The Dual PC Algorithm for Structure Learning

Enrico Giudice enrico.giudice@unibas.ch
Dep. of Mathematics and Computer Science, University of Basel, Basel, Switzerland

Jack Kuipers jack.kuipers@bsse.ethz.ch
Dep. of Biosystems Science and Engineering, ETH Zurich, Basel, Switzerland

Giusi Moffa giusi.moffa@unibas.ch
Dep. of Mathematics and Computer Science, University of Basel, Basel, Switzerland
and Division of Psychiatry, University College London, London, UK

Abstract

Learning the graphical structure of Bayesian networks is key to describing data generating mechanisms in many complex applications and it poses considerable computational challenges. Observational data can only identify the equivalence class of the directed acyclic graph underlying a Bayesian network model, and a variety of methods exist to tackle the problem. Under certain assumptions, the popular PC algorithm can consistently recover the correct equivalence class by reverse-engineering the conditional independence (CI) relationships holding in the variable distribution. Here, we propose the dual PC algorithm, a novel scheme to carry out the CI tests within the PC algorithm by leveraging the inverse relationship between covariance and precision matrices. By exploiting block matrix inversions we can efficiently supplement partial correlation tests at each step with those of complementary (or dual) conditioning sets. The multiple CI tests of the dual PC algorithm proceed by first considering marginal and full-order CI relationships and progressively moving to central-order ones. Simulation studies show that the dual PC algorithm outperforms the classic PC algorithm both in terms of run time and in recovering the underlying network structure, even in the presence of deviations from Gaussianity.

Keywords: Bayesian Networks; Graphical Models; Directed Acyclic Graphs; Structure Learning; PC Algorithm.

1. Introduction

Understanding and modelling the relationships among a set of random variables remains a central question in statistics. Probabilistic graphical models compactly describe a joint probability distribution and subsequently enable inferences about features of interest. Bayesian networks (BNs) are a particular class of probabilistic graphical models that employ directed acyclic graphs (DAGs) to describe the set of informational dependencies among variables (Pearl, 2009). Every node of the graph indicates a variable and edges between pairs of nodes encode conditional independence (CI) relations among the variables. All the nodes with an edge directed towards a certain variable X in the graph $\mathcal{G}$ constitute the parent set $\mathrm{Pa}(X)$ of X. Consider a set of random variables $\mathbf{X} = \{X_1, \dots, X_n\}$. A BN (Pearl, 1988; Koller and Friedman, 2009) is a pair $\langle \mathcal{G}, P \rangle$ of a DAG $\mathcal{G}$ and a joint probability distribution P, where P factorizes according to $\mathcal{G}$ into a product of conditional probability distributions: $P(\mathbf{X}) = \prod_{i=1}^{n} p(X_i|\mathrm{Pa}(X_i))$. When the graph $\mathcal{G}$ reflects all and only the CI relationships holding in the distribution P they are faithful to each other (Spirtes et al., 1993). Aiming

to imitate real-world data generating processes, BNs find application in a wide array of diverse fields, such as genomics (Friedman et al., 2000; Friedman, 2004; Kuipers et al., 2018), psychology (Moffa et al., 2017, 2021; Bird et al., 2018), text classification (de Campos and Romero, 2009), social sciences (Elwert, 2013) and epidemiology (Neil et al., 2020).

1.1 Structure Learning

Structure learning in BNs refers to the task of estimating their underlying graph $\mathcal{G}$ from a collection of observed realizations of the random vector $\mathbf{X} = \{X_1, \dots, X_n\}$. The number of all possible DAGs grows super-exponentially with the nodes n (Robinson, 1977), and recovering the network structure from observational data is NP-hard (Chickering, 1996). In particular, imposing the global acyclicity constraint constitutes a computational bottleneck. Given its practical relevance and the computational challenge, structure learning BNs is a very active area of research with new algorithms in continuous development. For a comparative study and an overview of well-established algorithms, we refer the reader to Constantinou et al. (2021) and Rios et al. (2021). Structure learning algorithms for BNs fall into two main categories: constraint- or score-based. Constraint-based methods employ CI tests to determine the presence or absence of an edge between each pair of nodes in the network structure. Score-based methods assign a global score to each network to quantify their overall ability to describe the data and search the space of all structures to find high-scoring networks. In this work, we focus on constraint-based methods.

