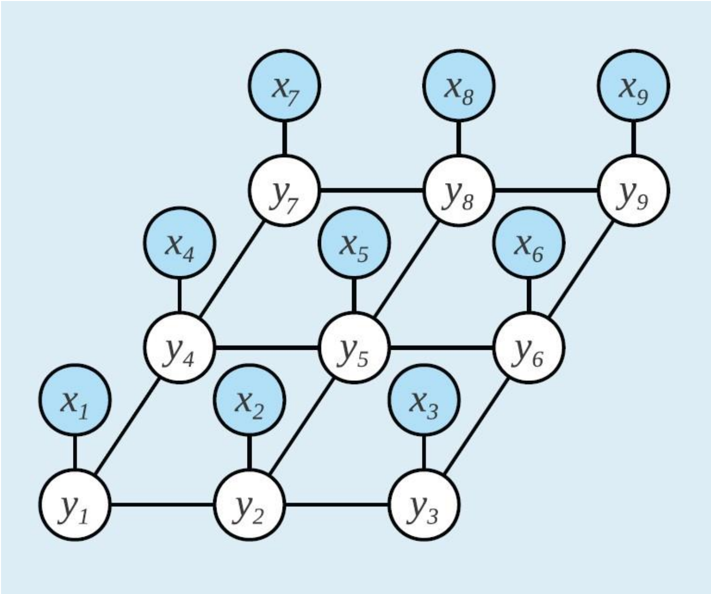
Probabilistic Graphical Models in Bioinformatics

Lecture 7: Parameter estimation and structure learning

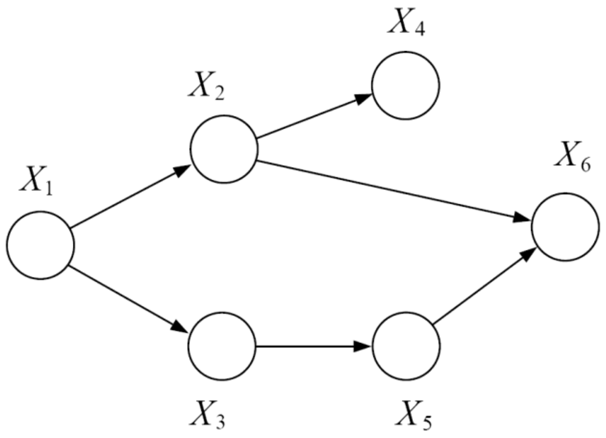


Hesam Montazeri
Department of Bioinformatics, IBB, University of Tehran

Farvardin 28, 1398
(April 17, 2019)

Review

- Representation
  - Factorization
  - Conditional independencies; D-separation
  - Local distributions



- Learning

	Known structure	Unknown structure
Fully observable	Global parameter decomposition MLE Bayesian methods	
Partially observable		

- Inference

Bayesian parameter estimation in Bayesian networks

A simple example

- Consider the network $X \to Y$
- Training data consists of observations $X[m], Y[m]$ for $m = 1, \ldots, M$.
- Question: what are the parameters?

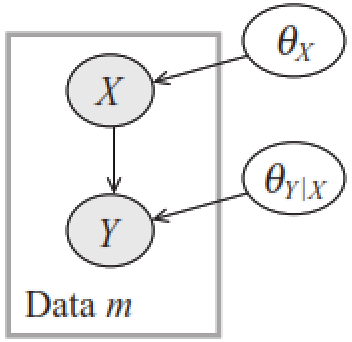
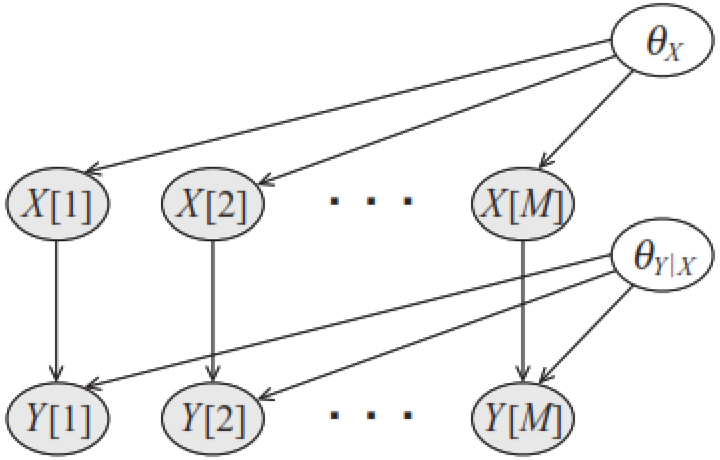


