Meeting Notes 7 23 2023

I. Dennis led us through his great notes on gCastle's generation of model training examples, an example of application using PC, and a major list of metrics that can be applied. Those notes and a paper outline of gCastle are attached.

gCastle also has a GUI, for which Dennis found a video here:

https://www.youtube.com/watch?v=5NOu2oApBgw

gCastle seems to instantiate all of the current frameworks for 'causal discovery': learning causal graphs from (non-temporal) observational data. If we take the 'ground truth' target as the world and the different causal discovery algorithms as ways of constructing scientific theories of the world, these are both (a) current machine learning techniques and (b) potential models of how science does or should optimally proceed in figuring out the world.

To review and supplement (from meeting notes last time), there are four basic approaches. The ones I understand are:

1. Constraint-based, including or based on Spirtes and Glymour PC. These start from a fully connected network, progressively removing links that the data shows are independent: first links between independent variables x and y, then links between variables and pairs x and {y, z}, etc.

2. Score-based GES (Chickering 2003). This starts from a blank graph—just nodes—then adds links which improve its score, continuing until no added link will further improve the score. Then, in a second round, it iteratively removes links as they improve the score.

The ones I don't yet understand:

3. Functional model LiNGAM (Shimizu et al. 2006). 'Where variables take non-Gaussian (non-normal distribution) values, the direction in which the estimated noise is independent from the hypothetical cause is the causal direction' (paraphrased from Glymour, Zhang and Spirtes 2019)

And a particularly intriguing family (though as I read further I'm not sure it is what I thought it was):

4. Gradient-based NOTEARS and successors (Zheng et al. 2018). '…the first algorithm to frame structure learning as a purely continuous optimization problem.' (Molak 2022)

Dennis led us through is great notes on gCastle's generation of model training examples, an example of application using PC, and all of the metrics that can be applied to judge 'scores' are attached.

II. One batch of questions is how good each of these is at capturing a ground truth, where (a) different kinds of ground truth may be relevant—random vs. scale-free graphs, for example and (b) success on different metrics may be relevant.

Another batch of questions is what the pattern of success is over time. Do some of these give us a smooth approximation of 'better'? Do some give a punctuated equilibrium with small improvements interspersed with major leaps?
The second of these may be relevant to taking these as (descriptive) models of the mechanisms of scientific discovery. The first may be relevant to taking these as (normative) models of how science might most optimally proceed in discovering the world.

III. With those in mind, here are some tasks for the next two weeks:

Patrick is going to try to figure out more about the models he doesn't understand. But Dennis may be of even more help in trying to read off what these are doing directly from the code.

Amber is going to get a handle on gCastle from Dennis's lead.

Dennis is going to see about implementing the other models in gCastle, but also is going to see about implementing a step-by-step image of how well these do, in line with the thought sin section II above.

With an eye to the descriptive task of modeling how science proceeds, Sophia is going to try to put together a map of where current (since Kuhn, etc.) discussion stands as to whether we should expect to see a pattern of discovery 'jumps' and the like.

Zhongming is going to take on a very specific background question. As outlined in Wikipedia on 'acyclic orientation,' any undirected graph can be made into an acyclic directed graph. The technique outlined there is:

(a) start with your undirected graph
(b) number the nodes in a sequence (any sequence)
(c) for each undirected link between nodes x and y:
    If y is higher in the sequence, draw a directed arrow from x to y.

The question: Is this sequence-technique (with different sequences) guaranteed to capture *all* the directed acyclic graphs with links matching the original undirected graph, or only some?

A plausible approach: look for a counter-example: a directed acyclic graph you couldn't get from its 'undirected' form in this way. That would answer the question one way. In looking for a counter-example, you may also figure out why there couldn't be one. That would answer the question the other way

IV. Patrick can't meet next week. We'll aim for Saturday, August 5^{th}, though it will probably have to be in the afternoon.