Since a DAG entails a set of CI relationships (Pearl, 2009), it is possible to at least partially learn the graphical structure by estimating the CIs holding between elements of $\mathbf{X}$ from a collection of its observed realizations. However, different graphical structures can describe an identical set of CI relations, since the same joint distribution P may factorize according to different DAGs. Such a set of DAGs form a Markov equivalence class, and in practice, the underlying graph of a BN is only identifiable up to its equivalence class. A completed partially directed acyclic graph (CPDAG) commonly describes an equivalence class of DAGs. A CPDAG is a graph with both directed and undirected edges, and it encodes all the CI statements of a Markov equivalence class (Andersson et al., 1997). Conveniently, CPDAGs uniquely represent a Markov equivalence class, therefore in the absence of prior information about the graph the objective of structure learning reduces to recovering the correct CPDAG. The skeleton of a DAG $\mathcal{G}$ is an undirected graph over the same set of nodes that contains an undirected edge for every edge in $\mathcal{G}$.

Skeletons are core components of hybrid approaches to structure learning. Hybrid methods (Tsamardinos et al., 2006) aim to combine the computational advantage of CI testing with the higher accuracy of score-based methods to achieve better overall performance. First they employ a constraint-based algorithm to restrict the DAG search space via CI testing. A search-and-score then runs over the reduced space of structures to find high-scoring networks. Analogously, reliable skeletons may also provide a good preliminary search space for sampling methods (Kuipers et al., 2021), so that developing fast and accurate constraint-based methods remains a relevant topic of research.

In section 2 we briefly review the PC algorithm and CI testing under Gaussian data. Section 3 introduces the *dual PC* algorithm, our novel variation on the scheme, which provides substantial improvements both in terms of accuracy and run-time by supplementing

the CI tests at each step with complementary (or dual) tests. Finally, section 4 reports on a comparative evaluation of the PC algorithm and its dual version on simulated data. To ease reproducibility we provide full `R` implementations of our simulation studies with the code available at `https://github.com/enricogiudice/dualPC`.

2. The PC Algorithm

One of the most popular constraint-based structure learning methods is the PC algorithm (Spirtes et al., 1993). The approach assumes faithfulness of the probability distribution of the observed variables $\mathbf{X}$ to the unknown DAG $\mathcal{G}$ and the absence of latent confounders (causal sufficiency) for the relationships among $\mathbf{X}$. The algorithm proceeds in two phases: first it estimates the skeleton by performing a series of CI tests between variables. Starting from zero-order (marginal) independence between all pairs of variables, it recursively deletes an edge from the complete graph every time it fails to reject a hypothesis of conditional independence. In a second phase, it directs as many edges as possible while preserving compatibility with the pattern of CIs learned in the first phase. Estimating the undirected skeleton is the most critical part of the algorithm since directing the edges depends entirely on the results of the CI tests in the first phase.

Under the special case of jointly Gaussian data, and as long as the faithfulness and causal sufficiency assumptions hold, the PC algorithm enjoys interesting consistency properties even for asymptotically limited data for sparse graphs, (see e.g. Kalisch and Bühlmann, 2007, for details and pseudo-code). In general, as the sample size goes to infinity it produces the correct CPDAG. Its consistency guarantees and relatively simple implementation make the PC algorithm one of the most popular for structure learning, though it is inefficient when applied to high-dimensional datasets such as gene expression data (Duy Le et al., 2015). As a result, several methods aim to improve its efficiency (Silverstein et al., 2000; Sondhi and Shojaie, 2019). However, they either learn local modules of the structures instead of producing an entire CPDAG, thus compromising the structural accuracy, or rely on additional assumptions concerning the DAG structure to ensure consistency.

Another limitation of the PC algorithm is that it may produce different CPDAGs depending on the order of the variables in the dataset. Colombo and Maathuis (2014) propose a modification to the original algorithm called PC-stable with the property of being order-independent for the skeleton. PC-stable implements no pruning of the graph until it moves to the next size of the conditioning set, so it needs to carry out more tests compared to its standard version. Consequently, the running time is longer, further exacerbating the complexity problem.

2.1 Sample Version

A common assumption is that an observed $N \times n$ data matrix follows a jointly Gaussian distribution; where N denotes the number of observations and n the number of variables. One reason is the availability of conventional testing procedures for CI (Glymour et al., 2019). In practice, however, deviations from Gaussianity are common and we study their impact on the results in section 4.2. Partial correlation extends Pearson's correlation to measure the degree of association between two random variables conditional on a set of other variables. Full-order partial correlation between two variables in a set measures their correlation when

conditioning on all other variables in the set. Under a multivariate Gaussian distribution an explicit relationship links the full-order partial correlations to the entries of the precision (inverse covariance) matrix. Let P be the precision matrix of the random vector $\mathbf{X}$, with P_{ij} the element in the i-th row and the j-th column. Then

$$\rho_{X_i X_j | \mathbf{X} \setminus \{X_i, X_j\}} = \frac{-P_{ij}}{\sqrt{P_{ii} P_{jj}}} \tag{1}$$

which mimics the form of the marginal correlation $\rho_{X_i X_j} = \frac{\Sigma_{ij}}{\sqrt{\Sigma_{ii}\Sigma_{jj}}}$ ensuing from the covariance matrix Σ. More generally the concept of partial correlation allows for conditioning on any set $S \subseteq \mathbf{X} \setminus \{X_i, X_j\}$ of a lower order. In the Gaussian case, a partial correlation coefficient of zero characterizes CI (Lauritzen, 1996) leading to an efficient way of testing for CIs within the PC algorithm framework.

