TUBAF
The University of Resources.
Since 1765.

Methods for Structure Identifiability

Seminar on causality and causal inference

Hanyue Gu

Institute of Stochastics, TU Bergakademie Freiberg

June 26, 2024

Content

Review

Markov property and faithfulness

The joint distribution $P_{\mathbf{X}}$ is said to be **Markov with respect to the DAG** $\mathcal{G}$ if

$$\mathbf{A}, \mathbf{B} \text{ d-sep. } by \ \mathbf{C} \quad \Rightarrow \quad \mathbf{A} \perp\!\!\!\perp \mathbf{B} | \mathbf{C}$$

for all disjoint set $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

The joint distribution $P_{\mathbf{X}}$ is said to be **faithful to the DAG** $\mathcal{G}$ if

$$\mathbf{A}, \mathbf{B} \text{ d-sep. } by \ \mathbf{C} \quad \Leftarrow \quad \mathbf{A} \perp\!\!\!\perp \mathbf{B} | \mathbf{C}$$

for all disjoint set $\mathbf{A}, \mathbf{B}, \mathbf{C}$.

Under the Markov condition and faithfulness, the Markov equivalence class of $\mathcal{G}$ is identifiable from $P_{\mathbf{X}}$.

Independence-Based methods

Idea

- Estimate the skeleton, that is, the undirected edges
- Orient as many edges as possible

Estimation of skeleton

Lemma [4]

The following two statements holds.

(i) Two nodes X, Y in a DAG $(\mathbf{X}, \mathcal{E})$ are adjacent if and only if they cannot be d-separated by any subsets $S \subseteq \mathbf{X} \backslash \{X, Y\}$.

(ii) If two nodes X, Y in a DAG $(\mathbf{X}, \mathcal{E})$ are not adjacent, then they are d-separated by either $\mathbf{PA}_X$ or $\mathbf{PA}_Y$.

Lemma (i): IC algorithm, SGS algorithm

Lemma (ii): PC algorithm

IC/SGS algorithm: Idea

For each pair of nodes (X, Y), these methods search through all possible subsets $\mathbf{A} \subseteq \mathbf{X} \backslash \{X, Y\}$ of variables neither containing X nor Y and check whether X and Y are d-separated given $\mathbf{A}$. After all those tests, X and Y are adjacent if and only if no set $\mathbf{A}$ was found that d-separates X and Y.

PC algorithm: Idea

The PC algorithm starts with a fully connected undirected graph and step-by-step increases the size of the conditioning set $\mathbf{A}$, starting with $\#\mathbf{A} = 0$. At iteration k, it considers sets $\mathbf{A}$ of size $\#\mathbf{A} = k$, using the following neat trick: to test whether X and Y can be d-separated, one only has to go through sets $\mathbf{A}$ that are subsets either of the neighbors of X or of the neighbors of Y.

PC algorithm(Example) [1]

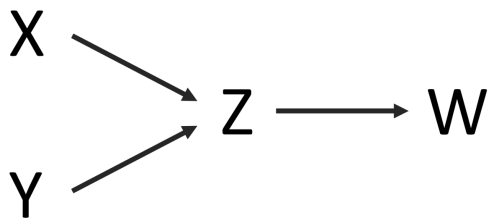

Figure: Original true causal graph.

PC algorithm: Estimation of skeleton

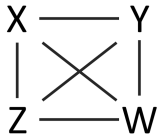

Figure: After step 1.

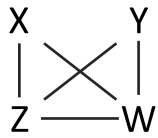

Figure: After step 2.

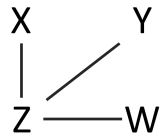

Figure: After step 3.

1. Form a complete undirected graph.
2. Eliminate edges between variables that are unconditionally independent.
3. For each pair of variables (A, B) having an edge between them, and for each variable C with an edge connected to either of them, eliminate the edge between A and B if $A \perp\!\!\!\perp B | C$.

PC algorithm: Estimation of skeleton

4. For each pair of variables (A, B) having an edge between them, and for each pair of variables C, D with edges both connected to A or both connected to B, eliminate the edge between A and B if $A \perp\!\!\!\perp B | \{C, D\}$.

Continue checking independencies conditional on subsets of variables of increasing size until there are no more adjacent pairs (A, B), such that there is a subset of variables such that all of the variables in the subset are adjacent to A or all adjacent to B.

PC algorithm: Orientation of edges

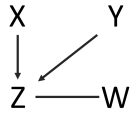

Figure: After step 5.

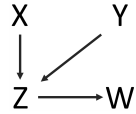

Figure: After step 6.