Plate model



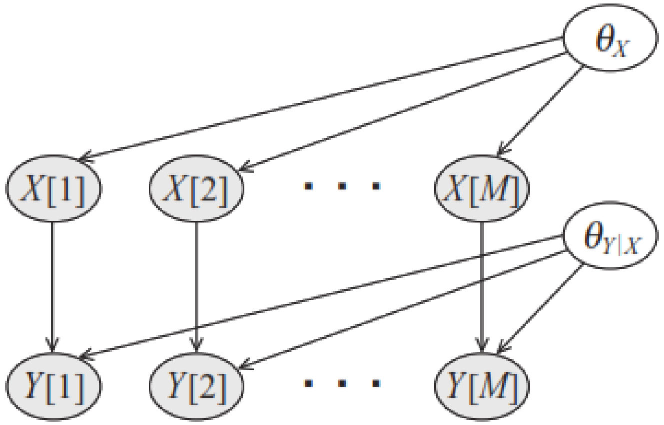
Ground Bayesian network

- Instances are independent given the unknown parameters.
  - $X[m]$ and Y$[m]$ are d-separated from $X[m']$ and Y$[m']$ given parameters

Global parameter independence

- If G is a Bayesian network with parameters

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_{X_1|\mathrm{Pa}_{X_1}}, \ldots, \boldsymbol{\theta}_{X_n|\mathrm{Pa}_{X_n}})$$



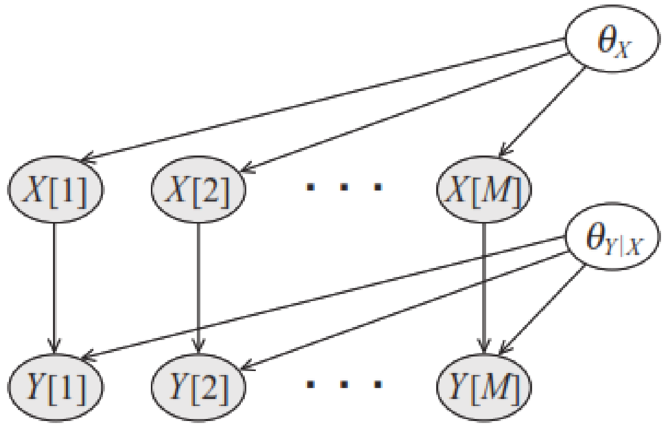
- Global parameter independence
    - Parameters for individual variables are independent a priori
    - Knowing the value of one parameter tells us nothing about another
    - A prior $P(\theta)$ satisfies global independence if

$$P(\boldsymbol{\theta}) = \prod_i P(\boldsymbol{\theta}_{X_i|\mathrm{Pa}_{X_i}})$$

- Not always an appropriate assumption

Bayesian estimation in BNs

- If we accept global parameter independence assumption, we have the following conclusion



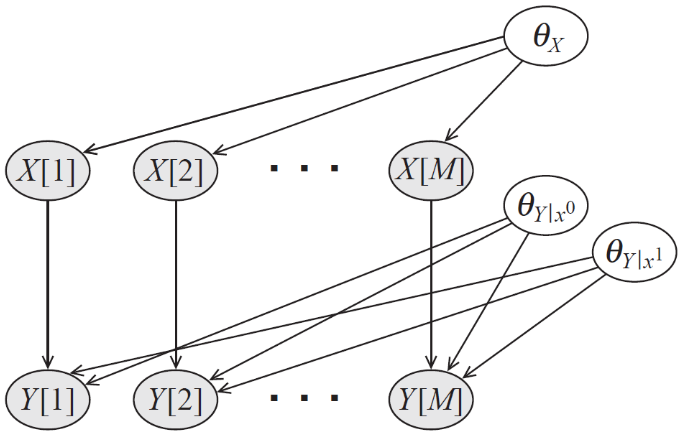
- Posterior of θ are independent given complete data
  - Complete data d-separates the parameters for different CPDs.

$$P(\boldsymbol{\theta}_X, \boldsymbol{\theta}_{Y|X} \mid \mathcal{D}) = P(\boldsymbol{\theta}_X \mid \mathcal{D}) P(\boldsymbol{\theta}_{Y|X} \mid \mathcal{D}).$$

- Practical ramification:
  - Given the data set D, we can determine the posterior over θ_X independently of posterior over $\theta_{Y|X}$.
  - We can solve each problem separately and then combine the results (analogous to the likelihood decomposition for MLE)

Local decomposition to table CPDs

- How to compute posterior for θ_X and $\theta_{Y|X}$?



