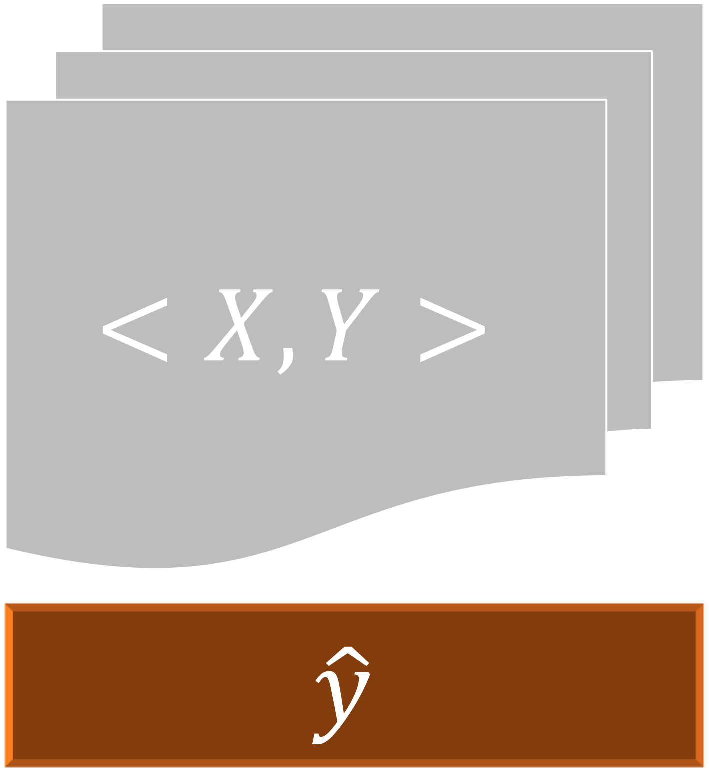
Causal inference for machine learning:
Generalization, Explanation, and Fairness

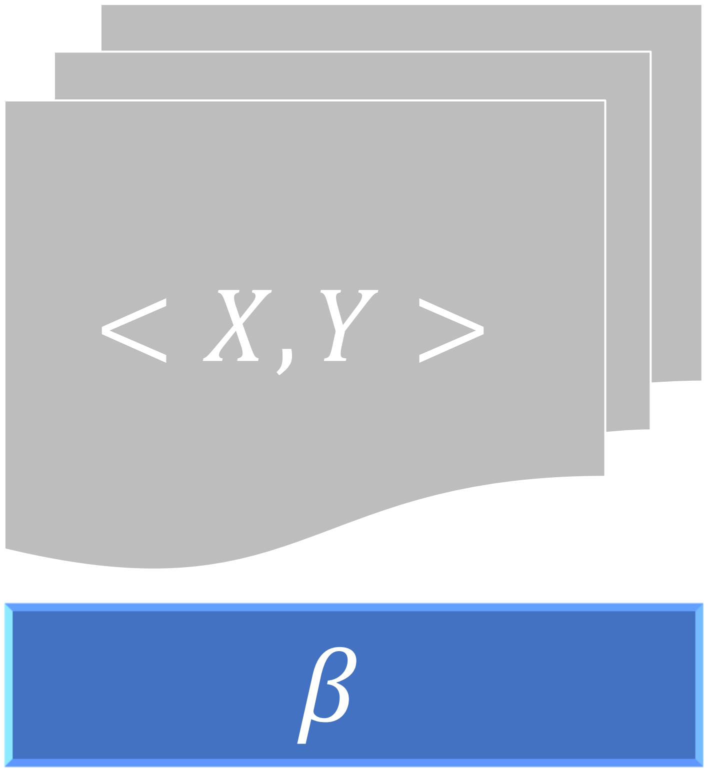
Amit Sharma
Microsoft Research India
@amt_shrma
www.amitsharma.in

