Thesis Draft

Daan Spijkers s1011382

Thesis Draft

Abstract

How much can we improve the accuracy of the resulting PAG from the BCCD algorithm using a greedy MAG search to optimise its probabilistic causal statements?

Introduction

Causal inference is taking a system of statistical independencies, and mining a system of causal relations. We then represent these causal relations in a causal graph.

In an ideal situation, we have a statistical test that determines whether x and y are independent with 100% accuracy. Given such a perfect test, complete algorithms exist; they give the total causal information possible from that system. Unfortunately, in the real world 100% accuracy is not possible, and we often have to make do with insufficient data.

That is why taking realistic data, and optimizing the result is a relevant problem. It is not always clear how an algorithm performs in these situations, even if it is complete. One such complete algorithm is BCCD, which uses a bayesian score to return a more robust and informative result than comparable procedures.

Preliminaries

Research

Problem details

The topic of my thesis will be an initial attempt to gain some insight into possible gains, by adding an additional step after BCCD. We will do this by defining a metric on its derived probabilistic causal statements, and running a MAG search to optimise it.

The specific research question is: how much can we improve the accuracy of the resulting PAG from the BCCD algorithm using a greedy MAG search to optimise its probabilistic causal statements?

For more insight we can vary the causal models we generate, and how much data we produce. Since BCCD is complete, if we feed it infinite data, we should see its result converge to the original causal model.

In our research, the main problem that we will need to solve is how exactly to define a metric on the causal statements. That is where the most involved effort will have to be.

Although other parts of the problem are simpler, the efficiency of a MAG search could be a limitation. Generating adjacent graphs is an expensive procedure, so we might have to restrict ourselves to smaller graphs.

Solution Details

Pseudocode Simple pseudocode for our process is as follows:

```
def run(pag):
  mag = pag_to_mag(pag)
  next_mag = next_mag(mag)

  while score(next_mag) > score(mag):
    mag = next_mag
    next_mag = next_mag(mag)

  return mag_to_pag(mag)

def next_mag(mag):
  return best_scoring_mag(adjacent_mags(mag))
```

Here we see that there are 4 main problems that we need to solve:

1. Transforming a PAG into a MAG.
2. Generating adjacent mags.
3. Scoring a MAG.
4. Transforming a MAG into a PAG

PAG to MAG The main difference is circle marks. The way to do this is by first orienting all semi-arcs into arcs, and then orienting all remaining edges into a DAG with no unshielded colliders. See Zhang paper.

MAG to PAG Turning a MAG back into a PAG is slightly more involved. We use d-separation and the FCI orientation rules to do so.

Generating adjacent MAGS This is the simplest problem to solve. We consider adjacent graphs to be graphs which have one edge changed compared to the original. Given two vertices u and v, there are four possibilities:

$$u \to v \tag{1}$$
$$u \leftarrow v \tag{2}$$
$$u \leftrightarrow v \tag{3}$$
$$(u, v) \notin E \tag{4}$$

Remembering that we ignored selection bias, and so there are no undirected edges. Our original graph has one of these four. All adjacent MAGS can then be easily generated by adding a graph which has one of the other 3 possibilities.

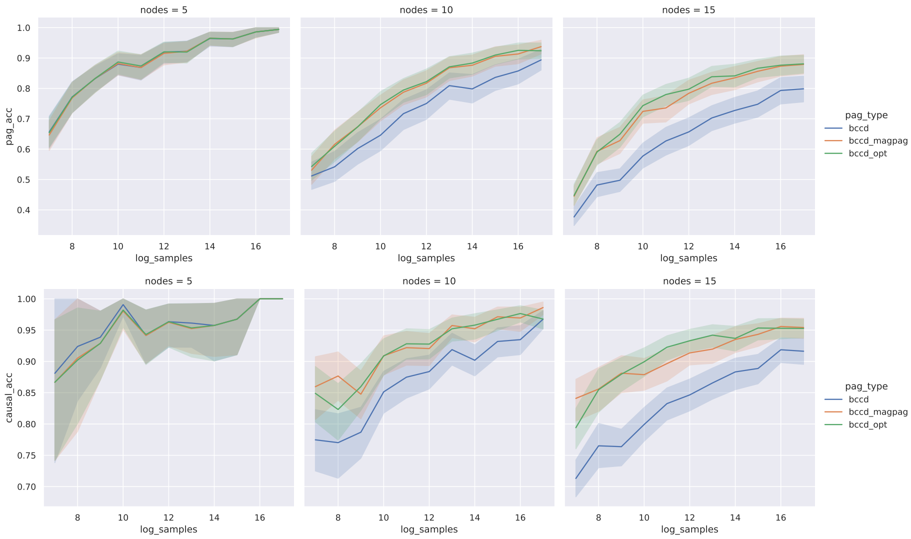
The real issue comes up when we want to check whether this MAG is also valid; that is whether it has any almost directed cycles.

