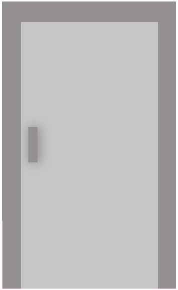
Robust Causal Domain Adaptation in a Simple Diagnostic Setting

Thijs van Ommen

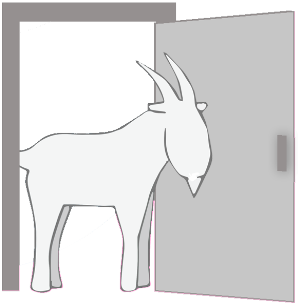
Utrecht University

Ghent, July 4, 2019

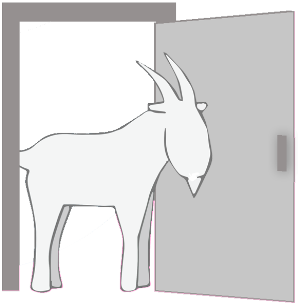
Background

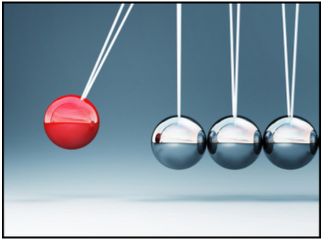


Background

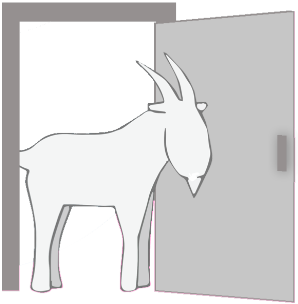


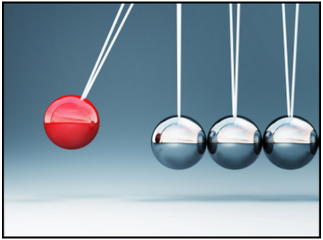
Background





Background





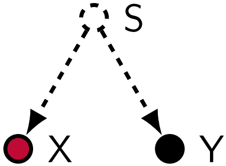
This work

Motivating example

- X: lung cancer — to be diagnosed

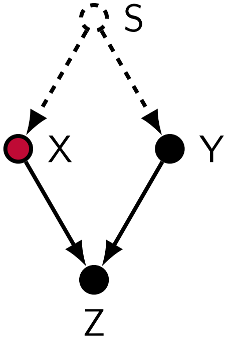
Motivating example

- X: lung cancer — to be diagnosed
- S: smoking (unobserved variable)
- Y: aspirin — may be prescribed to smokers due to their risk of heart disease



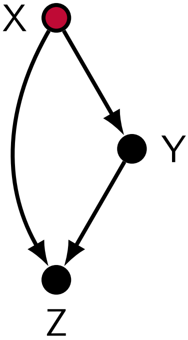
Motivating example

- X: lung cancer — to be diagnosed
- S: smoking (unobserved variable)
- Y: aspirin — may be prescribed to smokers due to their risk of heart disease
- Z: chest pain



Motivating example

- X: lung cancer — to be diagnosed
- S: smoking (unobserved variable)
- Y: aspirin — may be prescribed to smokers due to their risk of heart disease
- Z: chest pain



Motivating example

Two domains:

- source domain ($C = 0$) where we observe data
- target domain ($C = 1$) where we want to make decisions

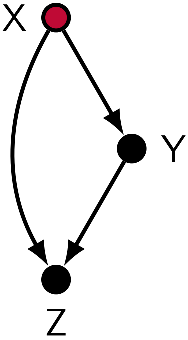
Same causal graph, different distributions:

source:

$P(X \mid C = 0)$

$P(Y \mid X, C = 0)$

$P(Z \mid X, Y, C = 0)$

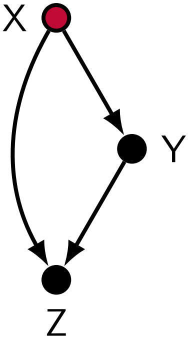


Motivating example

Two domains:

