Causal Inference with Graphical Models

Definition, Correlation vs. Causality, Challenges, Constraint-based learning using PC algorithm, Additive Noise Model: LiNGAM, Recent developments in causal discovery

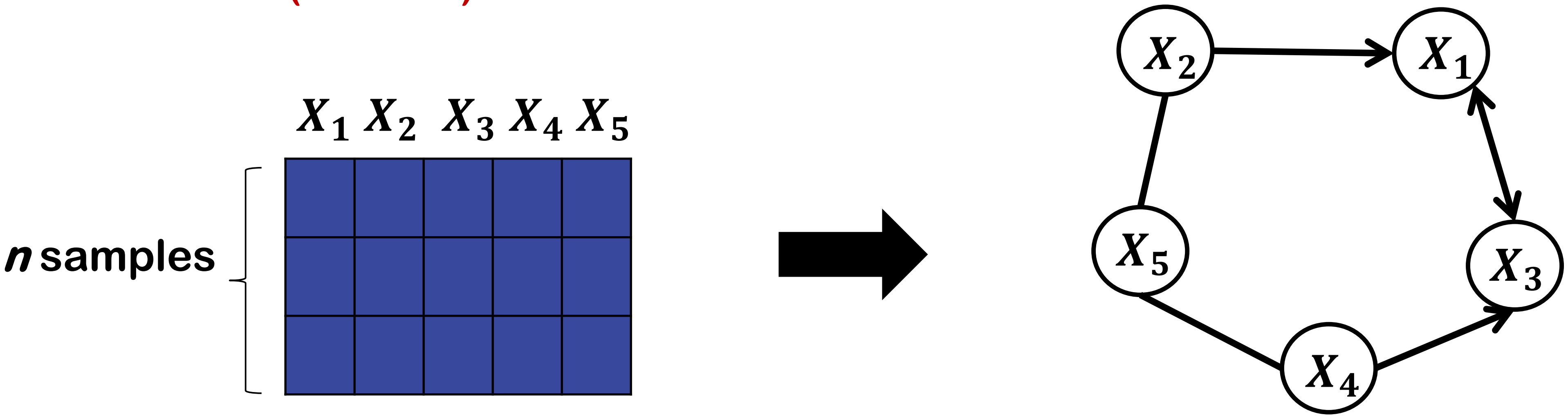
Mandar S. Chaudhary mschaudh@ncsu.edu

PhD student in Dr. Samatova's lab
Department of Computer Science
North Carolina State University

NC STATE UNIVERSITY
Department of Computer Science

What is Causal Inference?

- Causal Inference is the task of learning cause-effect relationships and estimating causal effects from the data.
- A directed edge represents a potential causal relationship, whereas an undirected or a bidirected edge represents an ambiguous relationship.
- For example, the directed edge $X_2 \rightarrow X_1$ indicates X_2 is a potential cause (or parent) and X_1 is the effect (or child).



An estimated causal structure from the PC algorithm

Correlation vs. Causality

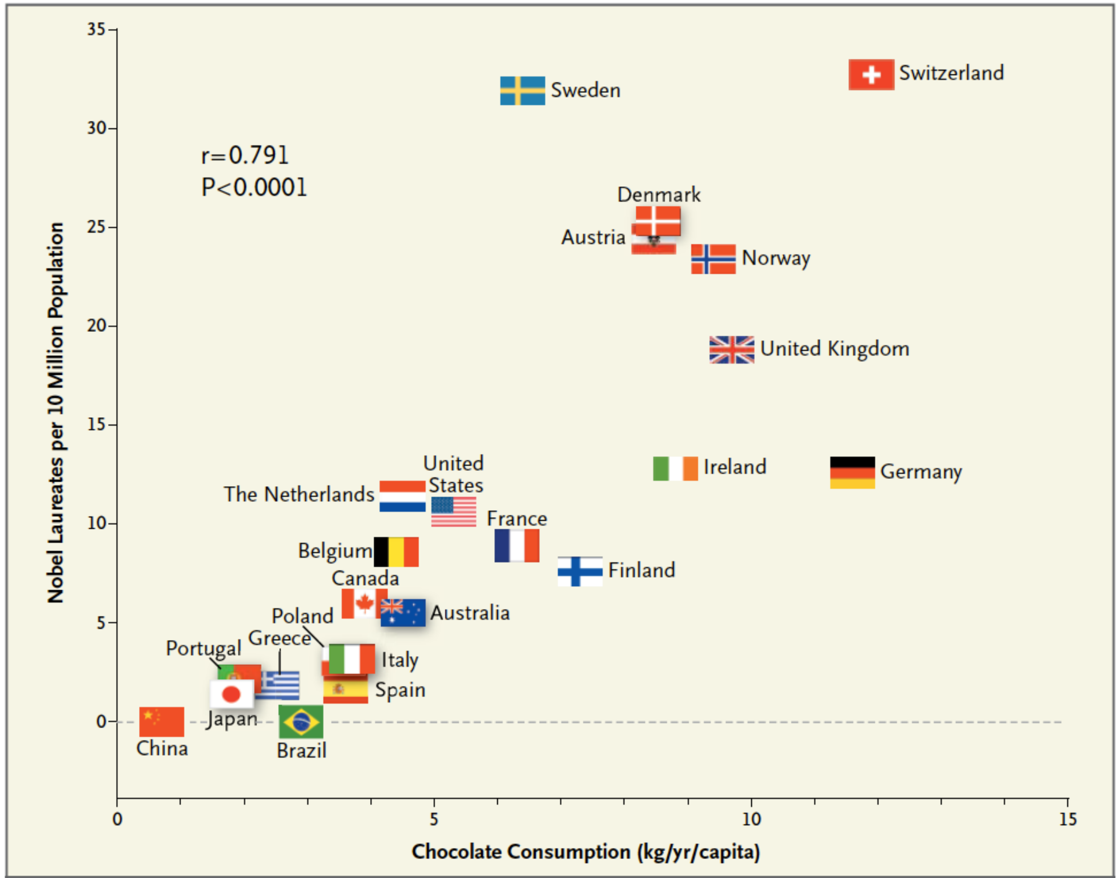


Figure 1. Correlation between Countries' Annual Per Capita Chocolate Consumption and the Number of Nobel Laureates per 10 Million Population.

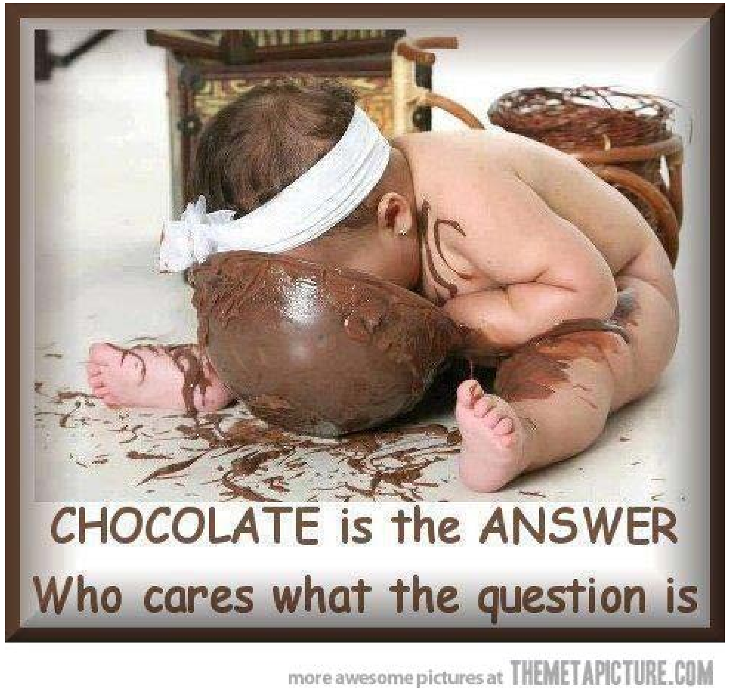
This research shows there is a strong correlation between chocolate consumption and winning Nobel prize!