3. The Dual PC Algorithm

In its established implementation, the PC algorithm tests for CI starting from zero-order (marginal) independence and incrementally moving to higher-order conditioning sets. The strategy is justified because computing partial correlation coefficients for large conditioning sets is generally computationally costlier. Estimating an ℓ-order partial correlation coefficient analogously to equation (1) requires inverting an $(\ell + 2) \times (\ell + 2)$ covariance matrix. The inversion step has a polynomial time complexity in the number of variables, slowing the overall procedure whenever evaluating high-order CIs is needed. For denser graphs, however, testing high-order partial correlation coefficients may be unavoidable in the skeleton estimation phase. In such a case variables may share a large number of parents requiring the PC algorithm to test a large number of subsets before finding one large enough to render a pair of variables conditionally independent.

To overcome the above limitation we propose an alternative ordering of the CI tests, prioritising certain high-order partial correlations. The idea is to start testing for CI from both zero-order (marginal) and full-order partial correlations on the basis of the covariance and precision matrices. Our algorithm then proceeds to test more central-order conditioning sets from both directions, starting with first-order and $(|S| - 1)$th order partial correlation coefficients, where S is the current set of neighbouring nodes of any pair of variables (see figure 1). Furthermore, we aim to improve upon the implementation efficiency of the classic PC algorithm by inverting the covariance and precision matrices in blocks to cheaply estimate the partial correlation coefficients.

Let $V_{[I]}$ denote the subset indexed by a set I of a vector V. If A is a matrix and I and J are two index sets, then $A_{[I],[J]}$ denotes the $|I| \times |J|$ submatrix of A formed by the entries located in the rows indexed by I and the columns indexed by J. For example $A_{[1,3],[1,3,4]} = \begin{pmatrix} A_{11} & A_{13} & A_{14} \\ A_{31} & A_{33} & A_{34} \end{pmatrix}$. Starting with the covariance matrix Σ of the data, our approach first identifies all pairs of variables for which marginal independence cannot be rejected, as in a classic PC procedure. After deleting all edges between such pairs of nodes from the initial complete undirected graph, we invert the covariance matrix to evaluate the precision matrix and additionally delete any edges between variables for which we cannot reject full-order CI. The following step tests pairwise independencies when conditioning on

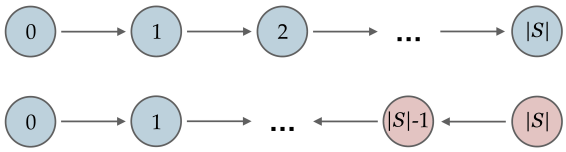

Figure 1: Order of testing of the different conditioning set sizes for a given pair of variables in the PC (top) and the dual PC algorithm (bottom). S is the current set of variables adjacent to the pair; blue and red sets are complementary to each other. The conditioning set size increases gradually for blue nodes; the dual PC algorithm proceeds to tests from both ends, in decreasing order for the red nodes.

sets of size 1, again as in the classic PC approach. For every ordered pair X_i, X_j of nodes still adjacent in the current instance of the skeleton, we find the set S of nodes adjacent to X_i, excluding X_j. Next we build the local covariance matrix $U = \Sigma_{[i,j,\zeta],[i,j,\zeta]}$ and the local precision matrix $T = U^{-1}$, where ζ is the index set of S: $\mathbf{X}_\zeta = S$. There are two advantages in computing T: first, it allows us to perform the tests for the complement sets of size $|S|-1$ more efficiently. The algorithm will typically need to test a large number of such dual sets, and we can reuse T in the same way we reuse U for testing multiple sets of size 1. Second, by applying equation (1) to T we directly obtain the estimated partial correlation coefficient $\hat{\rho}_{X_i X_j|S}$. If we reject the independence of X_i and X_j conditionally on the whole set S, we test conditionally on each variable $X_\mathcal{K} \in S$. Let k denote the index of variable $X_\mathcal{K}$ within the local covariance matrix U with $k \in \{3, \ldots, |S|+2\}$ and the shift mapping $\zeta_{(k-2)} = \mathcal{K}$. Using block inversion on $U_{[1,2,k],[1,2,k]}$ we extract its first 2×2 submatrix:

$$\left(U_{[1,2,k],[1,2,k]}\right)^{-1}_{[1,2],[1,2]} = \left(U_{[1,2],[1,2]} - U_{[1,2],[k]}\, U_{kk}^{-1}\, U_{[k],[1,2]}\right)^{-1}. \tag{2}$$