- source domain ($C = 0$) where we observe data
- target domain ($C = 1$) where we want to make decisions

Same causal graph, different distributions:



$$
\begin{array}{rl}
\text{target:} & \text{source:} \\
P(X \mid C = 1) = & P(X \mid C = 0) \\
P(Y \mid X, C = 1)\textbf{?} & P(Y \mid X, C = 0) \\
P(Z \mid X, Y, C = 1) = & P(Z \mid X, Y, C = 0)
\end{array}
$$

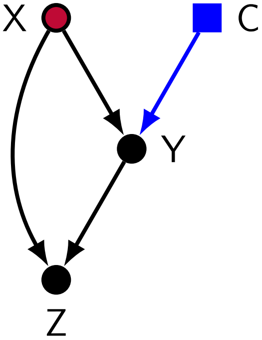
Motivating example

Two domains:

- source domain ($C = 0$) where we observe data
- target domain ($C = 1$) where we want to make decisions

Same causal graph, different distributions:

$$\begin{array}{rcl} & \text{target:} & \text{source:} \\ P(X \mid C = 1) & = & P(X \mid C = 0) \\ P(Y \mid X, C = 1) & \textbf{?} & P(Y \mid X, C = 0) \\ P(Z \mid X, Y, C = 1) & = & P(Z \mid X, Y, C = 0) \end{array}$$



Prior work on causal domain adaptation

Earlier approaches try find a set of features $\mathbf{A} \subseteq \{Y, Z\}$ s.t.

$$P(X \mid \mathbf{A}, C = 1) = P(X \mid \mathbf{A}, C = 0)$$

- Problem: in this graph, the only $\mathbf{A}$ that makes $X \perp\!\!\!\perp C \mid \mathbf{A}$ is $\mathbf{A} = \emptyset$
- That would mean: take the same decision for every patient
- Can we do better?

Robust approach

Let $\mathcal{P}$ be the credal set of all distributions for the target domain consistent with what we know from the source domain

- We want to take decisions that are good regardless of what $P \in \mathcal{P}$ is realized

Robust approach

Let $\mathcal{P}$ be the credal set of all distributions for the target domain consistent with what we know from the source domain

- We want to take decisions that are good regardless of what $P \in \mathcal{P}$ is realized
- Model as zero-sum game against adversary who chooses $P \in \mathcal{P}$
- For that, we need to fix a loss function, e.g. Brier or logarithmic loss

A theorem

Theorem (Existence and characterization of P^*)

For H_L finite and continuous, a $P \in \mathcal{P}$ maximizing the adversary's objective exists, and P^* is such a maximizer if and only if there exists a $\lambda^* \in \mathbb{R}^{\mathcal{X}}$ such that

(i) *for every $y \in \mathcal{Y}$ with $P^*(y) > 0$,*

$$\sum_{\substack{z \in \mathcal{Z}: \\ P^*(y,z) > 0}} P^*(z \mid y) H_L(P^*(\cdot \mid y, z)) = \sum_x P^*(x \mid y) \lambda^*_x;$$

(ii) *for every $y \in \mathcal{Y}$, for all $P' \in \Delta_{\mathcal{X}}$, let $P'(x, z \mid y) := P'(x) P(z \mid x, y)$; then*

$$\sum_{\substack{z \in \mathcal{Z}: \\ P'(z \mid y) > 0}} P'(z \mid y) H_L(P'(\cdot \mid y, z)) \leq \sum_x P'(x \mid y) \lambda^*_x.$$

Theorem applied to numerical example

We give a numerical example where all variables are binary, and find P^* analytically using the theorem:

- for Brier loss, and
- for logarithmic loss

The two solutions (and thus the resulting decisions) are different, even though both loss functions are strictly proper scoring rules

The end

☞ Come to the poster! ☜