Courtesy: Google Images

If Correlation == Causality

Courtesy: Google Images

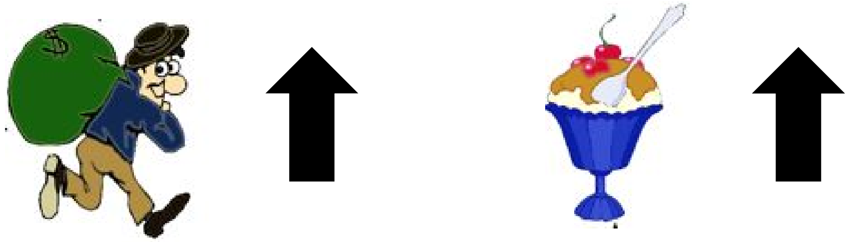
But the reality is…



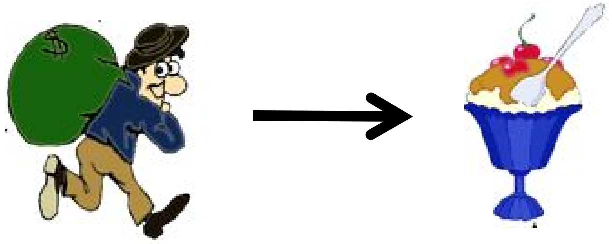
Courtesy: Google Images

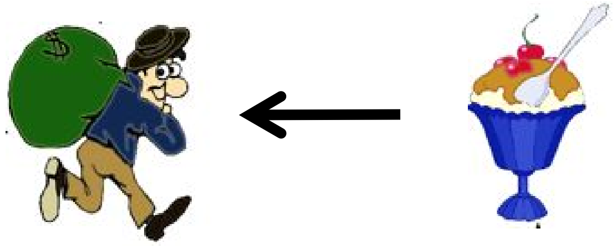
Example

- During summer, there is an increase in ice-cream sales and the crimes committed.



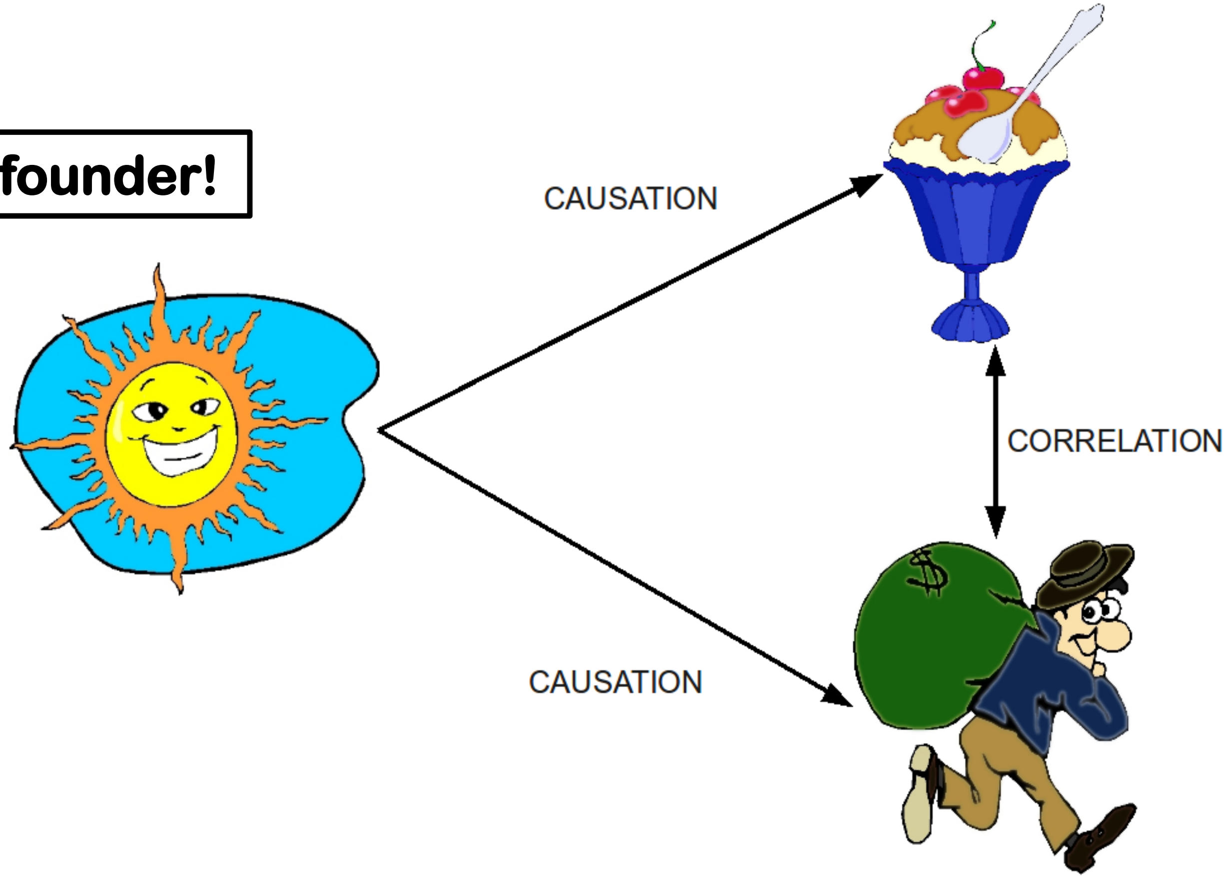
- Does this mean?





Example (cont.)

Hidden confounder!



Courtesy: Google Images

Challenges

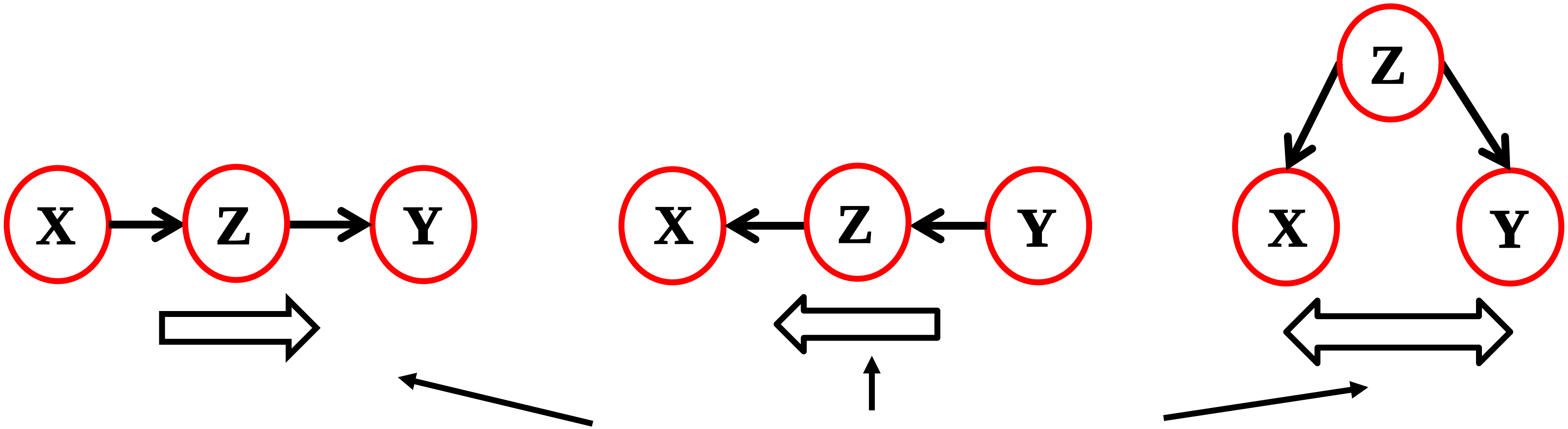
- Identifying accurate cause-effect relationships is obscured by spurious associations in the data.
- Difficult to test causality in real-world applications:
  - it involves performing randomized experiments that might not be feasible
- Underdetermined data sets
  - with fewer than hundred samples and thousands of variables
  - make it even more difficult to estimate cause-effect relationships

