Improving the Efficiency of the PC Algorithm by Using Model-Based Conditional Independence Tests

Erica Cai, Andrew McGregor, David Jensen
College of Information and Computer Sciences, University of Massachusetts Amherst

QR code for the paper!

Adding a pre-processing step to the PC algorithm can reduce the number of CI tests by up to 99%.

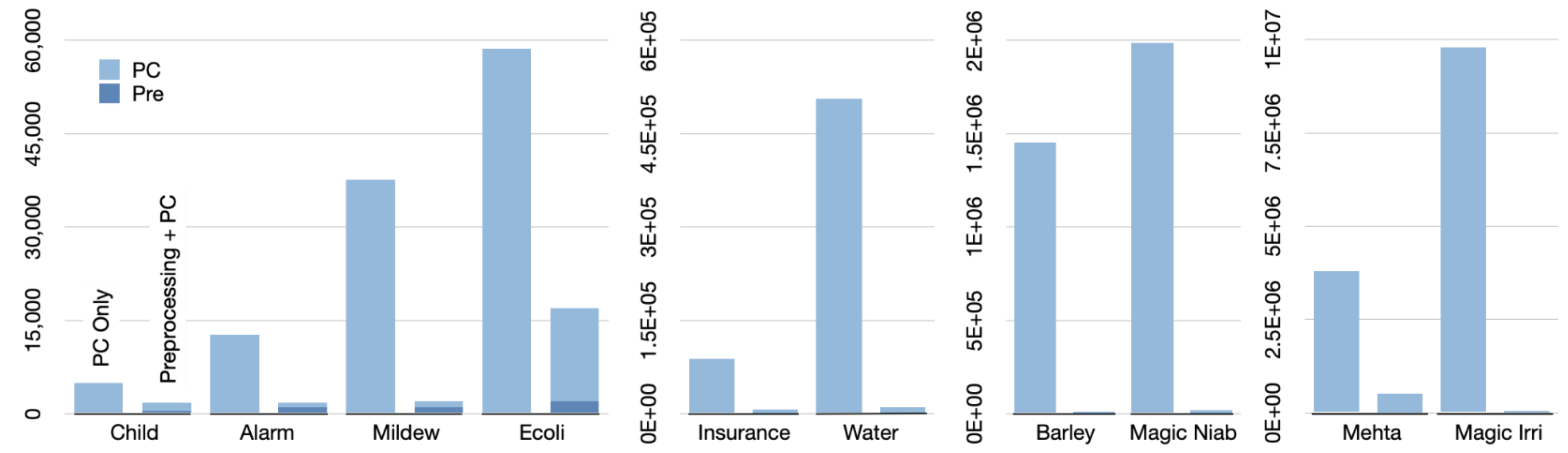

Figure 1. For DAGs that correspond to real-world systems in the *bnlearn* repository, the number of tests that P3PC performs and that PC alone performs to learn a structure.

What are constraint-based algorithms and PC?

Constraint-based algorithms infer a structure that corresponds to the conditional independence (CI) relationships within a population. These conditional independence relationships are inferred by performing CI tests on a data sample.

The PC algorithm [1] is a constraint-based algorithm with the following steps:
1. Initialize a complete undirected graph *G*.
2. For each pair of variables, perform CI tests in order of increasing size of conditioning sets *(0, 1, 2, …)* and stop only when a test infers that the pair of variables is CI (in this case, remove the edge corresponding to the pair in *G*) or after all feasible CI tests have been performed.
3. Use the information from those tests to orient the edge directions and return the final directed acyclic graph (DAG) *G*.

Worst-case runtime of the PC algorithm

The phase of PC and of other constraint-based algorithms that applies CI tests typically dominates their runtime. The number of CI tests applied in the worst case is equal to the cardinality of the power set of the possible conditioning variables.

Why use model-based CI tests?

Most constraint-based algorithms assume that CI tests perform more accurately on small conditioning sets. However, recent model-based CI tests [2,3,4] have the potential to work effectively on very large conditioning sets because they use well-regularized models. We present a pre-processing step for PC that uses large conditioning sets.

P3PC: Pre-processing Plus PC

```
P3PC($X, V$):        $X$ is a dataset and $V$ is a set of variables
1. Initialize $p$ as an $n \times n$ matrix with all 1 values
2. Initialize $L$ as a list of sets having length $c_1$
3. for $a, b$ in $V$:
  4. if $a \perp b$, set $p[a,b] = p[b,a] = 0$; continue
  5. for $i \in \{1, \dots, c_1\}$, set $L[i]$ as a random set of $n - c_2$ vars
     from $V \backslash \{a, b\}$
  6. for S $\in L$, if $a \perp b | S$, set $p[a,b] = p[b,a] = 0$; continue
7. Return PC($X$,$V$,$p$)    $p[i.j] = 0$ if vars $i,j$ could be made CI; 1 o/w
```

Analysis on DAGs w.r.t. to real-world phenomena

We performed experiments to estimate the raw number of CI tests performed by P3PC and by PC. The experiments use known DAG structure and standard d-separation rules to determine if variables could be made CI by a given conditioning set, rather than running CI tests on data generated by that DAG.