To compute the first-order partial correlations for every $X_\mathcal{K} \in S$ we can use equation (1). Note that applying equation (1) to the 2×2 matrix we obtain in equation (2) before the inversion yields the same partial correlation coefficient in absolute value. Because the sign of the partial correlation coefficient does not matter for our testing purposes, we can apply equation (1) directly without the final inversion, additionally speeding up the procedure. Therefore, we only need to compute

$$B(U,k) = U_{[1,2],[1,2]} - U_{[1,2],[k]}\, U_{kk}^{-1}\, U_{[k],[1,2]}\,, \qquad |\hat{\rho}_{X_i X_j|X_\mathcal{K}}| = \left|\frac{B_{12}}{\sqrt{B_{11}B_{22}}}\right|. \tag{3}$$

Every time the algorithm rejects the null hypothesis of independence for a variable $X_\mathcal{K}$, it proceeds to test the complementary (or dual) set $S \setminus X_\mathcal{K}$. Naively one could compute the first 2×2 block of $(U_{[-k],[-k]})^{-1}$, which would require inverting a matrix of size $(|S|-1) \times (|S|-1)$ for every variable $X_\mathcal{K}$. More efficiently we can write the desired 2×2 matrix in terms of T:

$$\begin{aligned}\left(U_{[-k],[-k]}\right)^{-1}_{[1,2],[1,2]} &= \left(T_{[-k],[-k]} - T_{[-k],[k]}\, T_{kk}^{-1}\, T_{[k],[-k]}\right)_{[1,2],[1,2]} \\ &= T_{[1,2],[1,2]} - T_{[1,2],[k]}\, T_{kk}^{-1}\, T_{[k],[1,2]} = B(T,k)\,.\end{aligned} \tag{4}$$

Therefore by simply applying equations (3) to the matrix T we can compute $\hat{\rho}_{X_i X_j | S \setminus X_\mathcal{K}}$. As soon as the algorithm finds a variable $X_\mathcal{K}$ or a dual set $S \setminus X_\mathcal{K}$ conditionally on which it cannot reject independence between X_i and X_j, it deletes the edge between the two nodes and it moves on to a new pair. After having tested all the remaining edges, the algorithm moves on to conditioning sets $X_\mathcal{K}$ of size 2, and their dual counterparts. The approach for testing CIs does not change for these new sets, since equations (2) and (4) hold for index sets k of any size. As for the PC algorithm, the size of the conditioning sets $X_\mathcal{K}$ progressively increases until there can be no higher-order CIs that would result in deleting an edge. Because inverting U_{kk} in equation (2) can be computationally expensive for larger sets $X_\mathcal{K}$, we proceed by computing its Cholesky decomposition $U_{kk} = C'C$, with C an upper triangular matrix. To solve the linear system $Cx = U_{[k],[1,2]}$ we can then use back substitution; and finally, compute the matrix block of interest as

$$B(U, k) \;=\; U_{[1,2],[1,2]} - x'x\,. \tag{5}$$

In our implementation, we test the whole set S before testing any of its subsets $X_\mathcal{K}$ since later dual tests anyway rely on the local precision matrix T. Indeed, if $|S| < 2|k|$ we use the local precision matrix T to avoid inverting the $|k| \times |k|$ matrix U_{kk} in equation (2). To compute the partial correlation coefficient $\hat{\rho}_{X_i X_j | X_\mathcal{K}}$ it is more efficient to set k equal to its complementary set $\{3, \dots, |S| + 2\} \setminus k$ in equation (4).

The core part of the dual PC algorithm outputs an undirected skeleton. To estimate a CPDAG we can use the same edge-orienting procedure as the classic PC algorithm. Just like the original PC algorithm, the dual PC version does not satisfy the order independence property, since the tested sets depend on the previous (potentially incorrect) edge deletions. Modifying the dual PC algorithm to achieve order-independence for the skeleton as in Colombo and Maathuis (2014) is straightforward. In its stable (order-independent) version, we test independencies between all pairs of variables for a given value of the conditioning set size ℓ before deleting an edge for each CI we cannot reject.

