Efficient identification of independence networks using mutual information

Davide Bacciu, Terence A. Etchells, Paulo J. G. Lisboa & Joe Whittaker

Computational Statistics

ISSN 0943-4062
Volume 28
Number 2

Comput Stat (2013) 28:621-646
DOI 10.1007/s00180-012-0320-6

ISSN 0943-4062

Computational Statistics 28·2

2013

ORIGINAL PAPERS

(*Contents continued on outside back cover*)

Springer

Springer

Comput Stat (2013) 28:621–646
DOI 10.1007/s00180-012-0320-6

ORIGINAL PAPER

Efficient identification of independence networks using mutual information

Davide Bacciu · Terence A. Etchells ·
Paulo J. G. Lisboa · Joe Whittaker

Received: 28 July 2011 / Accepted: 9 March 2012 / Published online: 20 March 2012

Abstract Conditional independence graphs are now widely applied in science and industry to display interactions between large numbers of variables. However, the computational load of structure identification grows with the number of nodes in the network and the sample size. A tailored version of the PC algorithm is proposed which is based on mutual information tests with a specified testing order, combined with false negative reduction and false positive control. It is found to be competitive with current structure identification methodologies for both estimation accuracy and computational speed and outperforms these in large scale scenarios. The methodology is also shown to approximate dense networks. The comparisons are made on standard benchmarking data sets and an anonymized large scale real life example.

Keywords Bayesian networks · Constraint based search · Dense networks · False discovery rate · False negative reduction · Graphical models · Mutual information · PC algorithm · Skeleton

D. Bacciu (✉)
Dipartimento di Informatica, Università di Pisa, Pisa, Italy
e-mail: bacciu@di.unipi.it

T. A. Etchells · P. J. G. Lisboa
School of Computing and Mathematical Sciences,
Liverpool John Moores University, Liverpool, UK
e-mail: t.a.etchells@ljmu.ac.uk

P. J. G. Lisboa
e-mail: P.J.Lisboa@ljmu.ac.uk

J. Whittaker
Department of Mathematics and Statistics, Lancaster University, Lancaster, UK
e-mail: joe.whittaker@lancaster.ac.uk

1 Introduction

Efficient identification of the skeleton of a Bayesian network is important in the context of large numbers of variables and large sample size. Applications such as this are found in data mining, where the co-occurrence and synchronization of events is analyzed to provide insight into business and marketing processes, and are also typical of biological and social networks in domains as diverse as public health and bioinformatics. Software performance is crucial for achieving both convergence speed and identification accuracy, that ideally should scale linearly in the number of variables and of observations.

The ultimate goal of structure identification algorithms is to maximize the quality of the reconstructed networks while maintaining a feasible computational complexity. The basic PC algorithm, Algorithm 1 below, suggests certain key aspects related to statistical hypothesis testing directly influence performance. These include (i) the choice of the independence test statistic, (ii) the issues raised by multiple testing relating to the rates of false negative and false positive decisions, (iii) the order in which the tests are executed, and (iv) the problems caused by a large sample size.

Algorithm 1 The Vanilla PC Algorithm (Spirtes et al. 2000)

```
Require: Dataset $D$, Test level $\alpha$
Initialize a fully connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
for all $i \in \mathcal{V}$ do {Marginal independence tests}
  for all $j \in \text{ne}(i)$ do
    IsIndependent $\Leftarrow$ TestIndependence($D, \alpha, i, j$)
    if IsIndependent then
      $\mathcal{E} \Leftarrow \mathcal{E} \setminus e_{ij}$ {Remove the edge from the graph}
      $\text{ne}(i) \Leftarrow \text{ne}(i) \setminus j$, $\text{ne}(j) \Leftarrow \text{ne}(j) \setminus i$ {Update the neighbors lists}
    end if
  end for
end for
$K \Leftarrow 1$
repeat {Order independence tests by size of conditioning set, $K$}
  for all $i \in \mathcal{V}$ do
    for all $j \in \text{ne}(i)$ do
      $\mathcal{A} \Leftarrow$ ConditionSet($\text{ne}(i) \setminus j, K$) {Construct potential conditioning sets}
      IsIndependent $\Leftarrow$ False
      repeat {Seeks a certificate of exclusion from neighbors of $i$}
        $A \Leftarrow$ select subset from $\mathcal{A}$
        IsIndependent $\Leftarrow$ TestCondIndependence($D, \alpha, i, j, A$)
        $\mathcal{A} \Leftarrow \mathcal{A} \setminus A$
      until $\mathcal{A} = \emptyset$ or isIndependent
      if IsIndependent then
        $\mathcal{E} \Leftarrow \mathcal{E} \setminus e_{ij}$ {Remove the edge from the graph}
        $\text{ne}(i) \Leftarrow \text{ne}(i) \setminus j$, $\text{ne}(j) \Leftarrow \text{ne}(j) \setminus i$ {Update the neighbors lists}
      end if
    end for
  end for
  $K \Leftarrow K + 1$
until there are no tests of order $K$ in $\mathcal{G}$
return Undirected graph $\mathcal{G}$
```

Our contribution is to provide an extensive experimental assessment of structure identification methodology intended to elucidate which procedures offer the best trade-off between computational effort and quality of the reconstructed network in large-scale scenarios. In particular we study the interactions arising from the combined use of such computational procedures. The procedures have been implemented in a Matlab package, herein called *CImap* for conditional independence map, and is evaluated against state-of-the-art models from the literature, mostly implemented as part of the *Causal Explorer* package (Aliferis et al. 2003). In particular, we consider constraint-based approaches such as the baseline PC algorithm (Spirtes and Meek 1995) and the computationally lightweight ARACNE method (Algorithm for the Reconstruction of Accurate Cellular Networks) (Margolin et al. 2006) that is the most widely used approach for the reconstruction of high dimensional gene regulatory networks. Greedy search-and-score approaches are also considered, including Greedy Search (GS) (Meek 1997) and the Sparse Candidate algorithm (SCA) (Friedman et al. 1999), where the former constraints the search space to that of the Markov equivalence classes for DAGs (Spirtes and Meek 1995), while the latter constraints the search procedure by allowing each node to have at most k parents. Finally, we also consider an hybrid algorithm, the Max-Min Hill Climbing (MMHC) (Tsamardinos et al. 2006), that uses an optimized constraint-based approach to reconstruct a candidate skeleton that is later refined by an hill-climbing search guided by a Bayesian score. The MMHC approach is, currently, the best structure finding algorithm in terms of tradeoff between computational complexity and quality of the reconstructed network. Both accuracy and scalability of all the models above have been evaluated in benchmark tests and on a proprietary large-scale scenario comprising some 300 variables and 40 K observations, extending the range of the experimental variables significantly beyond the scale of currently published performance evaluations for high order structure finding in dense networks.

Plan of the paper After a description of the PC algorithm in Sect. 2 we argue that for categorical data the mutual information is the most effective test statistic for conditional independence, and give a rationale for its use in Sect. 3.1. In Sects. 3.2 and 3.3, we discuss multiple test procedures based on rigorous methodologies for bounding Type I and Type II errors, respectively.

The issue of bounding Type I and Type II errors in skeleton identification is largely discussed in literature (see, e.g., Tsamardinos and Brown 2008 and Fast et al. 2008): Sects. 3.2 and 3.3 build on previous works to construct efficient computational routines that can be effectively applied within constraint based approach. Sections 3.5 and 3.4, on the other hand, present two novel methodological contribution of this work which have never been tackled with in literature, that are a non-arbitrary ordering for independence testing and a novel null independence hypothesis for large sample sizes, respectively.

If the order in which independence tests are performed is arbitrary, the discovered skeleton is (usually) not reproducible. Algorithm 1, for instance, implicitly assumes that an ordering exists among the nodes, but no guidance is given. In Sect. 3.5, we suggest a testing policy based on testing the weakest first (TWF), which imposes a non-arbitrary ordering of tests and results in a notable improvement in the network reconstruction performance.

The issue of large sample size is that small departures from an independence hypothesis are flagged as edges, leading to graphs with extremely dense connectivity patterns. Dense graphs also have a high computational cost. Section 3.4 discusses a modification of the null independence hypothesis that leads to an efficient sparse approximation of a dense networks. The experimental results are reported in Sect. 4 and the paper ends with a discussion in the final section.

2 The PC algorithm

Independence networks, also known as graphical models and discussed in the texts of Koller and Friedman (2009), Jensen and Nielsen (2007), Lauritzen (1996), and Whittaker (1990), have been widely applied to factorise the joint distribution of multivariate data into a product of conditional dependencies which are represented by edges in a graph. The topology of the graph serves to generate insights about relationships inherent in the data and, potentially, for inference modelling.

A Bayesian network (BN) is a graphical model $(\mathcal{G}, \mathsf{P})$, where P is a joint probability of random variables $X_{\mathcal{V}} = (X_1, \ldots, X_p)$ associated with nodes $\mathcal{V} = \{1, \ldots, p\}$. In the context of discrete data, each X_i takes values in a finite set $\mathcal{X}_i$ of size $|\mathcal{X}_i|$. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}^{dir})$ is a directed acyclic graph (DAG) whose edges $\mathcal{E}^{dir}$ encode the joint probabilistic relationships among the p random variables. The graph is a visual representation of the joint distribution of the data, where a directed edge e_{ij} from node i to j indicates that i is the parent of j as part of a conditional dependence relationship between the two nodes.