5. For each triple of variables (A, B, C) such that A and B are adjacent, B and C are adjacent, and A and C are not adjacent, orient the edges $A - B - C$ as $A \rightarrow B \leftarrow C$, if B was not in the set conditioning on which A and C became independent.
6. For each triple of variables such that $A \rightarrow B - C$, and A and C are not adjacent, orient the edge $B - C$ as $B \rightarrow C$. This is called orientation propagation.

PC algorithm: Notes

- There are other orientation propagation rules that are not illustrated here, such as Meek's orientation rules [2].
- In some examples, none of orientation rules will apply to a given undirected edge, and that edge will remain undirected in the output.

Conditional independence tests

- Statistical significance test
- Kernel-based test
- Gaussian distributed variables: vanishing partial correlation
- Non-Gaussian distributed variables: nonlinear extension of partial correlation

Nonlinear extension of partial correlation [4]

1. (Nonlinearly) regress X on Z and test whether the residuals are independent of Y
2. (Nonlinearly) regress Y on Z and test whether the residuals are independent of X
3. If one of those two independences hold, conclude that $X \perp\!\!\!\perp Y | Z$

Score-Based methods

Best scoring graph

Given data $\mathcal{D} = (\mathbf{X^1}, \cdots, \mathbf{X^n})$ from a vector $\mathbf{X}$ of variables, that is, a sample containing n i.i.d. observations, the idea is to assign a score $S(\mathcal{D}, \mathcal{G})$ to each graph $\mathcal{G}$ and search over the space of DAGs to find the graph with the highest score:

$$\hat{\mathcal{G}} := \underset{\mathcal{G} \text{ DAG over } \mathbf{X}}{\operatorname{argmax}} \; S(\mathcal{D}, \mathcal{G})$$

There are several possibilities to define such a scoring function S. Often a parametric model is assumed (e.g., linear Gaussian equations or multinomial distributions), which introduces a set of parameters $\theta \in \Theta$ [4].

(Penalized) likelihood

For each graph we may consider the maximum likelihood estimator $\hat{\theta}$ for θ and then define a score function by the *BIC*

$$S(\mathcal{D}, \mathcal{G}) := \log p(\mathcal{D}|\hat{\theta}, \mathcal{G}) - \frac{\#parameters}{2} \log n$$

where $\log p(\mathcal{D}|\hat{\theta}, \mathcal{G})$ is the log likelihood and n is the sample size [4].

Bayesian scoring functions

We define priors $p_{pr}(\mathcal{G})$ and $p_{pr}(\theta)$ over DAGs and parameters, respectively, and consider the log posterior as a score function (note that $p(\mathcal{D})$ is constant over all DAGs):

$$S(\mathcal{D}, \mathcal{G}) := \log p(\mathcal{G}|\mathcal{D}) \propto \log p_{pr}(\mathcal{G}) + \log p(\mathcal{D}|\mathcal{G}),$$

where $p(\mathcal{D}|\mathcal{G})$ is the marginal likelihood

$$p(\mathcal{D}|\mathcal{G}) = \int_{\theta \in \Theta} p(\mathcal{D}|\mathcal{G}, \theta) p_{pr}(\theta, \mathcal{G}) d\theta.$$

Here the resulting estimator $\hat{\mathcal{G}}$ is usually called a maximum a posteriori (MAP) estimator [4].

Greedy search techniques

At each step there is a candidate graph and a set of neighboring graphs. For all these neighbors, one computes the score and considers the best-scoring graph as the new candidate. If none of the neighbors obtains a better score, the search procedure terminates (not knowing whether one obtained only a local optimum).

A neighborhood relation: Starting from a graph $\mathcal{G}$, we may define all graphs as neighbors from $\mathcal{G}$ that can be obtained by removing, adding, or reversing one edge [4].

Exact methods

Here, “exact” means that they aim at finding (one of) the best scoring graphs for a given finite data set.

Due to the Markov factorization, we have for $\mathcal{D} = (\mathbf{X}^1, \cdots, \mathbf{X}^n)$ that

$$\log p(\mathcal{D}|\hat{\theta}, \mathcal{G}) = \sum_{j=1}^{d} \sum_{i=1}^{n} \log p(X_j^i | X^i_{\mathbf{PA}_j^{\mathcal{G}}}, \hat{\theta}),$$

which is a sum of d ’local’ scores.

Other techniques:

- ILP framework: represent graphical structures as vectors
- Restrict the number of parents [4]

Additive Noise Models

Score-based method combined with greedy search

Nonlinear Gaussian case:

For a given graph structure $\mathcal{G}$, we regress each variable on its parents and obtain the score

$$\log p(\mathcal{D}|\mathcal{G}) = \sum_{j=1}^{d} -\log \widehat{var}[R_j],$$

here, $\widehat{var}[R_j]$ is the empirical variance of the residuals R_j obtained from the regression of variable X_j on its parents [4].