True: y=f(x,u)+ϵ

Prediction ML:
y=h(x)+ϵ

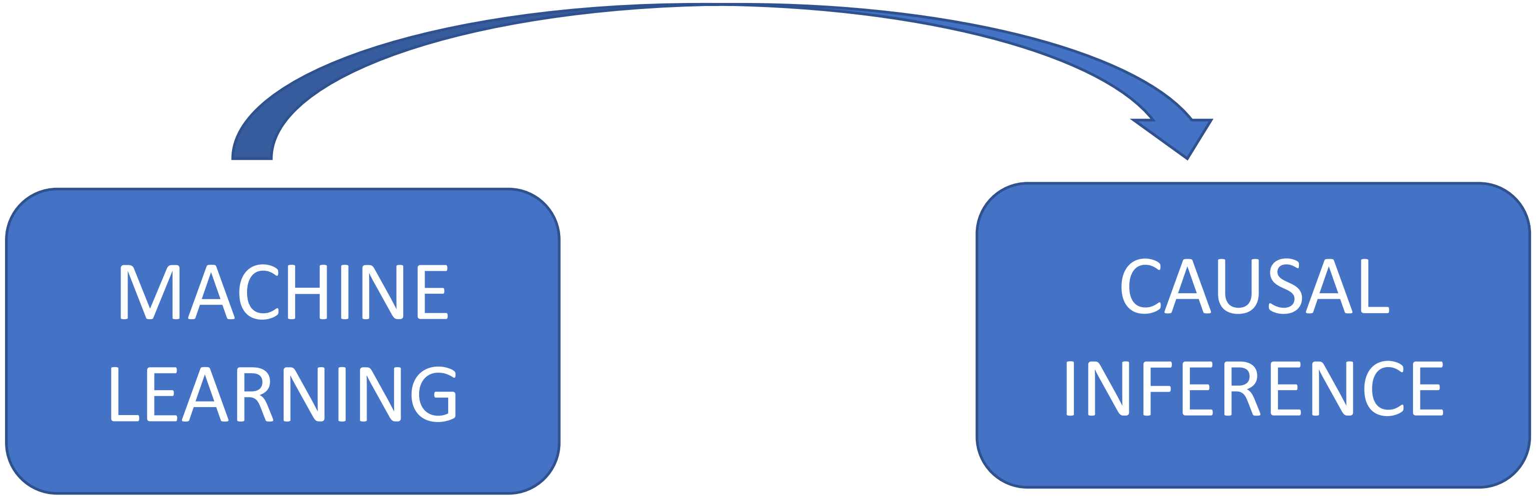


Causal inference:
$\partial f(x, u) / \partial x$

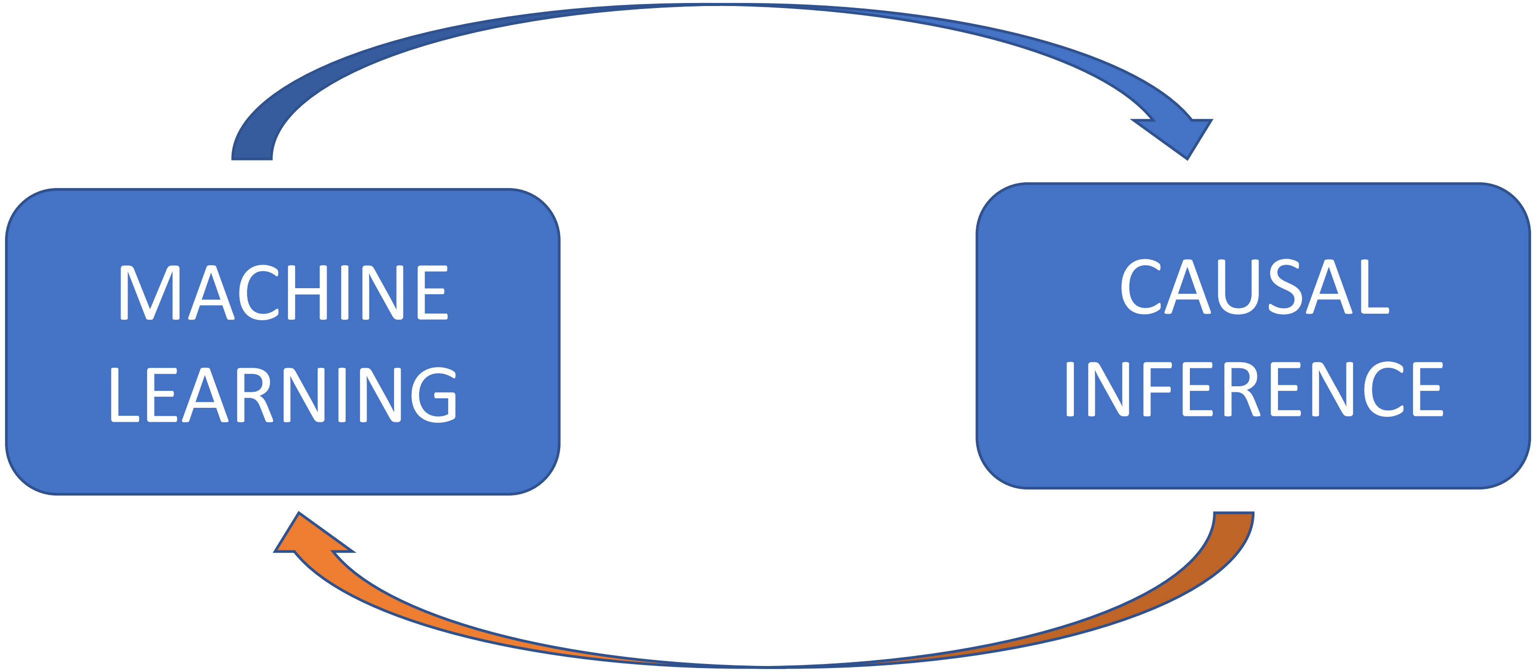


Hofman, **Sharma**, and Watts (2017). Prediction and Explanation in Social Systems. *Science*, 355.6324

Better Estimation and
Refutation of Causal Effect
github.com/microsoft/dowhy



Better Estimation and
Refutation of Causal Effect
github.com/microsoft/dowhy

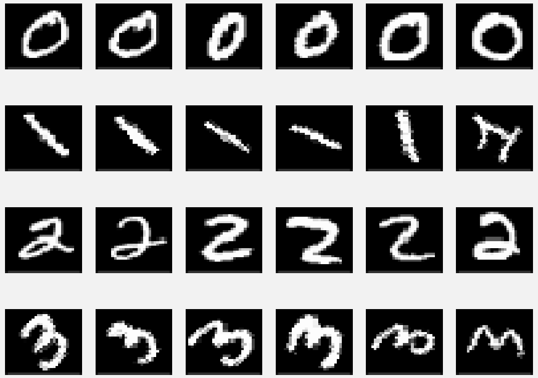


- Better Generalization & Robustness of ML models
- Principled framework for Fairness and Explanation

