Supplementary Material For: Order-Independent Structure Learning of Multivariate Regression Chain Graphs [⋆]

Mohammad Ali Javidian, Marco Valtorta, and Pooyan Jamshidi

University of South Carolina
javidian@email.sc.edu, mgv@cse.sc.edu, pjamshid@cse.sc.edu

Abstract. This supplementary document is organized as follows. Section 1 contains details of our experimental results and Section 2 contains the proofs of the theoretical results presented in Section 4.

Keywords: Chain graph · Multivariate Regression Chain Graph · Structural learning · Order-independence · High-dimensional data.

Table 1 summarizes the three order-dependence issues explained above and the corresponding modifications of the PC-like algorithm that removes the given order-dependence problem.

Table 1. Order-dependence issues and corresponding modifications of the PC-like algorithm that remove the problem. "Yes" indicates that the corresponding aspect of the graph is estimated order-independently in the sample version.

	skeleton	v-structures decisions	edges orientations
PC-like	No	No	No
stable PC-like	Yes	No	No
stable CPC/MPC-like	Yes	Yes	No
stable LCPC/LMPC-like	Yes	Yes	Yes

1 Evaluation

In this section, we evaluate the performance (measuring both time and several indicators of the quality of the learned CGs) of our algorithms in various setups using simulated data sets generated from synthetic CGs. We compare the performance of our algorithms with the original PC-like learning algorithm by running them on randomly generated MVR chain graphs in low-dimensional and

[⋆] Supported by organization x.

high-dimensional data, respectively. Empirical simulations show that our algorithm achieves competitive results with the original PC-like learning algorithm; in particular, in the Gaussian case the order-independent algorithms achieve output of better quality than the original PC-like algorithm, especially in high-dimensional settings. Algorithm **??** and the stable PC-like algorithms have been implemented in the R language ???cite???.

Performance Evaluation on Random MVR CGs (Gaussian case) To investigate the performance of the proposed learning methods in this paper, we use the same approach that [4] used in evaluating the performance of the LCD algorithm on LWF chain graphs. We run our algorithms on randomly generated MVR chain graphs and then we compare the results and report summary error measures in all cases.

Data Generation Procedure First we explain the way in which the random MVR chain graphs and random samples are generated. Given a vertex set V, let $p = |V|$ and N denote the average degree of edges (including bidirected, pointing out, and pointing in) for each vertex. We generate a random MVR chain graph on V as follows:

- Order the p vertices and initialize a $p \times p$ adjacency matrix A with zeros;
- Set each element in the lower triangle part of A to be a random number generated from a Bernoulli distribution with probability of occurrence $s = N/(p-1)$;
- Symmetrize A according to its lower triangle;
- Select an integer k randomly from $\{1, \dots, p\}$ as the number of chain components;
- Split the interval $[1, p]$ into k equal-length subintervals $I_1, \dots, I_k$ so that the set of variables into each subinterval I_m forms a chain component C_m;
- Set $A_{ij} = 0$ for any (i, j) pair such that $i \in I_l, j \in I_m$ with $l > m$.

This procedure yields an adjacency matrix A for a chain graph with $(A_{ij} = A_{ji} = 1)$ representing a bidirected edge between V_i and V_j and $(A_{ij} = 1, A_{ji} = 0)$ representing a directed edge from V_i to V_j. Moreover, it is not difficult to see that $\mathbb{E}[\text{vertex degree}] = N$, where an adjacent vertex can be linked by either a bidirected or a directed edge. In order to sample the artificial CGs, we first transform them into DAGs and then generate samples from these DAGs under marginalization, as indicated in [2], using Hugin.

