, \ldots, B^{X_{n-1}} \mid D) = \prod_{i=0}^{n-1} \Pr\left(B^{X_i} \middle| \{B^{X_j}\}_{j=0}^{i-1}, D \right). \tag{9}$$

In this way the posterior probability of each MB can be described in terms of conditional probabilities, using the training dataset D as evidence, together with the MB of the other variables.

The computation of $\Pr(B^{X_0}, \ldots, B^{X_{n-1}} \mid D)$ has to be done progressively, first calculating the posterior of the MB of a variable, and then, the knowledge obtained so far can be used as evidence to compute the posterior of the MB of other variables. However, this decomposition is not unique, since each possible ordering for the variables is associated to a particular decomposition. The basic idea underlying the computation of BJP is to sort the MBs by their size (that is, the degree of the nodes in the graph) in ascending order. This allows a series of inference steps, in order to avoid the computation of expensive and unreliable probabilities, and obtaining a more accurate data efficiency. This is due to the fact that as the size of the MB increases, greater amounts

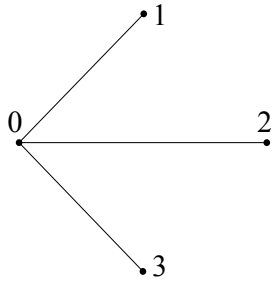

Fig. 1: Example of an undirected graph with 4 nodes and hub topology

of data are required for accurately estimating its posterior probability. By using the proposed strategy, the blanket posteriors of variables with fewer neighbors are computed first, and this information is used as evidence when computing the posteriors for variables with bigger blankets. As a result, the information obtained from the more reliable blanket posteriors is used for computing less reliable blankets posteriors.

Now consider an example probability distribution $\Pr(X_V)$ with four variables $X = \{X_0, X_1, X_2, X_3\}$, represented by a MN whose independence structure G is given by the graph of Figure 1. When sorting its nodes by their degree in ascending order, the vector (X_1, X_2, X_3, X_0) can be obtained, and the blankets joint posterior is decomposed as

$$\begin{aligned}\Pr(B^{X_0}, B^{X_1}, \ldots, B^{X_{n-1}}) = \Pr(B^{X_1}|D) \\ \times \Pr(B^{X_2}|B^{X_1}, D) \\ \times \Pr(B^{X_3}|B^{X_1}, B^{X_2}, D) \\ \times \Pr(B^{X_0}|B^{X_1}, B^{X_2}, B^{X_3}, D).\end{aligned}$$

This example allows us to illustrate the intuition behind BJP, since the sample complexity of the blanket posterior for variables X_1, X_2, and X_3 is lower than that of X_0. For the sake of clarity, Appendix B shows the complete computation of the BJP score for this example.

Given an undirected graph G, denote ψ the ordering vector which contains the variables sorted by their degree in ascending order. Therefore, we reformulate (9) as

$$BJP(G) = \prod_{i=0}^{n-1} \Pr\left(B^{\psi_i} \middle| \left\{B^{\psi_j}\right\}_{j=0}^{i-1}, D\right). \tag{10}$$

We now proceed to express the posterior of a MB in terms of probabilities of conditional independence and dependence assertions. The computation of $\Pr(B^{\psi_i}|\{B^{\psi_j}\}_{j=0}^{i-1}, D)$ can be derived from the posterior of the independences and dependences represented by each MB:

$$\begin{aligned}\Pr\left(B^{\psi_i} \middle| \left\{B^{\psi_j}\right\}_{j=0}^{i-1}, D\right) = \\ \prod_{\psi_k \notin B^{\psi_i}} \Pr\left(\langle \psi_i \perp \psi_k | B^{\psi_i} \rangle \middle| \left\{B^{\psi_j}\right\}_{j=0}^{i-1}, D\right) \times \\ \prod_{\psi_k \in B^{\psi_i}} \Pr\left(\langle \psi_i \not\perp \psi_k | B^{\psi_i} \setminus \{\psi_k\} \rangle \middle| \left\{B^{\psi_j}\right\}_{j=0}^{i-1}, D\right).\end{aligned} \tag{11}$$

The two factors in this equation will be interpreted as follows:

– The first product computes the probability of independence between ψ_i and its non-adjacent variables, conditioned on its MB, given the previously computed MBs and the dataset D. It can be computed as

$$\begin{aligned}\Pr\left(\langle \psi_i \perp \psi_k | B^{\psi_i} \rangle \middle| \left\{B^{\psi_j}\right\}_{j=0}^{i-1}, D\right) \\ = \begin{cases} \Pr(\langle \psi_i \perp \psi_k | B^{\psi_i} \rangle \mid D) & \\ \qquad\qquad \text{if } i < k, & \\ 1 \qquad\qquad \text{if } i > k. & \end{cases}\end{aligned} \tag{12}$$

  Here, $i < k$ indexes over the variables for which the MB posterior probability is not already computed. For the remaining variables the posterior of independence will be simply inferred as 1. This inference can be done since the independence is determined by the MB of ψ_k, which is in the evidence $\left\{B^{\psi_j}\right\}_{j=0}^{i-1}$. We discuss the correctness of this inference step in Appendix A.

– The second product in (11) computes the posterior probability of dependence between ψ_i and its adjacent variables, conditioned on its remaining neighbors, given the MBs computed previously and the dataset D. It can be computed as

$$\begin{aligned}\Pr\left(\langle \psi_i \not\perp \psi_k | B^{\psi_i} \setminus \{\psi_k\} \rangle \middle| \left\{B^{\psi_j}\right\}_{j=0}^{i-1}, D\right) \\ = \begin{cases} \Pr(\langle \psi_i \not\perp \psi_k | B^{\psi_i} \setminus \{\psi_k\} \rangle \mid D) & \\ \qquad\qquad \text{if } i < k, & \\ 1 \qquad\qquad \text{if } i > k. & \end{cases}\end{aligned} \tag{13}$$

  Here, again $i < k$ indexes over the variables for which the MB posterior is not already computed. For the remaining variables the posterior of dependence will be inferred as 1. Also, this inference can be done since the dependence is determined by the MB of ψ_k, which is in the evidence $\left\{B^{\psi_j}\right\}_{j=0}^{i-1}$. The correctness of this inference step is also discussed in Appendix A.

The only approximation in BJP is made in (11), by assuming that all the independence and dependence

assertions that determine the MB of a variable ψ_i are mutually independent. This is a common assumption, made implicitly by all the constraint-based MN structure learning algorithms [Schlüter, 2012], and also by the MPL score and the IB-score. For the computation of the posterior probabilities of independence $\Pr(\langle \psi_i \perp \psi_k | B^{\psi_i} \rangle \mid D)$ and dependence $\Pr(\langle \psi_i \not\perp \psi_k | B^{\psi_i} \setminus \{\psi_k\} \rangle \mid D)$ used in (12) and (13), respectively, BJP uses the Bayesian test of [Margaritis and Bromberg, 2009, Margaritis, 2005,Margaritis and Thrun, 2000], in the same way as the IB-score explained in the previous section. Precisely, this statistical test computes the posterior of independence and dependence assertions, and has been proven to be statistically consistent in the limit of infinite data.