- Independence of $\theta_{Y|x^0}$ and $\theta_{Y|x^1}$ given the data?
  - No d-separation between $\theta_{Y|x^0}$ and $\theta_{Y|x^1}$ given the data
  - However, we have context-specific independence between them

$$P(y[m] = y \mid x[m], \boldsymbol{\theta}_{Y|x^0}, \boldsymbol{\theta}_{Y|x^1}) = \begin{cases} \boldsymbol{\theta}_{y|x^0} & \text{if } x[m] = x^0 \\ \boldsymbol{\theta}_{y|x^1} & \text{if } x[m] = x^1. \end{cases}$$

- In this case, we have

$$P(\boldsymbol{\theta} \mid \mathcal{D}) = \prod_i \prod_{\mathrm{pa}_{X_i}} P(\boldsymbol{\theta}_{X_i|\mathrm{pa}_{X_i}} \mid \mathcal{D}).$$

- For multinomial $\theta_{X|u}$:
  - Prior: $Dirichlet(\alpha_{x^1|u}, \dots, \alpha_{x^k|u})$
  - Posterior: $Dirichlet(\alpha_{x^1|u} + M[x^1, u], \dots, \alpha_{x^k|u} + M[x^k, u])$

Priors for Bayesian network learning

- K2 prior
  - Use a fixed prior for all hyperparameters, e.g., $\alpha_{x_i^j|\mathrm{pa}_{X_i}} = 1$
  - Conceptually unsatisfying

- Bayesian Dirichlet equivalent (Bde) prior
  - Assume we have an imaginary dataset D' of prior examples.
  - Let $\alpha[x_i, \mathrm{pa}_{X_i}]$ denotes the number of observations in D' with respective values. Then, we may set

$$\cancel{\alpha_{x_i|\mathrm{pa}_{X_i}} = \alpha[x_i, \mathrm{pa}_{X_i}]}$$

Issue: storing a large dataset of pseudoinstances

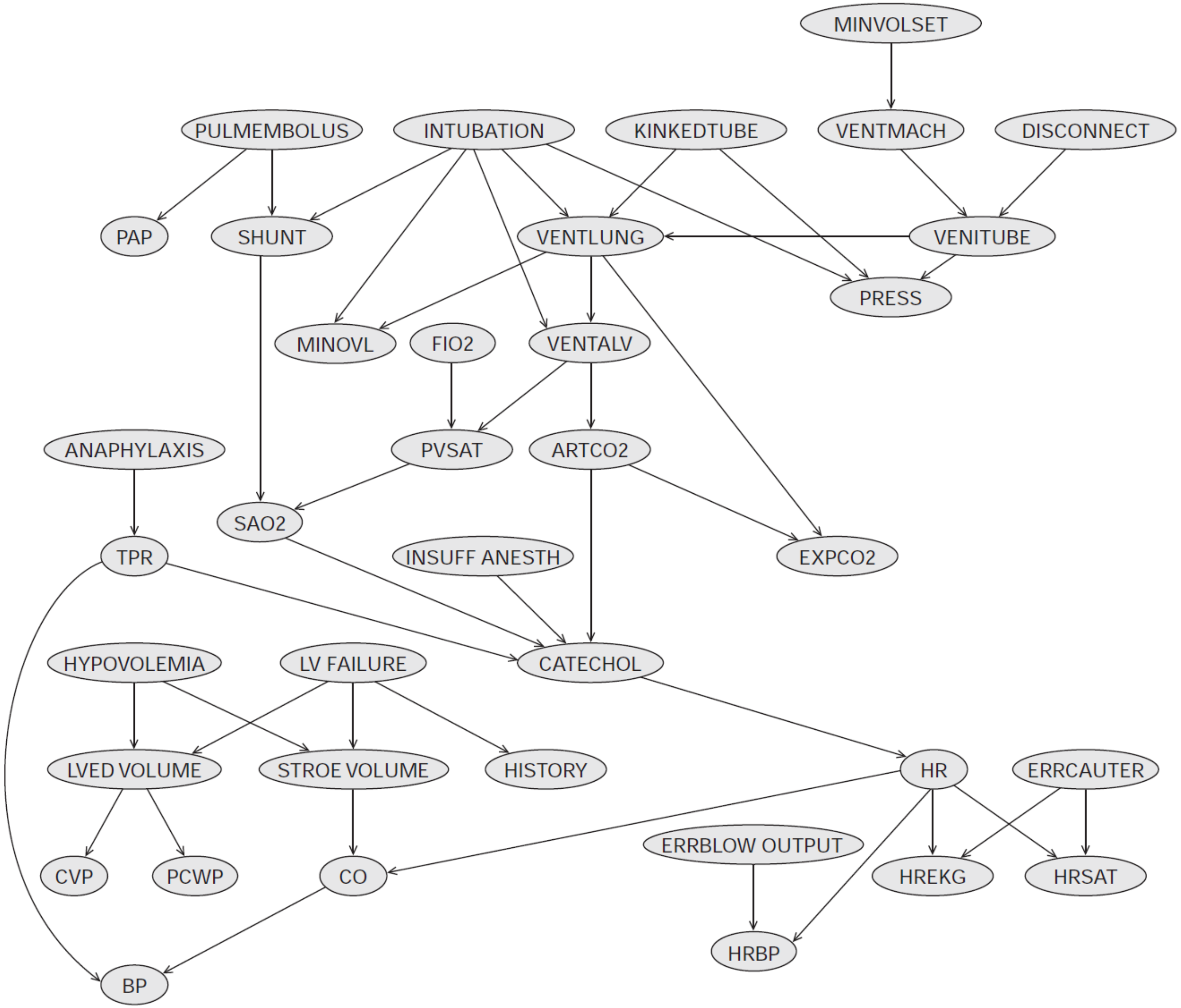
  - Instead, we can store α and a representation $P'(X_1, \dots, X_n)$ of D'