Key message: Causal reasoning is essential for machine learning

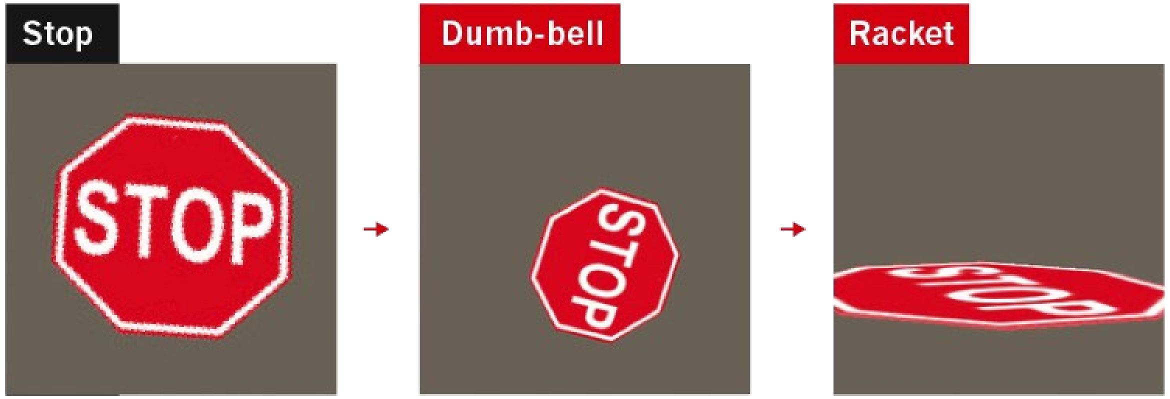
- Machine learning faces many fundamental challenges
  - Out of distribution generalization, Robustness, Fairness, Explainability, Privacy
- A causal perspective can help
  - Better definitions of the challenges
  - Theoretically justified algorithms
- “Matching” for out-of-distribution prediction
- “Counterfactuals” for explainable predictions
- “Missing data” for fairness

Correlational machine learning searches for patterns.
Often finds spurious ones



Accuracy on unseen angles (0, 90): 64%

[Piratla et al. ICML 2020]



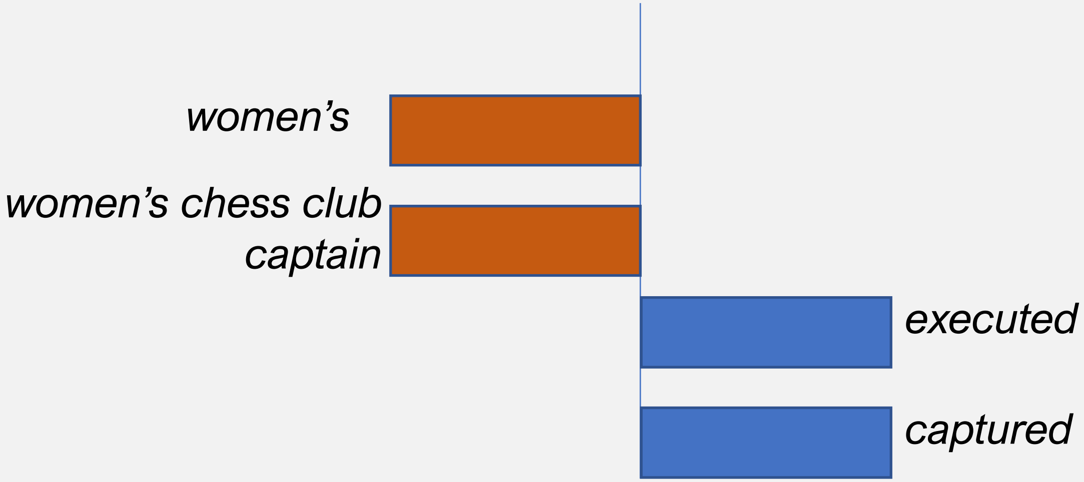
Incorrect predictions under changes in data

[Alcorn et al. CVPR 2019]

What color is the tray?	Pink
What colour is the tray?	Green
Which color is the tray?	Green
What color is it?	Green
How color is tray?	Green

Fooled by semantically equivalent perturbations

[Ribeiro et al. ACL 2018]



Bias in ML model for hiring decisions

[Reuters 2018, Weblink]

1. OOD generalization is a causal problem.

Domain Generalization using Causal Matching. ICML 2021.

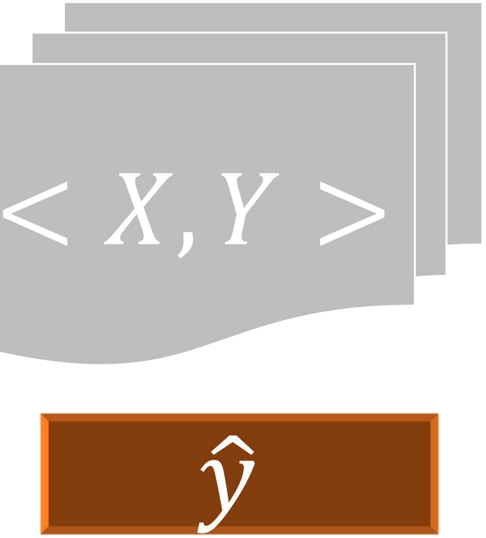
Alleviating Privacy Attacks using Causal Learning. ICML 2020.

Causal Regularization using Domain Priors. Arxiv.