Constraint-based method: PC Algorithm

- **Goal**: **Test every pair of adjacent variables by performing (un)conditional independence tests to remove as many spurious associations as possible.**

Conditional Independence: d-separation

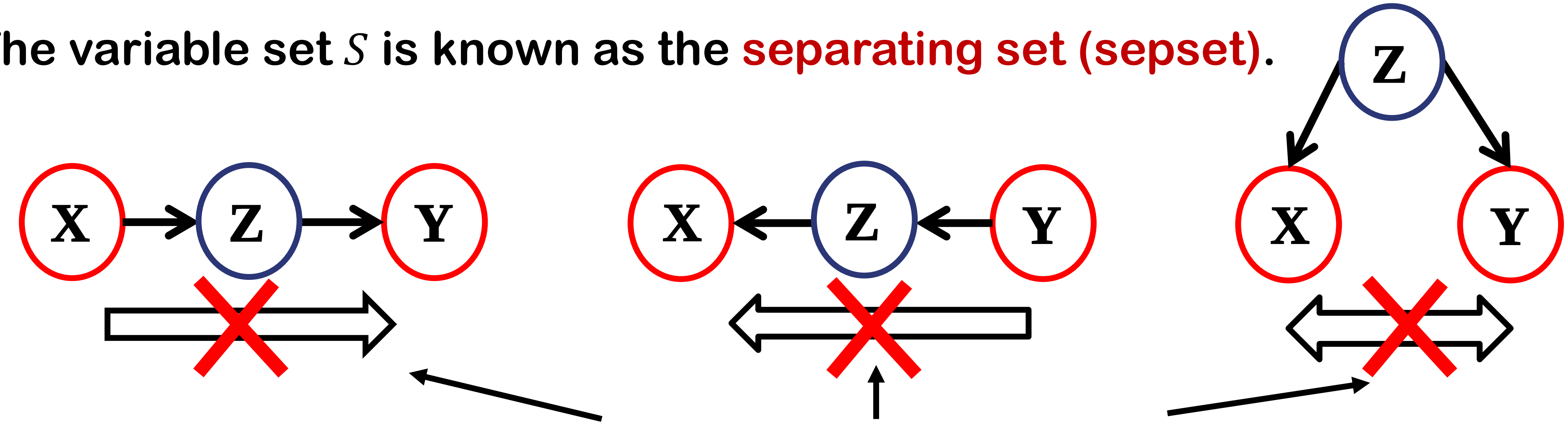
- Two variables X and Y are said to be d-separated by a set of variables S if X and Y are independent conditioned on S.
  - The idea is to block the information flow between X & Y using the set of variables in S.
- Consider the following three structures between X and Y, with $S = \{Z\}$:



The arrows indicate the flow of information between X and Y.

Conditional Independence: d-separation (cont.)

- Variable $Z \in S$:
  - d-separates X and Y as it blocks the information flow between the two variables
- The variables in S should d-separate X and Y for each path that exists between them.
- The variable set S is known as the separating set (sepset).

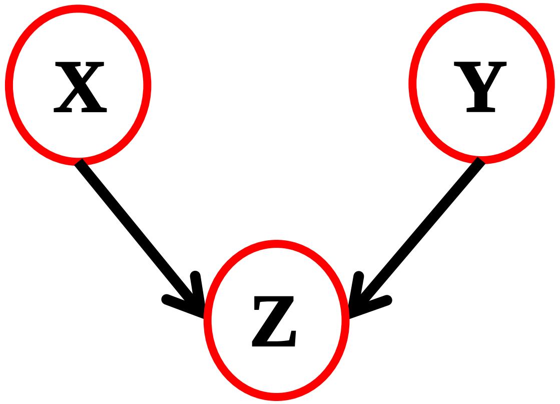


The information flow between X and Y is blocked by Z.

$X \perp\!\!\!\perp Y | Z \qquad \boldsymbol{sepset(X,Y) = Z}$

Conditional Independence: v-structure

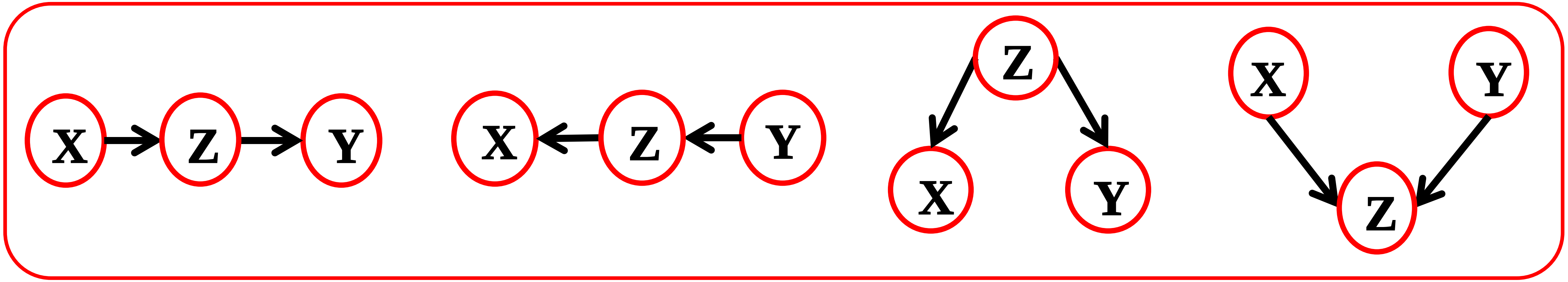
- A v-structure is a triple $X \rightarrow Z \leftarrow Y$ such that X and Y are not adjacent.
- X and Y are unconditionally independent but they are conditionally dependent given their child Z.



$$X \perp\!\!\!\perp Y$$

$$X \not\!\perp\!\!\!\perp Y \mid Z$$

Conditional Independence: Statistical tests



- In real-world the real causal structure is rarely known.
- To infer the structure we have a data set and statistical tests at our disposal.

There is a high chance that X and Y do not have a direct causal relationship.

Conditional Independence: Stat test: Continuous Variables

- **Continuous variables: Fisher's Z test**

$$Z = \sqrt{N - |W| - 3} * \frac{1}{2} log_e \left(\frac{1 + \tilde{\rho}_{X,Y|W}}{1 - \tilde{\rho}_{X,Y|W}} \right)$$

  - where $\tilde{\rho}_{X,Y|W}$ is the sample partial correlation between two variables X and Y given a set of variables W.

- The null hypothesis, $H_0: \tilde{\rho}_{X,Y|W} = 0$ is rejected if,

$$Z > \Phi^{-1}(1 - \frac{\alpha}{2})$$

  - where Φ is a cumulative distribution function of a Gaussian distribution with mean 0 and variance 1.