Experimental Results We evaluate the performance of the proposed algorithms in terms of six measurements: (a) the true positive rate (TPR) (also known as sensitivity, recall, and hit rate), (b) the false positive rate (FPR) (also known as fall-out), (c) the true discovery rate (TDR) (also known as precision or positive predictive value), (d) accuracy (ACC) for the skeleton, (e) the structural Hamming distance (SHD) (this is the metric described in [9] to compare the structure of the learned and the original graphs), and (f) run-time for the LCG recovery algorithms. In short, $TPR = \frac{\text{true positive } (TP)}{\text{the number of positive cases in the data } (Pos)}$ is the ratio of the number of correctly identified edges over total number of edges (in true graph), $FPR = \frac{\text{false positive } (FP)}{\text{the number of negative cases in the data } (Neg)}$ is the ratio of the number of incorrectly identified edges over total number of gaps, $TDR = \frac{\text{true positive } (TP)}{\text{the total number of edges in the recovered CG}}$ is the ratio of the number of correctly identified edges over total number of edges (both in estimated graph), $ACC = \frac{\text{true positive } (TP) + \text{ true negative } (TN)}{Pos+Neg}$ and SHD is the number of legitimate operations needed to change the current resulting graph to the true essential graph, where legitimate operations are: (a) add or delete an edge and (b) insert, delete or reverse an edge orientation. In principle, large values of TPR, TDR, and ACC, and small values of FPR and SHD indicate good performance. All of these six measurements are computed on the essential graphs of the CGs, rather than the CGs directly, to avoid spurious differences due to random orientation of undirected edges. We used Algorithm 1 in [8] to convert the true MVR CG to its essential graph.

In our simulation, for low-dimensional settings, we set N (expected number of adjacent vertices) to 2 and change the parameters p (the number of vertices) and n (sample size) and as follows:

- $p \in \{10, 20, 30, 40, 50\}$,
- $n \in \{500, 1000, 5000, 10000\}$.

For each (p, N) combination, we first generate 30 random MVR CGs. We then generate a random Gaussian distribution based on each graph (transformed DAG) and draw an identically independently distributed (i.i.d.) sample of size n from this distribution for each possible n. For each sample, four different significance levels ($\alpha = 0.001, 0.005, 0, 01, 0.05$) are used to perform the hypothesis tests. The *null hypothesis* H_0 is "two variables u and v are conditionally independent given a set C of variables" and alternative H_1 is that H_0 may not hold. We then compare the results to access the influence of the significance testing level on the performance of our algorithms.

For the high-dimensional setting, we generate 30 random MVR CGs with 1000 vertices for which the expected number of adjacent vertices for each vertex is 2. We then generate a random Gaussian distribution based on each graph (transformed DAG) and draw an identically independently distributed (i.i.d.) sample of size 50 from this distribution for each DAG. These numbers are similar to ones that could be encountered in gene regulatory network experiments [1, section 6].

Figure 1 shows that: (a) as we expected [4,3], all algorithms work well on sparse graphs ($N = 2$), (b) for all algorithms, typically the TPR, TDR, and ACC increase with sample size, (c) for all algorithms, typically the SHD and FPR decrease with sample size, (d) a large significance level ($\alpha = 0.05$) typically yields large TPR, FPR, and SHD, (e) while the stable PC-like algorithm has a better TDR and FPR in comparison with the original PC-like algorithm, the original PC-like algorithm has a better TPR (as observed in the case of DAGs [1]). This can be explained by the fact that the stable PC-like algorithm tends to perform more tests than the original PC-like algorithm, and (h) while the original PC-like algorithm has a (slightly) better SHD in comparison with the stable PC-like algorithm in low-dimensional data, the stable PC-like algorithm has a better SHD in high-dimensional data. Also, (very) small variances indicate that the order-independent versions of the PC-like algorithm in high-dimensional data are stable. When considering average running times versus sample sizes, as shown in Figure 1, we observe that: (a) the average run time increases when sample size increases; (b) generally, the average run time for the original PC-like algorithm is (slightly) better than that for the stable PC-like algorithm in both low and high dimensional settings.