Theoretical and empirical analysis on Erdős-Renyi DAGs

We show that if a pair of variables is CI, then for large conditioning sets of size $n-4$, the probability that all trails between the pair will be blocked is high in three statements:

Statement 1 (Trails of length at least 7): A conditioning set of size $n-4$ will block every trail of length 7 or greater.

Statement 2 (Trails of length at most 6): In an Erdős-Renyi network, the expected number of trails of length $\le$ 6 between a pair of nodes is at most $p(1+(pn)+(pn)^2+\cdots+(pn)^6) \le 7n^6p^7$ assuming that $p \ge 1/n$. If $p = \Theta(1)/n$, the expectation tends to 0 as $n -> \infty$.

Statement 3 (Expected number of colliders): The expected number of nodes that are colliders for some trail is $n(1-p)^n - (1-p)^n + (2((1-p)^n - 1)/p + n - p^2 + 1$. For example, when $p = 1/n$, the expression corresponds to $0.104n$.

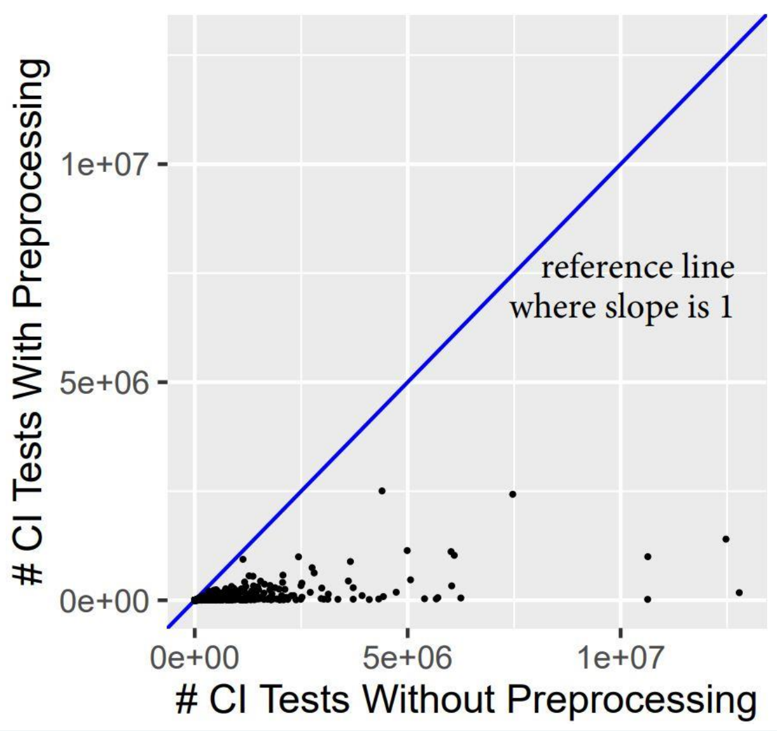

Figure 2. The number of tests performed by P3PC on the y-axis and the number of tests performed by the PC algorithm without pre-processing on the x-axis. Each point in the plot corresponds to a randomly generated ER DAG.

Takeaways about P3PC

1. We present P3PC, which performs model-based CI tests for variables pairs on large conditioning sets first.
2. We show that P3PC reduces the total number of CI tests performed by the PC algorithm in expectation for ER DAGs.
3. We perform experiments on DAGs that correspond to real-world systems, finding that P3PC performs between *0.5%* to *36%*, and often less than *10%*, of the number of tests performed by PC alone. We find similar surprising results on ER DAGs.
4. These results imply a new reason to prioritize research on accurate model-based tests of conditional independence, particularly those that perform well with large conditioning sets.

References

[1] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*, volume 81. 01 1993.

[2] Rajat Sen, Ananda Theertha Suresh, Karthikeyan Shanmugam, Alexandres G. Dimakis, and Sanjay Shakkettai. Model-Powered Conditional Independence Test. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 2955–2965, Red Hook, NY, USA, 2017. Curran Associates Inc

[3] Eric V. Strobl, Kun Zhang, and Shyam Visweswaran. Approximate Kernel-based Conditional Independence Tests for Fast Non-Parametric Causal Discovery. arXiv, 2017.

[4] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-Based Conditional Independence Test and Application in Causal Discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 804–813, Arlington, Virginia, USA, 2011. AUAI Press.

© Poster Template by Genigraphics® 1.800.790.4001 www.genigraphics.com