Supervised Ensemble-based Causal DAG Selection

**Abstract**

Xyz, Targeted?

**Keywords**

Causal Discovery; Model selection; Multi-label Classification; D-separation based similarity;

## 1. RESEARCH QUESTIONS

Given a set of n Causal Discovery algorithms we aim at leveraging their complementarity to select the single reconstructed DAG which is closest (in a causality related sense) to the ground truth DAG. The closeness between two DAGs is measured in terms of causally relevant similarities, e.g. in terms of the similarities of the d-connection structures.

## 2. SCHEMA of the DESIGN SPACE for the experimentation (in bold the default choices)

### 2.1 Structural Causal model (consists of a DAG, probabilities and functional dependencies)

DAG definition

* Type of network generation dynamics
  + **Erdős-Rényi model** (the G(n,p) variant), see my code
  + Barabasi-Albert Model, preferential attachment
* Number of nodes **5,10,15,20**, 25, 30, 40
* Number of connected pairs: 3 densities, low, medium, high CM[2]

Probability distribution for independent (parentless) variables

* **Gauss**, Cauchy, others in gCastle?
* Heterogeneous distributions of variables in the same DAG? CM[1]

Functional dependency

* parent-child
  + **Linear**
  + **Quadratic**
  + Exponential
  + Logarithmic
  + other in gCastle?
* Operation at collider (child of two or more parents)
  + Addition
  + Multiplication
  + what is gCastle doing?

Which number N of combinations to use overall? Which proportion of linear/gaussian?

### 2.2 Dataset generation

One dataset corresponds to a number of instances generated from a Structural Causal Model

Number of instances (number of rows) per dataset: 100, 500, 1000, **2000**, 5000

### 2.3 Causal Discovery Algorithms

* **PC, GOLEM, LINGAM, NoTears, GES**, all the other in gCastle
* prepare for use of subsets, in a “feature selection” style

### 2.4 Comparison with Ground Truth and between DAGs

Similarity metrics between graphs

* **Structural Hamming Distance** (Hamming distance between adjacency matrices)
* **D-connection similarity** (Hamming distance between d-connection matrices)
* Structural Intervention Distance (based on errors in predicting interventional distributions,   
  see Peters and Buhlmann 2015)
* Other causal distances (see A ladder of Causal Distances Maxime Peyrard and Robert West, 2021)

(TODO check correlation between D-connection similarity and other distances, to justify use of the former)

## 3. The dataset

Suppose we are using n=5 algorithms.

We start from a number N of rows, each row contains:

* the set of 10 pairwise distances among the DAGs produced by the 5 algorithms
* (we can add the 5 sum of distances of each DAG from the others, it was shown to be effective)
* the set of 5 distances from each DAG and the Ground-Truth DAG (GT-DAG)

At the beginning the row should contain all the elements defining the structural causal model (number of nodes, number of links, distributions, functional dependencies); those elements will not be visible in the training phase to the algorithms, except for the number of nodes, i.e. of the variables which has to be visible.

## 4. Learning phase

We want to train a special classifier that predicts which DAG/Algorithm is closest to the GT.

There can be several at the same distance, so we use multi-label classification

### 4.1 Multi-label classification

We use two methods

* Binary relevance (we train binary detectors for each possible “label” (the algorithm which has generated the DAGs closest to the GT))
* Classifier Chain (we train binary detectors incrementally, adding the outcome column of one detector to the features, as added feature)
* possible other Multi-label classification methods

### 4.2 Selection of model when multiple labels are predicted

When multiple labels are predicted (or no labels are predicted) one has to make a choice

* Use fixed criteria, prioritizing one label/algorithm over the other
* Learn from data a criterion from original features plus added features

NOTE

L’ultima parte può essere perfezionata riformulando il problema in modo da tener conto delle relazioni.

Ad esempio si consideri il caso degenere in cui due DAG (tra i 5 candidati) candidati hanno distanza zero tra di loro: di fatto (se la distanza 0 implica che sono uguali topologicamente) siamo in una condizione di scelta tra 4 DAG candidati (non tra 5 DAG candidati), e se (come è molto probabile) le 4 distanze dalla GT sono diverse la risposta su quale sia il DAG più vicino è una sola, quindi il multilabel è fuori luogo. Bisognerebbe separare in categorie diverse i casi degeneri secondo il tipo di coincidenza (queste distanze zero tra DAG candidati si vedono anche senza conoscere la GT) e addestrare un classificatore mono-etichetta per ciascuna condizione. Questo comporta un break-down del dataset di training in un numero di sotto tabelle e il training di altrettanti modelli di classificazione mono-label.

@Corrado: mi piacerebbe vedere il dataset di training prima di effettuare il training Multilabel

Se facciamo questo studio con il break-down e poi lo confrontiamo con il Multi-label e vediamo poca differenza possiamo affermare che il multilabel è la cosa più semplice anche se dal punto di vista logico rappresenta un’approssimazione.

NOTA 2

Dopo questa analisi si può pensare ad una vera fusione dei migliori per avvicinarsi alla GT con un NUOVO DAG.

NOTA CM[1]: C’e un trucco per farlo: per lo stesso DAG creiamo DIVERSI dataset con lo stesso numero di righe MA con diversi tipi di di dipendenze. A questo punto generiamo un nuovo dataset prendendo ogni colonna a caso da uno dei dataset generati precedentemente. Oppure possiamo farlo ‘a coppie’ (ogni coppia rappresenta un edge). Non e’ il massimo ma e’ semplice da implementare.

CM[2]: cosa intendi per *densities*: **low**, **medium**, **high**?   
Dato N numero di vertici, in un DAG il numero massimo di archi e’ M=N\*(N-1)/2. Servono 3 numeretti nel range [0,1]