Non-Standard Databases and Data Mining

Introduction to Causal Modeling and Reasoning

Dr. Özgür Özçep
Universität zu Lübeck
Institut für Informationssysteme

Presenter: Prof. Dr. Ralf Möller
Structural Causal Models

Slides prepared by Özgür Özçep

Part I: Basic Notions
(SCMs, d-separation)
Literature


• J. Pearl: Causality, CUP, 2000. *(The book on causality from the perspective of probabilistic graphical models)*

Motivation

- Usual warning:
  
  "Correlation is not causation"

- Bulk of data mining methods is about correlation

- But sometimes (if not very often) one needs causation to understand statistical data
A remarkable correlation? A simple causality!
Simpson’s Paradox (Example)

- Record recovery rates of 700 patients given access to a drug

<table>
<thead>
<tr>
<th></th>
<th>Recovery rate with drug</th>
<th>Recovery rate without drug</th>
</tr>
</thead>
<tbody>
<tr>
<td>Men</td>
<td>81/87 (93%)</td>
<td>234/270 (87%)</td>
</tr>
<tr>
<td>Women</td>
<td>192/263 (73%)</td>
<td>55/80 (69%)</td>
</tr>
<tr>
<td>Combined</td>
<td>273/350 (78%)</td>
<td>289/350 (83%)</td>
</tr>
</tbody>
</table>

- Paradox:
  - For men, taking the drug has benefit
  - For women, taking the drug has benefit, too.
  - But: for all persons taking the drug seems to have no benefit
Resolving the Paradox (Informally)

• We need to understand the causal mechanisms that lead to the data in order to resolve the paradox.

• In drug example
  – Why has taking the drug less benefit for women?
    Answer: Estrogen has negative effect on recovery
  – Data: Women more likely to take drug than men
  – So: Choosing randomly any person will rather give a woman – and for these, recovery is less beneficial

• In this case: Need to consider segregated data (not aggregated data)
Resolving the Paradox Formally (Look Ahead)

- We need to **understand the causal mechanisms** that lead to the data in order to resolve the paradox.

  ![Diagram]

  - Drug usage and recovery have common cause.
  - Gender is a confounder.
Simpson Paradox (Again)

- Record recovery rates of 700 patients given access to a drug w.r.t. blood pressure (BP) segregation

<table>
<thead>
<tr>
<th>BP Level</th>
<th>Recovery rate with drug</th>
<th>Recovery rate without drug</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low BP</td>
<td>234/270 (87%)</td>
<td>81/87 (93%)</td>
</tr>
<tr>
<td>High BP</td>
<td>55/80 (69%)</td>
<td>192/263 (73%)</td>
</tr>
<tr>
<td>Combined</td>
<td>289/350 (83%)</td>
<td>273/350 (78%)</td>
</tr>
</tbody>
</table>

- BP recorded at end of experiment
- This time segregated data recommends **not** using drug whereas aggregated does
Resolving the Paradox (Informally)

• We need to understand the causal mechanisms that lead to the data in order to resolve the paradox

• In this example
  – Drug effect: lowering blood pressure (but may have toxic effects)
  – Hence: In aggregated population drug usage recommended
    • In segregated data one sees only toxic effects
Resolving the Paradox Formally (Look Ahead)

- We need to **understand the causal mechanisms** that lead to the data in order to resolve the paradox.

![Diagram showing the relationship between blood pressure, drug usage, and recovery.](attachment:diagram.png)
Ingredients of a Statistical Theory of Causality

- Working definition of causation
- Method for creating causal models
- Method for linking causal models with features of data
- Method for reasoning over model and data
**Working Definition**

A (random) variable $X$ is a **cause** of a (random) variable $Y$ if $Y$ - in any way - relies on $X$ for its value.
A structural causal model (SCM) consists of

- A set $U$ of exogenous variables
- A set $V$ of endogenous variables
- A set $F$ of functions assigning each variable in $V$ a value based on values of other variables from $V \cup U$

- Only endogenous variables $V$ are those that are descendants of other variables
- Exogenous variables $U$ are roots of model.
- Value instantiations of exogenous variables completely determine values of all variables in SCM
Causality in SCMs

