# Local PC Algorithm

Stephen Smith

University of California, Los Angeles

April 5, 2019

1. Introduction

2. Review of PC Algorithm

3. Local PC Algorithm
   - Notation
   - Motivation
   - The Algorithm
   - Visualization

4. Conclusion

# Graphs

**Definition (Graph)**

A graph $G = (V, E)$ consists of a set of nodes $V = \{1, \ldots, p\}$ and a set of edges $E \subseteq V \times V$ with ordered pairs of distinct nodes.

An edge $(i, j) \in E$ is called directed if $(j, i) \notin E$, and we use the notation $i \to j$.

The set of nodes correspond to the components of a random vector $\mathbf{X} \in \mathbb{R}^p$

The structure of a graph $G$ can be represented in an adjacency matrix $A = (a_{ij})_{p \times p}$ where

$$a_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases}$$

# Directed Acyclic Graphs

**Definition (DAG)**

A Directed Acyclic Graph (DAG) is a graph $\mathcal{G}$ where all edges are directed and there are no cycles

- A *v-structure* in a DAG $\mathcal{G}$ is an ordered triple of nodes $(i, j, k)$ such that $\mathcal{G}$ contains the directed edges $i \rightarrow k$ and $j \rightarrow k$, where $i$ and $j$ are not adjacent. We will denote such a structure by $i \rightarrow k \leftarrow j$.
- Two DAGs are *equivalent* if and only if they have the same skeleton and v-structures.

# Bayesian Network

## Definition

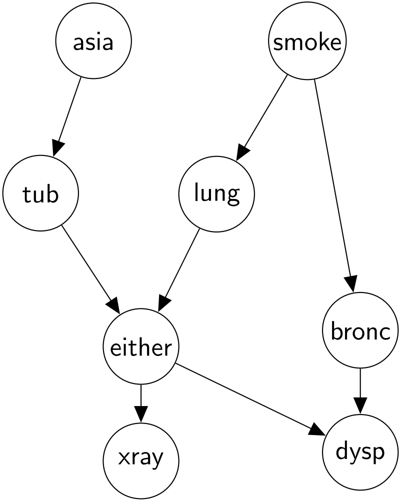
Random variables $X = (X_1, \ldots, X_p)$ are a Bayesian Network with respect to a DAG $\mathcal{G}$ if the joint probability density function $P$ can be written as

$$p(x) = \prod_{v \in V} p(x_v \mid x_{pa(v)})$$

where $pa(v) = \{v' \in V : v' \to v\}$.

# Example

This is an example of a Bayesian Network with 8 nodes (taken from the bnlearn online repository).



# Skeleton

**Definition (Skeleton)**

The skeleton of a DAG $\mathcal{G}$ ignores the direction of the edges by substituting undirected edges for directed edges

A probability distribution $P$ on $\mathbb{R}^p$ is faithful with respect to a graph $G$ if conditional independence information of the distribution may be inferred from *d-separation* in the graph $G$.

The skeleton is highly interpretable and yields insights into the dependence structure of the data, thus it is valuable to estimate.

# PDAGs and CPDAGs

## PDAG

A partially directed acyclic graph (PDAG) is a graph where some edges are directed and some are undirected, and no cycle may be traced following the directed edges and any direction for undirected edges.

## CPDAG

A CPDAG is a PDAG that is *completed*, that is, it satisfies the following:

1. Every directed edge exists in every DAG belonging to the equivalence class of the DAG
2. For every undirected edge $i - j$, there exists a DAG with $i \rightarrow j$ and a DAG with $i \leftarrow j$ in the equivalence class.

A CPDAG helps us to visualize *equivalence classes*, and estimating this graph is the primary structure learning task of the PC algorithm.