$$\alpha_{x_i|\mathrm{pa}_{X_i}} = \alpha \cdot P'(x_i, \mathrm{pa}_{X_i}).$$

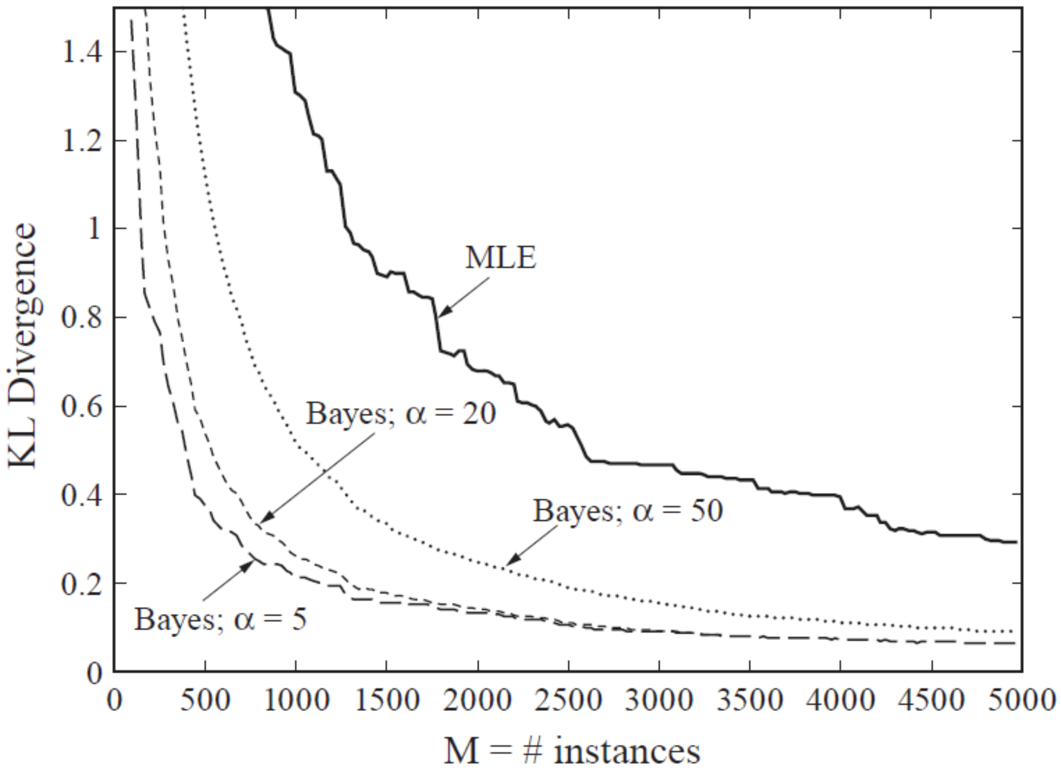
  - P' can be a set of independent marginals over the X_i's (called BDe prior in this case).

ICU-Alarm network

- 37 nodes
- 504 parameters



ICU Alarm network



Kullback–Leibler divergence is a asymmetric measure of the difference between two probability distributions

$$D_{KL}(p(x)||q(x)) = \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)}$$

Structure learning in Bayesian networks

Goals of learning

- Assume
  - P^*: underlying distribution
  - $M^* = (g^*, \theta^*)$: underlying graphical model with structure g^* and parameters θ^*

- Density estimation
  - We learn a network model to answer probabilistic queries (inference task).
  - Model evaluated on test data likelihood
  - We aim to recover P^*

- Knowledge discovery
  - We may hope that examination of the learned model can reveal some important properties of the domain structure.
  - Model evaluated by prior knowledge
  - We aim to recover g^*

Problem definition

- Example: two independent coins
  - We toss two standard X and Y independently.
  - A “typical” data set: 27 head/head, 22 head/tail, 25 tail/head, 26 tail/tail
  - In the *empirical* distribution, the two coins are not independent.