Conditional Independence: Stat test: Categorical Variables

- **Binary or Discrete variables**: G^2 test

$$G = 2 * \sum_i O_i log\left(\frac{O_i}{E_i}\right)$$

  - where O_i=observed frequencies and E_i=expected frequencies of the discrete variables.
- To test if X is independent of Y given Z, a contingency table is built for every level of Z and the G-statistic is the summation over all levels.

O_i

X\Y	0	1
0	N_{00}	N_{01}
1	N_{10}	N_{11}

$$N = N_{00} + N_{01} + N_{10} + N_{11}$$

E_i

X\Y	0	1
0	$(N_{+0} * N_{0+})$/N	$(N_{+1} * N_{0+})$/N
1	$(N_{+0} * N_{1+})$/N	$(N_{+1} * N_{1+})$/N

$$N_{+0} = N_{00} + N_{10}, N_{0+} = N_{00} + N_{01}$$
$$N_{+1} = N_{01} + N_{11}, N_{1+} = N_{10} + N_{11}$$

Conditional Independence: Stat test: G-statistic

- **Binary or Discrete variables**: G^2 test

$$\boldsymbol{G} = 2 * \sum_{\boldsymbol{i}} \boldsymbol{O_i} \boldsymbol{log}\left(\frac{\boldsymbol{O_i}}{\boldsymbol{E_i}}\right)$$

  - where O_i=observed frequencies and E_i=expected frequencies of the discrete variables.

- The G-statistic follows a Chi-square distribution with I*(J-1)*(K-1) degrees of freedom where I, J and K are the number of levels of X, Y and Z, respectively.

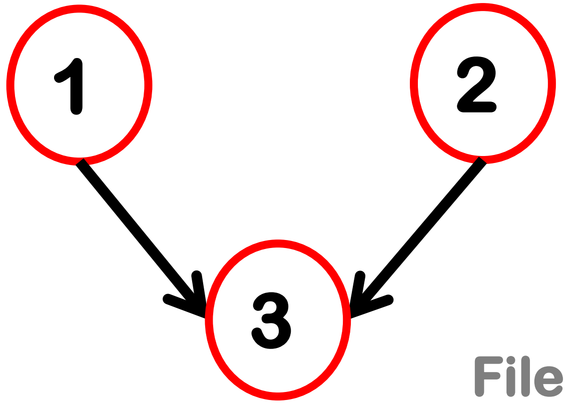
$$p - value = pchisq(G, df, lower.tail = FALSE)$$

- The null hypothesis of independence is rejected if $p - value < \alpha$; otherwise, we fail to reject the null hypothesis.

Conditional Independence: Statistical tests

- **Null hypothesis** Ho: Variables are independent → delete the edge
- **Alternative hypothesis** H1: Variables are not independent → keep the edge

Variable type	Test	R function
Continuous	Fisher's Z test	gaussCItest
Discrete	G^2 test	disCItest
Binary	G^2 test	binCItest



$1 \perp\!\!\!\perp 2$

File: causality.R

```
> gaussCItest(1,2,S=NULL,suffStat=list(C=cor(dat), n=nrow(dat)))
[1] 0.7581304
```

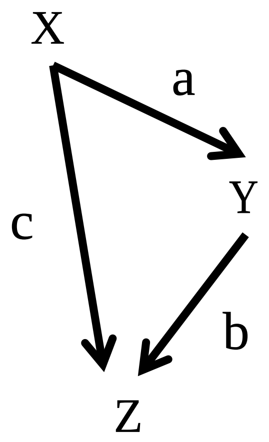
- If p-value < α we reject the null hypothesis, H_0 and if p-value > α we fail to reject the null hypothesis.

Constraint-based method: PC Algorithm

- **Goal: Test every pair of adjacent variables by performing (un)conditional independence tests to remove as many spurious associations as possible.**
- **Assumptions:**
  1. There are no cyclic causal relations
     - for example, $X \rightarrow Y$ and $Y \rightarrow X$
  2. Causal faithfulness: Dependencies found by PC algorithm are characteristic of the underlying data distribution.

Constraint-based method: PC Algorithm

- **Goal: Test every pair of adjacent variables by performing (un)conditional independence tests to remove as many spurious associations as possible.**
- **Assumptions:**
  1. There are no cyclic causal relations for example, $X \to Y$ and $Y \to X$
  2. Causal faithfulness: Dependencies found by PC algorithm are characteristic of the underlying data distribution.

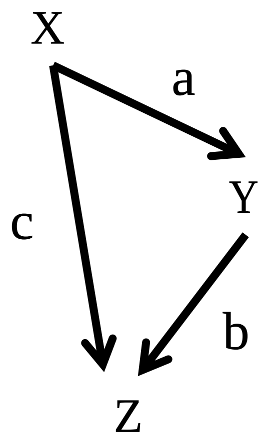


$$\boldsymbol{X} = \boldsymbol{\epsilon_X}$$
$$\boldsymbol{Y} = \boldsymbol{aX} + \boldsymbol{\epsilon_Y}$$
$$\boldsymbol{Z} = \boldsymbol{cX} + \boldsymbol{bY} + \boldsymbol{\epsilon_Z}$$

Applying d-separation does not give us any independence relations between X, Y and Z.

Constraint-based method: PC Algorithm

- **Goal: Test every pair of adjacent variables by performing (un)conditional independence tests to remove as many spurious associations as possible.**
- **Assumptions:**
  1. There are no cyclic causal relations for example, $X \rightarrow Y$ and $Y \rightarrow X$
  2. Causal faithfulness: Dependencies found by PC algorithm are characteristic of the underlying data distribution.

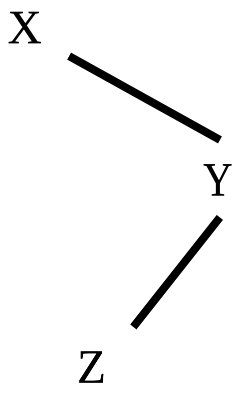


$$\boldsymbol{X = \epsilon_X}$$
$$\boldsymbol{Y = aX + \epsilon_Y}$$
$$\boldsymbol{Z = cX + bY + \epsilon_Z}$$

If

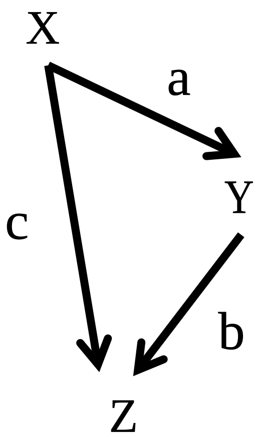
$$a.b + c = 0$$

then X is independent of Z!



Constraint-based method: PC Algorithm

- **Goal: Test every pair of adjacent variables by performing (un)conditional independence tests to remove as many spurious associations as possible.**
- **Assumptions:**
  1. There are no cyclic causal relations for example, $X \rightarrow Y$ and $Y \rightarrow X$
  2. Causal faithfulness: Dependencies found by PC algorithm are characteristic of the underlying data distribution.

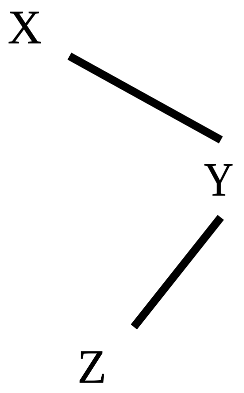


$$\boldsymbol{X = \epsilon_X}$$
$$\boldsymbol{Y = aX + \epsilon_Y}$$
$$\boldsymbol{Z = cX + bY + \epsilon_Z}$$

If

$$a.b + c = 0$$

then X is independent of Z!