True: y=f(x,u)+ϵ

Prediction ML:

y=h(x)+ϵ

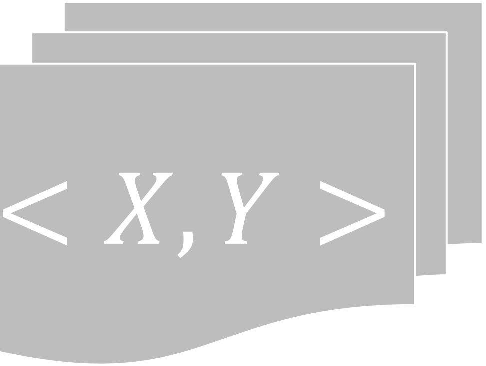


Hofman, **Sharma**, and Watts (2017). Prediction and Explanation in Social Systems. *Science*, 355.6324

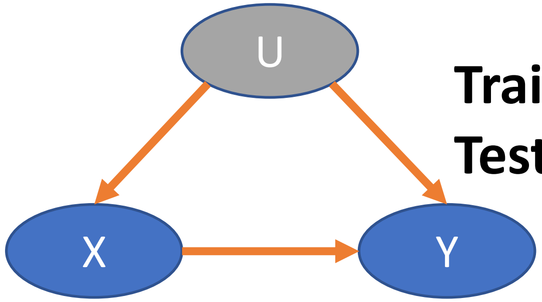
True: y=f(x,u)+ϵ

Prediction ML:

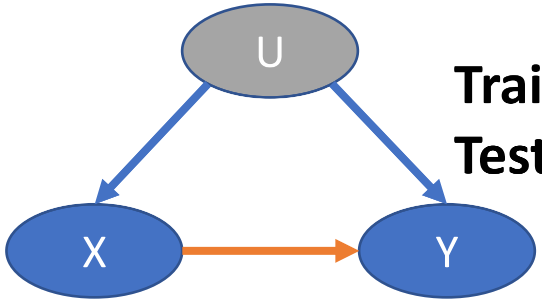
y=h(x)+ϵ



$\hat{y}$



Train: P(X, Y, U)
Test: P(X, Y, U)



Train: P(X, Y, U)
Test: P*(X, Y, U)

Typical supervised prediction

$$\min_{\mathrm{P}} (y - \hat{y})^2$$

Use cross-validation to select model.
No need to worry about u.

Out-of-distribution prediction:

$$\min_{\mathrm{P}^*} (y - \hat{y})^2$$

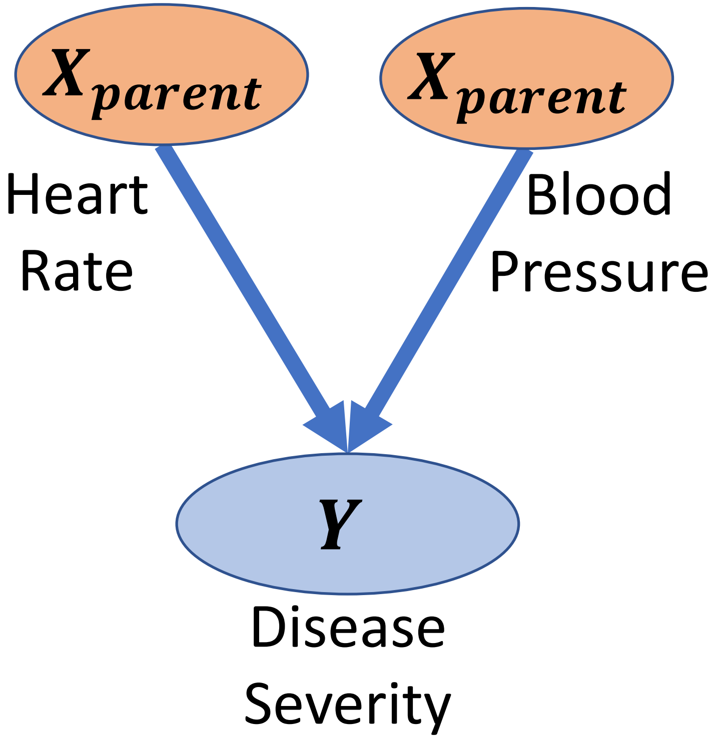
P* is not observed.
Cross-validation is not possible.

Hofman, **Sharma**, and Watts (2017). Prediction and Explanation in Social Systems. *Science*, 355.6324

Invariant causal learning: If you learn the causal function from X->Y, your model will be optimal across all unseen distributions.

Peters et al. (2015), Arjovsky et al. (2019)

Where's the catch? Learning causal models



$h(x_C)$ will lead to similar accuracy on both domains.

Blood Pressure ⬆ => Disease Severity ⬆

Structural Approach: Create a causal graph based on external knowledge.

Multiple Domains Approach: Find features whose effect stays invariant across many domains.

Constraints Approach: Identify the constraints that any causal model should satisfy.

I. Learning using causal structure

A dataset of people living with a chronic illness.

*<**Y:**disease_severity> <**X:**age, gender, blood pressure, heart rate>*

Associational ML: $\min_{h} \sum_{(x,y)} Loss(h(x), y)$

I. Learning using causal structure

A dataset of people living with a chronic illness.