- Now consider independence of football and rain
  - We scan the sports section of a newspaper for 100 days
  - $X = x^1$ if the word “rain” appears and $X = x^1$ otherwise.
  - $Y = y^1$ if the word “football” appears and $Y = y^1$ otherwise.
  - If we get the same data as the in the coins, we might suspect there is some weak connection.

Problem definition-2

- If our goal is to understand domain structure
  - We want to recover g^*
  - However, there can be many perfect maps for a distribution P^*
    - All of the networks in the same I-equivalence as g^*
    - Hence, g^* is not identifiable from the data.

  - In general, the goal of learning g^* (or an equivalent network) is hard to achieve.
    - The data sampled from P^* are noisy and is difficult to detect independencies reliably from the data.
    - We need to decide about our willingness to include in our leaned model edges which we are less sure.
      - Spurious correlation or spurious independencies?

Problem definition-3

- If our goal is to perform density estimation
  - In other words, the goal is to estimate a statistical model of the underlying distribution
  - We are looking for a network model to *generalize* to new instances.

  - **Question:** which network structure will lead to the best generalization?
    - Due to limited data, it is often better to prefer a sparser structure
    - Hence, g^* is not often the best model in term of generalization performance!

Structure learning methods

- Constraint-based structure learning
- Score-based structure learning
- Bayesian model averaging methods

Constraint-based approaches

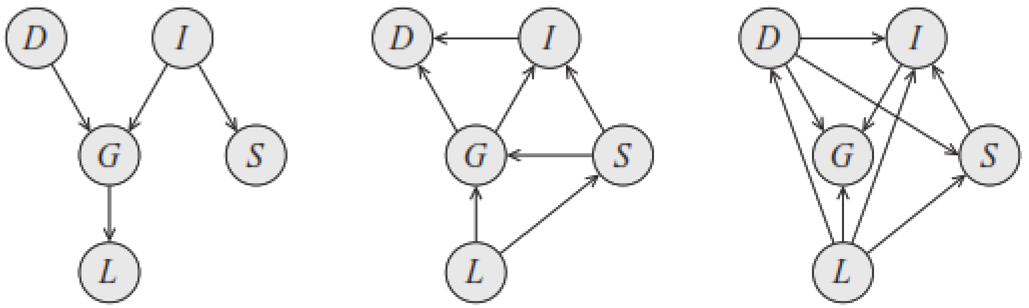
Motivation

- We attempt to reconstruct a network structure that best captures the independencies in the domain.
- This approach requires performing independence tests between variables in the data.
- **Question:** Assume a dataset with three variables is given. Explore on how to use independence tests to learn the optimal structure.
- Two approaches
  - Constructing a *minimal I-map*
  - Search for a *perfect map*

Constructing a minimal I-map

University of TEHRAN

- A graph $\mathcal{K}$ is a minimal I-map for a set of independencies I if
    A. it is an I-map for I, *i.e.* $\mathrm{I}(\mathcal{K}) \subseteq I$.
    B. and removal a single edge from graph $\mathcal{K}$ makes it not an I-map.



- Algorithm build-Minimal-I-Map has some issues:
    - The input order impacts on complexity of the network we find.
    - Conditional independence statements might involve large number of variables.

Algorithm 3.2 Procedure to build a minimal I-map given an ordering

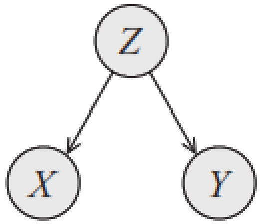
Procedure Build-Minimal-I-Map (
  $X_1, \ldots, X_n$ // an ordering of random variables in $\mathcal{X}$
  $\mathcal{I}$ // Set of independencies
)
  Set $\mathcal{G}$ to an empty graph over $\mathcal{X}$
  for $i = 1, \ldots, n$
    $\boldsymbol{U} \leftarrow \{X_1, \ldots, X_{i-1}\}$ // $\boldsymbol{U}$ is the current candidate for parents of X_i
    for $\boldsymbol{U}' \subseteq \{X_1, \ldots, X_{i-1}\}$
      if $\boldsymbol{U}' \subset \boldsymbol{U}$ and $\boxed{(X_i \perp \{X_1, \ldots, X_{i-1}\} - \boldsymbol{U}' \mid \boldsymbol{U}')} \in \mathcal{I}$ **then**
        $\boldsymbol{U} \leftarrow \boldsymbol{U}'$
    // At this stage $\boldsymbol{U}$ is a minimal set satisfying $(X_i \perp \{X_1, \ldots, X_{i-1}\} - \boldsymbol{U} \mid \boldsymbol{U})$
    // Now set $\boldsymbol{U}$ to be the parents of X_i
    for $X_j \in \boldsymbol{U}$
      Add $X_j \rightarrow X_i$ to $\mathcal{G}$
  return $\mathcal{G}$

Approach 2: search for a P-map

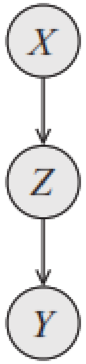
- In this approach, we learn an I-equivalence class rather than a single network.