“Spirtes et al. [2000, Theorem 3.2] show for linear models that this happens with probability zero if we assume that the coefficients are drawn randomly from positive densities.”

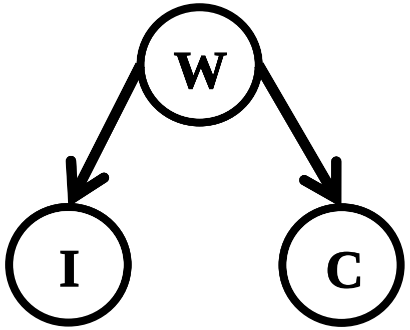
Constraint-based method: PC Algorithm

- **Goal: Test every pair of adjacent variables by performing (un)conditional independence tests to remove as many spurious associations as possible.**
- **Assumptions:**
  1. There are no cyclic causal relations
     - for example, $X \rightarrow Y$ and $Y \rightarrow X$
  2. Causal faithfulness: Dependencies found by PC algorithm are characteristic of the underlying data distribution.
  3. Causal sufficiency: Absence of hidden confounders in the data.

Constraint-based method: PC Algorithm

- **Goal: Test every pair of adjacent variables by performing (un)conditional independence tests to remove as many spurious associations as possible.**
- **Assumptions:**
  1. There are no cyclic causal relations for example, $X \rightarrow Y$ and $Y \rightarrow X$
  2. Causal faithfulness: Dependencies found by PC algorithm are characteristic of the underlying data distribution.
  3. Causal sufficiency: Absence of hidden confounders in the data.
- **Steps:**
  1. Start with a complete undirected graph.
  2. Perform a statistical test for every pair of adjacent variables to test whether an edge between them is spurious or not.
  3. Orient the remaining undirected edges.
- The output of PC algorithm is a partially directed graph which represents a Markov equivalent class of graphs.

PC Algorithm: An example

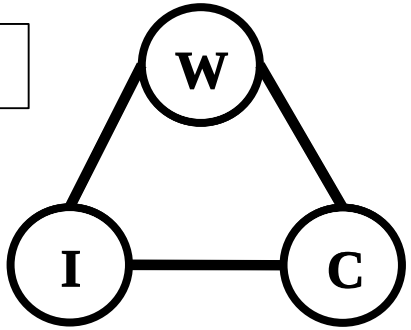


W: Weather
I: Ice-cream sales
C: Crime rate

True causal graph

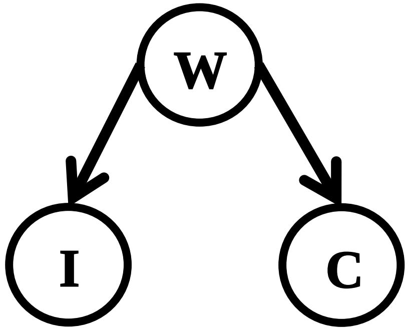
$$I \perp\!\!\!\perp C \mid W$$

Step 1



A complete undirected graph over variable set $\mathbf{X} = \{W, I, C\}$.

PC Algorithm: An example

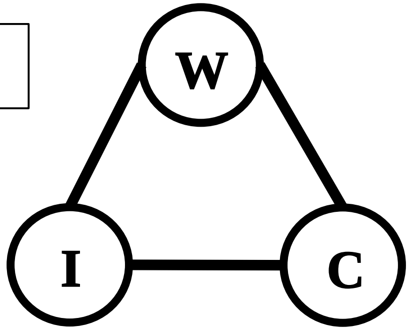


W: Weather
I: Ice-cream sales
C: Crime rate

True causal graph

$I \perp\!\!\!\perp C | W$

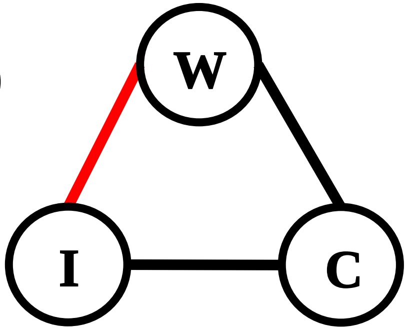
Step 1



A complete undirected graph over variable set $\mathbf{X} = \{W, I, C\}$.

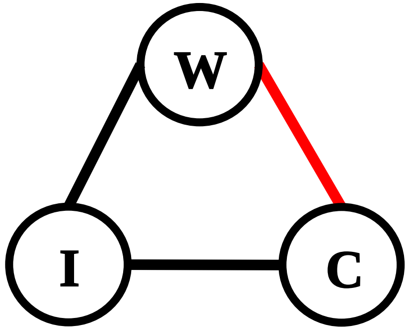
Step 2: Perform pairwise CI test with increasing conditioning set size, Z.

$|Z| = 0$



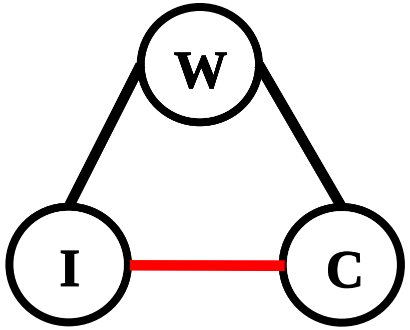
$I \perp\!\!\!\perp W$

No



$C \perp\!\!\!\perp W$

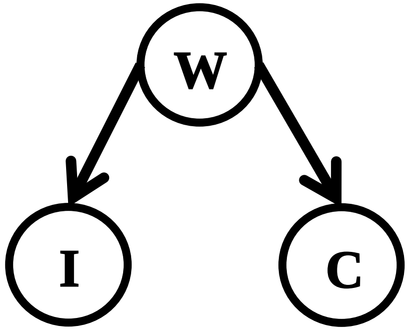
No



$I \perp\!\!\!\perp C$

No

PC Algorithm: An example

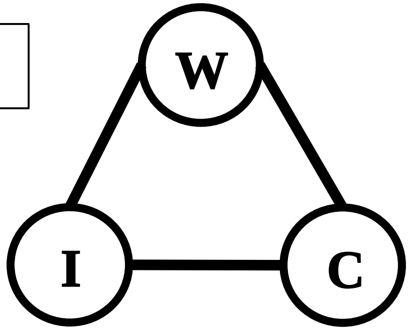


W: Weather
I: Ice-cream sales
C: Crime rate

True causal graph

$$\boxed{I \perp\!\!\!\perp C | W}$$

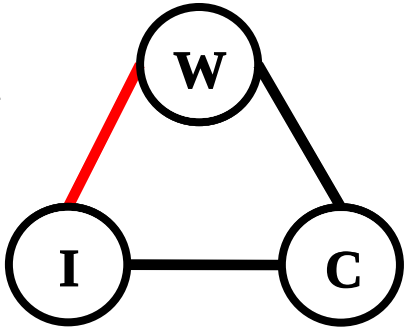
Step 1



A complete undirected graph over variable set $\mathbf{X} = \{W, I, C\}$.

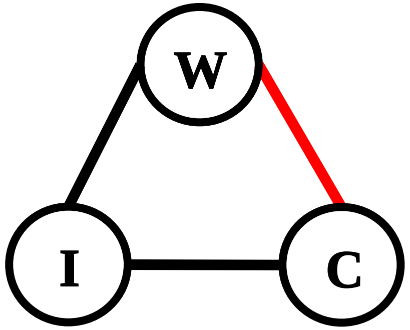
Step 2: Perform pairwise CI test with increasing conditioning set size, Z.