The skeleton of the graph is the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with undirected edges replacing directed edges. The predicate $I_{\mathsf{P}}(i, j|A)$ is used to denote the conditional independence, with respect to the joint probability P, of X_i and X_j given a subset of random variables X_A, $A \subset \mathcal{V}$. An alternative notation is $X_i \perp\!\!\!\perp_{\mathsf{P}} X_j | X_A$ (Dawid 1979). For each missing edge in the skeleton there is a triple i, j, A for which the conditional independence relationship $I_{\mathsf{P}}(i, j|A)$ is true. If all the conditional independencies in P, and only these, correspond to missing edges in the graph then $\mathcal{G}$ and P are faithful to each other. We consider the task of skeleton identification, learning the structure $\mathcal{E}$ of a Bayesian network from data $D = (d_1, d_2, \ldots, d_N)$ sampled from the distribution P. Skeleton identification maps the empirical conditional relationships between the random variables of $X_{\mathcal{V}}$ onto the graph.

The essential idea of the PC algorithm is the observation, first made by Spirtes et al. (1993), that a faithful BN has an edge between i and j in the skeleton if and only if $I_{\mathsf{P}}(i, j|A)$ does not hold for all conditioning subsets A of the remaining nodes. The problem is then to find an efficient procedure for computing such independence tests without incurring in a combinatorial explosion from the number of subsets A.

The starting point of our work is the vanilla implementation of the PC algorithm (Spirtes et al. 2000), described in Algorithm 1. This procedure takes as input the dataset D and the test level α, and returns an undirected graph $\mathcal{G}$ (the skeleton). The term ne (i) indicates the set of neighbors of node i, i.e. those sharing an edge with i. In particular, the function *ConditionSet()* in line 15 generates all the possible instantiations of a K-th order conditioning set A for an edge e_{ij}, by choosing the conditioning variables from the neighbors of i and j. The unconditional tests (lines 2–10) are separated from the higher order conditional independence tests (lines 12–29) highlighting the role of the first pass of unconditional tests in determining the efficiency of the algorithm. The procedure starts by generating a fully connected network whose edges are pruned, in the first pass, based on $O(p^2)$ pairwise independence tests. If such a preliminary pruning phase does not remove enough edges the second pass of the algorithm can result in an exponential number of conditional tests, i.e. $O(p^2 \cdot 2^{p-2})$.

3 Efficient structure identification with the PC algorithm

3.1 Tests based on mutual information

A popular approach in the literature for testing conditional independence between categorical variables are the likelihood ratio tests, known as G^2 statistics (Bishop et al. 1975; Spirtes et al. 2000),

$$G^2(i, j|A) = 2 \sum_{x_i, x_j, x_A} n_{ijA}(x_i, x_j, x_A) \log \frac{n_{ijA}(x_i, x_j, x_A) n_A(x_A)}{n_{iA}(x_i, x_A) n_{jA}(x_j, x_A)}$$

where $n_A(x_A)$ denotes the frequency of the value x_A taken by X_A in the data set $\mathcal{D}$. The $G^2(i, j|A)$ statistic, under the null hypothesis $I_P(i, j|A)$, is asymptotically distributed as chi-squared with $df = (|\mathcal{X}_i| - 1)(|\mathcal{X}_j| - 1)(|\mathcal{X}_A|)$ degrees of freedom where $\mathcal{X}_A$ denotes the domain for the conditioning variables. A different derivation of this independence test is based on the conditional mutual information

$$\mathcal{I}(i, j|A) = \sum_{x_i, x_j, x_A} p_{ijA}(x_i, x_j, x_A) \log_2 \frac{p_{ij|A}(x_i, x_j|x_A)}{p_{i|A}(x_i|x_A) p_{j|A}(x_j|x_A)}, \qquad (1)$$

where p is the joint mass function determined by P. When $\mathcal{I}(i, j|A) = 0$ the joint conditional in the numerator of the log term factorises into the denominator, otherwise $\mathcal{I}(i, j|A) > 0$. It measures the amount of dependency in bits, with the larger the value of $\mathcal{I}$ implying the stronger the dependency. A mutual information test retains the edge when the statistic is large.

The mutual information $\hat{\mathcal{I}}$ is estimated from a point estimate n_{ijA}/N of the mass function p_{ijA} that results in $\hat{\mathcal{I}}$ being a multiple of the G^2 statistic, $G^2/2N \log(2)$. The essential difference is the division by the sample size N, so while G^2 is easily interpretable in hypothesis testing, mutual information is an estimate of association strength.

The distribution of conditional mutual information estimates $\hat{\mathcal{I}}$ is approximated using the chi-squared distribution (see Goebel et al. 2005 for a detailed discussion), that is

$$2N \log(2) \hat{\mathcal{I}}(i, j|A) \sim \chi^2_{df} \qquad (2)$$

where N is the sample size and df the degrees of freedom given above. Hence it is possible to obtain a certificate of exclusion for an edge e_{ij} by performing a chi-squared test on the null hypothesis that the two random variables are independent given X_A. The p value for the hypothesis is

$$p_{ijA} = \mathsf{P}(\hat{\mathcal{I}}(i, j|A) > m|H_0), \tag{3}$$

where m is the observed value of the $\hat{\mathcal{I}}$ statistic. For instance, the test of the independence $I_{\mathsf{P}}(i, j|k)$ is performed by computing the p value p_{ijk}: given a significance level α the edge e_{ij} is deleted if $p_{ijk} > \alpha$ and retained otherwise.

Mutual information has been used to define several strategies for the identification of statistical dependencies in generic independence networks and there are several reasons favoring the choice of conditional mutual information as a test statistics,

1. Firstly, because of the principled information theoretic derivation of mutual information, it delivers a quantity measurable in scientific units (bits). It is intimately related to the Kullback–Leibler divergence and has universal application to random variables on different scales of measurement, for instance, binary, discrete, continuous or mixed.
2. Mutual information is a measure of strength which can be exploited to display edge relevance in the final graph and to consistently resolve testing order issues within constraint based algorithms. In Sect. 3.5, we define a strategy for eliminating weakest edges first using mutual information.
3. The mutual information statistic allows the specification of more interesting null distributions than complete independence, appropriate when there are small dependencies between all nodes, for instance, that might occur in a latent variable model where the underlying variable is not observed.

3.2 False negative reduction

False negatives (FN) result in over-constrained structures missing relevant dependencies between the random variables. Many FNs are caused by a failure in the independence test due to noise as a consequence of insufficient data to pick out the signal, the effect size. In structure identification, rules of thumb are proposed for false negative reduction (FNR) that either prevent testing the hypothesis if there is insufficient data, e.g. if the ratio of the sample size to the degrees of freedom of the test is less than 5 in G^2 tests (Spirtes et al. 2000); or adjust the independence threshold with respect to the sample size (Cheng et al. 2002). Such rules of thumb, although helping to reduce the impact of false negatives, take no account of the effect sizes present in the data.

Fast et al. (2008) suggest a procedure that tailors the power to a desired effect size. The procedure determines an acceptable number of degrees of freedom of the test to ensure that a hypothesis is tested with a minimum power of at least $1 - \beta$, given a false negative proportion β, test level α, sample size N, and a desired effect size w. Since the power decreases as the degrees of freedom increase, the procedure computes an upper bound for the degrees of freedom of the test: if this threshold is exceeded, the algorithm does not perform the test. The power of the conditional mutual information test statistic is evaluated from a non-central chi-squared distribution with non-centrality $\lambda = 2N \log(2)\, w^2$ (Goebel et al. 2005).

Algorithm 2 describes the power correction procedure implemented for the tests of conditional mutual information, where χ^{-2} and χ^2_{nc} are the inverse and non-central chi-squared distributions, respectively. This function, which is executed just once, outputs the allowed degrees of freedom dof for the conditional mutual information tests. The power is calculated for each degree of freedom until it drops below level $1-\beta$, which gives the maximum allowed value dof. The result of Algorithm 2 depends closely on the choice of the effect size w: this parameter can be set either by cross validation or by selecting the most appropriate value from the suggested effects described in Fast et al. (2008) for varying sample sizes.

Algorithm 2 Power Correction for Tests of Conditional Mutual Information

```
Require: Sample size $N$, test level $\alpha$, FN bound $\beta$, effect size $w$
$dof \Leftarrow 0$
while ($power \geq 1-\beta$) and ($dof \leq$ MAXDOF) do
    $dof \Leftarrow dof + 1$
    $c \Leftarrow \chi^{-2}(1-\alpha, dof)$
    $power \Leftarrow P(2N\log(2)\, w^2 > c)$
end while
return Degree of freedom bound $dof$
```

FNR is achieved by choosing not to perform the conditional independence test of $I_P(i, j|A)$ when the degrees of freedom $(|\mathcal{X}_i|-1)(|\mathcal{X}_j|-1)(\prod_{k\in A}|\mathcal{X}_k|)$ exceed dof, and thus includes the edge e_{ij} by default.

3.3 Bounding the false positive rate

A false positive results in the estimated network having an additional edge compared to the true network. While the probability of incurring in a false positive on a single test is given by the significance level α, the false positive rate accumulated over the whole network typically exceeds this. In structure identification multiple tests occur because of multiple edges and because tests may be repeated on the same edge for different conditioning sets.