3.1 Consistency

Under faithfulness of the distribution P to the true DAG $\mathcal{G}$, the classic PC algorithm is pointwise consistent (Spirtes et al., 1993) i.e., it constructs the CPDAG corresponding to the equivalence class of $\mathcal{G}$ as the sample size approaches infinity. If P is Gaussian, the estimated covariance matrix converges to the true covariance matrix, determining, in the given limit, CIs without errors corresponding to the so-called "population version" of the algorithm which ignores sampling variability. Furthermore, Kalisch and Bühlmann (2007) proved uniform consistency in the Gaussian case for certain sparse high-dimensional graphs under additional assumptions. Spirtes et al. (1993) formulated the original proof of consistency in the context of causal inference, relying on the concept of d-separation (Pearl, 1988). For BNs $\langle \mathcal{G}, P \rangle$ where P is faithful to $\mathcal{G}$, CI of X_i and X_j given $S \subseteq \mathbf{X} \setminus \{X_i, X_j\}$ is equivalent to d-separation of the nodes X_i and X_j given the set S (Verma and Pearl, 1988). Therefore the considerations in (Spirtes et al., 1993, theorem 5.1) also apply to the dual PC algorithm, since the properties of the output graph remain unaffected by the different ordering of the CI tests. The population version of the dual PC algorithm will share the same properties as the classic version, and under faithfulness of the Gaussian distribution P to the underlying DAG $\mathcal{G}$, it will produce the CPDAG corresponding to the Markov equivalence class of $\mathcal{G}$.

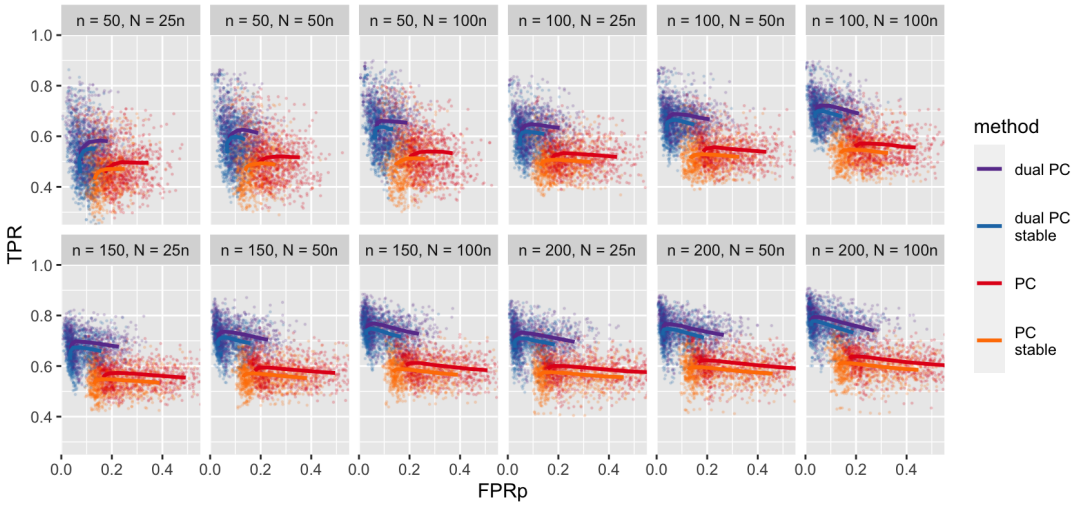

Figure 2: ROC-like curves illustrating the ability of the dual and classic PC algorithms respectively in recovering the correct CPDAG, with N the sample sizes, and n the number of nodes in the DAGs.

4. Simulation Study

To evaluate the performance of the dual PC algorithm we sample synthetic data from randomly generated BNs. To draw a DAG over n nodes we use the function `randDAG` from the `R` package `pcalg` (Kalisch et al., 2012), with the settings for the Erdős-Rényi model, to sample DAGs with iid probabilities of edge inclusion. For each DAG we then sample N instances for every node as a noisy linear function of its parents:

$$Y = \sum_{i=1}^{|\mathrm{Pa}(Y)|} w_i \, \mathrm{Pa}_i(Y) + \epsilon \,, \qquad \epsilon \sim \mathcal{N}(0,1) \tag{6}$$

with the weights w_i sampled from a uniform distribution on the interval $(0.4, 2)$. The data are standardised after generation. To assess how the algorithm behaves for networks of different sizes, we evaluate the performance across DAGs with 50, 100, 150 and 200 nodes. For every number of nodes n, we consider different scenarios with $25n$, $50n$ and $100n$ observations of each variable. For every combination of n and N, we generate 100 DAGs and a data matrix from each of them. In the software, we choose a setting designed to achieve an expected number of parents $d = 2$ for each node, corresponding to relatively dense networks, typically more challenging to learn than sparser ones and where we expect the dual PC to make more of a difference.