# PC Algorithm - Skeleton Estimation

**Algorithm 1** Population PC Algorithm

```
Input: Vertex set $V$, Conditional Independence Information, $\ell_{max}$
Form the complete undirected graph $\tilde{C}$ on the vertex set $V$
$\ell = -1$; $C = \tilde{C}$
while $\ell < \ell_{max}$ do
    $\ell = \ell + 1$
    do
        Select a new, adjacent pair of ordered nodes $(i, j)$ s.t. $|adj(C, i) \setminus \{j\}| \geq \ell$
        do
            Choose new set $\mathbf{k} \subseteq adj(C, i) \setminus \{j\}$ with $|\mathbf{k}| = \ell$
            if $i$ and $j$ are C.I. given $\mathbf{k}$ then
                Delete edge $i, j$; Save $\mathbf{k}$ in $S(i, j), S(j, i)$
            end if
        while $C_{ij} = 1$ or there are new $\mathbf{k}$ to be chosen
    while there remains ordered pairs with appropriately sized $\mathbf{k}$ to be tested for C.I.
end while
Output: estimated skeleton $C$ and separation sets $S$
```

# Population PC Algorithm - CPDAG Estimation

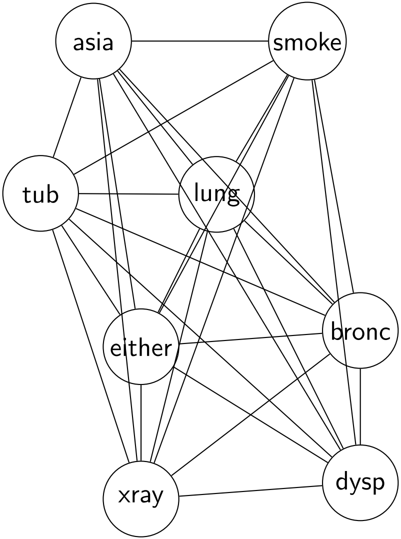
Once we have the skeleton and the separation sets, we can orient some of the edges by finding v-structures and using Meek's rules.

**Algorithm 2** V-Structure Orientation Algorithm

```
Input: Skeleton $G$, Separation sets $S$
for pairs of nonadjacent $i, j$ with common neighbor $k$ do
    if $k \notin S(i, j)$ then
        Replace $i - k - j$ in $G$ with by $i \rightarrow k \leftarrow j$
    end if
end for
Use Meek's Rules for edge orientation
Output: CPDAG $G$
```

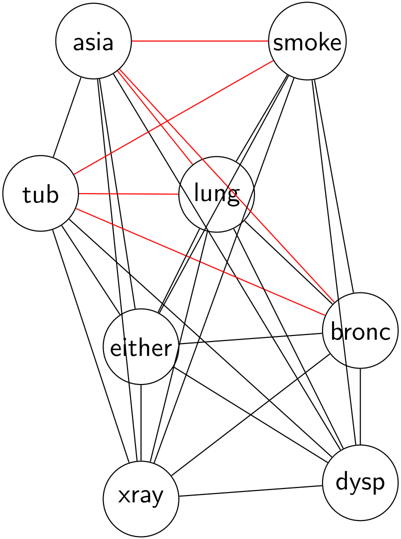
# Visual Demonstration of PC Algorithm

We start with a complete graph $\tilde{C}$



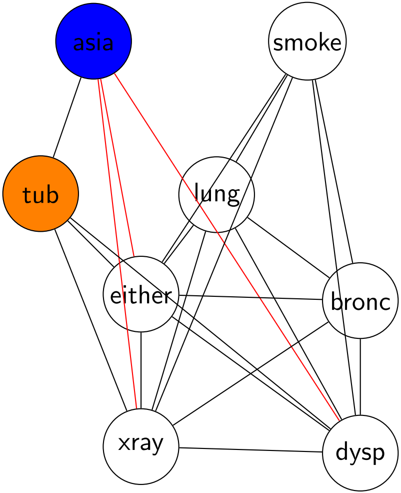
# Visual Demonstration of PC Algorithm

With $\ell = 0$, we delete edges of $C$ that have insignificant correlations.