We now discuss the computational complexity of the score. For a fixed structure, the computational cost is directly determined by the number of statistical tests that it is required to perform on data. Recall that the computational cost of each test is exponential in the number of variables involved [Cochran, 1954]. As stated in (10), BJP computes the posterior probability of the MB for the n variables of the domain. For each, it is required to perform $n-1$ statistical tests on data, by using (11). Then, one half of the tests are inferred when computing the posterior of independences and dependences of (12) and (13). Thus, only $\frac{n(n-1)}{2}$ tests are required for computing the BJP score of a structure.

We end this section with the optimization proposed in this work for learning the structure with the BJP score. The naïve optimization consists in maximizing over all the possible undirected graphs for some specific problem domain, as in (1), computing with (10) the score for each structure. Since the discrete optimization space of the possible graphs $\mathcal{G}$ grows rapidly with the number of variables n, the search is clearly intractable even for small domain sizes. Hence, in this work we test the performance of BJP with brute force only for small domains. For larger domains we use the IBMAP-HC algorithm, as an efficient approximate solution proposed in [Schlüter et al., 2014].

The optimization made by IBMAP-HC is a simple heuristic hill-climbing procedure. The search is initialized by computing the score for an empty structure with no edges, and n nodes. The hill-climbing search starts with a loop that iterates by selecting the next candidate structure at each iteration. A naïve implementation of hill-climbing would select the neighbor structure with maximum score, computing the score for the $\binom{n}{2}$ neighbors that differ in one edge. Such expensive computation is avoided by selecting the next candidate with a heuristic that flips the most promising edge. Once the next candidate is selected, its score is computed to be compared to the best scoring structure found so far. The algorithm stops when the neighbor proposed does not improve the current score.

4 Experimental evaluation

This section presents several experiments in order to determine the merits of BJP in practical terms. Two sets of experiments from low-dimensional and high-dimensional problems are presented. For the low-dimensional setting, we used brute force (i.e., exhaustive search) to study the convergence of the scoring functions to the exact solution. The goal is to prove experimentally that the sample complexity for successfully learning the exact structure is better for BJP than for the competitors. For the high-dimensional setting, we used hill-climbing optimization for all the scoring functions. This experiments were performed in order to prove that, by using a similar search strategy, BJP identifies structures with fewer structural errors than the selected competitors. The software to carry out the experiments has been developed in Java, and it is publicly available[1].

4.1 Consistency experiments

A MN scoring function is consistent when the structure which maximizes the score over all the possible structures is the correct one, in the limit of infinite data. However, in practice the data is often too scarce to satisfy this condition, and the sample size needed to reach the correct structure varies across different scoring functions. This is referred to as the *sample complexity* of the score. The experiments here presented were carried out in order to measure the sample complexity of three different scoring functions: MPL, IB-score and BJP. This is achieved by measuring their ability to return, by brute force, the exact independence structure of the MN which generated the data.

To make this comparative study, we selected the six different target structures shown in Figure 2. These graphs represent different cases of irregularity, according to (2). The first target structure is regular ($\text{irr} = 0$), the second has a little irregularity, the third and fourth structures are irregular structures with a hub topology, and the fifth and sixth target structures have maximum irregularity for $n = 6$. For constructing a probability distribution from these independence structures according to (3), random numeric values were assigned to their maximal clique factors, sampled independently from a

[1] http://dharma.frm.utn.edu.ar/papers/bjp

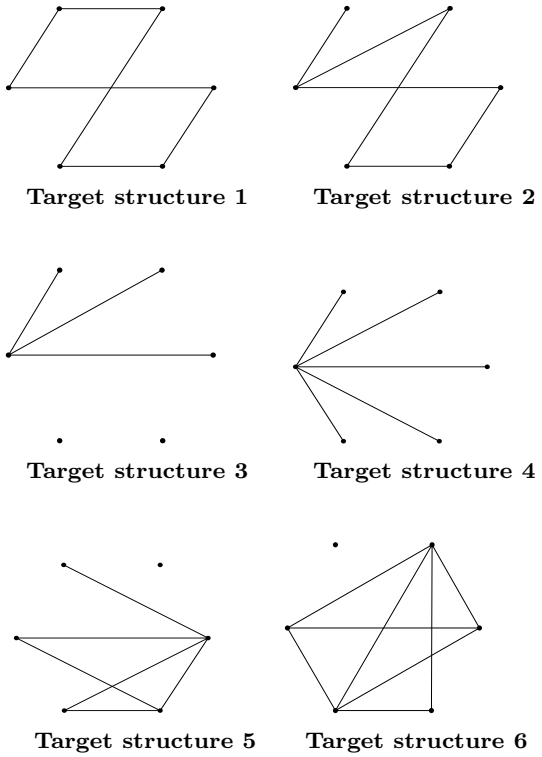

Fig. 2: Independence structures for the first set of experiments: model 1 is regular ($irr = 0$); model 2 has $irr = 10$; model 3 has $irr = 18$; model 4 has $irr = 20$; models 5 and 6 have the maximum irregularity for six variables ($irr = 26$).

uniform distribution over $(0, 1)$. Ten distributions were generated for each target structure, considering only binary discrete variables. Then, for each one, ten different random seeds were used to obtain 100 datasets for each graph, by using the Gibbs sampling tool of the open-source Libra toolkit [Lowd and Rooshenas, 2015]. The Gibbs sampler was run with 100 burn-in and 1000 sampling iterations, as commonly used in other works.

Since we have $n = 6$ variables, the search space consists of $2^{\binom{6}{2}} = 32768$ different undirected graphs. The experiment consisted of evaluating the number of true structures returned by each score over the 100 datasets. This is called here the success rate of the scoring function. The success rate is computed for increasing dataset sizes $\mathcal{N}_D = \{250, 500, 1000, 2000, 4000, 8000\}$. Of course, since greater sizes of the dataset lead to better estimations, $\mathcal{N}_D$ affects the quality of the structure learned. Therefore, a score is considered better than another score when its success rate converges to 1 with lower values of $\mathcal{N}_D$.

Table 1 shows the results of the experiment. The first column shows the target structures, the second shows their irregularity, the third shows each sample size $\mathcal{N}_D$ used, and the fourth shows the success rate. For all the cases, it can be seen how the success rate of the three scoring functions grows with the sample size $\mathcal{N}_D$. The results in the fourth column show that BJP has a better success rate in almost all cases. For structures 1 and 2, IB-score shows better convergence than BJP, but they would eventually converge similarly for greater $\mathcal{N}_D$ sizes. In contrast, for structures 3, 4, 5 and 6, BJP has in general the best success rate. For all the cases MPL has a slower convergence than IB-score and BJP. This is consistent with the experimental results shown in [Pensar et al., In Press], where the quality for MPL with irregular structures is reported as very low. Interestingly, BJP obtains improvements in success rate of up to 8.4% respect to IB-score, and up to 59% respect to MPL. In general, these results are consistent with the hypothesis of this work, since BJP has been designed to improve the sample complexity when learning irregular structures. The following section shows the performance of the three scoring functions for more complex domains.