4.1 Performance Metrics

A way to assess performance is to compare estimated and ground truth structures across all scenarios of different sample and graph sizes. Because we can only identify DAGs up to their equivalence class, we perform the comparison on the space of CPDAGs. The classic

PC algorithm from the `R` package `pcalg` serves as benchmark. For the comparative study, we build receiver operating characteristic (ROC)-like curves by varying the significance level α of the CI tests and counting true positive (TP) and false positive (FP) edges in the estimated graph. Then we scale them both by the number of true edges P to obtain the true positive edge rate (TPR $= \frac{\text{TP}}{\text{P}}$) and a modified false positive edge rate (FPRp $= \frac{\text{FP}}{\text{P}}$). Scaling FP by P rather than by the total true negative edges, which can be exceedingly large for DAGs, aims to have measures (TPR and FPRp) on comparable scales. Additionally, we measure and compare run times. Code to reproduce all simulations in `R` is available at `https://github.com/enricogiudice/dualPC`.

4.2 Results

The scatter plots of figure 2 display FPRp and TPR, on the space of CPDAGs, for values of the significance level α ranging from 0.002 to 0.25. The solid lines represent (partial) average ROC-like curves constructed by averaging the FPRps and TPRs of every method for each value of α. The comparison includes four distinct algorithms: the dual PC algorithm and its order-independent counterpart ("dual PC" and "dual PC stable"), as well as the standard and stable versions of the classic PC algorithm. The performance of the dual algorithms is on average superior in all simulated scenarios, reaching higher sensitivity for the same level of FPs. The dual PC achieves both a lower FPRp and higher TPR than the classic PC algorithm for the same value of the significance level α. One possible explanation is that the dual PC runs far fewer CI tests. In the simulated scenarios, the dual version of the PC algorithm performed on average at most one-third of the number of tests carried out by its classic counterpart. Performance evaluation in terms of structural Hamming distance (SHD) is also in agreement with the ROC-like curves. Results for the average number of tests and the SHD, as well as on sparser networks, and pseudo-code are available in an extended version of this manuscript at `https://arxiv.org/abs/2112.09036`.

Figure 3 shows the distribution of run-times for the different methods to estimate the CPDAG, with significance level $\alpha = .05$. On average the dual PC algorithm is roughly one order of magnitude faster than the classic implementation of the PC algorithm. The result persists across different simulation settings.

To determine the sensitivity to the assumption of strict Gaussianity of the data, we repeat the simulations with the noise ϵ in equation (6) following a Student's t-distribution instead. Figure 4 shows the ROC-like curves under varying degrees of freedom ν of the noise distribution, for networks with 50 nodes and 2500 observations per node. Additionally, we add greedy equivalence search (GES) (Chickering, 2003) as a benchmark, generally known to produce more FP edges than PC, while finding more TPs at higher FP rates (Rios et al., 2021). To define a skeleton for hybrid methods, the sparser output of the dual PC provides a smaller initial DAG space and therefore it may be preferable. The results indicate that deviations from Gaussianity do not greatly affect the performance of the dual PC algorithm relative to the other methods and it remains competitive.

5. Conclusions

In this work, we proposed a novel scheme for running the CI tests within the PC algorithm framework. Our strategy consists of running the tests proceeding from both sides, starting

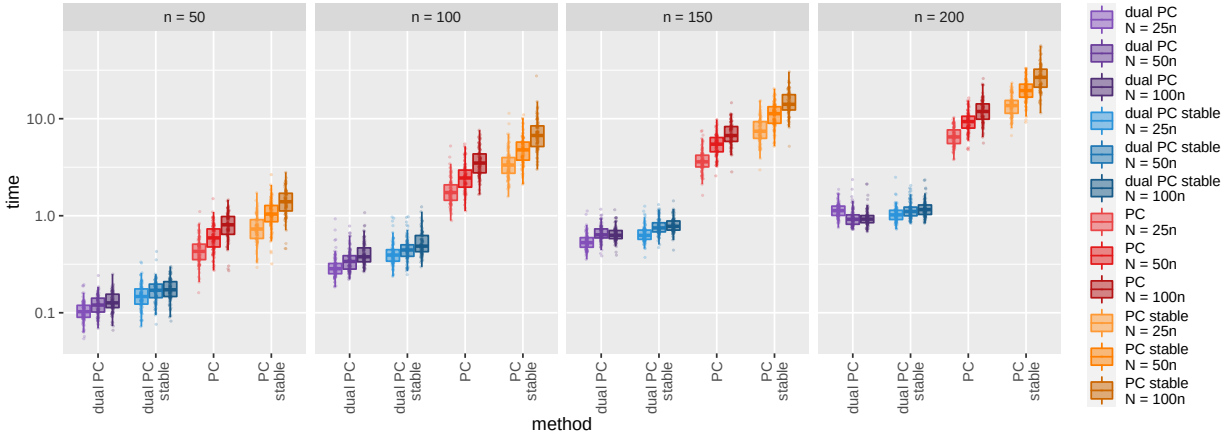

Figure 3: Running times for estimating the CPDAG for all variations of PC algorithms we considered (dual/classic, standard/stable), for a significance level $\alpha = 5\%$. Time is measured in seconds and is displayed on a log scale. N indicates the data sample sizes and n the number of nodes in the DAGs.

