

# 732A96/TDDE15 Advanced Machine Learning

## Graphical Models

Jose M. Peña  
IDA, Linköping University, Sweden

Lecture 4: Structure Learning

# Contents

- ▶ Structure Learning for BNs
  - ▶ Independence Test Based Approach
  - ▶ Score Based Approach
- ▶ Structure Learning for MNs
  - ▶ Independence Test Based Approach

# Literature

- ▶ Main source
  - ▶ Koski, T. J. T. and Noble, J. M. A Review of Bayesian Networks and Structure Learning. *Mathematica Applicanda* 40, 51-103, 2012.

## Structure Learning for BNs: Independence Test Based Approach

- ▶ We can get a DAG  $G$  to be used for **probabilistic** reasoning as follows:
- 

Let  $Y_{1:n}$  be any ordering of the random variables  $X_{1:n}$

For each  $Y_i$  do

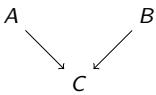
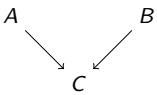
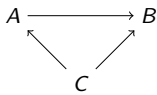
Set  $Pa_i$  to be any minimal subset of  $Y_{1:i-1}$  such that  $Y_i \perp_p Y_{1:i-1} \setminus Pa_i | Pa_i$

---

- ▶ **Exercise.** Prove the previous statement.

## Structure Learning for BNs: Independence Test Based Approach

- ▶ Note that  $G$  has the minimum number of edges among the DAGs that are consistent with the ordering considered.
- ▶ However,  $G$  may not have the minimum number of edges among all the DAGs, i.e. the ordering considered may not be optimal.

$A \perp_p B$	$G$ with ordering $A, B, C$	$G$ with ordering $C, A, B$
 <pre>graph TD; A --&gt; C; B --&gt; C;</pre>	 <pre>graph TD; A --&gt; C; B --&gt; C;</pre>	 <pre>graph TD; A --&gt; B; A --&gt; C; B --&gt; C;</pre>

- ▶ Since the ordering of the variables in the construction of the DAG is arbitrary, we cannot interpret the DAG as a causal structure and, thus, we should not use it for **causal** reasoning.
- ▶ As we will see next, we can get a DAG with minimum number of edges without searching over the  $n!$  orderings assuming that  $p$  is **faithful** to the true DAG  $G^*$ , i.e.  $U \perp_p V|Z$  if and only if  $U \perp_{G^*} V|Z$ . Yet the DAG learned should **only** be used for probabilistic reasoning.

# Structure Learning for BNs: Independence Test Based Approach

---

## Inductive Causation (IC) algorithm

---

Let  $G$  be the complete undirected graph

For each ordered pair of nodes  $X_i$  and  $X_j$  in  $G$  do

    If there is a set  $S \subseteq X \setminus \{X_i, X_j\}$  such that  $X_i \perp_p X_j | S$

        then remove the edge  $X_i - X_j$  from  $G$  and set  $S_{ij} := S_{ji} := S$

For each ordered pair of non-adjacent nodes  $X_i$  and  $X_j$  in  $G$  do

    If there is a node  $X_k \notin S_{ij}$  that is adjacent to both  $X_i$  and  $X_j$

        then orient  $X_i - X_k - X_j$  as  $X_i \rightarrow X_k \leftarrow X_j$  (a.k.a. unshielded collider) in  $G$

For each undirected edge  $X_i - X_j$  in  $G$  do

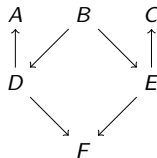
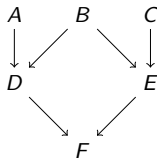
    Orient  $X_i - X_j$  as  $X_i \rightarrow X_j$  if the opposite orientation creates an unshielded collider or a directed cycle

---

- ▶ Parents and children (PC) algorithm: Refinement of the IC algorithm with efficient procedures to find the set  $S$  in line 3 and orient the edges in line 9.

## Structure Learning for BNs: Independence Test Based Approach

- **Exercise.** Run the IC algorithm assuming that  $p$  is faithful to the following DAGs.



