Structure Learning of DAGs

Qing Zhou

UCLA Department of Statistics

Stats 212 Graphical Models
Lecture Notes

Outline

1. Overview and assumptions
2. Equivalence class and CPDAG
3. Constraint-based learning
4. Score-based learning
5. Continuous relaxation of score
6. Learning with experimental data

Overview and assumptions

Structure learning: Let $(\mathcal{G}, \mathbb{P})$ be a causal DAG model over $X_1, \ldots, X_p$. Given data $x_i = (x_{i1}, \ldots, x_{ip}) \sim (\mathcal{G}, \mathbb{P})$, $i = 1, \ldots, n$, how to estimate the DAG $\mathcal{G}$?

- Constraint-based methods: Conditional independence tests against $X_i \perp X_j \mid X_S$ for all i, j, S.
- Score-based methods: Optimizing a scoring function over graph space.
- Hybrid methods: First use constraint-based method to prune the search space, and then apply a score-based method to search for the optimal DAG.

See, e.g. Aragam et al. (2019) Section 1 for recent literature.

Data types:

- Observational data (no intervention)
- Experimental data (intervention available)

Overview and assumptions

Main assumptions: (1) causal sufficiency; (2) faithfulness.

Definition 1 (Causal sufficiency)

A set of variables V is causally sufficient if every common cause of any two or more variables in V is also in V.

- For $\mathcal{G}$, this means that every common ancestor of two or more nodes is observed.
- In SEM $X_i = f_i(PA_i, \varepsilon_i)$, $i \in V$, causal sufficiency implies ε_i's are mutually independent.

Overview and assumptions

Definition 2 (Faithfulness)

For a graphical model $(\mathcal{G}, \mathbb{P})$, we say the distribution $\mathbb{P}$ is faithful to the graph $\mathcal{G}$ if for every triple of disjoint sets $A, B, S \subset V$,

$$X_A \perp X_B \mid X_S \Leftrightarrow S \text{ separates } (d\text{-separates}) \ A \text{ and } B.$$

- Conditional independence (CI) in $\mathbb{P} \Leftrightarrow$ d-separation in $\mathcal{G}$, i.e.

$$\mathcal{I}_{\mathbb{P}}(A, B|S) \Leftrightarrow \mathcal{D}_{\mathcal{G}}(A, B|S).$$

- Given $\mathcal{G}$, almost all parameter values in the SEMs will define a faithful $\mathbb{P}$.
- Structure learning: use CI relations learned from data to infer edges in $\mathcal{G}$.

Equivalence class and CPDAG

Suppose we only have observational data. What can be learned?

Definition 3 (Markov equivalence)

Two DAGs $\mathcal{G}$ and $\mathcal{G}'$ on the same set of nodes V are Markov equivalent if $\mathcal{D}_{\mathcal{G}}(X, Y|\boldsymbol{Z}) \Leftrightarrow \mathcal{D}_{\mathcal{G}'}(X, Y|\boldsymbol{Z})$ for any $X, Y \in V$ and $\mathbf{Z} \subseteq V \setminus \{X, Y\}$.

- Two DAGs are Markov equivalent if and only if they have the same skeletons and the same v-structures.
- A v-structure is a triplet $\{i, j, k\} \subseteq V$ of the form $i \rightarrow k \leftarrow j$: i and j are nonadjacent; k is called an *uncovered collider*.
- Equivalent DAGs form an equivalence class.
- DAGs in the same equivalence class cannot be distinguished from observational data. Thus we can only learn the equivalence class of $\mathcal{G}$ from observational data.

Equivalence class and CPDAG

How to represent an equivalence class? CPDAG (Completed partially DAG).