4.2 Structural errors analysis

In this section, experiments in higher-dimensional setting are presented. For this, we evaluate the quality of the structures learned by using an approximate search mechanism. The BJP score and the IB-score were tested with the IBMAP-HC algorithm proposed in [Schlüter et al., 2014], explained at the end of Section 3. The MPL scoring function was tested with the most efficient optimization algorithm proposed in [Pensar et al., In Press], described in Section 2.2.

The goal in the experiments is to show how the BJP score can improve the quality of the structures learned over the competitor scores, mainly for irregular underlying structures. For this, the selected graphs capture the properties of several real-world problems, where the target structure has fewer nodes with large degrees, and the remaining nodes have very small degree. Examples of problems with this characteristic include gene networks, protein interaction networks and social networks [Silva and Zhao, 2016]. Thus, for this comparative study, we used two types of structures: hubs and scale-free networks generated by the Barabasi-Albert model [Barabasi and Bonabeau, 2003]. These structures have an increasing complexity both in n and in irr. Additionally, we used four real-world networks, taken from the sparse matrix collection of [Davis and Hu, 2011]. The hub networks are shown in Figure 3, the scale-free networks are shown in Figure 4, and the real-world networks are shown in Figure 5.

Target structure	Irr	$\mathcal{N}_D$	Success rate MPL	IB-score	BJP
1 	0	250	0.000	0.000	0.000
		500	0.000	0.007	**0.013**
		1000	0.010	**0.057**	0.034
		2000	0.040	**0.150**	0.124
		4000	0.150	**0.257**	0.213
		8000	0.280	**0.350**	0.342
2 	10	250	0.000	0.000	0.000
		500	0.000	0.006	**0.014**
		1000	0.000	**0.040**	0.021
		2000	0.020	0.153	**0.164**
		4000	0.100	**0.271**	0.250
		8000	0.180	0.392	0.392
3 	18	250	0.000	**0.060**	0.040
		500	0.030	0.090	**0.120**
		1000	0.100	0.170	**0.190**
		2000	0.170	0.220	**0.270**
		4000	0.220	0.450	**0.490**
		8000	0.340	0.580	**0.610**
4 	20	250	0.000	0.000	0.000
		500	0.000	**0.033**	0.020
		1000	0.000	0.066	**0.100**
		2000	0.000	0.146	**0.180**
		4000	0.000	0.293	**0.360**
		8000	0.000	0.446	**0.500**
5 	26	250	0.000	0.015	0.015
		500	0.000	**0.023**	0.015
		1000	0.000	0.100	**0.115**
		2000	0.000	0.238	**0.261**
		4000	0.030	**0.561**	0.546
		8000	0.210	0.753	**0.769**
6 	26	250	0.000	0.000	0.000
		500	0.000	0.000	0.000
		1000	0.000	0.047	**0.131**
		2000	0.000	0.289	**0.372**
		4000	0.020	**0.663**	0.613
		8000	0.270	0.805	**0.820**

Table 1: Success rate of BJP, IB-score and MPL over 100 datasets for the target structures on Figure 2. Rates in bold face correspond to the best case.

For each target structure we generated 10 random distributions and 10 random samples for each distribution, with the Gibbs sampler tool of the Libra toolkit. Thus, a total of 100 datasets were obtained for each graph, with the same procedure explained in the previous section. As a quality measure, we report the average edge Hamming distance between the hundred learned structures and the underlying one, computed as the sum of false positives and false negatives in the learned structure. As in the previous section, the algorithms were executed for increasing dataset sizes $\mathcal{N}_D = \{250, 500, 1000, 2000, 4000, 8000\}$, to asses how their accuracy evolves with data availability.

Table 2 shows the comparison of BJP against MPL and IB-score for the hub structures of Figure 3. The table shows the structures, their sizes n, and their irregularities, in the first, second and third columns, respectively. The dataset sizes $\mathcal{N}_D$ are in the fourth column. The fifth column shows the average and standard deviation of the Hamming distance over the 100 repetitions. The sixth column shows the corresponding runtimes (in seconds)[2]. When analyzing these results, it can be seen that for all the algorithms the more complex the underlying structure (determined by n and irr), the larger is the number of structural errors for any value of $\mathcal{N}_D$. The results show that BJP obtains the best performance, reducing the number of errors of the structures learned for all the cases. When compared to IB-score, the improvements are more important as n and irr grow. This is because in those cases BJP uses a set of independence tests with lower sample complexity than IB-score to estimate the posterior of the structures. It can also be seen that, for all the target structures, MPL has the slowest convergence in $\mathcal{N}_D$. This is consistent with the results shown in the previous section, obtained by using brute force. In terms of the respective runtimes, the optimization using the BJP score obtains in general runtimes comparable to MPL and IB-score. For the case of Hub 4, BJP shows the best runtime for all the cases where $\mathcal{N}_D > 250$. This is because the more complex the underlying structure the better the convergence of the BJP score to correct structures.

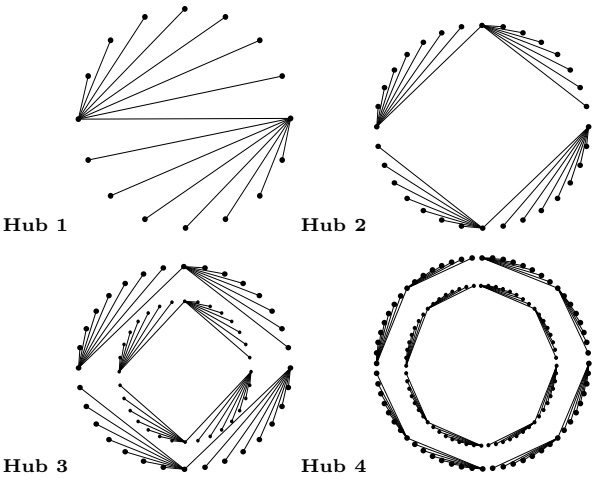

Fig. 3: Structures with a hub topology and 16, 32, 64 and 128 nodes

Table 3 shows the comparison of BJP against MPL and IB-score for the scale-free networks of Figure 4. The information of the table is organized in the same way as in Table 2. For all the scores, it can be seen that the trends in these results are similar to those of the

[2] All the experiments were performed on an Intel(R) Core(TM) i7-4770 CPU, with 3.40GHz, and 32 GB of main memory.