*<**Y:**disease_severity> <**X:**age, gender, blood pressure, heart rate>*

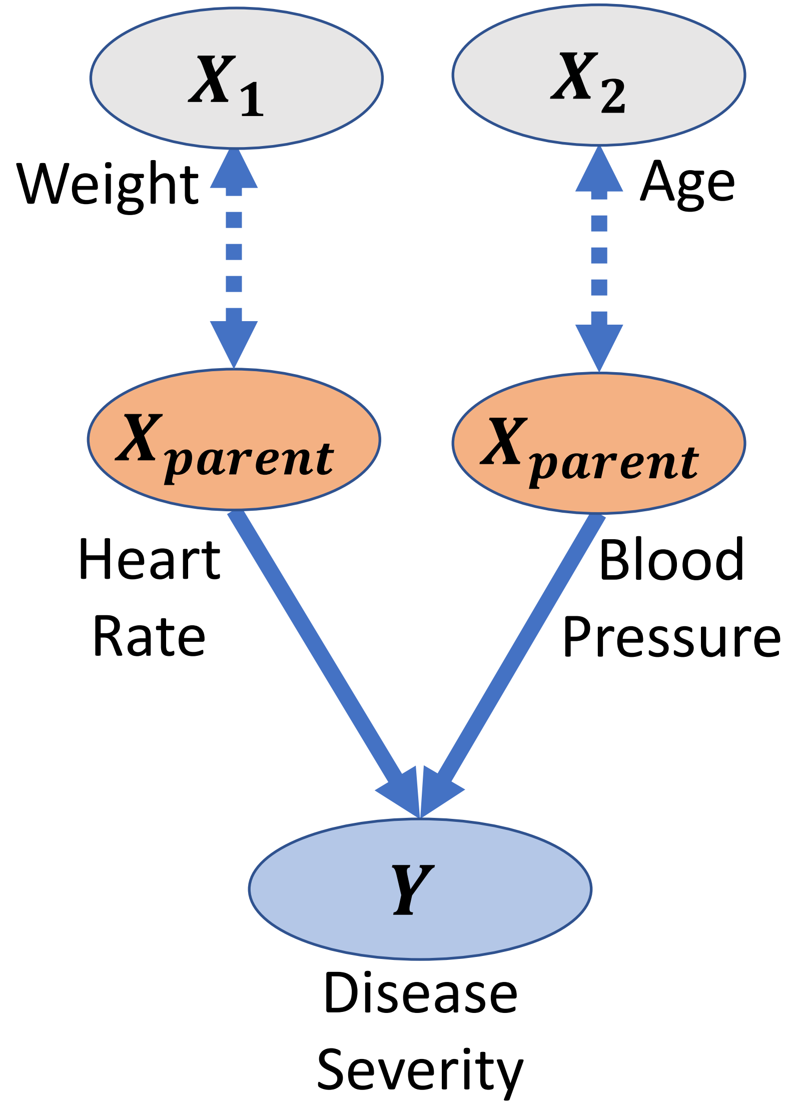
Associational ML: $\min_{h} \sum_{(x,y)} Loss(h(x), y)$

(Ideal) Causal learning:

1. Identify which features directly cause the outcome (parents of **Y** in the causal graph).
2. Build a predictive model using only those features.

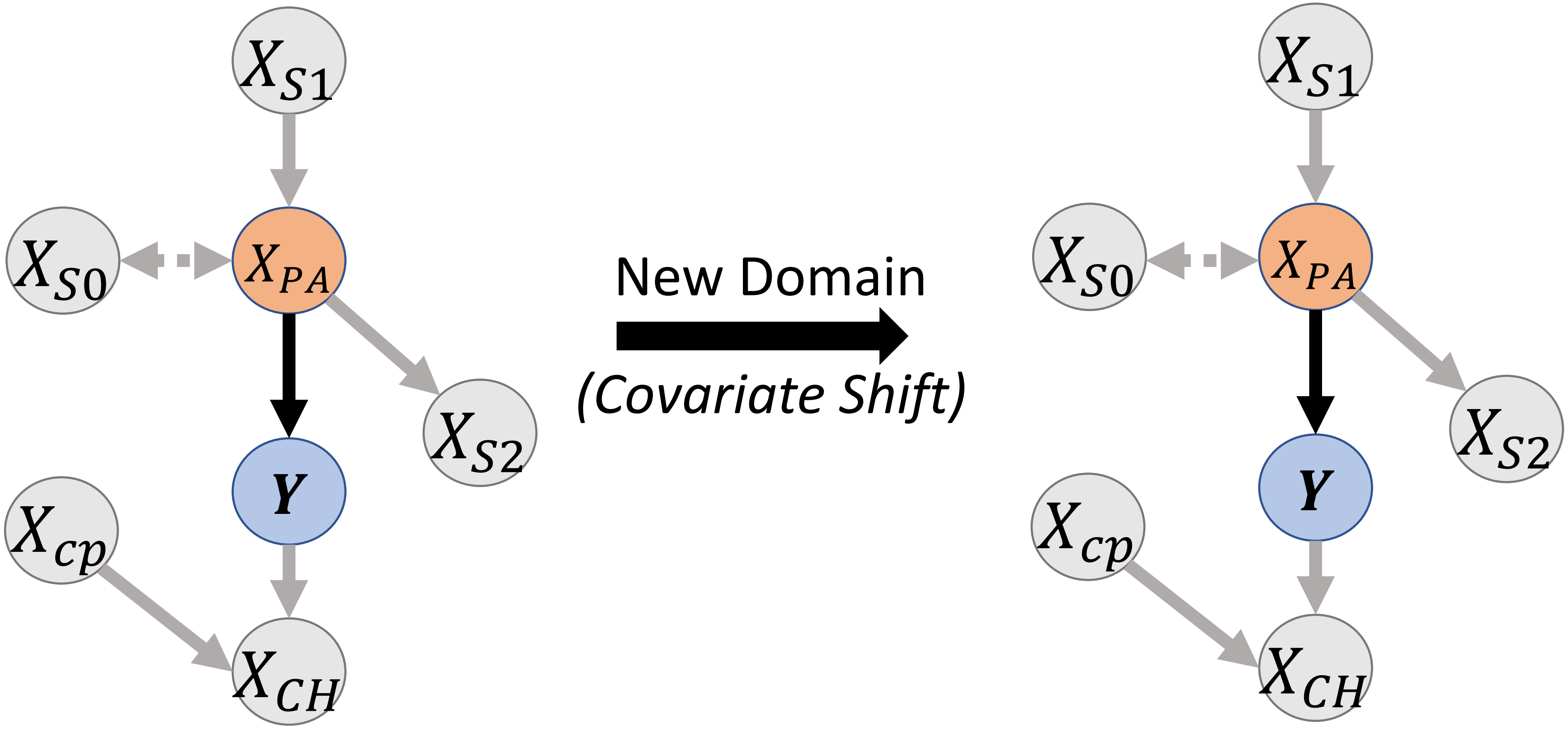
Causal ML: $\min_{h} \sum_{(x,y)} Loss(h(x_C), y)$