More formally, given a node i a constraint-based algorithm may perform a single test of independence of $I_{\mathsf{P}}(i, j|A_k)$ for each neighboring node j conditioned on a set A_k, with an associated p value p_{ijk}. The independence test $p_{ijk} > \alpha$ ensures that the probability of false detection on the single hypothesis $I_{\mathsf{P}}(i, j|A_k)$ is α, but does not provide a bound to the proportion of false positives in the cumulated tests

$$\{p_{ijk} > \alpha; \quad j \in V,\; A_k \subset V,\; k = 1, \ldots, k_o\}$$

for node i. The measurement of the overall error rate when testing such multiple hypotheses is consistently more complicated than for single tests. Tsamardinos and Brown (2008) describe a theoretical bound for such detection errors that uses the idea of False Discovery Rate (FDR) developed by Benjamini and Hochberg (1995). Here we exploit the work in Tsamardinos and Brown (2008) to provide three alternative testing procedures that can be used to reduce the number of false positives in BN structure learning.

An FDR procedure controls the expected proportion of false positives rather than the probability of at least one false positive in the set of all hypotheses tested. In network identification, the FDR-induced significance level α^* can, in principle, be different for each node or for each edge in the network. To apply the FDR at the node-level, we fix a target node i on which the multiple tests are performed. First consider the unconditional hypotheses $I_{\mathsf{P}}(i, j)$ with associated p values p_{ij}, and index these over $j (\neq i)$ into increasing order $p_{i(1)} \leq p_{i(2)} \leq, \ldots, \leq p_{i(J-1)}$. The Benjamini and Hochberg (1995) condition that enforces the desired FDR level α^{fdr}, requires rejecting all hypotheses $I_{\mathsf{P}}(i, j)$ whose p value p_{ij} is smaller than

$$\alpha_i^* = \max_j \left\{ p_{i(j)}; \quad p_{i(j)} \leq \frac{j}{J-1} \alpha^{fdr} \right\}. \tag{4}$$

An alternative, tighter bound to the FDR is the Benjamini and Yekutieli (2001) criterion

$$\alpha_i^* = \max_j \left\{ p_{i(j)}; \quad p_{i(j)} \leq \frac{j}{J \sum_{j'=1}^{J} \frac{1}{j'}} \alpha^{fdr} \right\}. \tag{5}$$

More generally, each edge e_{ij} is coupled with a set of values

$$\mathcal{P}_{ij} = \{p_{ijk}; \quad A_k \subset V, \; k = 1, \ldots, k_o\}$$

corresponding to the conditional independence hypotheses $I_{\mathsf{P}}(i, j|A_k)$, which includes the unconditional case $A_k = \emptyset$. However, Tsamardinos and Brown (2008) provide a theoretical argument showing that the p value set $\mathcal{P}_{ij}$ can be bounded by the smallest p value obtained when conditioning on a subset A_k for which independence $I_P(i, j|A_k)$ holds, i.e.

$$p_{ij}^* = \min_k \left\{ p_{ijk} \in \mathcal{P}_{ij}; \quad p_{ijk} > \alpha \right\},$$

and it suffices to apply the FDR procedure in (4) replacing p_{ij} with p_{ij}^*.

An alternative approach, not mentioned by Tsamardinos and Brown (2008), is to apply the FDR procedure at the edge-level by determining the most appropriate α_{ij}^* for each edge e_{ij} in the network. Consider the edge e_{ij} and the independence tests $I_{\mathsf{P}}(i, j|A_k)$ given candidate conditioning sets $A_1, \ldots, A_k, \ldots, A_{k_o}$. The p values p_{ijk} can be sorted in increasing order $p_{ij(k)}$ as before, and the FDR procedure in (4) can be applied to determine the α_{ij}^*. If the test level α_{ij}^* is undefined, i.e. when the set in (4) is empty, then edge e_{ij} is retained, otherwise it is pruned. Notice that with the edge-level approach, the unconditional tests (i.e. $A_k = \emptyset$) reduce to a single test of independence, and there is no advantage in applying an FDR procedure.

We have implemented three FDR control policies: *standard FDR*, *interleaved FDR* and *per-edge FDR*, where the first two are node-based and the latter one is edge-based.

The standard FDR procedure is a straightforward extension of the results in Tsamardinos and Brown (2008) and is applied after the convergence of Algorithm 1 (possibly extended with the FNR control) in order to get rid of eventual false positive edges that have not yet been pruned by the vanilla PC algorithm.

The interleaved FDR procedure, as the name suggests, is interleaved with the steps of the vanilla PC algorithm of order K. At the first pass of Algorithm 1, which computes all the unconditional tests, we apply the FDR procedure to their p values and prune the corresponding edges. Then, the order $|A_k| = K = 1$ tests are all performed and the edge p values are adapted by keeping track of the minimum p^*_{ij} at level 1; the FDR procedure is thus applied and the false positives are pruned. Such a process is iterated for all the orders of test until the algorithm converges. The rationale behind the design of the interleaved FDR procedure is to anticipate as much as possible the pruning of the false positives in order to (i) reduce the computational burden of the algorithm and (ii) reduce the risk of deleting a true edge because of the presence of false positives.

The per-edge FDR is a fairly straightforward implementation of the edge-level process described above. First, the vanilla procedure in Algorithm 1 is modified so that at order $K = 0$ all p values for the unconditional tests are computed, the FDR procedure is performed on these p values, and the network structure is pruned accordingly. This step has the same computational cost of the vanilla version. At orders $K \geq 1$ and for each tested edge e_{ij}, the algorithm accumulates the p values for the conditional tests and applies the edge-level FDR process to obtain the α^*_{ij} level, pruning edge e_{ij} if such a test level is undefined.

The three FDR procedures discussed above have been implemented, together with the FDR bound in (4). In Sect. 4, we compare the experimental performance of the different FDR settings on freely available structure learning benchmarks.

3.4 Approximating dense networks

With large samples it is possible that small departures from independence become noticeable so that the empirical distribution of the marginal mutual information differs consistently from chi-squared. One explanation is a small variation in the probability that $X_i = 1$ over the sampling units. A consequence is that the null distribution of the mutual information is no longer chi-squared and, at the first stage of the PC algorithm, most edges are retained in the graph which, in consequence, is dense. Such a result is not necessarily useful nor informative for prediction or understanding: rather than retaining these edges, a different approach is to retain only those edges that reach a given threshold. In this sense, the discovered network is only a sparse approximation to the true graph of the underlying skeleton.

Taking a thresholding perspective easily slots into the current search framework by replacing the assumption at (2) of a null chi-squared distribution by that of a Normal distribution. From (1), the sample estimate of the marginal mutual information is

$$m_{ij} = \frac{1}{N} \sum_{x_i, x_j} n_{ij}(x_i, x_j) \left[\log_2 \left(\frac{n_{ij}(x_i, x_j) N}{n_i(x_i) n_j(x_j)} \right) \right].$$

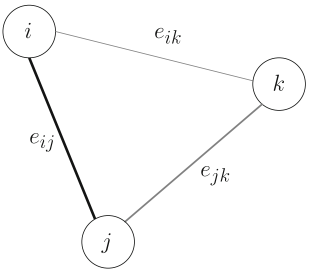

Fig. 1 Example of strong and weak edges: *edge thickness* indicates the strength of variable association. Notice that testing e_{ij} first might lead this edge to be pruned in place of e_{ik}

As the counts in the tables $n_{ij}(x_i, x_j)$ are sums of independent random variables and as m_{ij} is a function of these it has an asymptotic normal distribution. Its variance may be approximated by $s_{ij}^2 = \frac{1}{N}(m_{ij}^{(2)} - m_{ij}^2)$ where

$$m_{ij}^{(2)} = \frac{1}{N} \sum_{x_i, x_j} n_{ij}(x_i, x_j) \left[\log_2 \left(\frac{n_{ij}(x_i, x_j) N}{n_i(x_i) n_j(x_j)} \right) \right]^2 .$$

The p value is $1 - \Phi\left(\frac{m_{ij} - \mu_{\mathcal{I}}}{s_{ij}}\right)$ calculated from the standard $\mathsf{N}(0, 1)$ cumulative distribution, where $\mu_{\mathcal{I}}$ is the threshold value.

The convergence of m_{ij} to its mean under the thresholding assumption is $O(1/\sqrt{N})$ while under independence it is $O(1/N)$. There are obvious generalizations for the variance of the conditional mutual information in higher-way tables but, in practice, it is the behavior at the first pass of the PC algorithm that is problematic. This model is denoted as CImap-PN in the following.

3.5 Test the weakest first

The outcome of the vanilla PC algorithm is influenced by the order in which the conditional independence tests are executed. Consider, for instance, the toy scenario in Fig. 1, where edge thickness is used to denote the strength of conditional association between the random variables.

Suppose the algorithm performs the independence tests first on node i and, in particular, on edge e_{ij} conditioned on the realization of k, i.e. tests $I_{\mathsf{P}}(i, j|k)$. Here k is a neighbor of both i and j and it might be the case that it explains the association between these two nodes in terms of its weaker associations to i and j (i.e. e_{ik} and e_{jk}). This might lead the algorithm to pruning the strong edge e_{ij} that, conversely, ought to be preserved. Moreover, at the next step, the algorithm tests the weak edge e_{ik} without conditioning on j, given that it has been pruned from the neighbors of i (it is still in the neighborhood of k but will only be tested later). Hence, it might be the case that such a weak edge is preserved, thus resulting into two different errors: (i) the incorrect pruning of e_{ij} and (ii) the incorrect discovery of the e_{ik} dependency. The result of a test can alter the outcome of subsequent tests so that different node and edge orderings may lead to different final graphs from the same input data.