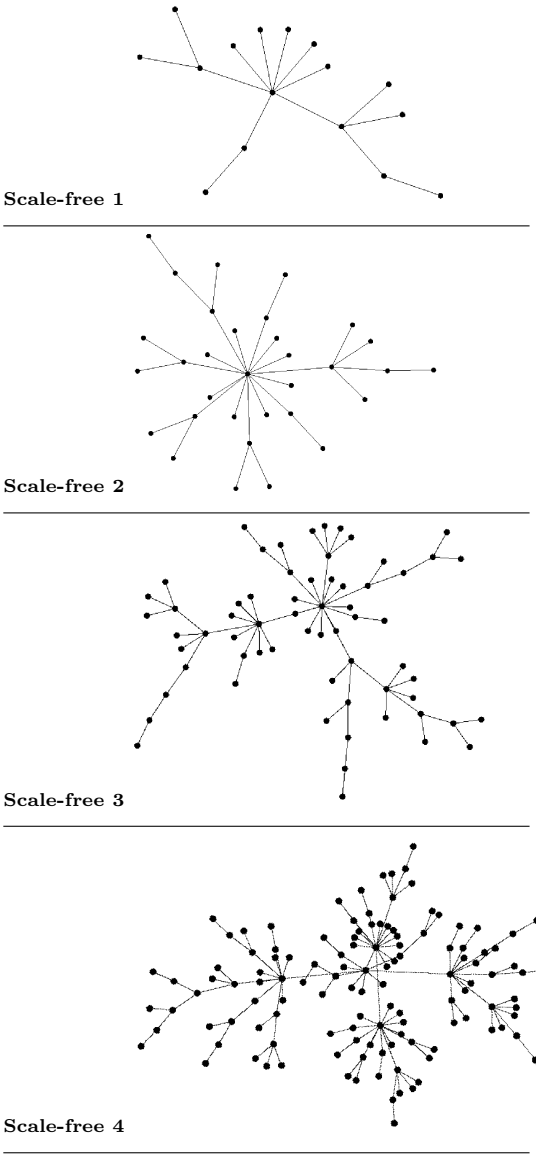

Fig. 4: Scale-free structures with 16, 32, 64 and 128 nodes

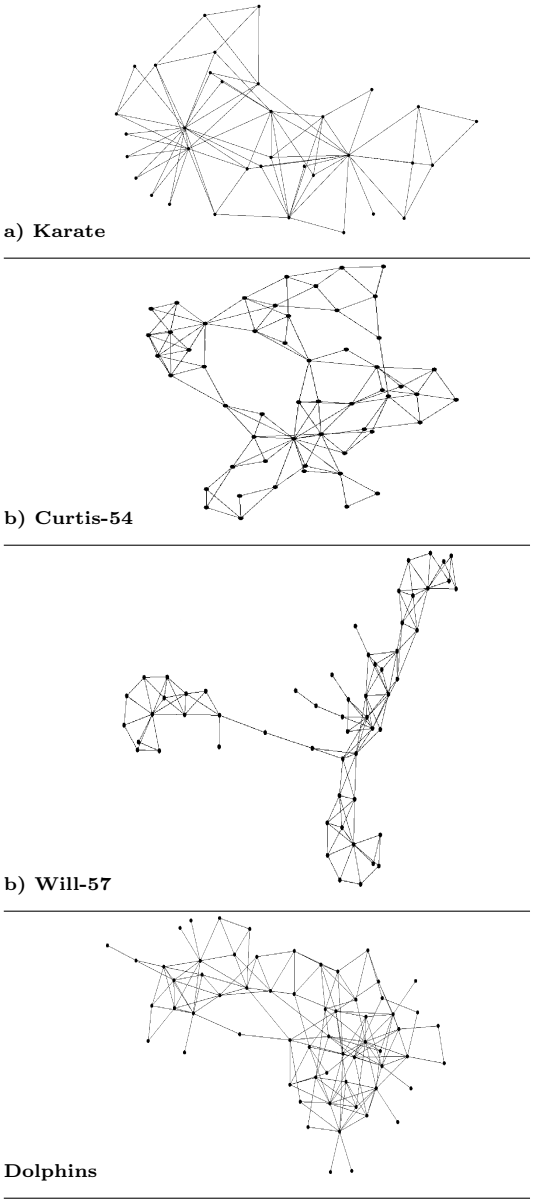

Fig. 5: Scale-free structures with 16, 32, 64 and 128 nodes

hub structures. In contrast with the hub structures, in the scale-free networks the size of the blankets is more variable. This can explain the diference in the trends of the Hamming distance, when compared with the results obtained for the hub networks. For the two most complex structures (scale-free 3 and 4), BJP reduces the number of errors of the structures learned in all the cases. In terms of the respective runtimes, BJP obtains the best runtimes for almost all the cases. Specifically, for scale-free 2, for all the cases where $\mathcal{N}_D > 500$; for scale-free 3, for all the cases; and for scale-free 4, for all the cases where $\mathcal{N}_D > 250$. As the complexity of the target structures grows, we can see a better convergence of the BJP score to correct structures.

Finally, Table 4 show the results for the real-world networks of Figure 5. Again, the information of this table is organized in the same way as in the previous tables. The real network structures are ordered by their complexity (in n and irr). The trends in these results are consistent to those in the previous tables, in terms of quality and runtime. For the Karate, Curtis-54 and Will-57 networks, BJP improves the quality of the structures learned for all the cases when $\mathcal{N}_D < 8000$. When $\mathcal{N}_D = 8000$ IB-score obtains the best qualities.

However, the differences in favor of IB-score are not statistically significant, and the runtime of the optimization is one or two orders of magnitude slower compared to BJP. For the Dolphins network, BJP improves the quality of the structure learned for all the cases. Regarding the runtimes, it can be seen again that BJP tends to improve the runtime over MPL and IB-score for almost all the cases.

In general, the results discussed confirm that BJP always outperforms the competitors when data are scarce. Also, the improvements are greater both in quality and runtime, for the more complex models. This confirms the hypothesis that the BJP score takes advantage of irregularities to optimize the sample complexity.

5 Conclusions

In this work we have introduced a novel scoring function for learning the structure of Markov networks. The BJP score computes the posterior probability of independence structures by considering the joint probability distribution of the collection of Markov blankets of the structures. The score computes the posterior of each Markov blanket progressively, using information of other blankets as evidence. The blanket posteriors of variables with fewer neighbors is computed first, and then this information is used as evidence for computing the posteriors for variables with bigger blankets. Thus, BJP can be useful to improve the data efficiency for problems with complex networks, where the topology exhibits irregularities, such as social and biological networks. In the experiments, BJP scoring proved to improve the sample complexity when compared with the state-of-the-art competitors. The score is tested by using exhaustive search for low-dimensional problems and by using a heuristic hill-climbing mechanism for higher-dimensional problems. The results show that BJP produces more accurate structures than the selected competitors.

We will guide our future work toward the design of more effective optimization methods, since the hill-climbing optimization has two inherent disadvantages: i) by only flipping one edge per step it scales slowly with the number of variables of the domain n, ii) it is prone to getting stuck in local optima. Moreover, we consider that the properties of BJP score have considerable potential for both further theoretical development, and applications.

6 Acknowledgements

This work was supported by Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) [PIP 2013 117], Universidad Nacional del Litoral (UNL) [CAI+D 2011 548] and Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT) [PICT 2014 2627] and [PICT-2012-2731].

Appendices

A - Correctness of BJP

Based on the developments in Section 3, and the analysis in Section 4, we see that the BJP score is a good measure of the fit of the estimated MN to the dataset. In this appendix we are concerned about the correctness of the method used by BJP to compute the posterior of structures. Thus, by correctness we mean that the probability computed by BJP is equivalent to the posterior probability of a MN structure.