Definition

1. $X$ is a **direct cause** of $Y$ iff $Y = f(\ldots,X,\ldots)$ for some $f$.
2. $X$ is a **cause** of $Y$ iff it is a direct cause of $Y$ or there is $Z$ s.t. $X$ is a direct cause of $Z$ and $Z$ is a cause of $Y$. 
Graphical Causal Model

- **Graphical causal model** associated with SCM
  - Nodes = variables
  - Edges = from $A$ to $B$ if $B = f(\ldots, A, \ldots)$

- Example SCM
  - $U = \{X, Y\}$
  - $V = \{Z\}$
  - $F = \{f_Z\}$
  - $f_Z : Z = 2X + 3Y$

- Associated graph

($Z =$ salary, $X =$ years of experience, $Y =$ years of profession)
Graphical Models

- Graphical models capture SCMs only partially

- But they are very intuitive and still allow for conserving much of the causal information of an SCM

- **Convention**: Consider only Directed Acyclic Graphs (DAGs)
SCMs and Probabilities

- Consider SCMs where all variables are random variables (RVs)

- Full specification of functions $f$ not always possible

- Instead: Use conditional probabilities as in BNs
  - $f_X(\ldots Y \ldots)$ becomes $P(X|\ldots Y \ldots)$
  - Technically: Non-measurable RVs $U$ model (probabilistic) indeterminism:
    \[ P(X|\ldots Y \ldots) = f_X(\ldots Y \ldots, U) \]

U not mentioned here
SCMs and Probabilities

- Product rule as in BNs used for full specification of joint distribution of all RVs $X_1, \ldots, X_n$

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{1 \leq i \leq n} P(x_i | \text{parents}(x_i))$$

- Can make same considerations on (probabilistic) (in)dependence of RVs

- Will be done in the following systematically
Bayesian Networks vs. SCMs

- **BNs** model statistical (in)dependencies
  - Directed, but not necessarily cause-relation
  - Inherently statistical
  - Very often used for RVs with discrete domains

- **SCMs** model causal relations
  - SCMs with random variables (RVs) induce BNs
  - Assumption: There is hidden causal (deterministic) structure behind statistical data
  - More expressive than BNs: Every BN can be modeled by SCMs but not vice versa
  - Default application: continuous variables
Reminder: Conditional Independence

• Event \( A \) independent of event \( B \) iff \( P(A \mid B) = P(A) \)

• RV \( X \) is independent of RV \( Y \) iff
  
  \[ P(X \mid Y) = P(X) \quad \text{iff} \quad \text{for every } x\text{-value of } X \text{ and for every } y\text{-value of } Y \text{ event } X = x \text{ is independent of event } Y = y \]

  Notation: \( (X \perp Y)_p \) or even shorter: \( (X \perp Y) \)

• \( X \) is conditionally independent of \( Y \) given \( Z \)
  
  iff \( P(X \mid Y, Z) = P(X \mid Z) \)

  Notation: \( (X \perp Y \mid Z)_p \) or even shorter: \( (X \perp Y \mid Z) \)
Independence in SCM graphs

- Almost all interesting independences of RVs in an SCM can be identified in its associated graph

- Relevant graph theoretical notion: d-separation

  Property
  \( X \) is independent of \( Y \) (conditioned on \( Z \)) iff
  \( X \) is d-separated from \( Y \) (by \( Z \))

- D-separation in turn rests on 3 basic graph patterns
  - Chains
  - Forks
  - Colliders

  We will develop a syntactic d-separation criterion that can be checked algorithmically
Independence in SCM graphs

**Property**

\( X \) is independent of \( Y \) (conditioned on \( Z \)) iff

\( X \) is d-separated from \( Y \) by \( Z \)

There are two conditions here due to “iff”:

- **Markov condition:**
  
  If \( X \) is d-separated from \( Y \) (by \( Z \))
  
  then \( X \) is independent of \( Y \) (conditioned on \( Z \))

- **Faithfulness:**
  
  If \( X \) is independent of \( Y \) (conditioned on \( Z \))
  
  then \( X \) is d-separated from \( Y \) (by \( Z \))
Chains

Example (SCM 1)

( \( X = \) school funding of high school, \( Y = \) its average satisfaction score, \( Z = \) average college acceptance )