A principled strategy that provides a non-arbitrary ordering of nodes and edges is based on a TWF policy, where the strength of the node i is defined as the sum of the marginal mutual information between i and its adjacent nodes $j \in \mathrm{ne}(i)$

$$\sigma_i = \sum_{j \in \mathrm{ne}(i)} \hat{\mathcal{I}}(i, j). \tag{6}$$

The outcome of the first pass of the algorithm (the unconditional tests) is invariant with respect to the test order, which only becomes relevant at the stage of conditional tests. The strengths σ_i can be computed as part of the first pass without any increase to the complexity of the algorithm. Once obtained for all nodes in the graph they can be ordered from the weakest node (i.e. having the lowest σ_i), that is tested first, to the strongest node, that is tested last. Similarly, the TWF policy can be applied to the edges of a candidate node i in order to select which association is tested first. In particular, given node i, its incident edges e_{ij} are sorted in increasing order of mutual information $\hat{\mathcal{I}}(i, j)$ so that the weakest edge is tested first.

Algorithm 3 describes the PC algorithm with the TWF policy: notice that node and edge ordering is updated before testing a new order of conditional dependencies (lines 17–18), so to keep sorting consistent with respect to the deleted edges (line 31). The computational cost of the TWF sorting up-to the K-th order tests is worst case $O(Kp^2 \log p)$ (presuming the use of a $O(p \log p)$ sorting algorithm), and this calculation can be made more efficient by exploiting the sorting obtained for order $K - 1$ to determine the new sorting at K.

The TWF–PC algorithm can be combined with any of the FNR and FDR policies described previously: for instance, the single-test edge pruning in lines 23–32 can be replaced by the per-edge FDR procedure. In the experimental evaluation we study the effect of the TWF policy on the quality of network reconstruction under different algorithmic setups and in combination with all the methodological tools discussed in the previous sections.

4 Results

4.1 Experimental comparisons

The software package *CImap* for high-dimensional structure finding with large datasets is based on the constraint-based structure of Algorithm 1, enhanced by procedures for false negative, false positive control and the TWF policy described in Algorithm 3. *CImap* is implemented by a software package comprising both Matlab scripts as well as open-source pre-compiled MEX routines to speed up mutual information computations. All the experiments have been performed using Matlab R2007b on a Dual Core Intel 1.83 GHz CPU equipped with 1 GB RAM.

The performance of the package is compared with respect to several state-of-the-art structure finding algorithms, mostly implemented as part of the *Causal Explorer* package (Aliferis et al. 2003). In particular, the following algorithms have been tested

Algorithm 3 The TWF–PC Algorithm

```
Require: Dataset $D$, Test level $\alpha$
Initialize a fully connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
for all $i \in \mathcal{V}$ do {Marginal independence tests}
  $\sigma_i \Leftarrow 0$
  for all $j \in \text{ne}\,(i)$ do
    $m \Leftarrow \hat{\mathcal{I}}(i, j)$
    IsIndependent $\Leftarrow$ TestIndependence($D, \alpha, m$)
    if IsIndependent then
      $\mathcal{E} \Leftarrow \mathcal{E} \setminus e_{ij}$ {Remove the edge from the graph}
      $\text{ne}\,(i) \Leftarrow \text{ne}\,(i) \setminus j$, $\text{ne}\,(j) \Leftarrow \text{ne}\,(j) \setminus i$ {Update the neighborhoods}
    else
      $m_{ij} \Leftarrow m$, $\sigma_i \Leftarrow \sigma_i + m$ {Update the node and edge strength}
    end if
  end for
end for
$K \Leftarrow 1$
repeat {Order $K$ independence tests}
  $\mathcal{V} \Leftarrow$ nodeSort($\mathcal{V}, \{\sigma_i\}$) {Sort nodes in increasing order of strength}
  $\text{ne}\,(\mathcal{V}) \Leftarrow$ edgeSort($\text{ne}\,(\mathcal{V}), \{m_{ij}\}$) {Sort edges in increasing order of $m_{ij}$ for each $i = 1, \ldots, p$}
  for all $i \in \tilde{\mathcal{V}}$ do
    for all $j \in \text{ne}\,(i)$ do
      $\mathcal{A} \Leftarrow$ ConditionSet($\text{ne}\,(i) \setminus j, K$)
      IsIndependent $\Leftarrow$ false
      repeat {Seeks a certificate of exclusion from neighbors of $i$}
        $A \Leftarrow$ select element from $\mathcal{A}$
        IsIndependent $\Leftarrow$ TestCondIndependence($D, \alpha, i, j, A$)
        $\mathcal{A} \Leftarrow \mathcal{A} \setminus A$
      until $\mathcal{A} = \emptyset$ or isIndependent
      if IsIndependent then
        $\mathcal{E} \Leftarrow \mathcal{E} \setminus e_{ij}$ {Remove the edge from the graph}
        $\text{ne}\,(i) \Leftarrow \text{ne}\,(i) \setminus j$, $\text{ne}\,(j) \Leftarrow \text{ne}\,(j) \setminus i$, {Update the neighbors lists}
        $\sigma_i \Leftarrow \sigma_i - m_{ij}$, $m_{ij} \Leftarrow 0$ {Update the node and edge strength}
      end if
    end for
  end for
  $K \Leftarrow K + 1$
until there are no tests of order $K$ in $\mathcal{G}$
return Undirected graph $\mathcal{G}$
```

- Two versions of the standard PC algorithm (Spirtes et al. 2000) described in Sect. 2: one, denoted as PC-G2, implementing significance testing on the G^2 statistics; the other, denoted as PC-MI, testing conditional independence based on mutual information thresholds, i.e. without significance testing.
- MMHC (Tsamardinos et al. 2006), an hybrid algorithm mixing an optimized constraint-based approach to reconstruct the skeleton with an edge orientation phase implemented by an hill-climbing search guided by a Bayesian score. The MMHC algorithm is, to the extent of our knowledge, the state of the art approach to structure identification in terms of tradeoff between reconstruction performance and computational requirements. In the tested configuration, no limits have been imposed on the cardinality and the maximum allowed size of the conditioning set has been fixed to $K = 10$;

- SCA (Friedman et al. 1999), a purely search-and-score algorithm that performs several hill-climbing searches in the full DAG space, constraining each node to have at most k parents and re-estimating the set of candidate parents at each GS iteration.
- The GS algorithm (Meek 1997), that performs a greedy optimization in the space of Markov equivalence classes for DAGs using a Bayesian score to evaluate the structures.
- A version of the ARACNE algorithm (Margolin et al. 2006) tailored for categorical variables: this method only tests pairwise interactions and considers, at most, all possible triplets of connected genes generated by an initial step. This makes the ARACNE method computationally very efficient and has proved to be very effective in the reconstruction of gene regulatory networks (Margolin et al. 2006). The implementation used in the tests exploits the same optimized MEX routines for mutual information computation employed in the *CImap* package.

Performance has been evaluated in terms of the skeleton reconstruction quality, i.e. the number of generated false positives and negatives, as well as in terms of completion time.

4.2 *CImap* configuration options

Several *CImap* configurations have been tested, e.g. CImap-P + iFDR + FNR indicates an algorithm with statistical tests, interleaved FDR and false negative control, listed in Table 1. The options listed in Table 1 have been combined in several ways to test a large number of configurations. For the sake of clarity, we use some shortcuts to denote particular configurations that are tested thoroughly: Table 2 provide a succinct summary of the shortcuts used.

The test level and the mutual information cut for the independence tests has been set to $\alpha = 0.05$ and $\mu_{MI} = 10$ mbits, respectively, in accordance to the testing guidelines in Tsamardinos et al. (2006). Experimentally, these values offer the best average reconstruction performance across different benchmarks for all the tested algorithms. The α^{fdr} value for the FDR procedures (see Sect. 3.3) is always set to be equal to the significance level α.

Table 1 *CImap* options tested in the experimental evaluation

Acronym	Definition
CImap-P	Basic PC algorithm with χ^2 tests on mutual information
CImap-MI	Basic PC algorithm with mutual information cuts
CImap-PN	Dense network approximation algorithm in Sect. 3.4
sFDR	Standard FDR with Benjamini and Yekutieli (2001) criterion
iFDR	Interleaved FDR with Benjamini and Yekutieli (2001) criterion
peFDR	Per-edge FDR with Benjamini and Yekutieli (2001) criterion
FNR	False negative control
TWF	Test-the-weakest-first policy

Table 2 List of shortcuts for specific *CImap* configurations tested in the experimental evaluation

Acronym	Configuration
CImap-P-1	CImap-P + TWF
CImap-P-2	CImap-P + iFDR + FNR + TWF
CImap-P-3	CImap-P + peFDR + FNR + TWF
CImap-P-4	CImap-P + iFDR + TWF
CImap-MI1	CImap-MI (basic algorithm)
CImap-MI2	CImap-MI + TWF
CImap-PN1	CImap-PN algorithm with threshold value $\mu_{\mathcal{I}} = 10$ mbits
CImap-PN2	CImap-PN algorithm with threshold value $\mu_{\mathcal{I}} = 5$ mbits

Table 3 Bayesian networks used in the experimental studies

Network	Variables	Edges degree	Max in/out range	Domain
Insurance	27	52	3/7	2–5
Alarm	37	46	4/5	2–4
Hailfinder	56	66	4/16	2–11
Barley	48	84	4/5	2–67

4.3 Benchmark datasets

The experimental setup comprises four benchmark networks that are freely available from the *Bayesian Network Repository*[1] and whose details are listed in Table 3. Experiments have been performed by varying the dataset size between 0.5 and 10 K samples: for each tested size, we performed 10 network simulations using the *Bayesian Network Toolbox* (*BNT*) (Murphy 1997) in order to sample 10 independent datasets. The *Barley* network has been tested only up to 5 K samples using the datasets available on the *Casual Explorer*[2] site.