Lower train accuracy but stays consistent with new domains.



$\mathbf{X_C} = \mathbf{X_{PA}} =$ {heart rate, blood pressure}

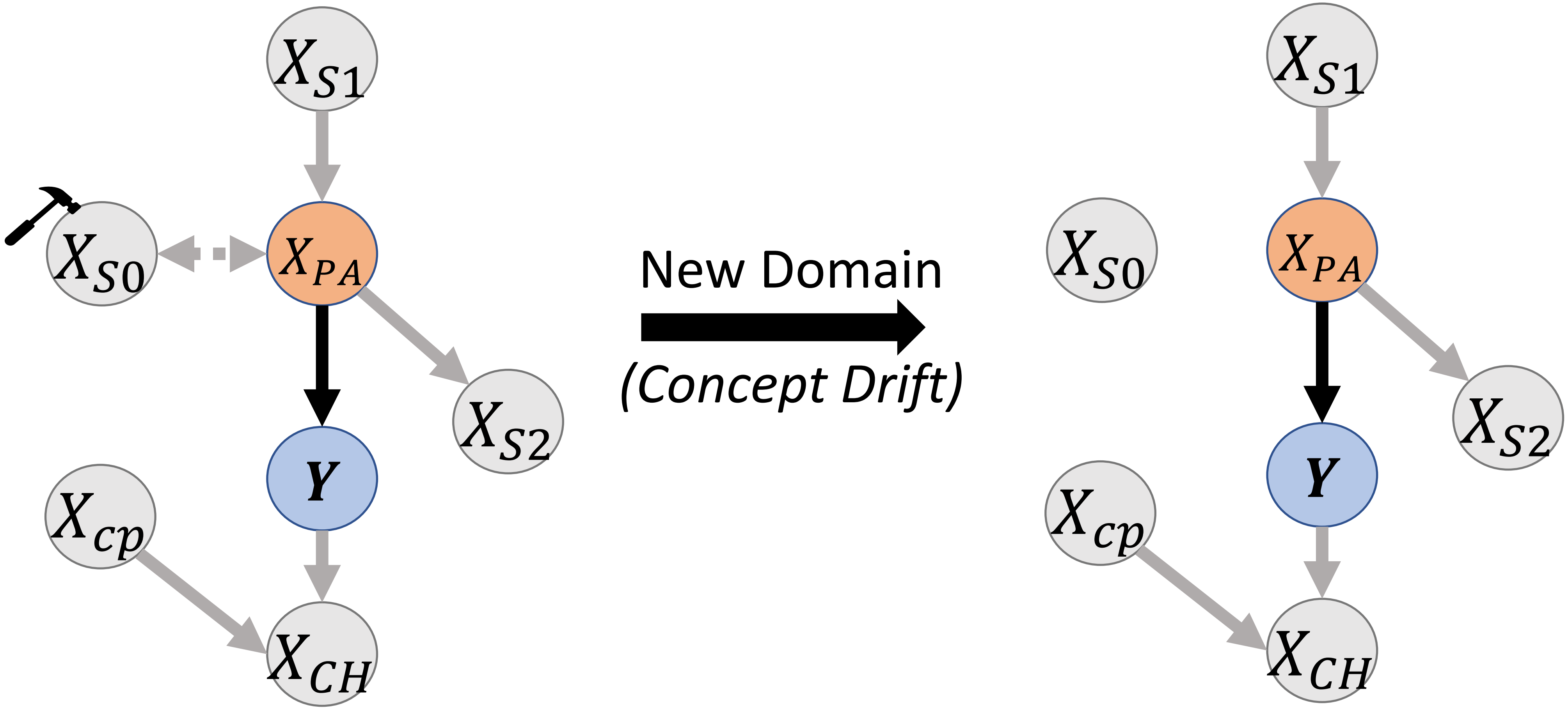
Why only parents of Y in the causal graph?