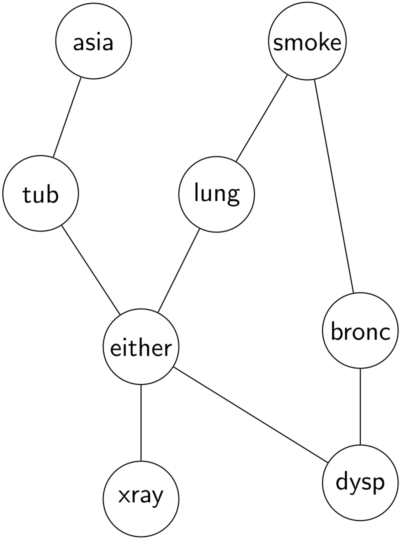
# Visual Demonstration of PC Algorithm

With $\ell = 1$, we delete edges $(i, j)$ with insignificant *partial* correlations given a separation set $S_{ij}$.



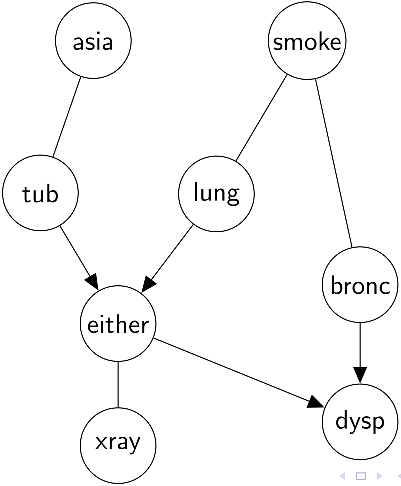
# Visual Demonstration of PC Algorithm

After completing the algorithm for estimating the skeleton, we obtain



# Visual Demonstration of PC Algorithm

Recall that $S_{tub,lung} = \emptyset$, which means we can introduce a v-structure $\text{tub} \rightarrow \text{either} \leftarrow \text{lung}$. We may also introduce the v-structure $\text{either} \rightarrow \text{dysp} \leftarrow \text{bronc}$.



# Notation

- Let $X$ be a node in a DAG $\mathcal{G}$ (denoted as $X \in \mathcal{G}_V$). We denote the parent set of $X$ as $Z = pa(X)$
- Our goal is to estimate the causal effect of $X$ on another node $Y$, i.e. $p(y \mid do(X = x))$, which can be calculated by

$$\begin{aligned} p(y \mid do(X = x)) &= \int_z p(y \mid do(x), z) p(z \mid do(x)) dz \\ &= \int_z p(y \mid x, z) p(z) dz \quad \text{Pearl (2000)} \end{aligned}$$

The last statement only requires conditional distributions that may be estimated from observational data.

# Notation

- For a linear Structural Equation Model (SEM), the causal effect of $X$ on $Y$ is defined by

$$\begin{aligned}\frac{\partial}{\partial x}\mathbb{E}(Y \mid do(X=x)) &= \frac{\partial}{\partial x}\mathbb{E}(\beta x + \gamma^\top Z \mid do(X=x)) \\ &= \frac{\partial}{\partial x}\left\{\beta x + \gamma^\top \mathbb{E}(Z)\right\} \\ &= \beta\end{aligned}$$

  where $(\beta, \gamma)$ is the regression coefficient of $Y$ on $(X, Z)$ (i.e. $\mathbb{E}(Y \mid X, Z) = \beta X + \gamma^\top Z$.

- To estimate the causal effect, we only need to learn the parent set of $X$.

# Motivation

- Since we only need a node's parent set to estimate the causal effect, if we could estimate the local structure around a node, and use this result to estimate causal effects with reduced computational complexity
- Nandy, Maathuis, and Richardson have found an algorithm to estimate causal effects with many *potential* parent sets. It would improve the efficiency of their algorithm to minimize the amount of potential parent sets using a local structure learning algorithm.