In the formulation of the BJP score, the joint distribution of the MBs of G is calculated by computing the probabilities of conditional independence and dependence assertions contained in the MB of each variable of the domain. Our discussion in this appendix follows by demonstrating that all the members and non-members of each MB are unequivocally determined in (11), and therefore, that the joint posterior over these dependences and independences is equivalent to the posterior of the blankets. From [Schlüter et al., 2014, Definition 2], the *Markov blanket closure* is a set of independence and dependence assertions that are formally proven to correctly determine a MN structure. This set is obtained by determining the MB of each variable $X_i \in X$ with the following set of conditional independence and dependence assertions:

$$\Big\{ \langle X_i \perp X_j | B^i \rangle \ : \ X_j \notin B^i \Big\} \ \cup$$
$$\Big\{ \langle X_i \not\perp X_j | B^i \setminus \{X_j\} \rangle \ : \ X_j \in B^i \Big\}. \tag{14}$$

Clearly, this is exactly the same set used by BJP in (11) to compute the posterior of the MB of each variable of the domain. Since this set determines all members and non-members of a MB, the posterior of this set of assertions is equivalent to the posterior of the MB. Then, we demonstrate that such probabilities are correctly estimated by (12) and (13). We proceed by discussing their correctness separately for independence and dependence assertions.

Equation (12) computes the probability of independence between a variable and a non-adjacent variable,

Target structure	n	irr	$\mathcal{N}_D$	Hamming distance MPL	Hamming distance IB-score	Hamming distance BJP	Runtime MPL	Runtime IB-score	Runtime BJP
			250	13.11 (0.07)	12.36 (0.11)	**12.14** (0.14)	0.16 (0.00)	0.20 (0.00)	**0.06** (0.01)
			500	11.76 (0.06)	9.92 (0.09)	**9.42** (0.11)	0.14 (0.02)	0.29 (0.05)	**0.11** (0.01)
			1000	10.46 (0.05)	7.80 (0.11)	**7.20** (0.12)	**0.19** (0.02)	0.74(0.02)	0.25 (0.04)
Hub 1	16	392	2000	9.40 (0.06)	6.04 (0.11)	**5.40** (0.11)	**0.41** (0.06)	2.39 (0.08)	0.60 (0.07)
			4000	8.19 (0.05)	4.06 (0.12)	**3.94** (0.10)	**1.09** (0.017)	6.75 (0.22)	1.34 (0.02)
			8000	7.26 (0.05)	3.16 (0.10)	**2.88** (0.10)	2.908 (0.052)	17.53 (0.59)	**2.59** (0.02)
			250	27.22 (0.12)	25.73 (0.09)	**25.02** (0.12)	0.42 (4.94)	0.81 (0.01)	**0.39** (0.00)
			500	24.34 (0.11)	22.00 (0.11)	**19.98** (0.15)	**0.59** (0.00)	1.50 (0.01)	0.92 (0.01)
			1000	21.53 (0.10)	17.50 (0.12)	**15.41** (0.16)	**1.35** (0.02)	3.87 (0.04)	2.15 (0.02)
Hub 2	32	1916	2000	18.96 (0.08)	12.86 (0.13)	**11.63** (0.11)	**3.00** (0.05)	11.39 (0.14)	5.28 (0.05)
			4000	16.68 (0.08)	9.36 (0.12)	**8.36** (0.11)	**7.67** (0.10)	29.32 (0.36)	11.63 (0.09)
			8000	14.56 (0.07)	7.06 (0.10)	**6.96** (0.10)	**22.45** (0.28)	76.584 (1.03)	23.75 (0.18)
			250	60.49 (0.21)	56.55 (0.12)	**54.03** (0.18)	3.09 (0.03)	1.79 (0.02)	**1.37** (0.00)
			500	52.92 (0.19)	50.60 (0.14)	**44.88** (0.20)	4.90 (63.37)	4.96 (0.07)	**3.86** (0.05)
			1000	46.17 (0.19)	42.33 (0.19)	**36.35** (0.25)	**10.33** (0.11)	17.24 (0.22)	10.39 (0.12)
Hub 3	64	6624	2000	40.31 (0.18)	33.49 (0.24)	**29.21** (0.29)	**24.73** (0.28)	57.95 (0.81)	25.991 (0.38)
			4000	34.97 (0.18)	26.31 (0.25)	**22.47** (0.30)	**61.75** (0.66)	180.92 (3.02)	63.64 (0.83)
			8000	30.55 (0.17)	20.87 (0.29)	**19.44** (0.31)	207.48 (2.08)	627.50 (11.27)	**156.24** (3.15)
			250	134.28 (0.35)	120.11 (0.14)	**112.43** (0.28)	58.92 (0.32)	**5.86** (0.13)	8.31 (0.13)
			500	113.96 (0.28)	110.03 (0.24)	**97.25** (0.37)	78.53 (0.49)	26.56 (0.42)	**25.14** (0.37)
			1000	98.24 (0.29)	95.01 (0.29)	**78.39** (0.44)	129.33 (0.77)	101.26 (1.05)	**74.80** (0.77)
Hub 4	128	24496	2000	84.27 (0.26)	78.78 (0.34)	**61.35** (0.54)	259.68 (1.74)	331.32 (3.19)	**198.77** (2.14)
			4000	72.70 (0.23)	65.17 (0.52)	**52.11** (0.75)	777.84 (6.36)	1252.88 (19.89)	**473.05** (6.97)
			8000	62.59 (0.26)	52.95 (0.78)	**47.04** (1.03)	3102.53 (28.43)	4913.07 (89.81)	**1185.91** (23.83)

Table 2: Structures with hub topology: average and standard deviation of the Hamming distance and runtime (in seconds) over the 100 repetitions