2 Proofs of Theorems in Section 4.

Two vertices x and y in chain graph G are said to be collider connected if there is a chain from x to y in G on which every non-endpoint vertex is a collider; such a chain is called a collider chain. Note that a single edge trivially forms a collider chain (path), so if x and y are adjacent in a chain graph then they are collider connected. The augmented graph derived from G, denoted $(G)^a$, is an undirected graph with the same vertex set as G such that

$$c \,\text{---}\, d \text{ in } (G)^a \Leftrightarrow c \text{ and } d \text{ are collider connected in } G.$$

Disjoint sets $X, Y \neq \emptyset$, and Z (Z may be empty) are said to be m^*-separated if X and Y are separated by Z in $(G_{an(X\cup Y\cup Z)})^a$. Otherwise X and Y are said to be m^*-connected given Z. The resulting independence model is denoted by $\Im_{m^*}(G)$. According to [6, Theorem 3.18.] and [2], for MVR chain graph G we have: $\Im_m(G) = \Im_{m^*}(G)$.

Theorem 1. Let the distribution of V be faithful to an MVR CG G, and assume that we are given perfect conditional independence information about all pairs of variables (u, v) in V given subsets $S \subseteq V \setminus \{u, v\}$. Then the output of the stable PC-like algorithm is an MVR CG that has exactly the minimum set of bidirected edges for its equivalence class.

Proof. The proof of Theorem 1 is completely analogous to the proof of Theorem 3 and 4 for the original PC-like algorithm in [7].

Theorem 2. The skeleton resulting from the sample version of the stable PC-like algorithm is order-independent.

Proof. We consider the removal or retention of an arbitrary edge $u \,\text{---}\, v$ at some level i. The ordering of the variables determines the order in which the edges (line 7 of Algorithm 2) and the subsets S of $a_H(u)$ and $a_H(v)$ (line 8 of Algorithm 2) are considered. By construction, however, the order in which edges are considered does not affect the sets $a_H(u)$ and $a_H(v)$.