4.4 Results for alternative configurations of *CImap*

First, we evaluated the performance of the *CImap* package for alternative configurations, to compare the effect of the different FDR and FNR corrections as well as of the TWF policy, using the *Insurance* and *Alarm* networks. Figures 2 and 4 show the corresponding reconstruction errors, for each *CImap* configuration, as a function of the sample size: left bars (blue) denote the total reconstruction errors, i.e. the sum of the false positive plus false negative edges, while right bars (red) show the number of false negatives. The plots show that the TWF policy produces a consistent reduction in the reconstruction errors. Moreover, this performance improvement is coupled with a consistent reduction in the time required by the algorithm to converge. Figure 3 indicate

[1] http://www.cs.huji.ac.il/~galel/Repository/.

[2] http://discover.mc.vanderbilt.edu/discover/public/supplements/mmhc_paper/mmhc_index.html.

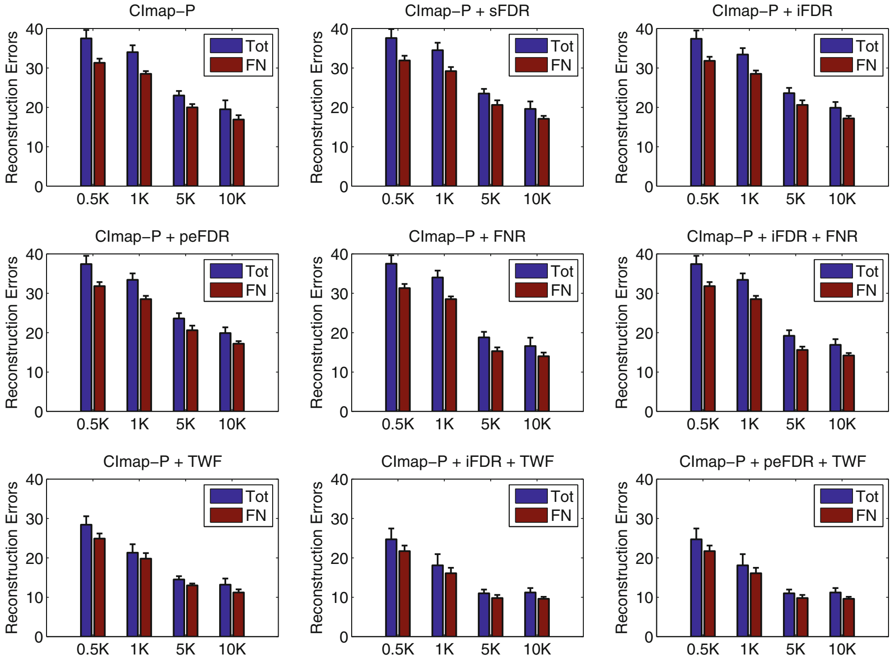

Fig. 2 Reconstruction errors for the *Insurance* network as a function of the dataset size with varying *CImap* configurations: *left bars* (*blue*) denote cumulative errors, while *right bars* (*red*) show the number of false negatives (color figure online)

a quadratic increase in time with sample size with no deterioration in performance by adding FDR; worse performance by adding FNR; and an improved performance by adding TWF. The explanation is that the TWF policy reduces the number of false positives in the network, which shrinks the size of the neighbourhood of each node, so reducing the number of independence tests performed by the algorithm which, therefore, converges earlier.

As regards the FDR and FNR corrections it is clear that the latter offers the most consistent increase in the network reconstruction performance: for instance, it obtains the best results when used jointly with the TWF policy. The FDR correction, on the other hand, seems to contribute less to network reconstruction quality, although its positive effect is seen for the Alarm network (see Fig. 4). Furthermore *interleaved* and *per-edge* FDR seem more effective than *standard* FDR both as regards reconstruction quality as well as for reducing the computational load of the algorithm. This, again, can be explained by the anticipated pruning of the false positives avoiding some tests on edges that should not be included in the final graph.

4.5 Comparative results

These results suggest we focus on comparing *Causal Explorer* and ARACNE with *CImap* with interleaved FDR, FNR control and TWF policy.

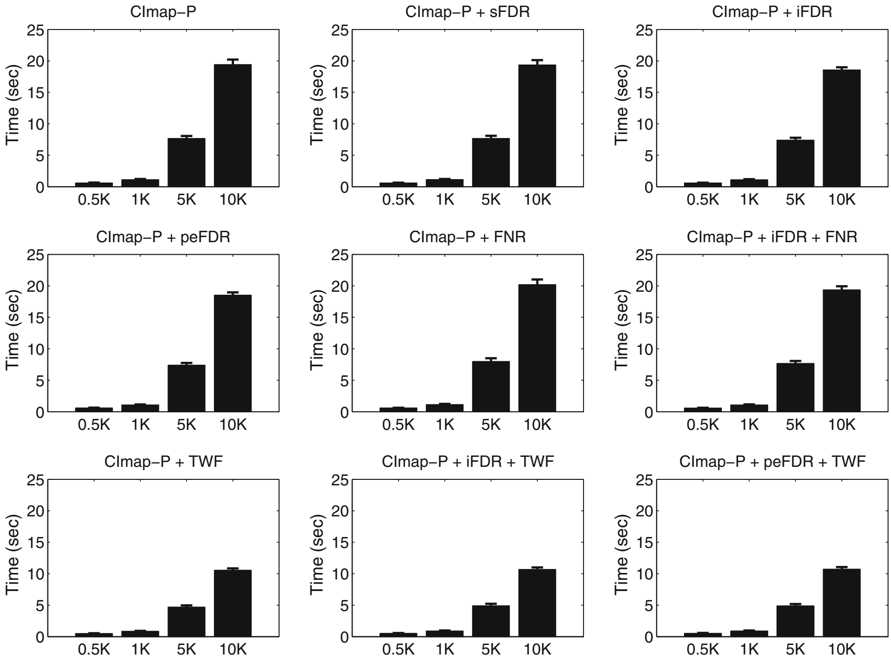

Fig. 3 Time to complete (in seconds) for the *Insurance* network as a function of the dataset size with varying *CImap* configurations

Figures 5, 6, 7, 8 show the reconstruction error on the *Insurance*, *Alarm*, *Hailfinder* and *Barley* networks as a function of the dataset size. Overall, it is clear that the MMHC gives the best reconstruction quality. *CImap* has a competitive performance with MMHC on the *Insurance* and *Barley* networks, especially when dealing with datasets comprising fewer samples. Figure 8 shows that reconstruction errors for CImap-P-2 are higher than for CImap-P-1 in the *Barley* network: the number of false positives increase with FNR switched on and the TWF policy alone cannot get rid of these. Compared to the standard implementation of the PC algorithm *CImap* shows a notable improvement in reconstruction performance. This latter result displays quite well the effectiveness of the TWF, FDR and FNR procedures, since *CImap* is essentially a PC algorithm, enhanced by the above methodology. ARACNE has the worse performance; it retains too many false positive edges because it only tests pairwise mutual information, hence missing more complex interactions. However, we expect the ARACNE algorithm to benefit from the FDR procedures resulting in a reduction of its false positives.

Table 4 shows the time to converge of the different algorithms on the four test networks. *CImap* gives the best performance with running times that, on larger networks such as *Hailfinder*, vary within 1–5 % of the time required by MMHC (see Fig. 9). The comparison with the performance of the standard PC algorithm and pure search-and-score algorithms such as SCA and GS shows clearly the marked advantage of *CImap* in terms of computational feasibility. The steep increase, with respect to sample size, in the time-to-converge of the PC algorithm for the Insurance

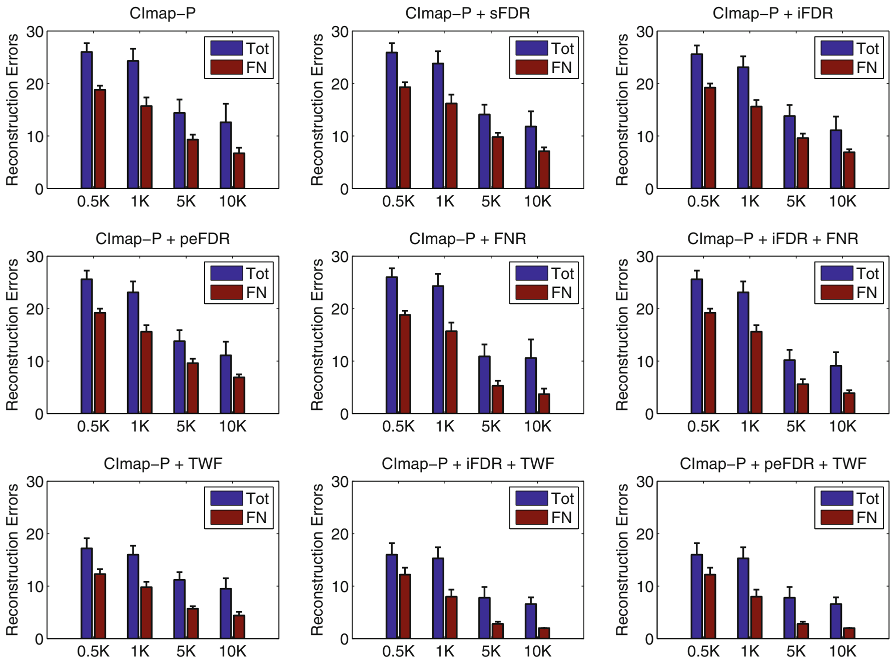

Fig. 4 Reconstruction errors for the *Alarm* network as a function of the dataset size with varying *CImap* configurations: *left bars* (*blue*) denote cumulative errors, while *right bars* (*red*) show the number of false negatives (color figure online)