Target structure	n	irr	$\mathcal{N}_D$	Hamming distance MPL	Hamming distance IB-score	Hamming distance BJP	Runtime MPL	Runtime IB-score	Runtime BJP
			250	12.35 (0.35)	11.30 (1.63)	**11.20** (1.23)	0.12 (0.01)	0.33 (0.11)	**0.12** (0.02)
			500	10.63 (0.26)	10.00 (1.45)	10.00 (1.59)	**0.11** (0.01)	0.40 (0.10)	0.16 (0.03)
			1000	**9.14** (0.28)	7.10 (1.64)	7.30 (1.15)	**0.25** (0.02)	0.76 (0.16)	0.35 (0.03)
Scale-free 1	16	364	2000	7.53 (0.23)	**5.10** (1.07)	5.20 (1.04)	**0.70** (63.75)	1.88 (0.41)	0.71 (0.10)
			4000	6.21 (0.22)	3.70 (0.75)	**3.50** (0.90)	1.90 (0.14)	3.87 (1.10)	**1.41** (0.13)
			8000	4.92 (0.22)	2.30 (1.00)	2.30 (1.11)	6.45 (0.71)	7.91 (1.85)	**2.91** (0.27)
			250	27.51 (0.45)	26.50 (1.74)	**25.88** (2.00)	**0.50** (0.03)	0.92 (0.22)	0.56 (0.24)
			500	24.08 (0.46)	22.40 (2.13)	**20.38** (2.72)	**0.78** (0.05)	1.34 (0.25)	0.95 (0.24)
			1000	20.82 (0.42)	18.30 (2.00)	**17.12** (2.15)	2.11 (0.16)	4.34 (1.15)	**1.73** (0.33)
Scale-free 2	32	1612	2000	18.27 (0.37)	13.60 (1.34)	**12.12** (1.60)	5.18 (0.35)	19.52 (8.80)	**5.20** (1.92)
			4000	16.13 (0.31)	**10.40** (1.77)	10.50 (2.03)	12.57 (0.81)	77.37 (36.80)	**10.51** (2.97)
			8000	14.41 (0.33)	**6.56** (1.70)	7.00 (1.28)	41.33 (3.86)	354.14 (207.98)	**25.32** (10.33)
			250	59.11 (0.91)	57.75 (5.67)	**55.33** (2.29)	4.73 (0.24)	3.83 (3.35)	**2.11** (1.10)
			500	50.14 (0.81)	52.00 (7.01)	**44.00** (8.64)	8.20 (0.45)	9.85 (4.38)	**6.06** (2.92)
			1000	43.05 (0.73)	43.25 (13.98)	**36.00** (11.04)	19.54 (1.03)	29.05 (21.09)	**13.87** (6.11)
Scale-free 3	64	6428	2000	36.71 (0.74)	33.50 (9.97)	**27.67** (8.26)	46.99 (2.34)	95.44 (49.23)	**46.06** (9.17)
			4000	31.37 (0.56)	26.25 (4.93)	**21.33** (2.29)	122.24 (6.49)	275.86 (69.80)	**99.06** (18.66)
			8000	27.52 (0.57)	19.00 (3.71)	**16.00** (7.15)	433.09 (22.47)	1124.33 (841.27)	**221.92** (4.05)
			250	131.42 (1.94)	123.20 (1.09)	**116.40** (2.73)	72.69 (3.47)	**6.71** (1.29)	12.50 (4.35)
			500	109.44 (1.75)	110.70 (2.96)	**101.00** (3.85)	106.37 (5.03)	36.51 (6.21)	**30.74** (12.14)
			1000	91.47 (1.58)	93.00 (4.02)	**83.10** (4.75)	196.18 (9.51)	140.10 (17.46)	**95.14** (31.07)
Scale-free 4	128	26188	2000	77.47 (1.40)	79.20 (5.12)	**64.50** (5.94)	429.12 (24.21)	403.99 (60.94)	**271.96** (93.17)
			4000	65.44 (1.30)	62.00 (4.99)	**46.50** (4.84)	1202.84 (92.93)	1469.52 (283.02)	**634.66** (95.49)
			8000	57.09 (1.21)	47.90 (3.87)	**34.30** (3.92)	5103.34 (531.26)	7650.26 (2037.82)	**1736.44** (437.66)

Table 3: Scale-free networks models: average and standard deviation of the Hamming distance and runtime (in seconds) over the 100 repetitions

conditioned on its MB, given the previously computed MBs and the dataset D. In this equation, for the case when $i < k$, which indexes over the variables for which the blanket posterior is not already computed, the posterior of the independence assertion $\langle \psi_i \perp \psi_k | B^{\psi_i} \rangle$ must be computed from data. It is performed by using the Bayesian statistical test of [Margaritis, 2005], that has been proven to be statistically consistent, since its mean square error tends to 0 as the dataset size tends to infinity. For the case when $i > k$, which indexes over the variables for which the blanket posterior is already computed, the independence assertion is inferred as 1, since its independence is determined by the MB of ψ_k, which is in the evidence $\left\{B^{\psi_j}\right\}_{j=0}^{i-1}$. By definition in (11), this case applies to all the variables $\psi_k \notin B^{\psi_i}$ (i.e., all the variables that are not connected to ψ_i). We argue the correctness for this inference by considering an intuitive equivalence commonly used by constraint-based approaches to perform independence tests that involve smaller number of variables [Koller and Friedman, 2009, p. 980]. If two variables X_i and X_k are not neighbors in G, then by applying the

Target structure	n	irr	$\mathcal{N}_D$	Hamming distance MPL	Hamming distance IB-score	Hamming distance BJP	Runtime MPL	Runtime IB-score	Runtime BJP
Karate	34	2044	250	58.60 (2.78)	51.91 (3.74)	**51.90** (3.59)	5.30 (1.85)	5.01 (1.20)	**1.78** (0.24)
			500	49.80 (2.26)	44.00 (4.92)	**42.20** (3.33)	12.01 (4.97)	14.59 (6.37)	**2.95** (0.29)
			1000	44.00 (2.18)	27.25 (5.25)	**26.00** (4.55)	22.78 (4.47)	213.57 (75.01)	**11.07** (3.09)
			2000	40.50 (1.24)	17.12 (4.89)	**11.30** (3.15)	**40.74** (3.74)	1220.47 (656.76)	51.54 (10.90)
			4000	38.00 (0.68)	7.88 (2.11)	**5.80** (1.93)	**118.66** (12.99)	9557.99 (3025.38)	195.52 (48.87)
			8000	36.60 (0.65)	**2.60** (0.49)	3.20 (0.82)	**320.12** (26.48)	30963.00 (3032.01)	665.70 (89.15)
Curtis-54	54	3140	250	76.50 (1.50)	77.00 (2.72)	**71.20** (2.37)	12.09 (0.51)	11.64 (1.57)	5.42 (0.34)
			500	64.40 (1.31)	59.10 (2.33)	**56.60** (2.00)	28.82 (1.79)	28.49 (3.23)	**11.33** (0.42)
			1000	52.40 (0.86)	40.10 (1.63)	**39.40** (2.15)	83.15 (3.25)	83.29(5.36)	**29.48** (1.10)
			2000	40.10 (0.82)	22.70 (2.33)	**18.90** (3.75)	244.77 (9.30)	278.23 (37.50)	**86.36** (6.45)
			4000	30.30 (1.12)	7.50 (1.55)	**4.40** (1.47)	689.46 (26.14)	1466.95 (599.15)	**240.60** (7.56)
			8000	24.00 (0.57)	**2.12** (0.81)	2.20 (0.64)	2015.04 (54.97)	4665.29 (788.77)	**742.01** (23.51)
Will-57	57	4156	250	79.50 (1.82)	81.90 (4.17)	**79.40** (4.25)	13.14 (0.52)	9.67 (1.38)	**5.69** (0.49)
			500	66.80 (1.34)	63.60 (2.27)	**60.70** (3.77)	31.49 (1.93)	25.19 (2.42)	**12.33** (0.92)
			1000	55.60 (1.08)	44.50 (2.21)	**42.50** (3.78)	85.83 (3.36)	75.41 (6.13)	**31.91** (2.33)
			2000	47.60 (0.53)	25.30 (2.40)	**23.80** (2.94)	232.10 (8.80)	245.99 (25.06)	**87.38 (4.42)**
			4000	38.30 (0.89)	10.70 (2.03)	**9.40** (4.22)	672.76 (19.39)	886.78 (123.50)	**274.24** (24.75)
			8000	28.90 (0.54)	**2.70** (0.53)	3.90 (1.75)	2383.00 (72.06)	3077.88 (418.76)	**787.45** (53.24)
Dolphins	62	6480	250	126.70 (4.00)	126.90 (5.18)	**125.20** (4.14)	24.02 (2.64)	12.46 (3.02)	**7.11** (1.35)
			500	106.60 (4.32)	106.10 (6.06)	**102.10** (5.10)	48.47 (5.52)	30.53 (5.83)	**16.73** (2.34)
			1000	88.50 (1.90)	71.60 (4.64)	**65.90** (3.84)	126.57 (12.52)	120.44 (13.49)	**55.37** (3.75)
			2000	74.20 (2.02)	50.60 (3.41)	**47.30** (3.52)	349.26 (23.90)	337.56 (26.13)	**144.14** (12.24)
			4000	63.00 (1.99)	32.50 (2.93)	**27.70** (3.14)	981.07 (65.15)	1092.66 (102.50)	**386.27** (25.09)
			8000	50.80 (1.60)	20.60 (1.94)	**12.90** (2.06)	3591.12 (153.37)	4171.51 (173.19)	**1331.72** (44.67)