$$\boldsymbol{P(X,Y) = P(X)P(Y|X)}$$

$$\boldsymbol{P^*(X,Y) = P^*(X)P(Y|X)}$$

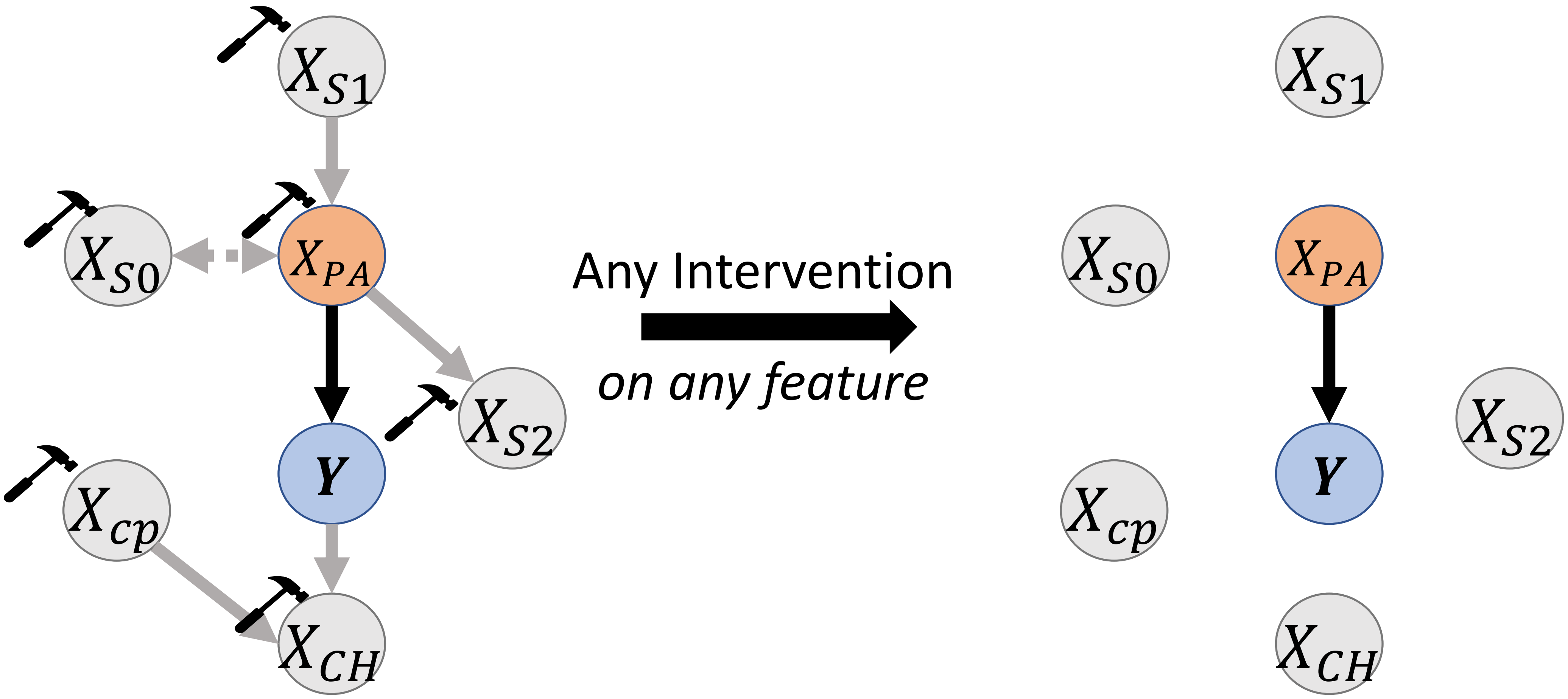
Why only parents of Y in the causal graph?



$$\boldsymbol{P(X,Y) = P(X)P(Y|X)}$$

$$\boldsymbol{P^*(X,Y) = P(X)P^*(Y|X)}$$

Why only parents of Y in the causal graph?



$P(Y|X_{PA})$ is invariant across different distributions, unless there is a change in true data-generating process for **Y**.

Any other benefits?

* **Result 1:** Better out-of-distribution generalization
- **Result 2:** Stronger differential privacy guarantees

Theorem: When equivalent Laplace noise is added and models are trained on same dataset, causal mechanism M_C provides ϵ_C-DP and associational mechanism M_A provides ϵ_A-DP guarantees such that:

$$\boldsymbol{\epsilon_c \leq \epsilon_A}$$

Causal models are more robust to privacy attacks like membership inference.

Alleviating Privacy Attacks using Causal Learning. ICML 2020.

Result 1: Worst-case out-of-distribution error of a causal model is lower than an associational model.

For any model h, and P^* such that $P^*(Y|X_{PA}) = P(Y|X_{PA})$,

In-Distribution Error (IDE)= $\mathbf{IDE_P}(\boldsymbol{h}, \boldsymbol{y}) = \mathbf{L}_{\boldsymbol{P}}(\boldsymbol{h}, \boldsymbol{y}) - \mathbf{L}_{\boldsymbol{S}\sim\mathrm{P}}(\boldsymbol{h}, \boldsymbol{y})$

Expected loss on the same distribution as the train data

Out-of-Distribution Error (ODE)=$\mathbf{ODE_{P,P^*}}(\boldsymbol{h}, \boldsymbol{y}) = \mathbf{L}_{\boldsymbol{P}^*}(\boldsymbol{h}, \boldsymbol{y}) - \mathbf{L}_{\boldsymbol{S}\sim\mathrm{P}}(\boldsymbol{h}, \boldsymbol{y})$

Expected loss on a different distribution P^* than the train data

Result 1: Worst-case out-of-distribution error of a causal model is lower than an associational model.