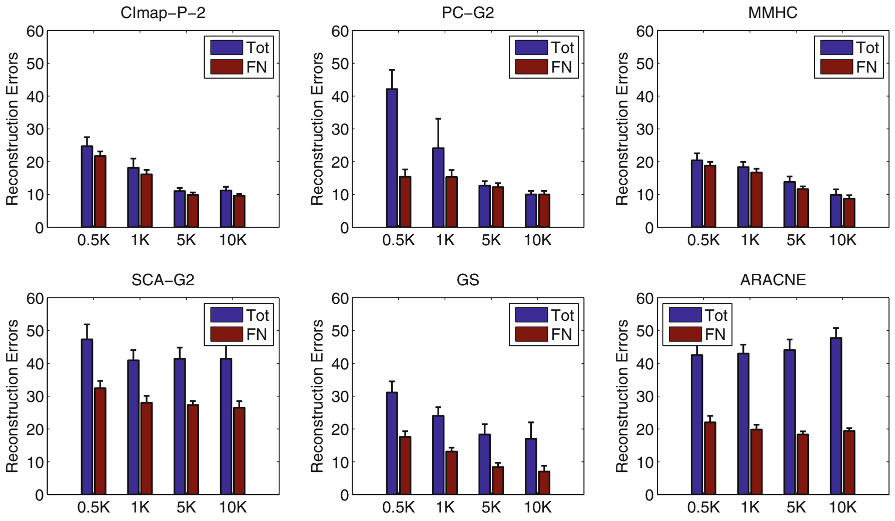

Fig. 5 Reconstruction errors for the *Insurance* network using p value test with significance $\alpha = 0.05$. *Left bars* (*blue*) denote cumulative errors, while *right bars* (*red*) show the number of false negatives (color figure online)

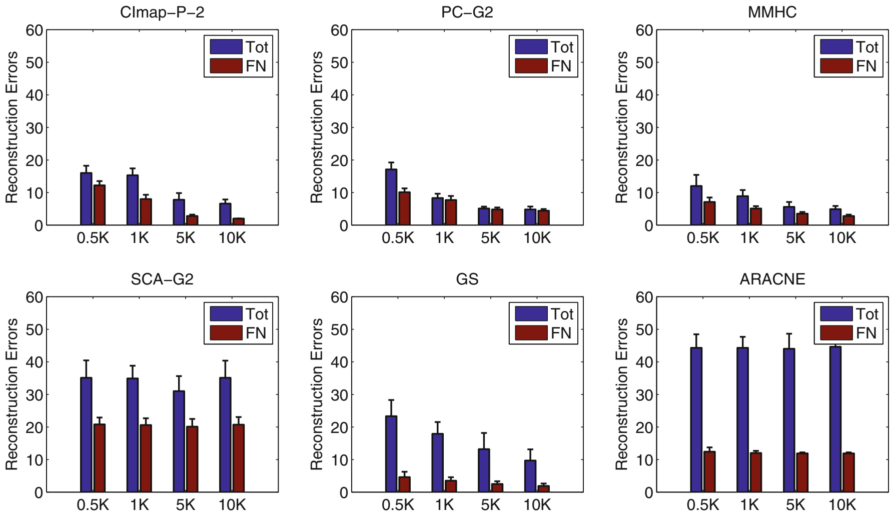

Fig. 6 Reconstruction errors for the *Alarm* network using *p* value test with significance $\alpha = 0.05$. *Left bars* (*blue*) denote cumulative errors, while *right bars* (*red*) show the number of false negatives (color figure online)

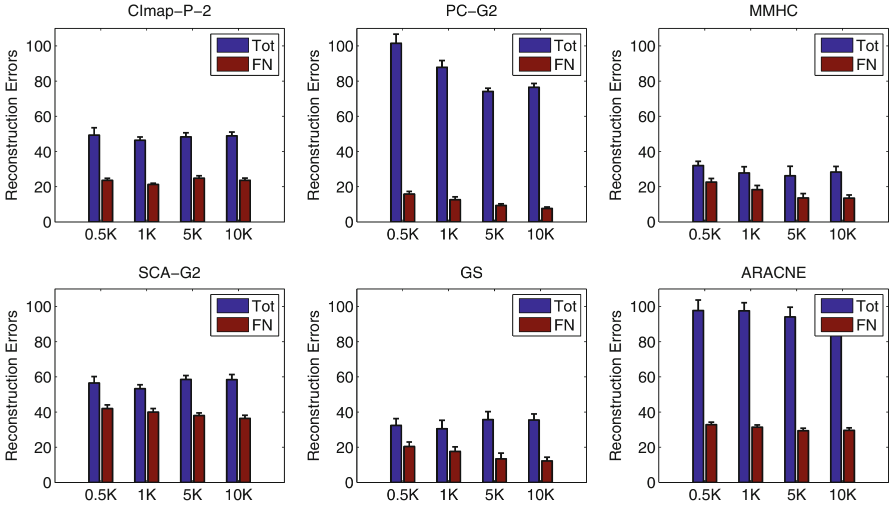

Fig. 7 Reconstruction errors for the *Hailfinder* network using *p* value test with significance $\alpha = 0.05$. *Left bars* (*blue*) denote cumulative errors, while *right bars* (*red*) show the number of false negatives (color figure online)

network has been documented in Tsamardinos et al. (2006) and appears to be connected with a late pruning of false positive edges, so resulting in the algorithm performing independence tests on a larger number of conditioning sets. The standard PC algorithm did not converge within 7 days on the *Barley* network, while GS ran out of memory when reconstructing the *Barley* network from 5 K samples. The ARACNE algorithm has, notably, the lowest time computational requirements between tested methods, due

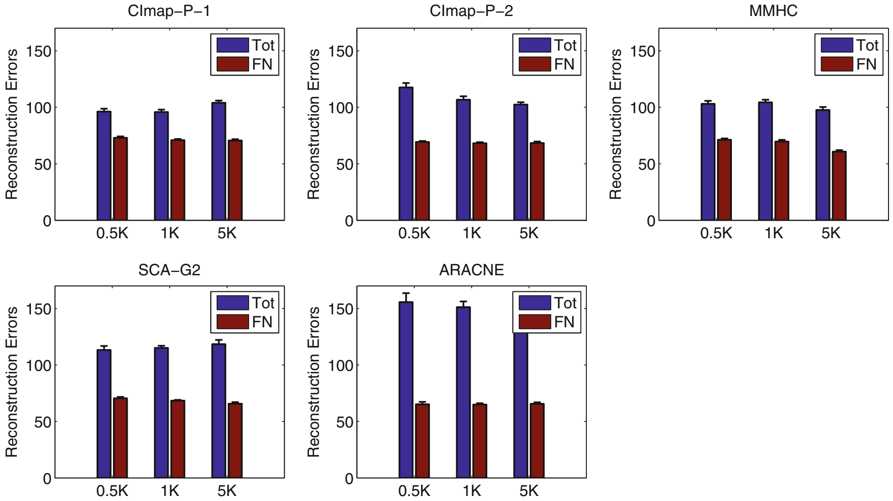

Fig. 8 Reconstruction errors for the *Barley* network using *p* value test with significance $\alpha = 0.05$. *Left bars* (*blue*) denote cumulative errors, while *right bars* (*red*) show the number of false negatives (color figure online)

to its simplified testing strategy addressing only pairwise dependencies. However, as seen previously, such simplification results in a consistent increase of the amount of false positive errors. Overall, the experimental results suggest that *CImap* offers the best trade-off in terms of reconstruction quality and computational load when dealing with high dimensional networks.

4.6 Large datasets

We compare the performance of *CImap* with respect to the algorithms in the *Causal Explorer* package on a large-scale (*LargeScale*) case-study using anonymized real-world data. By exploiting proprietary data comprising more than 30 K samples, we build a ground truth Bayesian network comprising 328 nodes with 154 edges: the network structure is sparse with over 212 nodes being marginally independent. The ground truth network is built from the original data using *CImap* with mutual information cuts and a threshold $\mu_{MI} = 8$ mbits. Network parameters are learnt by maximum likelihood estimation using *BNT*. The experimental setup comprises datasets of different size, including 10, 20, 30 and 40 K samples. For each dataset size, we generate 10 independent datasets by sampling from the ground truth network above, using the facilities in *BNT*.

The proposed scenario poses challenging computational issues as the sample size is large and the network is high-dimensional, yet sparse, being characterized by a modular structure with low inter-cluster connectivity and strong intra-cluster connectivity. The high dimensionality of the network causes greedy methods to fail because of the combinatorial explosion of computational complexity. The large number of observations inflates chi-squared statistics, preventing most of the constraint-based and hybrid algorithm to consistently estimate the network structure.