Table 4: Real networks: average and standard deviation of the Hamming distance and runtime (in seconds) over the 100 repetitions

local Markov property of (4) once for each, we have that $\langle X_i \perp X_k | B^{X_i} \rangle$ and $\langle X_i \perp X_k | B^{X_k} \rangle$ hold. Therefore, the inference made is correct.

A similar argument can be given for the case of the dependence assertions. Equation (13) computes the probability of dependence between a variable and an adjacent variable conditioned on its remaining neighbors, given the previously computed MBs and the dataset D. Again, for the case when $i < k$, which indexes over the variables for which the blanket posterior is not already computed, the posterior of the dependence assertion must be computed from data. For the case when $i > k$, which indexes over the variables for which the blanket posterior is already computed, the dependence assertion is inferred as 1, since its dependence is determined by the MB of ψ_k, which is again in the evidence $\left\{B^{\psi_j}\right\}_{j=0}^{i-1}$. By definition in (11), this case applies to all the variables $\psi_k \in B^{\psi_i}$ (i.e., all the variables that are connected to ψ_i). Clearly, if two variables X_i and X_k are neighbors in G, there are no sets separating them in the graph. Therefore, the dependence assertion inferred is true.

B - Example of BJP score computation

This appendix shows a complete example of the computation of the BJP score for the graph of Figure 1. Consider this graph as the independence structure of a probability distribution $\Pr(V)$, with $n = 4$ variables $V = \{X_0, X_1, X_2, X_3\}$, represented by a MN. Given a dataset D, the BJP score can be computed by following the next steps:

a) Build the vector ψ, with the nodes sorted by their degree in ascending order: $\psi = (X_1, X_2, X_3, X_0)$.

b) By following (10), the computation of $BJP(G)$ is given by:

$$\begin{aligned} BJP(G) = \Pr\left(B^{X_1}\middle|D\right) &\times \Pr\left(B^{X_2}\middle|B^{X_1}, D\right) \\ &\times \Pr\left(B^{X_3}\middle|B^{X_1}, B^{X_2}, D\right) \\ &\times \Pr\left(B^{X_0}\middle|B^{X_1}, B^{X_2}, B^{X_3}, D\right). \end{aligned}$$

c) Compute each term of the above expression by following (11), resulting in:

$$\begin{aligned} \Pr\left(B^{X_1}\middle|D\right) = \Pr\left(\langle X_1 \perp X_2 | X_0\rangle\middle|D\right) &\times \Pr\left(\langle X_1 \perp X_3 | X_0\rangle\middle|D\right) \\ &\times \Pr\left(\langle X_1 \not\perp X_0 | \emptyset\rangle\middle|D\right). \end{aligned}$$

$$\begin{aligned} \Pr\left(B^{X_2}\middle|B^{X_1}, D\right) = \Pr\left(\langle X_2 \perp X_1 | X_0\rangle\middle|B^{X_1}, D\right) &\times \Pr\left(\langle X_2 \perp X_3 | X_0\rangle\middle|B^{X_1}, D\right) \\ &\times \Pr\left(\langle X_2 \not\perp X_0 | \emptyset\rangle\middle|B^{X_1}, D\right). \end{aligned}$$

$$\Pr\left(B^{X_3}\middle|B^{X_1},B^{X_2},D\right) = \Pr\left(\langle X_3 \perp X_1|X_0\rangle\middle|B^{X_1},B^{X_2},D\right) \times \Pr\left(\langle X_3 \perp X_2|X_0\rangle\middle|B^{X_1},B^{X_2},D\right) \times \Pr\left(\langle X_3 \not\perp X_0|\emptyset\rangle\middle|B^{X_1},B^{X_2},D\right).$$

$$\Pr\left(B^{X_0}\middle|B^{X_1},B^{X_2},B^{X_3},D\right) = \Pr\left(\langle X_0 \not\perp X_1|X_2,X_3\rangle\middle|B^{X_1},B^{X_2},B^{X_3},D\right) \times \Pr\left(\langle X_0 \not\perp X_2|X_1,X_3\rangle\middle|B^{X_1},B^{X_2},B^{X_3},D\right) \times \Pr\left(\langle X_0 \not\perp X_3|X_1,X_2\rangle\middle|B^{X_1},B^{X_2},B^{X_3},D\right).$$

d) By replacing Equations (12) and (13) in the factors of the above expression, one half of the tests can be inferred, and only the following probabilities must be computed from data by using the Bayesian statistical test:

$$\Pr\left(B^{X_1}\middle|D\right) = \Pr\left(\langle X_1 \perp X_2|X_0\rangle\middle|D\right) \times \Pr\left(\langle X_1 \perp X_3|X_0\rangle\middle|D\right) \times \Pr\left(\langle X_1 \not\perp X_0|\emptyset\rangle\middle|D\right).$$

$$\Pr\left(B^{X_2}\middle|B^{X_1},D\right) = 1 \times \Pr\left(\langle X_2 \perp X_3|X_0\rangle\middle|D\right) \times \Pr\left(\langle X_2 \not\perp X_0|\emptyset\rangle\middle|D\right).$$

$$\Pr\left(B^{X_3}\middle|B^{X_1},B^{X_2},D\right) = 1 \times 1 \times \Pr\left(\langle X_3 \not\perp X_0|\emptyset\rangle\middle|D\right).$$

$$\Pr\left(B^{X_0}\middle|B^{X_1},B^{X_2},B^{X_3},D\right) = 1 \times 1 \times 1.$$

The inferred tests are the 1s at each equation.

References

Agresti, 2002. Agresti, A. (2002). Categorical Data Analysis. Wiley, 2nd edition.

Albertson, 1997. Albertson, M. O. (1997). The irregularity of a graph. Ars Combinatoria, 46:219–225.

Aliferis et al., 2010. Aliferis, C., Statnikov, A., Tsamardinos, I., Mani, S., and Koutsoukos, X. (2010). Local Causal and Markov Blanket Induction for Causal Discovery and Feature Selection for Classification Part I: Algorithms and Empirical Evaluation. JMLR, 11:171–234.