# Idea

In order to perform this estimation, we would need to be given the neighbors of $X$, denoted as $N$, where $N = N_1 \cup N_2$, and

$$N_1 = pa(X) \cup ch(X) \cup sp(X)$$

is the set of first-order neighbors, with

- $ch(X) = \{x' \in \mathcal{G}_V : X \rightarrow x'\}$
- $sp(X) = \{X' \in \mathcal{G}_V : ch(X) \cap ch(X') \neq \emptyset\}$
- For a node $v \in \mathcal{G}_V$, let $N_1(v)$ denote the first-order neighbors of $v$

$N_2$ is the set of second-order neighbors, which we define as

$$N_2 = \bigcup_{n \in N_1} N_1(n)$$

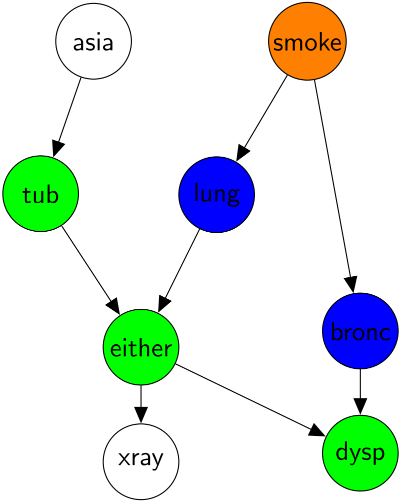
That is, $N_2$ is the union of the first-order neighbors for each of the first-order neighbors of the node of interest.

# Idea

- Given the neighbors for our node of interest, we would proceed to use the PC algorithm to estimate the local structure.
- It may also be of interest to have a set of target nodes to consider. In this case, when proceeding through the PC Algorithm, we would *not* have a complete graph on all the nodes being considered. Rather, we would only create complete graphs for each neighborhood.

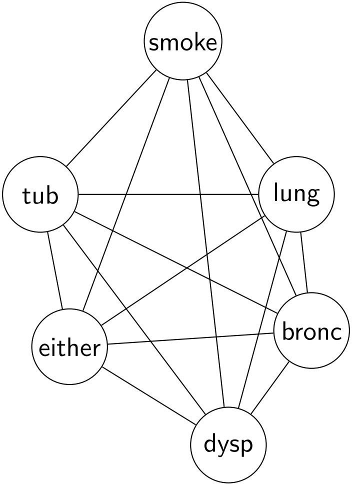
# Visualizing the Local PC Algorithm

Suppose the variable of interest is smoke.



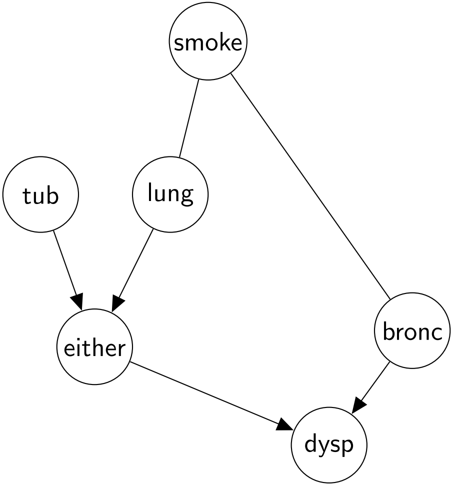
# Visualizing the Local PC Algorithm

We start with a complete graph, only considering the neighborhood of smoke



# Visualizing the Local PC Algorithm

After completing the PC algorithm on this local set of nodes, we obtain



# Future Research

- Find an efficient way to estimate the neighborhood of a node
- Compare the population version of the Local PC algorithm to that of the PC algorithm to see how accurate we are with only a few target nodes
- Use generated data and compare the accuracy between the normal PC algorithm and the local algorithm. We expect to make some gains in this area because of fewer significance tests

# References

1. Kalisch, M. and Bühlmann, P. (2007), "Estimating high-dimensional directed acyclic graphs with the PC-algorithm," *Journal of Machine Learning Research*.
2. Nandy, P., Maathuis, M. H., and Richardson T.S. (2017), "Estimating the Effect of Joint Interventions From Observational Data in Sparse High-Dimensional Settings," *Annals of Statistics*.

# Questions

Thank you