$|\boldsymbol{Z}| = 1$



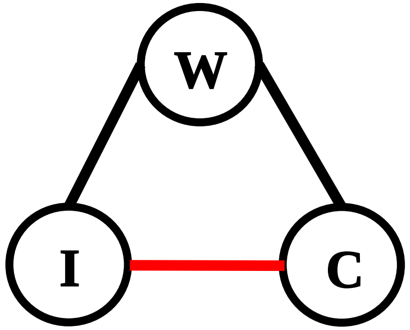
$I \perp\!\!\!\perp W | C$

No



$C \perp\!\!\!\perp W | I$

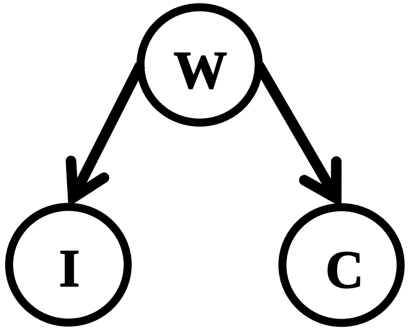
No



$$\boxed{I \perp\!\!\!\perp C | W}$$

Yes

PC Algorithm: An example

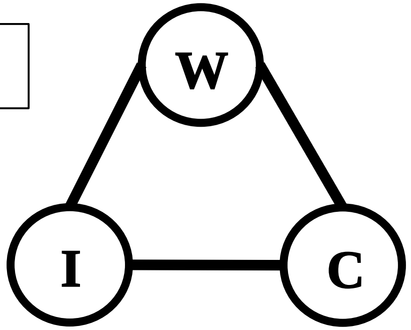


W: Weather
I: Ice-cream sales
C: Crime rate

True causal graph

$\boxed{I \perp\!\!\!\perp C \mid W}$

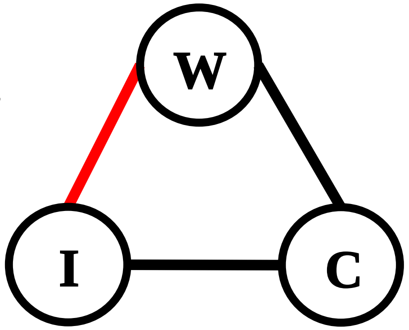
Step 1



A complete undirected graph over variable set $\mathbf{X} = \{W, I, C\}$.

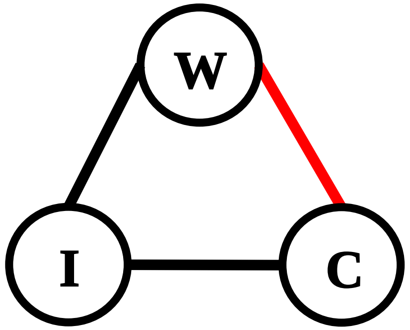
Step 2: Perform pairwise CI test with increasing conditioning set size, Z.

$|Z| = 1$



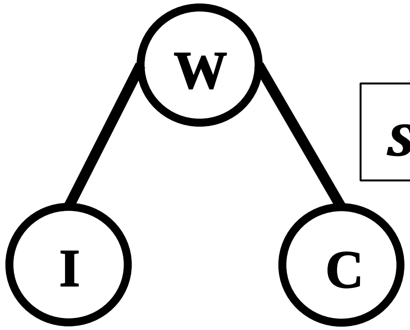
$I \perp\!\!\!\perp W \mid C$

No



$C \perp\!\!\!\perp W \mid I$

No

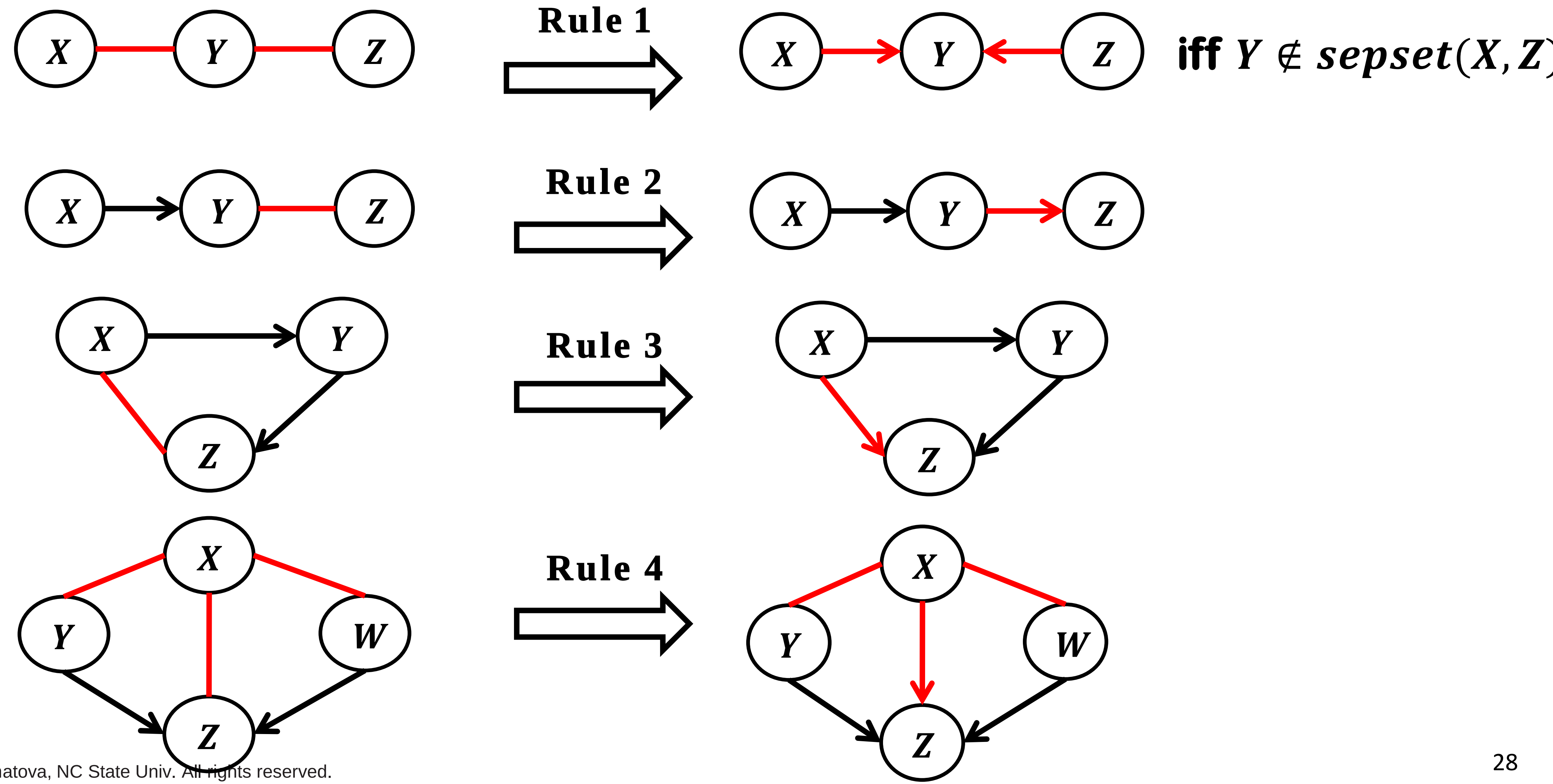


$\boxed{I \perp\!\!\!\perp C \mid W}$

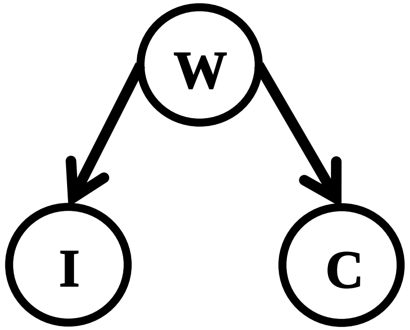
Yes

$sepset(I, C) = W$

PC algorithm: Orientation Rules



PC Algorithm: An example



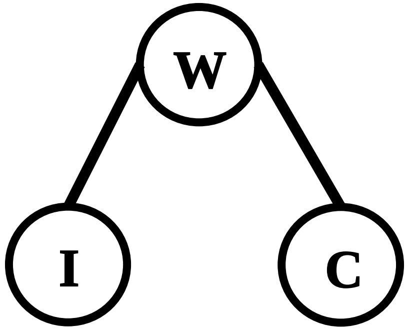
W: Weather
I: Ice-cream sales
C: Crime rate