Table 4 Time to complete (in seconds) for each algorithm configuration as a function of the dataset size for p value tests of level $\alpha = 0.05$

Algorithm	500		1000		5000		10000	
	Sec.	SD	Sec.	SD	Sec.	SD	Sec.	SD
Insurance								
CImap-P-1	0.70	0.04	1.07	0.07	4.38	0.34	9.09	0.46
CImap-P-2	0.74	0.04	1.19	0.07	4.86	0.51	9.79	0.56
PC-G2	8.09	0.43	9.68	0.27	26.50	3.32	43.28	4.27
MMHC-G2	4.32	0.62	5.32	0.25	19.11	2.07	36.23	2.41
SCA-G2	19.32	3.18	26.73	3.53	88.75	22.72	160.83	36.05
GS	7.61	1.17	10.12	1.67	35.55	5.57	74.31	13.92
ARACNE	0.16	0	0.19	0.01	0.48	0.01	0.92	0.14
Alarm								
CImap-P-1	0.50	0.03	0.86	0.05	4.70	0.26	10.56	0.28
CImap-P-2	0.53	0.04	0.90	0.06	4.91	0.31	10.69	0.30
PC-G2	69.56	22.85	108.26	163.41	34.73	2.74	93.84	8.61
MMHC-G2	3.01	0.60	4.06	0.25	20.29	1.38	57.56	3.79
SCA-G2	10.57	3.46	15.55	3.04	40.18	11.07	58.20	14.75
GS	3.58	0.45	5.07	1.08	15.61	2.09	27.47	2.36
ARACNE	0.30	0.02	0.3	0.01	1.12	0.33	1.99	0.37
Hailfinder								
CImap-P-1	1.2	0.042	1.49	0.06	4.57	0.39	9.47	1.07
CImap-P-2	1.78	0.58	2.91	1.44	12.70	8.67	32.47	22.08
PC-G2	40	5	78	8	460.23	25.95	1048	72
MMHC-G2	8	1	14	1	172.97	10.28	561	50
SCA-G2	22	5	36	11	111.68	27.48	225	53
GS	14	3	22	3	100.47	14.65	187	33
ARACNE	0.73	0.01	0.84	0.01	2.16	0.10	3.96	0.14
Barley								
CImap-P-1	0.92	0.03	1.23	0.04	5.01	0.47	n.a.	n.a.
CImap-P-2	1.20	0.17	1.56	0.40	5.04	0.26	n.a.	n.a.
CImap-P-3	1.15	0.11	1.55	0.41	5.10	0.35	n.a.	n.a.
PC-G2	–	–	–	–	–	–	n.a.	n.a.
MMHC-G2	7.94	0.59	11.86	2.24	45.19	9.46	n.a.	n.a.
SCA-G2	28.51	5.87	45.57	12.69	129.19	37.69	n.a.	n.a.
GS	16.08	1.89	36.62	9.15	–	–	n.a.	n.a.
ARACNE	0.56	0.02	0.65	0.01	1.64	0.05	n.a.	n.a.

Results show the mean and standard deviation over 10 independently sampled datasets. Results for algorithms that did not converge within a week are marked as "–", "*n.a.*" is used to mark the missing datasets for the *Barley* network

Several *CImap* configurations, *Causal Explorer* algorithms and ARACNE are compared in these large scale scenarios. Initially, we run structure learning using basic tests of mutual information cuts, i.e. we use a threshold $\mu_{MI} = 10\,\text{mbits}$ (a standard

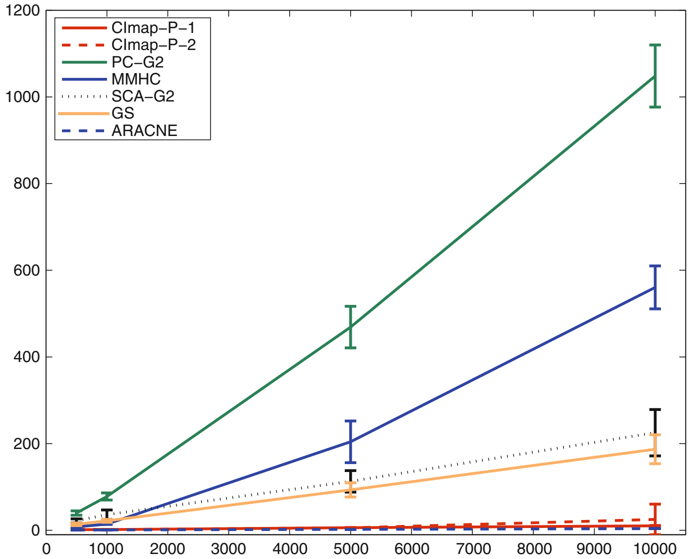

Fig. 9 Comparison of the time to complete (in seconds) for the Hailfinder network as a function of the dataset size

threshold value from literature) and run the structure learning algorithms by pruning those edges having conditional mutual information lower that μ_{MI}. Such an approach has the advantage of keeping the computational complexity of edge testing down as it does not require computing χ^2 distributions.

Edge presence testing based on cuts does not allow false positive and false negative control to be exploited, and also, does not obviously allow one to tailor testing levels to the size of the conditioning set. Hypothesis testing, however, may implement FDR and FNR and accounts for the sizes of the conditioning set through the appropriate null distribution. We run some experiments using the CImap-P algorithm with χ^2 tests of mutual information discussed at level $\alpha = 0.05$.

Figure 10 shows the behavior of the total reconstruction errors (averaged over the 10 re-samplings) as a function of the dataset size. Surprisingly, the algorithms using simple MI cuts are performing better than those exploiting more refined statistical tests. From the error breakdown in Table 5, it is clear that the latter are generating a large amount of FPs and the FDR procedure in CImap-P-4 manages to reduce them only partially, leaving almost 180 false positives out of about 300 discovered edges.

The origin of such degenerate behavior is in the first pass of unconditional edge testing, where p values calculated from the χ^2 distribution are significant for almost all edges, resulting in extremely dense graphs. As discussed in Sect. 3.4, this is due to the fact that with large samples small departures from independence become noticeable, resulting in the null distribution of the mutual information differing consistently from χ^2. The impact of this issue on the computational performance of the algorithm is well illustrated by Fig. 11, where the time-to-complete of the χ^2-based algorithms follows an exponential behavior with respect to dataset size. The results from Causal

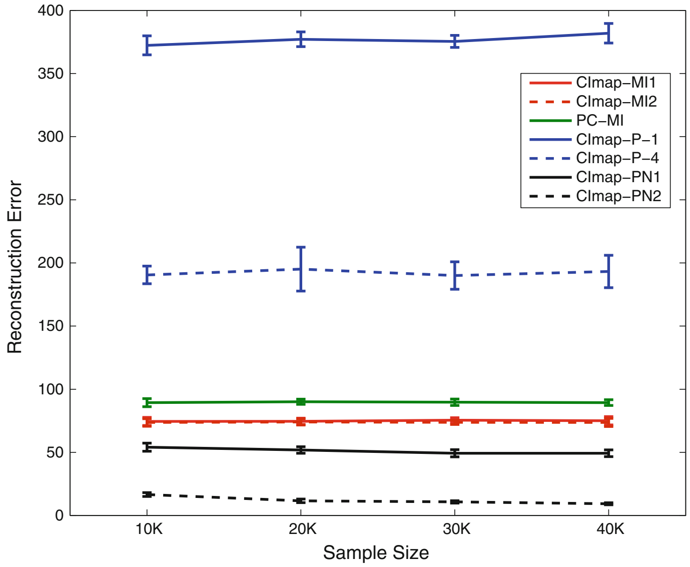

Fig. 10 Cumulative reconstruction errors (false negatives and positives) for the *LargeScale* scenario using mutual information cuts (with $\mu_{MI} = 10$ mbits) and p value tests (with significance $\alpha = 0.05$)

Table 5 False positives and false negatives on the *LargeScale* scenario for each algorithm configuration as a function of the dataset size for mutual information cuts $\mu_{MI} = 10$ mbits, p value tests of level $\alpha = 0.05$ and Normal approximation thresholds $\mu_{\mathcal{I}} = 10$ mbits and $\mu_{\mathcal{I}} = 5$ mbits

Algorithm	10000		20000		30000		40000	
	FN	FP	FN	FP	FN	FP	FN	FP
Insurance								
CImap-MI1	74.5	0	74.6	0	75.5	0	75.0	0
CImap-MI2	73.7	0	74.1	0	73.8	0	73.7	0
PC-MI	89.4	0	90.1	0	89.7	0	89.4	0
CImap-P-1	23.4	348.9	19.0	358.1	16.0	359.4	14.2	367.7
CImap-P-4	20.8	169.7	15.9	179.2	13.2	176.8	11.2	182.0
PC-G2	10.2	210.7	–	–	–	–	–	–
CImap-PN1 (10 Mbit)	54.1	0	51.9	0	49.3	0	49.3	0
CImap-PN2 (5 Mbit)	16.6	0	11.5	0.1	10.8	0	9.3	0
ARACNE	54.4	2,121.4	53.9	2130.1	54.7	2114.9	54.3	2122.5

Results show the mean over 10 independently sampled datasets. Results for algorithms that did not converge within a week are marked as "–"

Explorer are somewhat disappointing because the program failed to reach a result within a reasonable run-time apart from the PC-MI algorithm discussed previously. Such a failure has different explanations, depending on the algorithms. For instance, search-and-score models such as GS and SCA do not converge due to the large search space (i.e. graphs with more than 300 nodes). Both the PC algorithm with G^2 statistics