- \( V = \{X, Y, Z\} \)
- \( U = \{U_X, U_Y, U_Z\} \)
- \( F = \{f_X, f_Y, f_Z\} \)

- \( f_X: X = U_X \)
- \( f_Y: Y = x/3 + U_Y \)
- \( f_Z: Z = y/16 + U_Z \)
Chains

Example (SCM 2)

( X = switch, Y = circuit, Z = light bulb )

- $V = \{X,Y,Z\}$
- $U = \{U_X,U_Y,U_Z\}$
- $F = \{f_X,f_Y,f_Z\}$
- $f_X : X = U_X$
- $f_Y : Y = \begin{cases} 
\text{closed} & \text{if } (X = \text{up} \land U_Y = 0) \text{ or } (X = \text{down} \land U_Y = 1) \\
\text{open} & \text{otherwise}
\end{cases}$
- $f_Z : Z = \begin{cases} 
\text{on} & \text{if } (Y = \text{closed} \land U_Z = 0) \text{ or } (Y = \text{open} \land U_Z = 1) \\
\text{off} & \text{otherwise}
\end{cases}$
**Example (SCM 3)**

(X = work hours, Y = training, Z = race time)

- \( V = \{X,Y,Z\} \quad U = \{U_X,U_Y,U_Z\} \quad F = \{f_X,f_Y,f_Z\} \)
- \( f_X: X = U_X \)
- \( f_Y: Y = 84 - x + U_Y \)
- \( f_Z: Z = \frac{100}{y} + U_Z \)
(In)Dependences in Chains

- **Z and Y** are likely dependent
  (For some $z,y$: $P(Z=z \mid Y = y) \neq P(Z = z)$)
- **Y and X** are likely dependent
  (…)
- **Z and X** are likely dependent
- **Z and X** are independent, conditioned on **Y**
  (For all $x,z,y$: $P(Z=z \mid X=x,Y = y) = P(Z = z \mid Y = y)$)
Dependence not Transitive

**Example (SCM 4)**

\[ V = \{X,Y,Z\} \quad U = \{U_X,U_Y,U_Z\} \quad F = \{ f_X,f_Y,f_Z \} \]

- \( f_X: X = U_X \)
- \( f_Y: Y = \begin{cases} a & \text{if } X = 1 \text{ & } U_Y = 1 \\ b & \text{if } X = 2 \text{ & } U_Y = 1 \\ c & \text{if } U_Y = 2 \end{cases} \)
- \( f_Z: Z = \begin{cases} i & \text{if } Y = c \text{ or } U_Z = 1 \\ j & \text{if } Y \neq c \text{ & } U_Z = 2 \end{cases} \)

- \( Y \) depends on \( X \), \( Z \) depends on \( Y \) but \( Z \) does not depend on \( X \)
- “Variable level” graph hides *independence*
Rule 1 (Conditional Independence in Chains)

Variables $X$ and $Z$ are independent given set of variables $Y$ iff there is only one path between $X$ and $Z$ and this path is unidirectional and $Y$ intercepts that path.
**Example (SCM 5)**

(X = Temperature, Y = Ice cream sale, Z = Crime)

- V = \{X,Y,Z\}  
  U = \{U_X,U_Y,U_Z\}  
  F = \{f_X,f_Y,f_Z\}

- f_X: X = U_X
- f_Y: Y = 4x + U_y
- f_Z: Z = x/10 + U_Z
Example (SCM 5)

\( X = \text{switch}, \ Y = \text{light bulb 1}, \ Z = \text{light bulb 2} \)

- \( V = \{X,Y,Z\} \)
- \( U = \{U_X,U_Y,U_Z\} \)
- \( F = \{f_X,f_Y,f_Z\} \)

- \( f_X: X = U_X \)

- \( f_Y: Y = \begin{cases} 
\text{on} & \text{if } (X = \text{up} & U_Y = 0) \text{ or } (X = \text{down} & U_Y = 1) \\
\text{off} & \text{otherwise}
\end{cases} \)

- \( f_Z: Z = \begin{cases} 
\text{on} & \text{if } (X = \text{up} & U_Z = 0) \text{ or } (X = \text{down} & U_Z = 1) \\
\text{off} & \text{otherwise}
\end{cases} \)
(In)Dependences in Forks

- X and Z are likely dependent
  \(\exists x,z: P(X=x \mid Z = z) \neq P(X = x)\)
- X and Y are likely dependent
  ...
- Z and Y are likely dependent
- Y and Z are independent, conditional on X
  \(\forall x,y,z: P(Y=y \mid Z=z,X = x) = P(Y = y \mid X = x)\)
Independence Rule in Forks

**Rule 2** (Conditional Independence in Forks)

If variable $X$ is a common cause of variables $Y$ and $Z$ and there is only one path between $Y, Z$, then $Y$ and $Z$ are independent conditional on $X$. 