Two types of edges in a DAG $\mathcal{G}$:

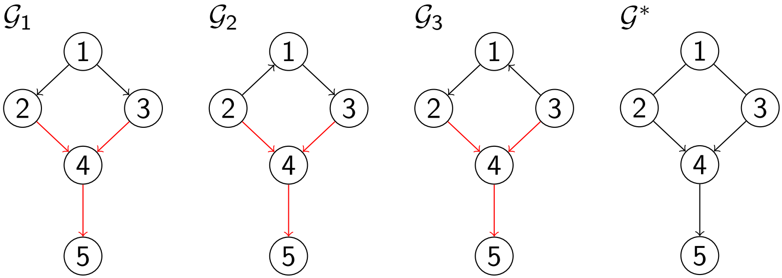
- A directed edge $i \to j$ is *compelled* in $\mathcal{G}$ if for every DAG $\mathcal{G}'$ equivalent to $\mathcal{G}$, the edge $i \to j$ exists in $\mathcal{G}'$.
- If an edge is not compelled in $\mathcal{G}$, then it is *reversible*.

Definition 4 (CPDAG or essential graph)

The CPDAG of an equivalence class is the PDAG consisting of a directed edge for every compelled edge in the equivalence class, and an undirected edge for every reversible edge in the equivalence class.

Equivalence class and CPDAG

Equivalence class $[\mathcal{G}_1] = \{\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3\}$ and CPDAG $\mathcal{G}$:



Red: compelled edges, same orientation in all equivalent DAGs.
Black: reversible edges, either direction occurs in at least one equivalent DAG.

Equivalence class and CPDAG

Characterization of CPDAGs (or essential graphs):

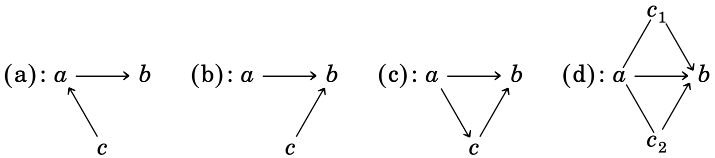
Theorem 1 (Andersson et al. (1997))

A graph $\mathcal{G}$ is a CPDAG for some DAG if and only if $\mathcal{G}$ satisfies the following conditions:

1. $\mathcal{G}$ is a chain graph.
2. $\mathcal{G}_\tau$ is chordal for every chain component τ of $\mathcal{G}$.
3. The configuration $a \to b - c$ does not occur as an induced subgraph of $\mathcal{G}$.
4. Every arrow $a \to b$ in $\mathcal{G}$ is strongly protected.

Equivalence class and CPDAG

- Chordal graph: An undirected graph is chordal if every cycle of length $n \geq 4$ possesses a chord, that is an edge between two nonconsecutive vertices on the cycle. (Triangulated graph)
- An arrow $a \to b$ is strongly protected in $\mathcal{G}$ if it occurs in at least one of the following configurations as an induced subgraph:



Constraint-based learning

Theorem 2 (Spirtes et al. (1993))

Suppose $(\mathcal{G}, \mathbb{P})$ satisfies the faithfulness assumption. Then there is no edge between a pair of nodes $X, Y \in V$ if and only if there exists a subset $\boldsymbol{Z} \subseteq V \setminus \{X, Y\}$ such that $\mathcal{I}_P(X, Y|\boldsymbol{Z})$.

Constraint-based methods:

1. Find the skeleton of $\mathcal{G}$ by CI tests;
2. Identify v-structures;
3. Orient other edges.

Output: CPDAG (or PDAG)

Constraint-based learning

Outline of PC algorithm (Spirtes and Glymour 1991):

1: $E \leftarrow$ edge set of the complete undirected graph on V.
2: **for** $(i, j) \in E$ **do**
3: Search for a subset S_{ij} of either $N_i(E)$ or $N_j(E)$ such that $X_i \perp X_j \mid S_{ij}$. If found, $E \leftarrow E \setminus \{(i,j),(j,i)\}$ and store S_{ij}.
4: **end for**
5: Identify v-structures based on E and $\{S_{ij}\}$.
6: Orient as many edges in E as possible by Meek's rules.

Notes:

1. Line 3: $N_i(E) = \{X_k : (i,k) \in E\}$.
2. For loop: implemented in ascending order of $|S_{ij}| = \ell$ for $\ell = 0, \ldots, \ell_{\max}$.
3. Line 1 to 4: Estimate skeleton $sk(\widehat{\mathcal{G}})$ of $\mathcal{G}$.