## Structure Learning for BNs: Independence Test Based Approach

- ▶ In practice, we do not have access to  $p$  but to a finite sample from it. Then, replace  $X_i \perp_p X_j | S$  in the IC algorithm with an independence test, preferably with one that is consistent so that the algorithm is **asymptotically** correct.
- ▶ Let  $d_{1:N}$  be a complete sample. Then,  $X_i \perp_p X_j | S$  implies that  $p(x_i, x_j | s) = p(x_i | s)p(x_j | s)$  and thus that

$$N_{x_i, x_j, s} \approx N_{x_i, s} N_{x_j, s} / N_s$$

where  $N_{x_i, x_j, s}$  is the number of instances in  $d_{1:N}$  where  $x_i$ ,  $x_j$  and  $s$ , and  $N_{x_i, s} = \sum_{x_j} N_{x_i, x_j, s}$  and  $N_{x_j, s} = \sum_{x_i} N_{x_i, x_j, s}$  and  $N_s = \sum_{x_i, x_j} N_{x_i, x_j, s}$ .

- ▶ We can measure the deviance from the expected situation above by

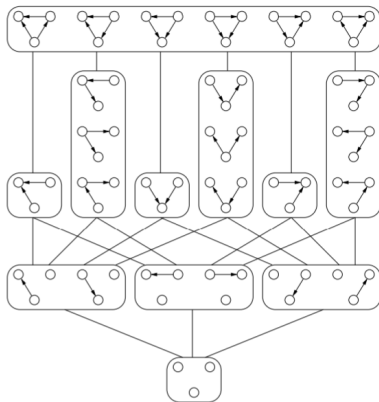
$$deviance = \sum_{x_i, x_j, s} \frac{[N_{x_i, x_j, s} - N_{x_i, s} N_{x_j, s} / N_s]^2}{N_{x_i, s} N_{x_j, s} / N_s}$$

- ▶ If the deviance is too large, then reject the hypothesis that  $X_i \perp_p X_j | S$ .
- ▶ Asymptotically, the deviance follows a  $\chi^2$  distribution with the appropriate number of degrees of freedom, i.e.  $|S|(|X_i| - 1)(|X_j| - 1)$ . Then, we can control the probability of falsely rejecting the hypothesis, a.k.a.  $p$ -value.



## Structure Learning for BNs: Independence Test Based Approach

- Two DAGs represent the same independencies according to the separation criterion (i.e. they are **equivalent**) if and only if they have the same adjacencies and **unshielded colliders**, i.e. subgraphs  $X_i \rightarrow X_k \leftarrow X_j$  where  $X_i$  and  $X_j$  are not adjacent.



Hasse diagram of the space of Markov equivalence classes of Bayesian network structures over three variables.

## Structure Learning for BNs: Independence Test Based Approach

- ▶ The output of the IC algorithm is not a DAG in general, but an **essential graph** (EG):
  - ▶ The EG  $G$  has an edge  $X_i \rightarrow X_j$  if and only if  $X_i \rightarrow X_j$  is in **every** DAG that is equivalent to the true DAG  $G^*$ .
  - ▶ In other words,  $G$  has an edge  $X_i - X_j$  if and only if  $X_i \rightarrow X_j$  is in some DAG that is equivalent to  $G^*$  and  $X_i \leftarrow X_j$  is in some other DAG that is equivalent to  $G^*$ .
- ▶ A naive way to convert  $G$  into a DAG that is equivalent to  $G^*$  is as follows:

---

Repeat while possible

    Replace any edge  $X_i - X_j$  in  $G$  with  $X_i \rightarrow X_j$  if this does not create a directed cycle or a new unshielded collider

    If  $G$  is not a DAG, then backtrack

---

- ▶ Again,  $G$  can be used for probabilistic reasoning but not for causal reasoning.

## Structure Learning for BNs: Score Based Approach

- ▶ Alternatively, we can choose the DAG  $G$  with maximum posterior probability (a.k.a **Bayesian score**):

$$p(G|d_{1:N}) = p(d_{1:N}|G)p(G)/P(d_{1:N}) \propto p(d_{1:N}|G)p(G)$$

where  $p(d_{1:N}|G)$  is the marginal likelihood of  $d_{1:N}$  given  $G$ ,  $p(G)$  is a prior probability distribution, and  $p(d_{1:N})$  is a normalization constant.