![Diagram showing a fork with variables $X$, $Y$, and $Z$ and their common cause $U$.]
Example (SCM 6)

( X = musical talent, Y = grade point, Z = scholarship)

- V = {X,Y,Z} \hspace{1cm} U = \{U_X, U_Y, U_Z\} \hspace{1cm} F = \{f_X, f_Y, f_Z\}
- f_X: X = U_X
- f_Y: Y = U_Y
- f_Z: \{ yes \hspace{1cm} if \hspace{1cm} X = \text{yes} \hspace{1cm} or \hspace{1cm} Y > 80% \\
  \hspace{1cm} no \hspace{1cm} otherwise \}

U_X \hspace{1cm} U_Y

X

U_Z

Y

Z
(In)dependence in Colliders

- **X** and **Z** are likely dependent
  \( \exists z,y: P(X=x \mid Z = z) \neq P(X = x) \)

- **Y** and **Z** are likely dependent

- **X** and **Y** are independent

- **X** and **Y** are likely dependent, conditional on **Z**
  \( \exists x,z,y: P(X= x \mid Y=y,Z = z) \neq P(X = x \mid Z = z) \)

If scholarship received (**Z**) but low grade (**Y**), then must be musically talented (**X**)

**X-Y dependence** (conditional on **Z**) is statistical but not causal
(In)dependence in Colliders (Extended)

Example (SCM 7)

(X = coin flip, Y = second coin flip, Z = bell rings, W = bell witness)

- V = \{X,Y,Z,W\}  
  U = \{U_X, U_Y, U_Z, U_W\}  
  F = \{f_X, f_Y, f_Z, f_W\}

- f_x: X = U_X
- f_y: Y = U_Y

- f_z: Z =
  \[
  \begin{cases} 
  \text{yes} & \text{if } X = \text{head} \text{ or } Y = \text{head} \\
  \text{no} & \text{otherwise}
  \end{cases}
  \]

- f_w: W =
  \[
  \begin{cases} 
  \text{yes} & \text{if } Z = \text{yes} \text{ or } (Z = \text{no} \text{ and } U_W = \frac{1}{2}) \\
  \text{no} & \text{otherwise}
  \end{cases}
  \]

X and Y are dependent conditional on Z and on W.
Independence Rule in Colliders

**Rule 3** (Conditional Independence in Colliders)

If a variable $Z$ is the collision node between variables $X$ and $Y$ and there is only one path between $X$, $Y$, then $X$ and $Y$ are unconditionally independent, but are dependent conditional on $Z$ and any descendant of $Z$.
**Recap: Property**

X independent of Y (conditional on Z) w.r.t. a probability distribution iff

X d-separated from Y (by Z) in graph

**Definition (informal)**

X is d-separated from Y by Z iff

Z blocks every possible path between X and Y

- Z (possibly a set of variables) prohibits the "flow" of statistical effects/dependence between X and Y
  - Must block every path
  - Need only one blocking variable for each path

**Pipeline metaphor**
Blocking Conditions

**Definition (formal)**

A path \( p \) in \( G \) (between \( X \) and \( Y \)) is **blocked by** \( Z \) iff

1. \( p \) contains chain \( A \rightarrow B \rightarrow C \) or fork \( A \leftarrow B \rightarrow C \) s.t. \( B \in Z \) or
2. \( p \) contains collider \( A \rightarrow B \leftarrow C \) s.t. \( B \notin Z \) and all descendants of \( B \) are \( \notin Z \)

If \( Z \) blocks every path between \( X \) and \( Y \), then \( X \) and \( Y \) are **\( d \)-separated conditional on** \( Z \), for short: \((X \perp Y \mid Z)_G\)

