Algorithmic causal discovery faces challenges

User provides a

Latency

However, operators often only have access to textual logs.

Have non

Users can define

Three key challenges:

Set special variables

Start with

Ignoring bias makes

Failures are a daily phenomenon when operating large complex systems.

Generate synthetic logs for each of 1000 "machines" with a

We pick the aggregate that

Sawmill suggests

Leverage GPT

[1]

Log information can be too granular for meaningful analysis.

Have faulty users artificially be on a

The best aggregate to pick for each variable can be unclear a priori.

Causal reasoning [1] has helped scientists across domains pose, discuss and test hypotheses.

User evaluates them and revises graph.

The process repeats while increasing the

Vary performance

Challenge B

With a Handful of User Interactions, Sawmill Uncovers Highly Accurate Effects

Sawmill Provides Solutions for Each Challenge

Challenge B: Aggregate Data to Maximize Entropy

• Log information can be too granular for meaningful analysis.

• Users can define causal units over which to aggregate log information, depending on the context (e.g. machines, users etc.).

• The best aggregate to pick for each variable can be unclear a priori.

• We pick the aggregate that maximizes empirical entropy between the causal units, in order to maximize downstream usefulness.

Challenge A: Utilize LLMs for Tagging

• Use Drain [4] to create the unlabeled parsed table.

• Leverage GPT-4 [5] to assign human-understandable tags to each variable.

Challenge C: Leverage Exploration-based Causal Discovery

• Algorithmic causal discovery faces challenges due to dependencies.

• Hand-crafting a full causal model is also daunting, but only part of it is needed.

• Use a human in the loop:

  • User provides a variable of interest.

  • Sawmill suggests candidate causes.

  • User evaluates them and revises graph.

  • The process repeats while increasing the exploration score.

With a Handful of User Interactions, Sawmill Uncovers Highly Accurate Effects

Battling Confounding in Real-World Logs

• Collect logs from PostgreSQL running TPC-DS.

• Vary performance-affecting parameters and bias parameter combinations to trade off work_mem and max_parallel_workers.

• Ignoring bias makes mean latency increase for more parallelism.

Discerning Subtle Semi-Synthetic Effects

• Start with real logs from a mobile application and generate similar logs for 1000 users. Label a varying fraction of them (1% to 50%) as faulty.

• Have faulty users artificially be on a different OS version and have them fail HTTP requests at varying rates (20% to 100% of the time).

• Have non-faulty users fail HTTP requests 10% of the time.

Overcoming Noisiness in Synthetic Logs

• Generate synthetic logs for each of 1000 "machines" with a varying number of variables (V in 10-1000).

• Have most of the variables take a random value between 0-100.

• Set special variables x,y,z such that z confounds the effect of x on y.

• Add Gaussian noise to x and y, with a varying standard deviation (1-10).