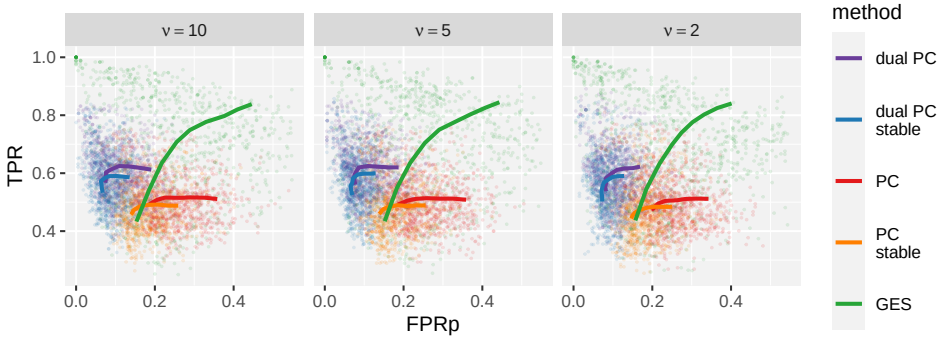

Figure 4: The ability of different methods in recovering the correct CPDAG for varying levels of non-Gaussianity. The data generating model at each node is a linear function of its parents plus a Student's t-distributed noise with ν degrees of freedom.

from zero and full-order conditioning sets and progressively moving to central-order tests. We harness the properties of the precision matrix to efficiently compute partial correlation coefficients for the dual conditioning sets. This new approach results in faster run times compared to the classic PC by performing CI tests at a lower computational cost, while also requiring overall fewer tests. As a consequence, the dual PC strategy accurately identifies high-order CI relations more effectively than the classic version of the PC algorithm. Our simulation studies show that the dual PC algorithm achieves a better performance in terms of both structure accuracy and run time than the classic PC algorithm. Hence the dual PC will extend more easily to efficiently estimate graphs in high-dimensional settings.

BNs provide a natural representation for causally induced CIs and are extensively used for modelling causal relationships between variables (Pearl, 2009). Unsurprisingly, constraint-based structure learning algorithms such as PC are popular tools for learning causal diagrams (Spirtes et al., 1993). Improving on their speed and accuracy will benefit and extend the possibilities for exploratory causal analyses in high-dimensional settings. Notably, constrained-based methods also constitute an important component of some hybrid methods for structure learning of Bayesian networks. The PC algorithm in fact may serve to restrict the search space of DAGs for a search-and-score strategy. Pruning the space with a constrained-based algorithm before proceeding to search-and-score may help to explore the space more efficiently (Tsamardinos et al., 2006). The dual PC may also provide an opportunity to further improve sampling methods since its increased accuracy and speed compared to the classic PC algorithm would enable a more efficient sampling from the bulk of the posterior probability mass. As such, combining the dual PC algorithm with state-of-the-art MCMC sampling schemes (Viinikka et al., 2020; Kuipers et al., 2021) holds the potential to improve the Bayesian treatment of larger networks.

A limitation of the dual PC algorithm is that a (partial) correlation coefficient of zero only characterizes (conditional) independence in the jointly Gaussian case. Nevertheless, our simulations show that moderate deviations from Gaussianity do not affect the relative performance of our method, though more general data types may require non-parametric conditional independence testing procedures. Thanks to its versatility, there have been several successful attempts at extending the PC algorithm to handle non-Gaussian continuous data (Zhang et al., 2012; Chakraborty and Shojaie, 2021). In its current form, the dual PC algorithm relies for computational efficiency on the relationship holding under Gaussianity between the precision matrix and the partial correlation. However, it is reasonable to expect improved network learning performance when we supplement low-order CI tests with tests of their complement sets. This argument is supported by our simulations showing that the dual PC requires fewer tests compared to the classic PC algorithm. Therefore, we anticipate that the reduction in the number of tests might translate into shorter run times and higher accuracy despite the additional computational burden of non-parametrically testing high-order CIs. On these grounds we speculate that there may be a value in adapting the dual PC algorithm to deal with non-Gaussian data, holding the potential to improve the structure learning accuracy and lower the computational cost.

Acknowledgments

The authors are grateful to acknowledge partial funding support for this work from the two Cantons of Basel through project grant PMB-02-18 granted by the ETH Zurich.