- ▶ Moreover

$$p(d_{1:N}|G) = \int p(d_{1:N}|\theta, G)p(\theta|G)d\theta$$

where  $p(d_{1:N}|\theta, G)$  is the likelihood function of  $d_{1:N}$  given  $G$  and  $\theta$ , and  $p(\theta|G)$  is a prior probability distribution.

- ▶ **Assuming** that  $p(\theta|G) = \prod_i \prod_j p(\theta_{x_i|Pa_i=j}|G)$  and  $p(\theta_{x_i|Pa_i=j}|G) \sim \text{Dirichlet}(\alpha_{ij1}, \dots, \alpha_{ijk_i})$ , we have that

$$p(d_{1:N}|G) = \prod_i \prod_j \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij} + N_{ij})} \prod_k \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

where  $\alpha_{ij} = \sum_k \alpha_{ijk}$ ,  $N_{ijk}$  is the number of instances in  $d_{1:N}$  where  $X_i = k$  and  $Pa_i = j$ , and  $N_{ij} = \sum_k N_{ijk}$ .

# Structure Learning for BNs: Score Based Approach

- ▶ The Bayesian score is **score equivalent** (i.e. it gives the same score to equivalent DAGs) if

$$\alpha_{ijk} = \frac{\alpha}{|X_i| \prod_{X_l \in Pa_i} |X_l|}$$

where  $\alpha$  is the user-defined imaginary sample size (the higher the less regularization). This is called the BDeu score. the higher, you have more preference to the graph

- ▶ Under the Dirichlet parameter prior assumption and when  $N \rightarrow \infty$ , we have that

$$\log p(d_{1:N}|G) \approx \log p(d_{1:N}|\theta^{ML}, G) - \frac{\log N}{2} \dim(G)$$

1st term: how well you fit the data

where  $\dim(G)$  is the dimension or number of free parameters of  $G$ , i.e.  $\sum_i (|X_i| - 1) \prod_{X_l \in Pa_i} |X_l|$ .

- ▶ This approximation is called **Bayesian information criterion** (BIC), and it shows that the Bayesian score favours models that trade off fit of data and model complexity.

## Structure Learning for BNs: Score Based Approach

- ▶ Number of DAGs with 1-12 nodes: 1, 3, 25, 543, 29281, 3781503, 1138779265, 783702329343, 1213442454842881, 4175098976430598143, 31603459396418917607425, 521939651343829405020504063
- ▶ Then, an exhaustive search is prohibitive. Then, a heuristic search must be performed instead.

---

### Hill-climbing (HC)

---

Let  $G$  be the empty DAG

Repeat until no change occurs

    Add, remove or reverse any edge in  $G$  that improves the Bayesian score the most

---

- ▶ Unlike the IC algorithm, HC is not asymptotically correct under faithfulness, i.e. it may get trapped in local optima. Still, HC is very popular.

## Structure Learning for MNs: Independence Test Based Approach

- ▶ We can get an UG  $G$  to be used for probabilistic reasoning as follows:

---

For each  $X_i$  do

Set  $Ad(X_i)$  to be any minimal subset of  $X \setminus X_i$  such that  $X_i \perp_p X \setminus Ad(X_i) | Ad(X_i)$

---

- ▶ Luckily, we can get  $G$  without searching over the  $2^{n-1}$  possible adjacent sets for each node if we assume that  $p$  is **faithful** to the true MN  $G^*$ , i.e.  $U \perp_p V | Z$  if and only if  $U \perp_{G^*} V | Z$ .

---

Incremental associative Markov boundary algorithm (IAMB)

---

For each  $X_i$  do

$Ad(X_i) := \emptyset$

Repeat until no change occurs

if there exists  $X_j \notin Ad(X_i) \cup X_i$  such that  $X_i \not\perp_p X_j | Ad(X_i)$  then

$Ad(X_i) := Ad(X_i) \cup X_j$

Repeat until no change occurs

if there exists  $X_j \in Ad(X_i)$  such that  $X_i \perp_p X_j | Ad(X_i) \setminus X_j$  then

$Ad(X_i) := Ad(X_i) \setminus X_j$

---

- ▶ **Exercise.** How would you perform score based structure learning for MNs ?

# Contents

- ▶ Structure Learning for BNs
  - ▶ Independence Test Based Approach
  - ▶ Score Based Approach
- ▶ Structure Learning for MNs
  - ▶ Independence Test Based Approach

Thank you