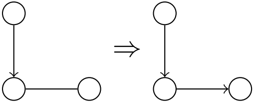
Constraint-based learning

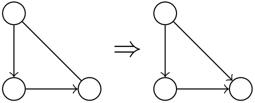
Edge orientation steps:

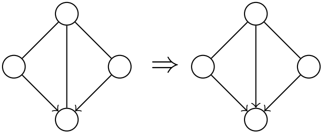
1. Identify v-structures (Line 5) given $sk(\widehat{\mathcal{G}})$:
   For all nonadjacent pair (i, j) with a common neighbor k, orient $i - k - j$ as $i \to k \leftarrow j$ if $k \notin S_{ij}$.
   Because otherwise, $X_i \not\perp X_j \mid S_{ij}$, contradiction. After this step, we obtain a PDAG.
2. Meek's rules (Line 6): In the resulting PDAG, orient as many undirected edges as possible by repeated application of four rules (Meek 1995).
   Basic idea: If orienting an undirected edge $i - j$ into $i \to j$ would result in additional v-structures or a directed cycle, then orient it into $i \leftarrow j$.

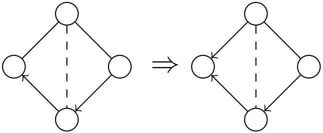
Constraint-based learning

Meek's rules:

R1: 

R2: 

R3: 

R4: 

dashed line in R4: undirected or directed with either orientation

Constraint-based learning

Conditional independence tests ($H_0 : X \perp Y \mid S$):

- Gaussian data: partial correlation $\text{cor}(X, Y \mid S) = 0$.
  1. Sample covariance matrix $\widehat{\Sigma}$ from data columns of (X, Y, S).
  2. $\widehat{\Omega} = (\omega_{ij}) \leftarrow \widehat{\Sigma}^{-1}$ and $\widehat{\rho}_{XY|S} = -\omega_{12}/\sqrt{\omega_{11}\omega_{22}}$.
  3. Fisher z-transformation,

$$z(X, Y|S) = \frac{1}{2} \log \left(\frac{1 + \widehat{\rho}_{XY|S}}{1 - \widehat{\rho}_{XY|S}} \right)$$

  and $\sqrt{n - |S| - 3} \cdot z(X, Y|S) \mid H_0 \sim \mathcal{N}(0, 1)$.
- Discrete data: G^2 or χ^2 test for conditional independence.

$$G^2(X, Y; S = s) = 2 \sum_{x,y} O_{xys} \log(O_{xys}/E_{xys}),$$

$$G^2(X, Y; S) = \sum_{s} G^2(X, Y; S = s) \mid H_0 \sim \chi^2_{(|X|-1)(|Y|-1)|S|},$$

E_{xys}: expected counts under H_0; O_{xys}: observed counts.

Constraint-based learning

Correctness and consistency:

Let $\widehat{\mathcal{G}}_n$ be the estimated graph by PC from a sample of size n and $\mathcal{C}$ be the CPDAG of $\mathcal{G}$. Suppose that $\mathbb{P}$ is faithful to $\mathcal{G}$.

1. CI oracles (Spirtes et al. 1993; Meek 1995): If all CI tests are perfect (CI oracles), then $\widehat{\mathcal{G}}_n = \mathcal{C}$ and all found separating sets $|S_{ij}| \leq \max\{|PA_i|, |PA_j|\}$.
2. Large-sample limit: When the sample size $n \to \infty$, all CI tests involved will be perfect (no type I or II error) with high probability. Then the PC algorithm estimates the CPDAG of $\mathcal{G}$ consistently, i.e.

$$\lim_{n\to\infty} \mathbb{P}(\widehat{\mathcal{G}}_n = \mathcal{C}) = 1.$$

Score-based learning

Score-based methods:

$$\widehat{\mathcal{G}} = \underset{G \in Space}{\text{argmax}}\, S(G, \mathbf{D}). \tag{1}$$

1. $\mathbf{D} = (x_{ij})_{n \times p} = [X_1 \mid \ldots \mid X_p]$ i.i.d. data from $(\mathcal{G}, \mathbb{P})$.
2. $S(G, \mathbf{D})$ is a scoring function: log-likelihood of $\mathbf{D}$ given a graph G with a penalty term on model complexity (number of edges or number of free parameters). For example,

$$S_{\text{BIC}}(G, \mathbf{D}) = \log p(\mathbf{D} \mid \widehat{\theta}, G) - \frac{d}{2} \log n, \tag{2}$$

   $\widehat{\theta}$: MLE of parameters under G, d = dimension of θ.
3. Space of graphs: DAGs, equivalence class (CPDAGs) or topological sorts.