References

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

J. C. Bird, R. Evans, F. Waite, B. S. Loe, and D. Freeman. Adolescent paranoia: prevalence, structure, and causal mechanisms. *Schizophrenia Bulletin*, 45:1134–1142, 2018.

S. Chakraborty and A. Shojaie. Nonparametric causal structure learning in high dimensions. arXiv 2106.11415, 2021.

D. M. Chickering. *Learning Bayesian networks is NP-complete*, pages 121–130. Springer-Verlag, Learning from data: artificial intelligence and statistics v edition, 1996.

D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3:507–554, 2003.

D. Colombo and M. H. Maathuis. Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15:3921–3962, 2014.

A. C. Constantinou, Y. Liu, K. Chobtham, Z. Guo, and N. K. Kitson. Large-scale empirical validation of Bayesian network structure learning algorithms with noisy data. *International Journal of Approximate Reasoning*, 131:151–188, 2021.

L. M. de Campos and A. E. Romero. Bayesian network models for hierarchical text classification from a thesaurus. *International Journal of Approximate Reasoning*, 50:932 – 944, 2009. Special section on graphical models and information retrieval.

T. Duy Le, T. Hoang, J. Li, L. Liu, and H. Liu. A fast PC algorithm for high dimensional causal discovery with multi-core PCs. arXiv 1502.02454, 2015.

F. Elwert. *Graphical causal models*, pages 245–273. Springer Netherlands, 2013.

N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303: 799–805, 2004.

N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000.

C. Glymour, K. Zhang, and P. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10:524, 2019.

M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, 8:613–636, 2007.

M. Kalisch, M. Mächler, D. Colombo, M. Maathuis, and P. Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software, Articles*, 47:1–26, 2012.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques - adaptive computation and machine learning*. The MIT Press, 2009.

J. Kuipers, T. Thurnherr, G. Moffa, P. Suter, J. Behr, R. Goosen, G. Christofori, and N. Beerenwinkel. Mutational interactions define novel cancer subgroups. *Nature Communications*, 9:4353, 2018.

J. Kuipers, P. Suter, and G. Moffa. Efficient sampling and structure learning of Bayesian networks. arXiv 1803.07859, 2021.

S. L. Lauritzen. *Graphical models*. Oxford University Press, 1996.

G. Moffa, G. Catone, J. Kuipers, E. Kuipers, D. Freeman, S. Marwaha, B. R. Lennox, M. R. Broome, and P. Bebbington. Using directed acyclic graphs in epidemiological research in psychosis: an analysis of the role of bullying in psychosis. *Schizophrenia Bulletin*, 43: 1273–1279, 2017.

G. Moffa, J. Kuipers, G. Carrà, C. Crocamo, E. Kuipers, M. Angermeyer, T. Brugha, M. Toumi, and P. Bebbington. Longitudinal symptomatic interactions in long-standing schizophrenia: a novel five-point analysis based on directed acyclic graphs. *Psychological Medicine*, pages 1–8, 2021.

M. Neil, N. Fenton, M. Osman, and S. McLachlan. Bayesian network analysis of covid-19 data reveals higher infection prevalence rates and lower fatality rates than widely reported. *Journal of Risk Research*, 23:866–879, 2020.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc., 1988.

J. Pearl. *Causality: models, reasoning and inference*. Cambridge University Press, 2nd edition, 2009.

F. L. Rios, G. Moffa, and J. Kuipers. Benchpress: a scalable and platform-independent workflow for benchmarking structure learning algorithms for graphical models. arXiv 2107.03863, 2021.

R. W. Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial Mathematics V*, pages 28–43. Springer Berlin Heidelberg, 1977.

C. Silverstein, S. Brin, R. Motwani, and J. Ullman. Scalable techniques for mining causal structures. *Data Mining and Knowledge Discovery*, 4:163–192, 2000.

A. Sondhi and A. Shojaie. The reduced PC-algorithm: improved causal structure learning in large random networks. *Journal of Machine Learning Research*, 20:1–31, 2019.

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

I. Tsamardinos, L. Brown, and C. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006.

T. Verma and J. Pearl. Causal networks: semantics and expressiveness. In *Proceedings of the Fourth Annual Conference on Uncertainty in Artificial Intelligence*, pages 69–78. North-Holland Publishing Co., 1988.

J. Viinikka, A. Hyttinen, J. Pensar, and M. Koivisto. Towards scalable Bayesian learning of causal DAGs. In *Advances in Neural Information Processing Systems*, volume 33, pages 6584–6594. Curran Associates, Inc., 2020.

X. Zhang, X.-M. Zhao, K. He, L. Lu, Y. Cao, J. Liu, J.-K. Hao, Z.-P. Liu, and L. Chen. Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. *Bioinformatics*, 28:98–104, 2012.