In particular: \( X \) and \( Y \) are unconditionally independent iff all \( X-Y \) paths contain colliders.
Example 1 (d-separation)

- Unconditional relation between $Z$ and $Y$?
  - D-separated because of collider on single $Z$-$Y$ path.
  - Hence unconditionally independent
Example 1 (d-separation)

- Relation between $Z$ and $Y$ conditional on $\{W\}$?
  - Not d-separated
    - because fork $X \notin \{W\}$
    - and collider $\in \{W\}$
  - Hence conditionally dependent on $\{W\}$ (and $\{T\}$)
Example 1 (d-separation)

- Relation between $Z$ and $Y$ conditional on $\{W,X\}$?
  - d-separated
    - Because fork $X$ blocks
  - Hence conditionally independent on $\{W,X\}$
Example 2 (d-separation)

- Relation between $Z$ and $Y$?
  - Not d-separated because second path not blocked (no collider)
  - Hence not unconditionally independent
Example 2 (d-separation)

- Relation between $Z$ and $Y$ conditional on $\{R\}$?
  - d-separated by $\{R\}$ because
    - First path blocked by fork $R$
    - Second path blocked by collider $W \notin \{R\}$
  - Hence independent conditional on $\{R\}$
Example 2 (d-separation)

• Relation between $Z$ and $Y$ conditional on $\{R,W\}$?
  – Not d-separated by $\{R,W\}$ because $W$ unblocks second path
  – Hence not independent conditional on $\{R,W\}$
Example 2 (d-separation)

- Relation between $Z$ and $Y$ conditional on $\{R,W,X\}$?
  - d-separated by $\{R,W,X\}$ because
    - Now second path blocked by fork $X$
  - Hence independent conditional on $\{R,W,X\}$
Using D-separation

- Verifying/falsifying causal models on observational data
  1. $G = SCM$ to test for
  2. Calculate independencies $I_G$ entailed by $G$ using d-separation
  3. Calculate independencies $I_D$ from data (by counting and estimating probabilities) and compare with $I_G$
  4. If $I_G = I_D$, SCM is a good solution. Otherwise identify problematic $I \in I_G$ and change $G$ locally to fit corresponding $I' \in I_D$
Using D-separation

• This approach is local
  – If $I_G$ not equal $I_D$, then can manipulate $G$ w.r.t. RVs only involved in incompatibility
  – Usually seen as benefit w.r.t. global approaches via likelihood with scores, say

• Approach is qualitative and constraint-based

• Known algorithms:
  – PC (Peter Spirtes & Clark Glymour)
  – IC (Verma & Pearl)
Equivalent Graphs

• One learns graphs that are (observationally) equivalent w.r.t. entailed independence assumptions

• Formalization
  
  – $v(G) = v$-structure of $G$ = set of colliders in $G$ of form $A \rightarrow B \leftarrow C$ where $A$ and $C$ not adjacent
  
  – $sk(G) =$ skeleton of $G$ = undirected graph resulting from $G$

**Definition**

$G_1$ is equivalent to $G_2$ iff $v(G_1) = v(G_2)$ and $sk(G_1) = sk(G_2)$
Equivalent Graphs

**Theorem**
Equivalent graphs entail same set of d-separations

Proof sketch:
- Forks and chains have similar role w.r.t. independence
  (Hence forgetting about the direction in skeleton does not lead to loss of information)
- Collider has different role (hence need v-structure)
Equivalent Graphs

- $v(G) = v$-structure of $G = \text{set of colliders in } G \text{ of form } A \rightarrow B \leftarrow C \text{ where } A \text{ and } C \text{ not adjacent}$
- $sk(G) = \text{skeleton of } G = \text{undirected graph resulting from } G$

**Definition**

$G_1$ is equivalent to $G_2$ iff $v(G_1) = v(G_2)$ and $sk(G_1) = sk(G_2)$
Equivalent Graphs

- $v(G) = v$-structure of $G = \text{set of colliders in } G \text{ of form } A \rightarrow B \leftarrow C \text{ where } A \text{ and } C \text{ not adjacent}$
- $sk(G) = \text{skeleton of } G = \text{undirected graph resulting from } G$