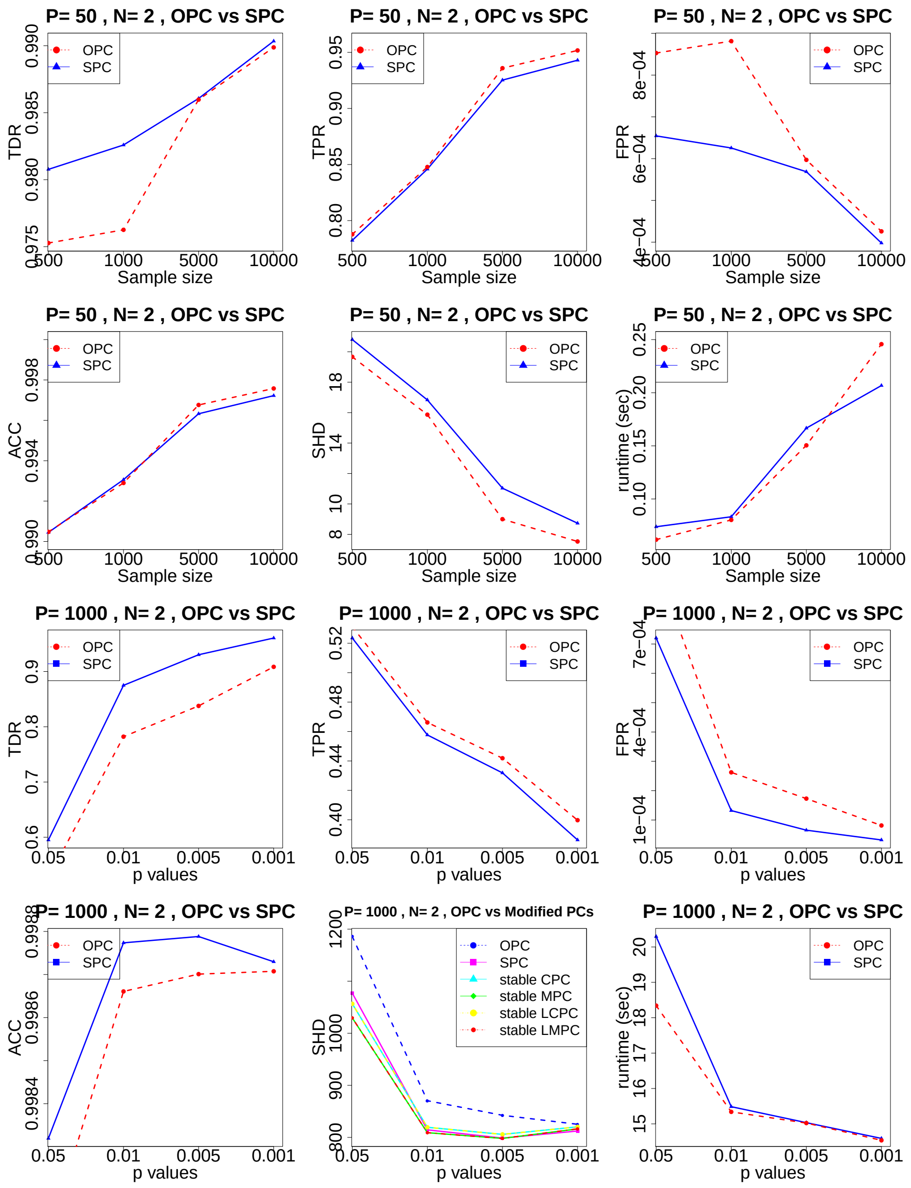

Fig. 1. The first two rows show the performance of the original (OPC) and stable PC-like (SPC) algorithms for randomly generated Gaussian chain graph models: average over 30 repetitions with 50 variables correspond to N = 2, and the significance level $\alpha = 0.001$. The last two rows show the performance of the original (OPC) and stable PC-like (SPC) algorithms for randomly generated Gaussian chain graph models: average over 30 repetitions with 1000 variables correspond to N = 2, sample size S=50, and the significance level $\alpha = 0.05, 0.01, 0.005, 0.001$. The results are shown as averages, computed over randomly generated graphs and lexicographical variable ordering per graph.

If there is at least one subset S of $a_H(u)$ or $a_H(v)$ such that $u \perp\!\!\!\perp_p v|S$, then any ordering of the variables will find a separating set for u and v. (Different orderings may lead to different separating sets as illustrated in Example 2, but all edges that have a separating set will eventually be removed, regardless of the ordering). Conversely, if there is no subset S' of $a_H(u)$ or $a_H(v)$ such that $u \perp\!\!\!\perp_p v|S'$, then no ordering will find a separating set.

Hence, any ordering of the variables leads to the same edge deletions, and therefore to the same skeleton.

Theorem 3. Let the distribution of V be faithful to an MVR CG G, and assume that we are given perfect conditional independence information about all pairs of variables (u, v) in V given subsets $S \subseteq V \setminus \{u, v\}$. Then the output of the (stable) CPC/MPC-like algorithm is an MVR CG that is Markov equivalent with G that has exactly the minimum set of bidirected edges for its equivalence class.

Proof. The skeleton of the learned CG is correct by Theorem 1. Now, we prove that for any unshielded triple (X_i, X_j, X_k) in an MVR CG G, X_j is either in all sets that m-separate X_i and X_k or in none of them. Since X_i, X_k are not adjacent and any MVR chain graph is a maximal ancestral graph [2], they are m-separated given some subset $S \setminus \{X_i, X_k\}$ due to the maximal property. Based on the pathwise m-separation criterion for MVR CGs (see section 2), X_j is a collider node in G if and only if $X_j \notin An(S)$. So, $X_j \notin S$. On the other hand, if X_j is a non-collider node then $X_j \in S$, for all S that m-separate X_i and X_k. Because in this case, $X_j \in An(X_i \cup X_k \cup S)$ and so there is an undirected path X_i — X_j — X_k in $(G_{An(X_i \cup X_k \cup S)})^a$. Any set $S \setminus \{X_i, X_k\}$ that does not contain X_j will fail to m-separate X_i and X_k because of this undirected path. As a result, unshielded triples are all unambiguous. Since all unshielded triples are unambiguous, the orientation rules are as in the (stable) PC-like algorithm. Therefore, the output of the (stable) CPC/MPC-like algorithm is an MVR CG that is Markov equivalent with G that has exactly the minimum set of bidirected edges for its equivalence class, and soundness and completeness of these rules follows from Sonntag and Peña [7].