If the noise cannot be assumed to have a Gaussian distribution [3]

For each DAG $\mathcal{G}_i$ we follow the three-step procedure:

1. For each node k estimate the residuals $\hat{\epsilon}_k$ by nonparametrically regressing X_k on $\{X_l\}_l \in \mathrm{pa}_{\mathcal{G}_i(k)}$. If $\mathrm{pa}_{\mathcal{G}_i(k)} = \emptyset$, set $\hat{\epsilon}_k = x_k$.
2. For each node k estimate the residual densities $\hat{p}_{\epsilon k}$ from the estimated residuals $\hat{\epsilon}_k$.
3. Compute the penalized likelihood score

$$S_i^n = \frac{1}{n} \sum_{j=1}^{n} \sum_{k=1}^{d} \log \hat{p}_{\epsilon k}(\hat{\epsilon}_k^j) - \#(edges)_i \cdot a_n,$$

where a_n controls the strength of the penalty.

Independence tests

Algorithm 1 Regression with subsequent independence test (RESIT)

Input: I.i.d. samples of a p-dimensional distribution on $(X_1, \ldots, X_p)$
$S := \{1, \ldots, p\}, \pi := [\,]$
PHASE 1: Determine causal order.
repeat
  for $k \in S$ **do**
    Regress X_k on $\{X_i\}_{i \in S \setminus \{k\}}$.
    Measure dependence between residuals and $\{X_i\}_{i \in S \setminus \{k\}}$.
  end for
  Let k^* be the k with the weakest dependence.
  $S := S \setminus \{k^*\}$
  $\mathrm{pa}(k^*) := S$
  $\pi := [k^*, \pi]$ (π will be the causal order, its last component being a sink)
until $\#S = 1$

Figure: PHASE 1

14: PHASE 2: Remove superfluous edges.
15: **for** $k \in \{2, \dots, p\}$ **do**
16: **for** $\ell \in \mathrm{pa}(\pi(k))$ **do**
17: Regress $X_{\pi(k)}$ on $\{X_i\}_{i \in \mathrm{pa}(\pi(k)) \setminus \{\ell\}}$.
18: **if** residuals are independent of $\{X_i\}_{i \in \{\pi(1), \dots, \pi(k-1)\}}$ **then**
19: $\mathrm{pa}(\pi(k)) := \mathrm{pa}(\pi(k)) \setminus \{\ell\}$
20: **end if**
21: **end for**
22: **end for**
23: **Output:** $(\mathrm{pa}(1), \dots, \mathrm{pa}(p))$

Figure: PHASE 2 [5]

Note: This method is based on the fact that for each node X_i the corresponding noise variable N_i is independent of all non-descendants of X_i.

Intervention

Intervention

- Known intervention targets
- Unknown intervention targets

Unknown intervention targets

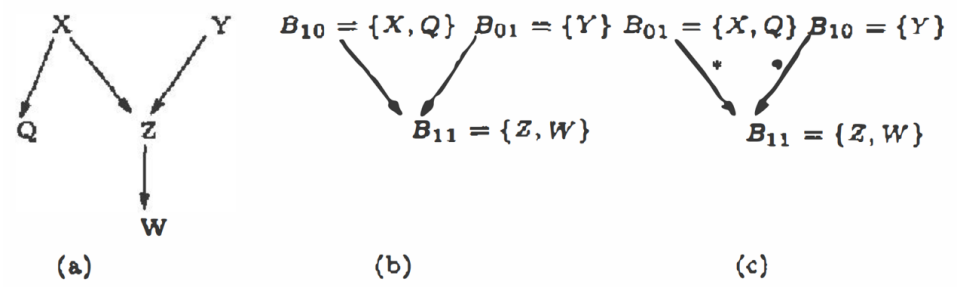

Figure: (a) A causal diagram; (b) The order graph without knowing the intervention target; (c) The marked order graph.[6]

References

References

[1] Clark Glymour, Kun Zhang, and Peter Spirtes. “Review of causal discovery methods based on graphical models”. In: *Frontiers in genetics* 10 (2019), p. 524.

[2] Chris Meek. *Complete orientation rules for patterns*. Carnegie Mellon [Department of Philosophy], 1995.

[3] Christopher Nowzohour and Peter Bühlmann. “Score-based causal learning in additive noise models”. In: *Statistics* 50.3 (2016), pp. 471–485.

[4] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[5] Jonas Peters et al. “Causal discovery with continuous additive noise models”. In: (2014).

[6] Jin Tian and Judea Pearl. “Causal discovery from changes”. In: *arXiv preprint arXiv:1301.2312* (2013).