**Definition**

$G_1$ is equivalent to $G_2$ iff $v(G_1) = v(G_2)$ and $sk(G_1) = sk(G_2)$

- $v(G) \neq v(G')$
- $sk(G) = sk(G')$
- Hence not equivalent
IC-Algorithm (Verma & Pearl, 1990)

**Input**
- $P$ resp.
- $P$-independencies

- $(C \perp A \mid B)$
- $(C \perp D \mid B)$
- $(D \perp A \mid B)$
- $(E \perp A \mid B)$
- $(E \perp B \mid C,D)$

**Output**
- Pattern
  - (represents compatible class of equivalent DAGs)

**Algorithm**
- Steps 1-3

**Definition**

*Pattern* = partially directed DAG
  = DAG with directed and non-directed edges

**Directed edge** $A \rightarrow B$ in pattern: In any of the DAGs the edge is $A \rightarrow B$

**Undirected edge** $A-B$ in pattern: There exists (equivalent) DAGs with $A \rightarrow B$ in one and $B \rightarrow A$ in the other

IC-Algorithm (Informally)

1. Find all pairs of variables that are dependent of each other (applying standard statistical methods on the database) and eliminate indirect dependencies

2. + 3. Determine directions of dependencies
Note: „Possible“ in step 3 means: if you can find two patterns such that in the first the edge A-B becomes A->B but in the other A<-B, then do not orient.

IC-Algorithm (schema)

1. Add (undirected) edge $A-B$ iff there is no set of RVs $Z$ such that $(A \perp B | Z)_P$. Otherwise let $Z_{AB}$ denote some set $Z$ with $(A \perp B | Z)_P$.

2. If $A \rightarrow B \rightarrow C$ and not $A \leftarrow C$, then $A \rightarrow B \leftarrow C$ iff $B \notin Z_{AC}$

3. Orient as many of the undirected edges as possible, under the following constraints:
   - Orientation should not create a new v-structure and
   - Orientation should not create a directed cycle.

Steps 1 and step 3 leave out details of search
- Hierarchical refinement of step 1 gives PC algorithm (next slide)
- A refinement of step 3 possible with 4 rules (thereafter)
PC algorithm (Spirtes & Glymour, 1991)

- Remember Step 1 of IC
  1. Add (undirected) edge $A-B$ iff there is no set of RVs $Z$ such that $(A \perp B | Z)_P$. Otherwise let $Z_{AB}$ denote some set $Z$ with $(A \perp B | Z)_P$.

- Have to search all possible sets $Z$ of RVs for given nodes $A, B$
  - Done systematically by sets of cardinality 0, 1, 2, 3…
  - Remove edges from graph as soon as independence found
  - Polynomial time for graphs of finite degree (because can restricted search for $Z$ to nodes adjacent to $A, B$)

IC-Algorithm (with rule-specified last step)

1. as before
2. as before
3. Orient undirected edges as follows
   - B → C into B→C if there is an arrow A→B s.t. A and C are not adjacent;
   - A → B into A→B if there is a chain A→C→B;
   - A → B into A→B if there are two chains A→C→B and A→D→B such that C and D are nonadjacent;
   - A → B into A→B if there are two chains A→C→D and C→D→B s.t. C and B are nonadjacent;
Theorem

The 4 rules specified in step 3 of the IC algorithm are necessary (Verma & Pearl, 1992) and sufficient (Meek, 95) for getting a maximally oriented DAG compatible with the input-independencies.


Stable Distribution

• The IC algorithm accepts stable distributions $P$ (over set of variables) as input, i.e., distribution $P$ s.t. there is DAG $G$ giving exactly the $P$-independencies.

• Extension IC* works also for sampled distributions generated by so-called latent structures:
  – A latent structure (LS) additionally specifies a (subset) of observation variables for a causal structure.
  – A LS not determined by independencies.
  – For IC* please refer to, e.g.,
Definition
The problem of ignorance denotes the fact that there are RVs A, B and sets of RVs Z such that it is not known whether \((A \perp B|Z)_p\) or not \((A \perp B|Z)_p\).

- Problem of ignorance ubiquitous in science practice
- IC faces the problem of ignorance (Leuridan 2009)
- (Leuridan 2009) approaches this with adaptive logic
  - An adaptive logic supposes that all formulas behave normally unless and until proven otherwise.