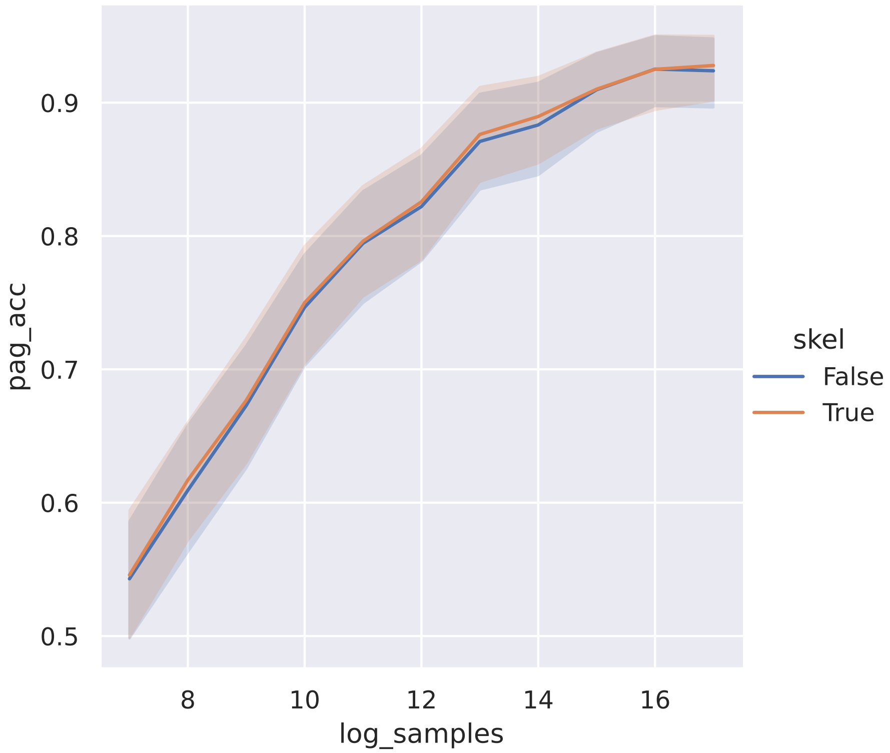
Scoring a MAG Scoring is not necessarily the most difficult part, but it is more unique to our problem, and did not have a readily available implementation. We have implemented 3 checks:

1. Ancestor(x, y)
2. Edge(x, y)
3. Cofounder(x, y)

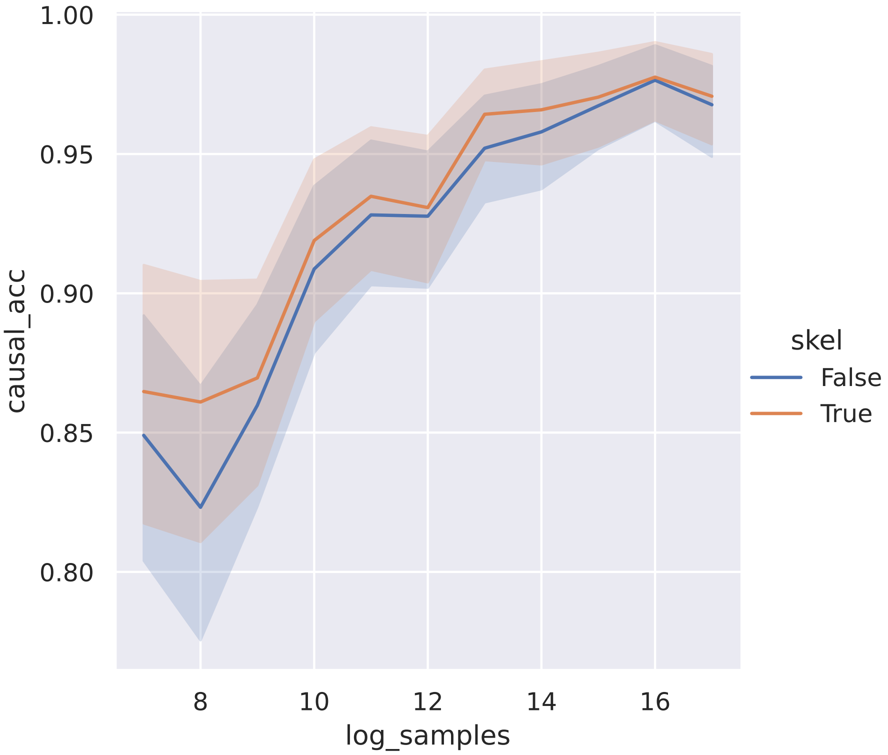
Benchmarks While the bccd portion, as well as the fci portion are both measured in seconds, my part is only measured in miliseconds.

Results





pag_acc
0.5
0.6
0.7
0.8
0.9
8
10
12
14
16
log_samples
skel
False
True



Related Work

Conclusions

References