- I-equivalence: same set of conditional independence assertions for different BN structures

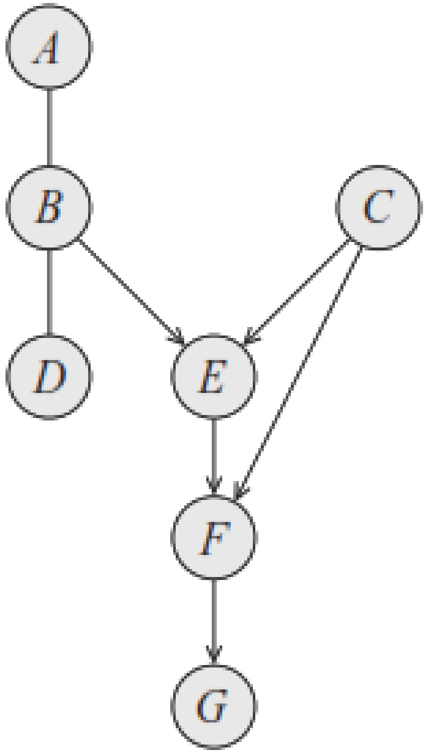
G1:



G2:



$$I(G_1) = I(G_2) = \{X \perp Y \mid Z\}$$



An example PDAG

- We use *acyclic partially directed graphs* (known as PDAG) to represent equivalence classes of DAGs.

Finding perfect maps

- **Theorem:** Two BN graphs have the **same skeleton** and the **same set of immoralities** if and only if they are I-equivalent.

- Algorithm to find a P-map:
  - Identify the undirected skeleton
  - Identify immoralities → results in a PDAG
  - We can orient more edges according to some rules → results in a complete PDAG

Identify the undirected skeleton

- Basic idea:
  - To use independence queries of the form $X \perp Y \mid U$ for different sets of variables U.
  - if X and Y are connected in g^*, we cannot separate them with any set of variables

Question: find PDAG if we have

$A \perp B,\ A \perp D \mid B\ , C \perp D \mid B$

Algorithm 3.3 Recovering the undirected skeleton for a distribution P that has a P-map

Procedure Build-PMap-Skeleton (
  $\mathcal{X} = \{X_1, \ldots, X_n\}$, // Set of random variables
  P, // Distribution over $\mathcal{X}$
  d // Bound on witness set
)
  Let $\mathcal{H}$ be the complete undirected graph over $\mathcal{X}$
  for X_i, X_j in $\mathcal{X}$
    $\boldsymbol{U}_{X_i,X_j} \leftarrow \emptyset$
    for $\boldsymbol{U} \in$ Witnesses$(X_i, X_j, \mathcal{H}, d)$
      // Consider $\boldsymbol{U}$ as a witness set for X_i, X_j
      if $P \models (X_i \perp X_j \mid \boldsymbol{U})$ **then**
        $\boldsymbol{U}_{X_i,X_j} \leftarrow \boldsymbol{U}$
        Remove X_i—X_j from $\mathcal{H}$
        break
  return $(\mathcal{H}, \{\boldsymbol{U}_{X_i,X_j} : i, j \in \{1, \ldots, n\}\})$

If X and Y are not adjacent in g^*, we can find a set U, called witness set, so that $X \perp Y \mid U$.

Identify immoralities

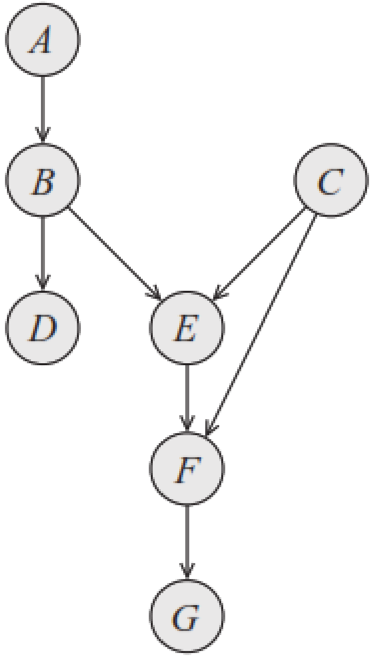
- The main cue for learning edge directions in g^* are *immoralities*.
- According the theorem 3.8, all DAGs in the equivalence class of g^* share the same set of immoralities.