Barabasi and Bonabeau, 2003. Barabasi, A. and Bonabeau, E. (2003). Scale-free netwroks. Scientific American.

Barahona, 1982. Barahona, F. (1982). On the computational complexity of Ising spin glass models. Journal of Physics A: Mathematical and General, 15(10):3241–3253.

Besag, 1972. Besag, J. E. (1972). Nearest-neighbour systems and the auto-logistic model for binary data. Journal of the Royal Statistical Society. Series B (Methodological), pages 75–83.

Bromberg et al., 2009. Bromberg, F., Margaritis, D., and Honavar, V. (2009). Efficient Markov network structure discovery using independence tests. JAIR, 35:449–485.

Claeskens et al., 2015. Claeskens, G., Pircalabelu, E., and Waldorp, L. (2015). Constructing graphical models via the focused information criterion. In Modeling and Stochastic Learning for Forecasting in High Dimensions, pages 55–78. Springer.

Cochran, 1954. Cochran, W. (1954). Some methods of strengthening the common χ tests. Biometrics., page 10:417451.

Cover and Thomas, 1991. Cover, T. and Thomas, J. (1991). Elements of information theory. Wiley-Interscience, New York, NY, USA.

Davis and Domingos, 2010. Davis, J. and Domingos, P. (2010). Bottom-up learning of Markov network structure. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), pages 271–278.

Davis and Hu, 2011. Davis, T. A. and Hu, Y. (2011). The university of florida sparse matrix collection. ACM Transactions on Mathematical Software (TOMS), 38(1):1.

Della Pietra et al., 1997. Della Pietra, S., Della Pietra, V., and Lafferty, J. (1997). Inducing Features of Random Fields. IEEE Trans. PAMI., 19(4):380–393.

Dimitrov et al., 2014. Dimitrov, D., Brandt, S., and Abdo, H. (2014). The total irregularity of a graph. Discrete Mathematics & Theoretical Computer Science, 16.

Ganapathi et al., 2008. Ganapathi, V., Vickrey, D., Duchi, J., and Koller, D. (2008). Constrained Approximate Maximum Entropy Learning of Markov Random Fields. In Uncertainty in Artificial Intelligence, pages 196–203.

Hammersley and Clifford, 1971. Hammersley, J. and Clifford, P. (1971). Markov fields on finite graphs and lattices.

Hwang and Kim, 2015. Hwang, W. and Kim, J. (2015). Markov network-based unified classifier for face recognition. IEEE Transactions on Image Processing, 24(11):4263–4275.

Koller and Friedman, 2009. Koller, D. and Friedman, N. (2009). Probabilistic Graphical Models: Principles and Techniques. MIT Press.

Larrañaga and Lozano, 2002. Larrañaga, P. and Lozano, J. (2002). Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Pubs.

Lauritzen, 1982. Lauritzen, S. L. (1982). Lectures in contingency tables. University of Aalborg Press, Aalborg, Denmark, 2nd edition.

Lee et al., 2006. Lee, S., Ganapathi, V., and Koller, D. (2006). Efficient structure learning of Markov networks using L1-regularization. In NIPS.

Li, 2001. Li, S. (2001). Markov random field modeling in image analysis. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Li et al., 2015. Li, Y., Pearl, S. A., and Jackson, S. A. (2015). Gene networks in plant biology: approaches in reconstruction and analysis. Trends in plant science, 20(10):664–675.

Lowd and Davis, 2014. Lowd, D. and Davis, J. (2014). Improving markov network structure learning using decision trees. Journal of Machine Learning Research, 15:501–532.

Lowd and Rooshenas, 2015. Lowd, D. and Rooshenas, A. (2015). The libra toolkit for probabilistic models. arXiv preprint arXiv:1504.00110.

Margaritis, 2005. Margaritis, D. (2005). Distribution-Free Learning of Bayesian Network Structure in Continuous Domains. In Proceedings of AAAI.

Margaritis and Bromberg, 2009. Margaritis, D. and Bromberg, F. (2009). Efficient Markov Network Discovery Using Particle Filter. Comp. Intel., 25(4):367–394.

Margaritis and Thrun, 2000. Margaritis, D. and Thrun, S. (2000). Bayesian network induction via local neighborhoods. In Proceedings of NIPS.

McCallum, 2003. McCallum, A. (2003). Efficiently inducing features of conditional random fields. In Proceedings of Uncertainty in Artificial Intelligence (UAI).

Nyman et al., 2014. Nyman, H., Pensar, J., Koski, T., and Corander, J. (2014). Context-specific independence in graphical log-linear models. Computational Statistics, pages 1–20.

Pearl, 1988. Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers, Inc.

Peng et al., 2016. Peng, F., Lu, J., Wang, Y., Yi-Da Xu, R., Ma, C., and Yang, J. (2016). N-dimensional markov random field prior for cold-start recommendation. Neurocomputing, 191:187–199.

Pensar et al., In Press. Pensar, J., Nyman, H., Niiranen, J., and Corander, J. (In Press). Marginal pseudo-likelihood learning of Markov network structures. Bayesian analysis.

Rautenbach and Schiermeyer, 2006. Rautenbach, D. and Schiermeyer, I. (2006). Extremal problems for imbalanced edges. Graphs and Combinatorics, 22(1):103–111.

Schlüter, 2012. Schlüter, F. (2012). A survey on independence-based Markov networks learning. Artificial Intelligence Review, pages 1–25.

Schlüter et al., 2014. Schlüter, F., Bromberg, F., and Edera, A. (2014). The IBMAP approach for Markov network structure learning. Annals of Mathematics and Artificial Intelligence, pages 1–27.

Schmidt et al., 2008. Schmidt, M., Murphy, K., Fung, G., and Rosales, R. (2008). Structure learning in random fields for heart motion abnormality detection. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pages 1 –8.

Shakya et al., 2012. Shakya, S., Santana, R., and Lozano, J. (2012). A markovianity based optimisation algorithm. Genetic Programming and Evolvable Machines, 13(2):159–195.

Silva and Zhao, 2016. Silva, T. and Zhao, L. (2016). Machine Learning in Complex Networks. Springer International Publishing.

Spirtes et al., 2000. Spirtes, P., Glymour, C., and Scheines, R. (2000). Causation, Prediction, and Search. Adaptive Computation and Machine Learning Series. MIT Press.

Tsamardinos et al., 2003. Tsamardinos, I., Aliferis, C., and Statnikov, A. (2003). Algorithms for large scale Markov blanket discovery. In FLAIRS.

Van Haaren and Davis, 2012. Van Haaren, J. and Davis, J. (2012). Markov network structure learning: A randomized feature generation approach. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence.

Van Haaren et al., 2013. Van Haaren, J., Davis, J., Lappenschaar, M., and Hommersom, A. (2013). Exploring disease interactions using markov networks. In Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence.

Wan et al., 2015. Wan, Y.-W., Allen, G. I., Baker, Y., Yang, E., Ravikumar, P., Liu, Z., and Wan, M. Y.-W. (2015). Package xmrf.