True causal graph

$I \perp\!\!\!\perp C \mid W$

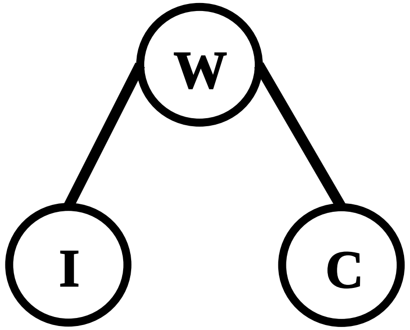
Step 2: Output

$sepset(I, C) = W$

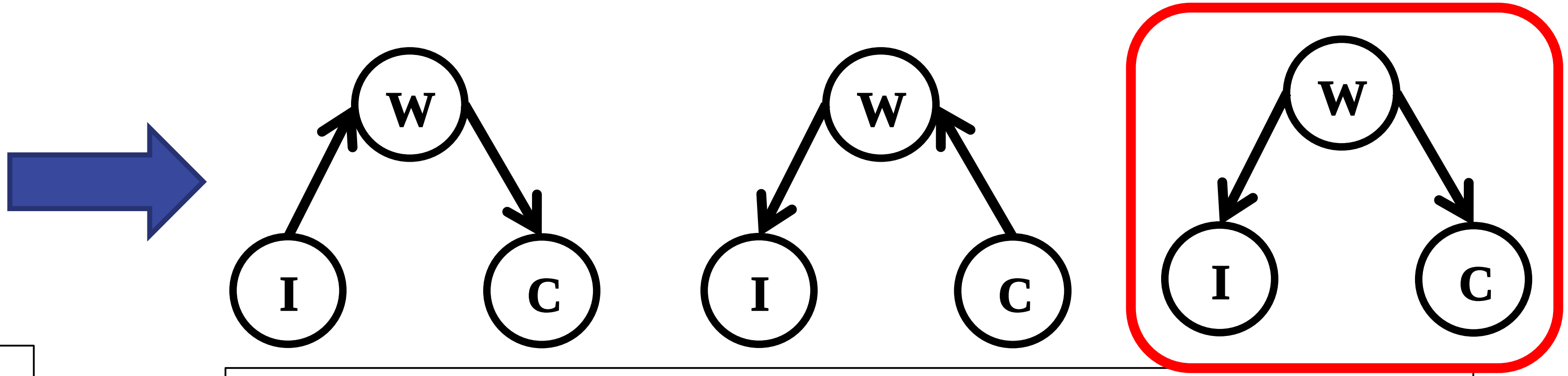


None of the orientation rules apply to this structure.

Step 3: Output



A partially directed graph.



A set of Markov equivalent graphs which encode the same conditional independence relationship.

PC Algorithm: Simulation

File: causality.R

```
# PC algorithm example
# Causal structure: I <- W -> C
W <- rnorm(1000)
I <- 4*W + rnorm(1000)
C <- 2*W + rnorm(1000)
dat_3 <- cbind(W, I, C)
pc_fit <- pc(suffStat=list(C=cor(dat_3), n=nrow(dat_3)), indepTest=gaussCItest,
alpha=0.05, labels=colnames(dat_3), verbose=TRUE)
plot(pc_fit, main="Estimated PDAG")
```

Additive Noise Model (ANM): Linear Non-Gaussian Acyclic Model (LiNGAM)

- Assuming causal sufficiency, exact causal graph can be identified if the data is generated from an additive noise model such that at most X or ϵ is Gaussian.

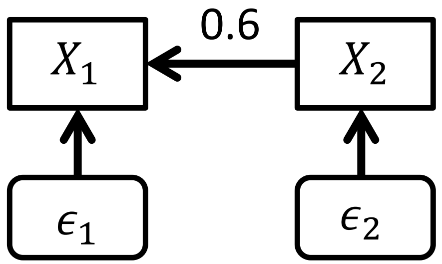
$$Y = f(X) + \epsilon$$

- Consider a variable set $X = \{X_1, X_2, .., X_n\}$ generated from a structural equation model where X is a linear function of the disturbance variables $E = \{E_1, E_2, \dots, E_n\}$, which are mutually independent and non-Gaussian.

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 0 & 0.6 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

$$X = BX + E$$

$$E = WX$$



$$\boldsymbol{X_1 = 0.6 * X_2 + \epsilon_1}$$
$$\boldsymbol{X_2 = \epsilon_2}$$

  - where $W = I - B$
- How do we estimate W? Independent Component Analysis (ICA)!

Independent Component Analysis (ICA)

- ICA estimates the mixing matrix, W, however, the rows in the matrix can be in random order.

$$\begin{bmatrix} E_2 \\ E_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & -0.6 \end{bmatrix}}_{} \begin{bmatrix} X_2 \\ X_1 \end{bmatrix} \xrightarrow{\text{Permute rows}} \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} = \begin{bmatrix} 1 & -0.6 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

The order of rows is incorrect, as E_2 corresponds to X_1 and E_1 to X_2

- An estimate of the coefficient matrix, $\bar{B}$, can now be computed using $\bar{B} = I - \bar{W}$

$$\bar{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & -0.6 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0.6 \\ 0 & 0 \end{bmatrix}$$

32

Estimation Errors in Coefficient Matrix

- However, it is possible to get an estimated coefficient matrix as follows,

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0.65 \\ -0.05 & 0 \end{bmatrix}}_{\bar{B}} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} E_1 \\ E_2 \end{bmatrix}$$

- In such cases, we need to identify which path coefficients are zero by first permuting the rows and columns of $\bar{B}$ using a permutation matrix Q, such that it minimizes the sum of elements in the upper triangular part.

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 0 & 0.65 \\ -0.05 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \Rightarrow \begin{bmatrix} X_2 \\ X_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -0.05 \\ 0.65 & 0 \end{bmatrix}}_{Q\bar{B}Q^T} \begin{bmatrix} X_2 \\ X_1 \end{bmatrix} + \begin{bmatrix} E_2 \\ E_1 \end{bmatrix}$$

- Finally, set the values in the upper triangular matrix to zero.

$$\begin{bmatrix} X_2 \\ X_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & -0.05 \\ 0.65 & 0 \end{bmatrix}}_{Q\bar{B}Q^T} \begin{bmatrix} X_2 \\ X_1 \end{bmatrix} + \begin{bmatrix} E_2 \\ E_1 \end{bmatrix} \Rightarrow \begin{bmatrix} X_2 \\ X_1 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 \\ 0.65 & 0 \end{bmatrix}}_{\tilde{B} \text{ is strictly lower triangular}} \begin{bmatrix} X_2 \\ X_1 \end{bmatrix} + \begin{bmatrix} E_2 \\ E_1 \end{bmatrix}$$

Permutation Algorithm for Large Dimensions

- Computing coefficient matrix using exhaustive search becomes computationally expensive in large dimensions.
- An alternative approach,
  1. Set the $m(m+1)/2$ smallest (absolute) values in $\widetilde{B}$ to zero where m=# of variables in the data.
  2. Test whether the resulting matrix can be permuted to a strict lower triangular matrix.
     - If yes, return the matrix as it contains the causal order of the variables.
     - If not, then set the next smallest (absolute value) element in the matrix to zero and repeat Step 2.