Theorem 4. The decisions about v-structures in the sample version of the stable CPC/MPC-like algorithm is order-independent.

Proof. The stable CPC/MPC-like algorithm have order-independent skeleton, by Theorem 2. In particular, this means that their unshielded triples and adjacency sets are order-independent. The decision about whether an unshielded triple is unambiguous and/or a v-structure is based on the adjacency sets of nodes in the triple, which are order independent.

Theorem 5. Let the distribution of V be faithful to an MVR CG G, and assume that we are given perfect conditional independence information about all pairs of variables (u, v) in V given subsets $S \subseteq V \setminus \{u, v\}$. Then the output of the (stable) LCPC/LMPC-like algorithm is an MVR CG that is Markov equivalent with G that has exactly the minimum set of bidirected edges for its equivalence class.

Proof. By Theorem 3, we know that the (stable) CPC-like and (stable) MPC-like algorithms are correct. With perfect conditional independence information, there are no conflicts between orientation rules in the essential graph recovery phase of the algorithms. Therefore, the (stable) LCPC-like and (stable) LMPC-like algorithms are identical to the (stable) CPC-like and (stable) MPC-like algorithms.

Theorem 6. The sample versions of (stable) CPC-like and (stable) MPC-like algorithms are fully order-independent.

Proof. This follows straightforwardly from Theorems 2 and 4 and the procedure with lists and bi-directed edges discussed above.

Conclusion In this article we proposed several modifications of the PC-like algorithm [7] for learning a CG from a probability distribution faithful to a multivariate regression CG. The algorithms presented are constraint-based and inspired by [1]. We showed that the PC-like algorithm is order-dependent, in the sense that the output can depend on the order in which the variables are given. This order-dependence is a minor issue in low-dimensional settings. As Colombo and Maathuis [1] showed, however, it can be very pronounced in high-dimensional settings, where it can lead to highly variable results. We proposed several modifications of the original PC-like algorithm that remove part or all of this order-dependence. We showed that our modifications yield similar performance in low-dimensional settings and improved performance in high-dimensional settings. We believe that our approach is extendable to the structural learning of other CG interpretations (i.e., LWF and AMP CGs) and marginal AMP chain graphs [5].

References

1. Colombo, D., Maathuis, M.H.: Order-independent constraint-based causal structure learning. The Journal of Machine Learning Research **15**(1), 3741–3782 (2014)
2. Javidian, M.A., Valtorta, M.: On the properties of MVR chain graphs. In: Workshop proceedings of the 9th International Conference on Probabilistic Graphical Models. pp. 13–24 (2018)
3. Kalisch, M., Bühlmann, P.: Estimating high-dimensional directed acyclic graphs with the pc-algorithm. J. Mach. Learn. Res. **8**, 613–636 (2007)
4. Ma, Z., Xie, X., Geng, Z.: Structural learning of chain graphs via decomposition. Journal of Machine Learning Research **9**, 2847–2880 (2008)
5. Peña, J.M., Gómez-Olmedo, M.: Learning marginal AMP chain graphs under faithfulness revisited. International Journal of Approximate Reasoning **68**, 108 – 126 (2016)
6. Richardson, T.S., Spirtes, P.: Ancestral graph markov models. The Annals of Statistics **30**(4), 962–1030 (2002)

7. Sonntag, D., Peña, J.M.: Learning multivariate regression chain graphs under faithfulness. Proceedings of the 6th European Workshop on Probabilistic Graphical Models pp. 299–306 (2012)
8. Sonntag, D., Peña, J.M., Gómez-Olmedo, M.: Approximate counting of graphical models via mcmc revisited. International Journal of Intelligent Systems **30**(3), 384–420 (2015)
9. Tsamardinos, I., , Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. Machine Learning **65**(1), 31–78 (Oct 2006)