Score-based learning

BIC score for Gaussian DAGs:

- Liner SEM for data columns $X_j \in \mathbb{R}^n, j \in [p]$:

$$X_j = \sum_{i \in PA_j} \beta_{ij} X_i + \varepsilon_j, \qquad \varepsilon_j \sim \mathcal{N}_n(0, \omega_j^2 I_n).$$

- Decomposable:

$$\begin{aligned} S_{\text{BIC}}(G, \mathbf{D}) &= \sum_{j=1}^{p} s(X_j, PA_j^G) \qquad (3) \\ &= \sum_j \log p(X_j \mid \widehat{\beta}_j, \widehat{\omega}_j^2, PA_j^G) - \frac{1}{2} |PA_j^G| \log n. \end{aligned}$$

$(\widehat{\beta}_j, \widehat{\omega}_j^2)$: MLEs in Gaussian regression $X_j \sim PA_j^G$.

Score-based learning

Bayesian Dirichlet score for discrete DAGs (Heckerman et al. 1995):

- Multinomial distribution: $\theta_{ijk} = \mathbb{P}(X_i = k \mid PA_i = j)$. Parameter for $[X_i \mid PA_i]$ is a $q_i \times r_i$ table:

$$\Theta_i = \left\{ \theta_{ijk} : j \in [q_i], k \in [r_i], \text{such that} \sum_{k=1}^{r_i} \theta_{ijk} = 1 \right\}.$$

- Assume a conjugate prior over Θ_i given G

$$\Theta_i \mid PA_i \sim \text{Product-Dirichlet}((\alpha_{ijk})_{q_i \times r_i}) \Leftrightarrow$$
$$\theta_{ij} = (\theta_{ij1}, \ldots, \theta_{ijr_i}) \mid PA_i \sim_{ind} \text{Dirichlet}(\alpha_{ij1}, \ldots, \alpha_{ijr_i}).$$

  Choose $\alpha_{ijk} = \alpha/(r_i \cdot q_i)$.

- Assume a prior over G: $P(G) \propto \lambda^{d(G)}$, $\lambda \in (0,1)$ and $d(G) = \sum_{i=1}^{p} r_i q_i$ number of parameters.

Score-based learning

Given $(G, \mathbf{D})$, how to compute the BD score: $(PA_i \equiv PA_i^G)$

- Contingency tables: $N_{ijk} = \#\{PA_i = j \,\&\, X_i = k\}$ in $\mathbf{D}$. For each node, a $q_i \times r_i$ table: $N_i = \{N_{ijk} : j \in [q_i], k \in [r_i]\}$.
- Marginal likelihood of N_{ij} (one row) given PA_i:

$$P(N_{ij} \mid PA_i) = \int P(N_{ij} \mid \theta_{ij})\pi(\theta_{ij} \mid PA_i) d\theta_{ij}$$
$$= \frac{\Gamma(\alpha/q_i)}{\Gamma(N_{ij\bullet} + \alpha/q_i)} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha/(q_i r_i))}{\Gamma(\alpha/(q_i r_i))},$$

  where $N_{ij\bullet} = \sum_k N_{ijk}$ (row sum).
- Marginal likelihood of N_i (the whole table):

$$P(N_i \mid PA_i) = \prod_{j=1}^{q_i} P(N_{ij} \mid PA_i).$$

Score-based learning

- Marginal likelihood of $\mathbf{D}$ (all p tables, one for each node):

$$P(\mathbf{D} \mid G) = \prod_{i=1}^{p} P(N_i \mid PA_i).$$

Posterior distribution

$$\begin{aligned} P(G \mid \mathbf{D}) &\propto P(G)P(\mathbf{D} \mid G) \\ &= \prod_{i=1}^{p} \lambda^{q_i r_i} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha/q_i)}{\Gamma(N_{ij\bullet} + \alpha/q_i)} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha/(q_i r_i))}{\Gamma(\alpha/(q_i r_i))}. \end{aligned}$$