Algorithm 3.4 Marking immoralities in the construction of a perfect map

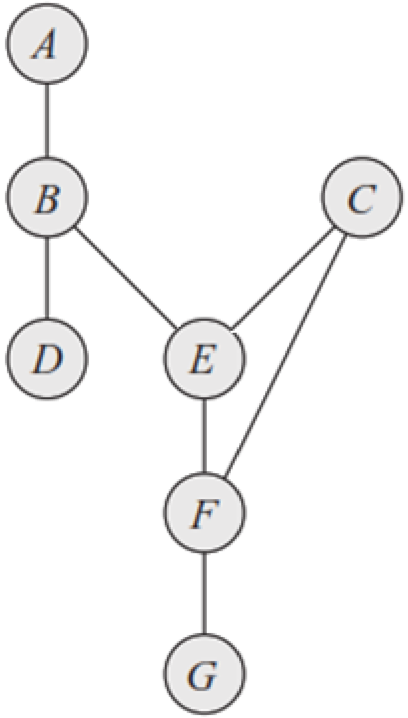
Procedure Mark-Immoralities (
  $\mathcal{X} = \{X_1, \ldots, X_n\}$,
  S // Skeleton
  $\{\boldsymbol{U}_{X_i,X_j} : 1 \leq i, j \leq n\}$ // Witnesses found by Build-PMap-Skeleton
)

```
1   K ← S
2   for X_i, X_j, X_k such that X_i—X_j—X_k ∈ S and X_i—X_k ∉ S
3       // X_i—X_j—X_k is a potential immorality
4     if X_j ∉ U_{X_i,X_k} then
5       Add the orientations X_i → X_j and X_j ← X_k to K
6   return K
```