34

Prune the coefficient matrix

- The LiNGAM algorithm has additional statistical tests to further prune the edges in the causal structure.
- The steps are as follows,
  1. Find non-significant edges by applying Wald test at a significance level of 0.05.
  2. For all the non-significant edges identified by the Wald test.
     a. Set the least significant edge in the lower triangular matrix $\tilde{B}$ to zero.
     b. Test if the reduced model with one reduced edge fits as good as the full model.
     c. Further, perform a Chi-squared test on the current model with reduced edge.
     d. If both the hypotheses are accepted, prune the edge.
  3. Return the updated coefficient matrix $\tilde{B}$.

State-of-the-Art: Causal Discovery

Improvements to PC algorithm – PC-stable, Conservative PC, PC-Max [3,6]

Improvements to the PC algorithm were proposed to overcome: (i) variable order-dependent output and (ii) errors in orienting edges.

Causal effect estimation from CPDAGs – IDA [5]

An approach to estimate lower bound of causal effect of a gene on a phenotype was developed to prioritize randomized control experiments. Assuming linearity causal effects can be estimated using linear regression.

Divide-and-conquer based approaches– Rec, SADA, SAR, CDCD [10,1,4,2]

Divide-and-conquer based approaches perform conditional independence test only on edges that connect different graph partitions thereby reducing errors in statistical tests and improving the quality of output causal graphs.

State-of-the-Art: Causal Discovery (cont.)

Additive Noise Models (ANM) – LiNGAM, DirectLiNGAM, PNL [8,9,11,12]

Data generated from the following additive noise model could lead to exact identification of the causal graph if at most one of X or ϵ is Gaussian.

$$Y = g(f(X) + \epsilon\ )$$

Robust Conditional Independence tests – CCI, KCIT, RCIT [7,13,14]

Conditional independence tests that utilize kernel matrices and nonlinear regression to better detect independencies between variables.

Publicly Available Tools for Causal Discovery

Software/Package	Language	Link
pcalg	R	https://cran.r-project.org/package=pcalg
bnlearn	R	http://www.bnlearn.com/
Bayes Net Toolbox	MATLAB	https://github.com/bayesnet/bnt
Causal Explorer	MATLAB	http://proceedings.mlr.press/v3/supplemental/Software_WCCI08_challenge.pdf
Tetrad	Java	http://www.phil.cmu.edu/tetrad/

PC Algorithm: R Example

```
library(pcalg)
# Initialize the number of variables
rDAG <- randomDAG(10, prob=0.3, lB=0.1, uB=1)

# Visualize the generated DAG
plot(rDAG, main="A sample random DAG")
# Generate a multivariate data according to rDAG
data <- rmvDAG(1000, rDAG, errDist="normal")

# Now let's run the PC algorithm using the data
suffStat=list(C=cor(data), n=nrow(data))
pc_fit <- pc(suffStat, indepTest=gaussCItest, alpha=0.05, labels=colnames(data), verbose=TRUE)
plot(pc_fit, main="PC Output")

# Evaluate the estimated graph with the true graph
shd(rDAG , pc_fit)
compareGraphs(pc_fit@graph, rDAG)
```

LiNGAM algorithm: R Example

```
library(pcalg)

# Initialize the number of variables
x1 <- runif(1000)
x2 <- 0.8*x1 + runif(1000)
data <- cbind(x1, x2)

# Visualize the distribution
plot(data[, 1], data[, 2])

# Now let's run the LiNGAM algorithm using the data
lingam_fit <- lingam(data, verbose=TRUE)
show(lingam_fit)

pc_fit <- pc(suffStat=list(C=cor(data), n=nrow(data)), indepTest=gaussCItest, alpha=0.05, p=ncol(data))
plot(pc_fit)
```

References

1. Cai, R., Zhang, Z. and Hao, Z., 2013, February. SADA: A general framework to support robust causation discovery. In *International Conference on Machine Learning* (pp. 208-216).
2. Chaudhary, M.S., Ranshous, S. and Samatova, N.F., 2017, November. A Community-Driven Graph Partitioning Method for Constraint-Based Causal Discovery. In *International Workshop on Complex Networks and their Applications* (pp. 253-264). Springer, Cham.
3. Colombo, D. and Maathuis, M.H., 2014. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, *15*(1), pp.3741-3782.
4. Liu, H., Zhou, S., Lam, W. and Guan, J., 2017. A new hybrid method for learning bayesian networks: Separation and reunion. *Knowledge-Based Systems*, *121*, pp.185-197.
5. Maathuis, M.H., Kalisch, M. and Bühlmann, P., 2009. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, *37*(6A), pp.3133-3164.
6. Ramsey, J., 2016. Improving Accuracy and Scalability of the PC Algorithm by Maximizing P-value. *arXiv preprint arXiv:1610.00378*.
7. Ramsey, J.D., 2014. A scalable conditional independence test for nonlinear, non-gaussian data. *arXiv preprint arXiv:1401.5031*.
8. Shimizu, S., Hoyer, P.O., Hyvärinen, A. and Kerminen, A., 2006. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, *7*(Oct), pp.2003-2030.
9. Shimizu, S., Inazumi, T., Sogawa, Y., Hyvärinen, A., Kawahara, Y., Washio, T., Hoyer, P.O. and Bollen, K., 2011. DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research*, *12*(Apr), pp.1225-1248.
10. Xie, X. and Geng, Z., 2008. A recursive method for structural learning of directed acyclic graphs. *Journal of Machine Learning Research*, *9*(Mar), pp.459-483.

References

11. Zhang K, Hyvärinen A (2009) On the identifiability of the post-nonlinear causal model. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, Montreal
12. Zhang K, Hyvärinen A (2010) Distinguishing causes from effects using nonlinear acyclic causal models. In: JMLR Workshop and Conference Proceedings, Presented at NIPS 2008 workshop on causality. 6:157–164
13. Zhang K, Janzing D, Schölkopf B (2011) Kernel-based conditional independence test and application in causal discovery. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011), Barcelona
14. Zhang, H., Zhou, S., Zhang, K. and Guan, J., 2017, February. Causal Discovery Using Regression-Based Conditional Independence Tests. In *AAAI* (pp. 1250-1256).

Constraint-based method: PC Algorithm

- **Goal: Test every pair of adjacent variables by performing (un)conditional independence tests to remove as many spurious associations as possible.**
- **Assumptions:**
  1. Causal faithfulness: Dependencies found by PC algorithm are characteristic of the underlying data distribution.
  2. Causal sufficiency: Absence of hidden confounders in the data.
- **Steps:**
  1. Start with a complete undirected graph.
  2. Perform a statistical test for every pair of adjacent variables to test whether an edge between them is spurious or not.
  3. Orient the remaining undirected edges.

- **The output of PC algorithm is a partially directed graph which represents a Markov equivalent class of graphs.**