- BD score is decomposable:

$$S_{BD}(G, \mathbf{D}) := \log P(G) + \log P(\mathbf{D} \mid G) = \sum_{i=1}^{p} s(N_i, PA_i). \quad (4)$$

Score-based learning

Properties of the scoring functions (3) and (4):

- Score-equivalent: For any two Markov equivalent DAGs G_1 and G_2, we have $S(G_1, \mathbf{D}) = S(G_2, \mathbf{D})$.
- Consistent (Chickering 2002): A scoring function $S(G, \bullet)$ is *consistent* if the following two properties hold for $\mathbf{D}_n \sim_{iid} \mathbb{P}$:
  1. If $\mathbb{P} \in G \setminus H$, then $\lim_n \mathbb{P}\{S(G, \mathbf{D}_n) > S(H, \mathbf{D}_n)\} = 1$.
  2. If $\mathbb{P} \in G \cap H$ and $d(G) < d(H)$, i.e. G has fewer parameters, then $\lim_n \mathbb{P}\{S(G, \mathbf{D}_n) > S(H, \mathbf{D}_n)\} = 1$.

Haughton (1988) established:

1. $S_{\text{BIC}}(G, \bullet)$ (2) is consistent for exponential family.
2. $S_{BD}(G, \mathbf{D}_n) = S_{\text{BIC}}(G, \mathbf{D}_n) + O_p(1) = O_p(n) + O_p(1)$.

Thus, both (3) and (4) are consistent scoring functions.

Score-based learning

Consistency of score-based learning:

Theorem 3

Suppose $\mathbb{P}$ is faithful to $\mathcal{G}$ and $\mathbf{D}_n \sim_{iid} \mathbb{P}$. If $S(G, \bullet)$ is consistent and score-equivalent, then

$$\lim_{n\to\infty} \mathbb{P}\left\{\underset{G}{\operatorname{argmax}}\, S(G, \mathbf{D}_n) = \mathcal{C}\right\} = 1,$$

where $\mathcal{C} = [\mathcal{G}] := \{G : G \simeq \mathcal{G}\}$ is the Markov equivalence class of $\mathcal{G}$.

Score-based learning

Space and search:

- DAG space: greedy hill climbing (Heckerman et al. 1995; Gámez et al. 2011), stochastic search (e.g. Zhou (2011)).
- Topological sorts: Larranaga et al. (1996); Teyssier and Koller (2005).
  Define score for a sort $\pi \in \mathcal{P}$ (space of permutations): Then search for $\widehat{\pi} = \text{argmax}_{\pi \in \mathcal{P}} S(\pi, \mathbf{D})$.
- Equivalence classes: Greedy Equivalence Search (GES) (Chickering 2002).

Score-based learning

Search over topological sorts:

- Define score for a sort $\pi \in \mathcal{P}$ (space of permutations):

$$S(\pi, \mathbf{D}) := \max_{G \in \mathcal{D}(\pi)} S(G, \mathbf{D}),$$

  where $\mathcal{D}(\pi)$ is the set of DAGs that can be sorted by π.
- $S(\pi, \mathbf{D})$ can be calculated by dynamic programming when $|PA_i| \leq d$ (small) for all i.
- Then search for $\widehat{\pi} = \text{argmax}_{\pi \in \mathcal{P}} S(\pi, \mathbf{D})$ by optimization over permutation space.