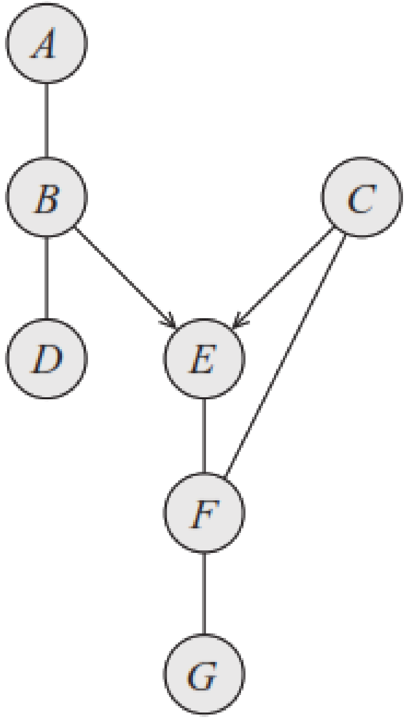
Example



Original DAG g^*

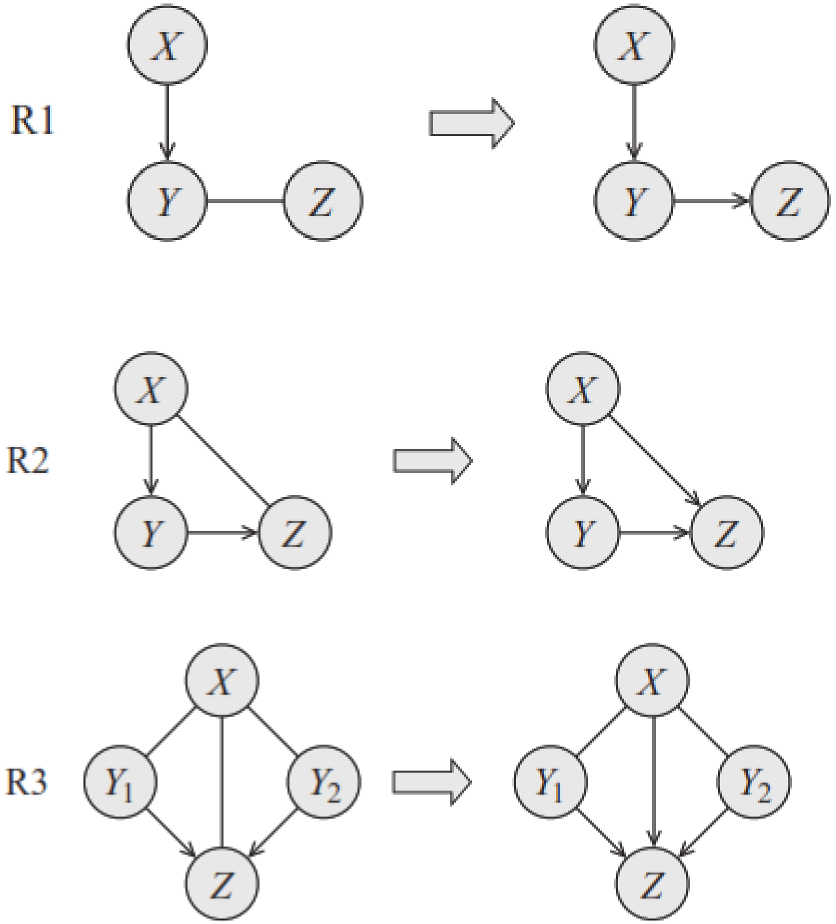


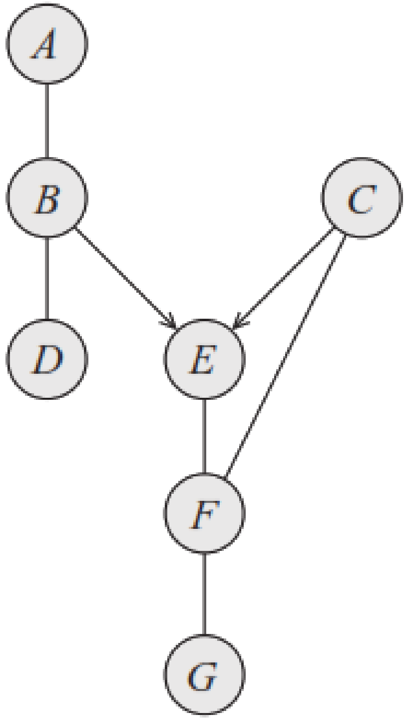
Skeleton of g^*



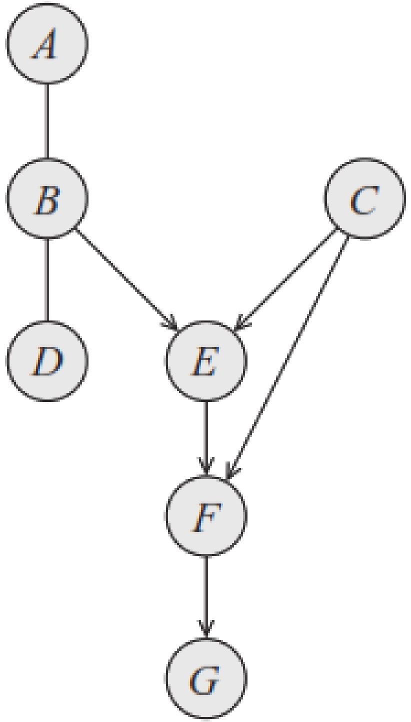
Skeleton of g^* annotated with immoralities

Rules for orienting edges in PDAG





Skeleton of g^* annotated with immoralities



Complete PDAG

Important: repeated application of these three local rules is guaranteed to capture all edge orientations in the equivalence class.

Algorithm that implements this process

Algorithm 3.5 Finding the class PDAG characterizing the P-map of a distribution P

```
Procedure Build-PDAG (
    𝒳 = {X_1, ..., X_n}    // A specification of the random variables
    P    // Distribution of interest
)
    S, {U_{X_i,X_j}} ← Build-PMap-Skeleton(𝒳, P)
    𝒦 ← Find-Immoralities(𝒳, S, {U_{X_i,X_j}})
    while not converged
        Find a subgraph in 𝒦 matching the left-hand side of a rule R1–R3
        Replace the subgraph with the right-hand side of the rule
    return K
```

Independence tests

- The only remaining question is how to answer queries about conditional independencies between variables in the data.

- Testing $X \perp Y$:
  - **Null hypothesis:** X and Y are independent.
  - **Test statistic:** deviance measure from the null hypothesis

$$d_{\chi^2}(\mathcal{D}) = \sum_{x,y} \frac{(M[x,y] - M \cdot \hat{P}(x) \cdot \hat{P}(y))^2}{M \cdot \hat{P}(x) \cdot \hat{P}(y)}.$$

  - Under the null hypothesis, χ^2 statistic follows χ^2 distribution.

- Testing conditional independence $X \perp Y \mid Z$

$$d_{\chi^2}(\mathcal{D}) = \sum_{x,y,z} \frac{(M[x,y,z] - M \cdot \hat{P}(z)\hat{P}(x \mid z)\hat{P}(y \mid z))^2}{M \cdot \hat{P}(z)\hat{P}(x \mid z)\hat{P}(y \mid z)}.$$

Independence tests-2

- **Question:** why is multiple hypothesis testing an issue here?

- Hence, some of the independence tests results can be wrong

- One misleading test result can produce multiple errors in the resulting PDAG.