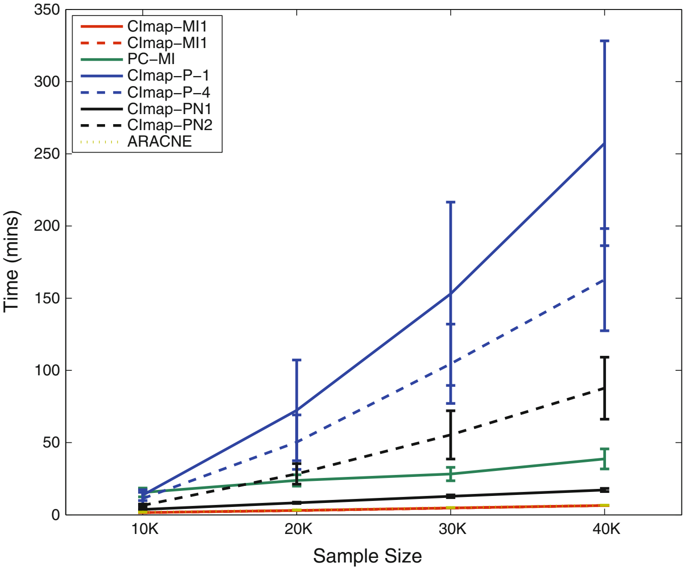

Fig. 11 Comparison of the time to complete (in minutes) for the *LargeScale* network as a function of the dataset size

and the MMHC algorithm fail to provide results within a 2 week run-time limit, probably because they are both affected by the χ^2 issue. Notice that the PC-G2 algorithm managed to provide results within the 2 weeks, but only for the 10 K samples datasets: Table 5 reports the corresponding reconstruction error, which has been obtained with an average time-to-complete of 3202.7 min. For a 20 K samples dataset, PC-G2 crashes (after 1 week running time) because of excessive memory usage when testing order 10 conditional dependencies. The ARACNE algorithm managed to provide timely results, with a time to complete that is comparable to that the *CImap* algorithms based on simple mutual information cuts (see Fig. 11). However, Table 5 shows that ARACNE produces an large number of false positives, with several thousand false edges. It seems most of these are produced by the failure of the χ^2 tests leading to extremely dense networks at the first step of the algorithm. The second ARACNE step, which prunes the edge with the lowest MI among all triplets of connected nodes, is able to consistently reduce the network connectivity, but still retains too many false positive edges.

The algorithms based on simple cuts do not generate such a huge number of false positive edges, with the *CImap* implementation providing results within an outstanding 6.4 min time (on 40 K datasets) as compared to the 38.68 min required by PC-MI. However, neither provides a satisfactory reconstruction of the ground truth network with over 70 errors, almost all false negatives, so returning extremely sparse and poorly connected networks.

To improve the reconstruction quality in this *LargeScale* scenario we implemented the Normal approximation for dense networks discussed in Sect. 3.4. In particular, we extended the *CImap* algorithm to perform unconditional statistical testing based on a Normal mutual information distribution, while conditional tests are still performed based on a χ^2 distribution. The significance level of the tests is set to $\alpha = 0.05$ for both configurations. Notice that, in contrast with $\alpha_{\mathcal{I}}$, $\mu_{\mathcal{I}}$ does not define a hard MI threshold under which edges are severed from the network (e.g. $\alpha_{\mathcal{I}} = 10$ mbits). Rather, it relocates the expected MI for independent edges from the expectation under the χ^2 distribution to $\mu_{\mathcal{I}}$, performing a statistical test of independence based on test level α.

The results in Fig. 10 and Table 5 shows that the CImap-PN approach produces a notable increase in the quality of the reconstructed network, with the CImap-PN2 approach yielding to as few as 9 missing edges with no false positives. The computational effort is larger but not intolerable as the CImap-PN1 configuration is converging in about 87 min (see Fig. 11). Comparing CImap-PN1 with CImap-PN2 shows the more conservative nature of the latter configuration which tends to preserve more edges at the first step of unconditional edge testing. This, on the one hand, reduces the number of false negatives enhancing the overall reconstruction quality; but increases the time-to-converge of the algorithm as more conditional tests need to be performed.

5 Concluding remarks

Constraint-based approaches, such as the PC algorithm (Spirtes et al. 2000), are important tools for dealing with structure identification on high-dimensional problems with large sample sizes. To this end, we propose a set of general computational procedures for enhancing the reconstruction performance of constraint-based methods, without affecting their computational complexity. Some of them, including FDR and FNR control, build on previous work in the field to construct computationally efficient routines for generic constraint-based approaches. The TWF policy and the Normal approximation for dense networks, on the other hand, are novel contributions of this paper, addressing issues that have not yet been tackled with in literature.

The experimental assessments show how even a fairly simple algorithm such as the standard PC algorithm can be made competitive with the best hybrid algorithms, such as the state-of-the-art MMHC model (Tsamardinos et al. 2006). This model has an hybrid structure finding strategy that mixes a constraint-based skeleton identification phase with a search-and-score refinement of the structure, that makes it extremely competitive from the point of view of the trade-off between computational load and quality of the reconstructed network. However, the proposed *CImap* approach is shown, across several experimental tests on benchmark data, to match MMHC reconstruction quality with computational speeds comparable with one of the fastest current algorithms, ARACNE. The ARACNE approach simplifies notably the testing strategy since it addresses only pairwise dependencies; however, such simplification results in a consistent increase of the amount of false positive errors. Overall, the proposed methodology proved to be the best in terms of trade-off between computational requirements and quality of the reconstructed network, already with small-to-medium sized problems.

We show that application of error rate bounding in statistical hypothesis testing enhances the quality of the reconstructed network. Also we show how mutual information provides reliable test statistics for edge tests, which can naturally be exploited to provide a measure of edge strength. Such a measure is used to implement a test-the-weakest-first policy, that is shown to provide a notable increase in the quality of the reconstructed network by reducing both false positives and false negatives.

A critical issue affecting the applicability of some algorithms in the literature is the inflation of test statistics with large sample size, leading to dense networks. We propose a novel procedure for approximating networks that eliminates an edge when the mutual information is no bigger than a given threshold, and we show that this leads to a high quality reconstructed network. To our knowledge, such an issue has not been raised in literature, as it is strongly linked to the peculiarities of large scale applications (e.g. social networks, data mining, etc.), where a large network search space is associated to extensive data collections..

References

Aliferis CF, Tsamardinos I, Statnikov AR, Brown LE (2003) Causal explorer: a causal probabilistic network learning toolkit for biomedical discovery. In: Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS'03), pp 371–376

Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. J R Stat Soc Ser B Methodol 57(1):289–300

Benjamini Y, Yekutieli D (2001) The control of the false discovery rate in multiple testing under dependency. Ann Stat 29(4):1165–1188

Bishop YMM, Fienberg SE, Holland PW (1975) Discrete multivariate analyses: theory and practice. MIT Press, Cambridge, MA

Cheng J, Greiner R, Kelly J, Bell D, Liu W (2002) Learning Bayesian networks from data: an information-theory based approach. Artif Intell 137(1–2):43–90

Dawid AP (1979) Conditional independence in statistical theory (with discussion). J R Stat Soc Ser B 41:1–31

Fast A, Hay M, Jensen D (2008) Improving accuracy of constraint-based structure learning. Technical report 08-48, University of Massachusetts Amherst, Computer Science Department

Friedman N, Nachman I, Pe'er D (1999) Learning Bayesian network structure from massive datasets: the "sparse candidate" algorithm. In: Proceedings of the Fifteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-99), Morgan Kaufmann, San Francisco, CA, pp 206–221

Goebel B, Dawy Z, Hagenauer J, Mueller J (2005) An approximation to the distribution of finite sample size mutual information estimates. In: ICC 2005. 2005 IEEE International Conference on Communications, vol 2, pp 1102–1106

Jensen F, Nielsen T (2007) Bayesian networks and decision graphs. Springer, Berlin

Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT Press, Cambridge, MA

Lauritzen S (1996) Graphical models. Oxford University Press, Oxford

Margolin A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera R, Califano A (2006) ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinf 7(Suppl 1):S7

Meek C (1997) Graphical models: selecting causal and statistical models. Ph.D. thesis, Carnegie Mellon University

Murphy K (1997) Bayes net toolbox for matlab. http://people.cs.ubc.ca/~murphyk/Software/BNT/bnt.html

Spirtes P, Meek C (1995) Learning Bayesian networks with discrete variables from data. In: Usama M, Fayyad, Ramasamy Uthurusamy (eds). Proceedings of the first international conference on knowledge discovery and data mining. AAI Press, Navrangpura, Ahmedabad, pp 294–299

Spirtes P, Glymour C, Scheines R (1993) Causation, prediction, and search. Springer, New York

Spirtes P, Glymour C, Scheines R (2000) Causation, prediction and search, 2nd edn. MIT Press, New York, NY

Tsamardinos I, Brown LE (2008) Bounding the false discovery rate in local Bayesian network learning. In: Proceedings of the Twenty-third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, IL, USA, 13–17 July 2008, pp 1100–1105

Tsamardinos I, Brown LE, Aliferis CF (2006) The max- min hill-climbing Bayesian network structure learning algorithm. Mach Learn 65(1):31–78

Whittaker J (1990) Graphical models in applied multivariate statistics. Wiley, Chichester