Score-based learning

GES (Greedy Equivalence Search):

- Define score for an equivalence class $\mathcal{E}$:

$$S(\mathcal{E}, \mathbf{D}) := S(G, \mathbf{D}), \qquad \forall G \in \mathcal{E}.$$

  $S(\mathcal{E}, \mathbf{D})$ is well-defined if $S(G, \mathbf{D})$ is score-equivalent.
- Neighbors: $\mathcal{E}' \in \mathcal{N}^+(\mathcal{E})$ iff there is $G \in \mathcal{E}$ to which a single edge addition results in a $G' \in \mathcal{E}'$. Similarly define $\mathcal{N}^-(\mathcal{E})$ via single edge deletion.
- Two phases of greedy search from an initial empty graph:
  Phase 1: $\mathcal{E}^{t+1} \leftarrow \text{argmax}\{S(\mathcal{E}, \mathbf{D}) : \mathcal{E} \in \mathcal{N}^+(\mathcal{E}^t)\}$.
  Phase 2: $\mathcal{E}^{t+1} \leftarrow \text{argmax}\{S(\mathcal{E}, \mathbf{D}) : \mathcal{E} \in \mathcal{N}^-(\mathcal{E}^t)\}$.
- In the large sample limit $n \to \infty$, $\widehat{\mathcal{E}}$ found by GES with the BIC or the BD score is the true equivalence class (pr $\to 1$).

Continuous relaxation of score

Continuous relaxation of the scoring function:

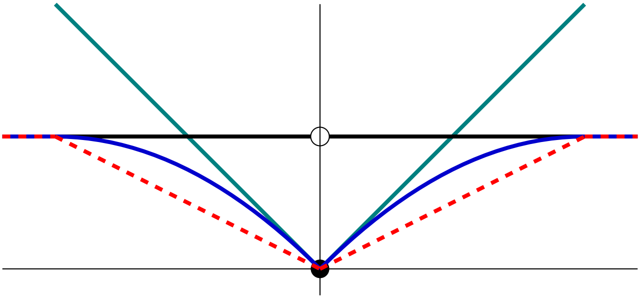
- Consider Gaussian DAGs for simplicity. The BIC score $S_{BIC}(G, \mathbf{D})$ (3) is over a discrete space and hard to optimize.
- $B = (\beta_{ij}) = [\beta_1 \mid \cdots \mid \beta_p]$ and $\Omega = \text{diag}(\omega_j^2)$.
  Maximum regularized likelihood (Fu and Zhou 2013; Aragam and Zhou 2015):

$$(\widehat{B}, \widehat{\Omega}) = \underset{B \in \mathcal{B}, \Omega}{\text{argmax}} \sum_{j=1}^{p} \log p(X_j \mid X\beta_j, \omega_j^2) - \lambda_n \rho(\beta_j). \qquad (5)$$

1. $\mathcal{B}$: weighted adjacency matrices of DAGs, so that $PA_j = \text{supp}(\beta_j)$ and $\text{supp}(B)$ defines a DAG G.
2. $\rho(\beta_j) = \sum_i \rho(|\beta_{ij}|)$: continuous function, e.g. ℓ_1 or concave.

Continuous relaxation of score

Compare regularizers: ℓ_1, concave, and ℓ_0.



Black: ℓ_0 penalty; Teal: ℓ_1 penalty; Blue: MCP; Red, dashed: Capped-ℓ_1 penalty.

Continuous relaxation of score

Maximizing regularized log-likelihood (5)

- Apply continuous optimization, such as block-wise coordinate descent, subject to acyclicity constraint (supp(B) defines a DAG), e.g. Fu and Zhou (2013); Aragam and Zhou (2015).
- Considering maximizing over topological sorts:

$$S(\pi, \mathbf{D}) := \max_{B \in \mathcal{B}(\pi), \Omega} \sum_{j=1}^{p} \log p(X_j \mid X\beta_j, \omega_j^2) - \lambda_n \rho(\beta_j).$$

  $\mathcal{B}(\pi)$: weighted adjacency matrices compatible with π. Computed via p regularized regression problems (lasso or MCP) (Ye et al. 2021).

Reformulation of acyclicity constraint (Zheng et al. 2018): $B \in \mathcal{B}$ if and only if $h(B) = 0$, where $h(\cdot)$ is differentiable.

Learning with experimental data

Score-based learning with experimental data:

- If X_i is under intervention, i.e. $do(X_i = x^*)$: delete edges $X_k \to X_i$ for all $k \in PA_i$.
- Let $\mathcal{O}_i$ be the row indices of the data matrix $\mathbf{D}$ for which node X_i is *not* under intervention (i.e. observational). Replace $p(X_i \mid PA_i)$ by $p(X_{\mathcal{O}_i i} \mid PA_{\mathcal{O}_i i})$.
  1. Gaussian data: log-likelihood in (3) and (5) replaced by

$$\ell(B, \Omega; \mathbf{D}) = \sum_{j=1}^{p} \log p(X_{\mathcal{O}_j j} \mid X_{\mathcal{O}_j} \beta_j, \omega_j^2). \tag{6}$$

  2. Multinomial data: Replace N_{ijk} by

$$N_{ijk}(\mathcal{O}_i) = \#\{rows \in \mathcal{O}_i : PA_i = j \,\&\, X_i = k\}.$$

Identifiability of causal DAGs:

Assumptions:

(A1) The true parameter Θ^* is faithful to $\mathcal{G}$.

(A2) The parameter for $[X_j \mid PA_j]$ is identifiable.

(A3) Each node X_j is under intervention for $n_j \gg \sqrt{n}$ data points.

Theorem 4 (Gu et al. (2019))

Assume (A1), (A2) and (A3). Denote by $\ell(\Theta; \mathbf{D}_n)$ the log-likelihood of the data $\mathbf{D}_n$. For any $\Theta \neq \Theta^*$,

$$\lim_{n\to\infty} \mathbb{P}\{\ell(\Theta^*; \mathbf{D}_n) > \ell(\Theta; \mathbf{D}_n)\} = 1.$$

1. Gaussian data, $\ell(\Theta; \mathbf{D}_n) = (6)$.
2. Discrete data, $\ell(\Theta; \mathbf{D}_n) = \sum_{i=1}^{p} \sum_{j,k} N_{ijk}(\mathcal{O}_i) \log \theta_{ijk}$.

References I

S.A. Andersson, D. Madigan, and Michael D Perlman. A characterization of markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25:505–542, 1997.

Bryon Aragam and Qing Zhou. Concave penalized estimation of sparse Gaussian Bayesian networks. *Journal of Machine Learning Research*, 16:2273–2328, 2015.

Bryon Aragam, Jiaying Gu, and Qing Zhou. Learning large-scale bayesian networks with the sparsebn package. *Journal of Statistical Software*, 91(11):issue 11, 1–38, 2019.

David Maxwell Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3: 507–554, 2002.

References II

Fei Fu and Qing Zhou. Learning sparse causal Gaussian networks with experimental intervention: Regularization and coordinate descent. *Journal of the American Statistical Association*, 108 (501):288–300, 2013.

José A Gámez, Juan L Mateo, and José M Puerta. Learning Bayesian networks by hill climbing: Efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, 22(1-2):106–148, 2011.

Jiaying Gu, Fei Fu, and Qing Zhou. Penalized estimation of directed acyclic graphs from discrete data. *Statistics and Computing*, 29:161–176, 2019.

Dominique M.A. Haughton. On the choice of a model to fit data from an exponential family. *Annals of Statistics*, 16:342–355, 1988.

References III

David Heckerman, Dan Geiger, and David M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

P. Larranaga, M. Poza, Y. Yurramendi, R.H. Murga, and C. Kuijpers. Structure learning of Bayesian networks by genetic algorithms: a performance analysis of control parameters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18: 912–926, 1996.

Christopher Meek. Causal inference and causal explanation with background knowledge. *Uncertainty in Artificial Intelligence*, 11: 403–410, 1995.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer, 1993.

References IV

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1): 62–72, 1991.

Marc Teyssier and Daphne Koller. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. *Proceedings of the 21st Conferences on Uncertainty in Artificial Intelligence*, pages 584–590, 2005.

Q. Ye, A.A. Amini, and Qing Zhou. Optimizing regularized cholesky score for order-based learning of Bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:3555–3572, DOI: 10.1109/TPAMI.2020.2990820, 2021.

Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric Xing. Dags with no tears: Smooth optimization for structure learning. *NIPS*, 2018.

References V

Qing Zhou. Multi-domain dampling with applications to structural inference of Bayesian networks. *Journal of the American Statistical Association*, 106(496):1317–1330, 2011.