For any model h, and P^* such that $P^*(Y|X_{PA}) = P(Y|X_{PA})$,

In-Distribution Error (IDE)= $\mathbf{IDE_P}(\boldsymbol{h}, \boldsymbol{y}) = \mathbf{L}_{\boldsymbol{P}}(\boldsymbol{h}, \boldsymbol{y}) - \mathbf{L}_{\boldsymbol{S}\sim\mathrm{P}}(\boldsymbol{h}, \boldsymbol{y})$

Expected loss on the same distribution as the train data

Out-of-Distribution Error (ODE)=$\mathbf{ODE_{P,P^*}}(\boldsymbol{h}, \boldsymbol{y}) = \mathbf{L}_{\boldsymbol{P}^*}(\boldsymbol{h}, \boldsymbol{y}) - \mathbf{L}_{\boldsymbol{S}\sim\mathrm{P}}(\boldsymbol{h}, \boldsymbol{y})$

Expected loss on a different distribution P^* than the train data

Discrepancy b/w P and P^* distributions

Simple case: Assume $y = f(\boldsymbol{x})$ is deterministic.

Causal Model $\mathbf{ODE_{P,P^*}}(\boldsymbol{h}_\mathbf{c}, \boldsymbol{y}) \leq \mathbf{IDE_P}(\boldsymbol{h}_{\boldsymbol{c}}, \boldsymbol{y}) + \boldsymbol{disc}_\mathbf{L}(\boldsymbol{P}, \boldsymbol{P}^*)$

Result 1: Worst-case out-of-distribution error of a causal model is lower than an associational model.

For any model h, and P^* such that $P^*(Y|X_{PA}) = P(Y|X_{PA})$,

In-Distribution Error (IDE)= $\mathbf{IDE_P}(\boldsymbol{h}, \boldsymbol{y}) = \mathbf{L}_{\boldsymbol{P}}(\boldsymbol{h}, \boldsymbol{y}) - \mathbf{L}_{\boldsymbol{S}\sim\mathrm{P}}(\boldsymbol{h}, \boldsymbol{y})$

Expected loss on the same distribution as the train data

Out-of-Distribution Error (ODE)=$\mathbf{ODE_{P,P^*}}(\boldsymbol{h}, \boldsymbol{y}) = \mathbf{L}_{\boldsymbol{P^*}}(\boldsymbol{h}, \boldsymbol{y}) - \mathbf{L}_{\boldsymbol{S}\sim\mathrm{P}}(\boldsymbol{h}, \boldsymbol{y})$

Expected loss on a different distribution P^* than the train data

Simple case: Assume $y = f(\boldsymbol{x})$ is deterministic.

Causal Model $\mathbf{ODE_{P,P^*}}(\boldsymbol{h_c}, \boldsymbol{y}) \le \mathbf{IDE_P}(\boldsymbol{h_c}, \boldsymbol{y}) + \boldsymbol{disc}_{\mathrm{L}}(\boldsymbol{P}, \boldsymbol{P^*})$

Assoc. Model $\mathbf{ODE_{P,P^*}}(\boldsymbol{h_a}, \boldsymbol{y}) \le \mathbf{IDE_P}(\boldsymbol{h_a}, \boldsymbol{y}) + \boldsymbol{disc}_{\mathrm{L}}(\boldsymbol{P}, \boldsymbol{P^*}) + \mathbf{L}_{\boldsymbol{P^*}}(\boldsymbol{h_{a,P}^{OPT}}, \boldsymbol{y})$

Discrepancy b/w P and P^* distributions

Optimal h_a on P is not optimal on P^*

$$\Rightarrow \max_{\mathbf{P^*}} \mathbf{ODEBound_{P,P^*}}(\boldsymbol{h_c}, \boldsymbol{y}) \le \max_{\mathbf{P^*}} \mathbf{ODEBound_{P,P^*}}(\boldsymbol{h_a}, \boldsymbol{y})$$

Result 2: A causal model has stronger differential privacy guarantees than associational model

How much do trained model parameters change based on changing one data point?

Differential Privacy [DR'14]**:** A learning mechanism M satisfies ϵ-differential privacy if for any two datasets, S, S' that differ in one data point, $\frac{\Pr(\mathrm{M}(S)\in H)}{\Pr(\mathrm{M}(S')\in H)} \le e^{\epsilon}$.

(Smaller ϵ values provide better privacy guarantees)

Theorem: When equivalent Laplace noise is added and models are trained on same dataset, causal mechanism $\mathrm{M_C}$ provides ϵ_C-DP and associational mechanism $\mathrm{M_A}$ provides ϵ_A-DP guarantees such that:

$$\boldsymbol{\epsilon_c \le \epsilon_A}$$

Result 3: Causal models are more robust to membership inference (MI) attacks

Advantage of an MI adversary: *(roughly)* Given black-box access to ML model, accuracy of detecting if an input belongs to the training data.

[From Yeom et al. CSF'18] Membership advantage of an adversary is bounded by $e^{\epsilon} - 1$.

Theorem: When trained on the same dataset of size n, membership advantage of a causal model is lower than the membership advantage for an associational model.

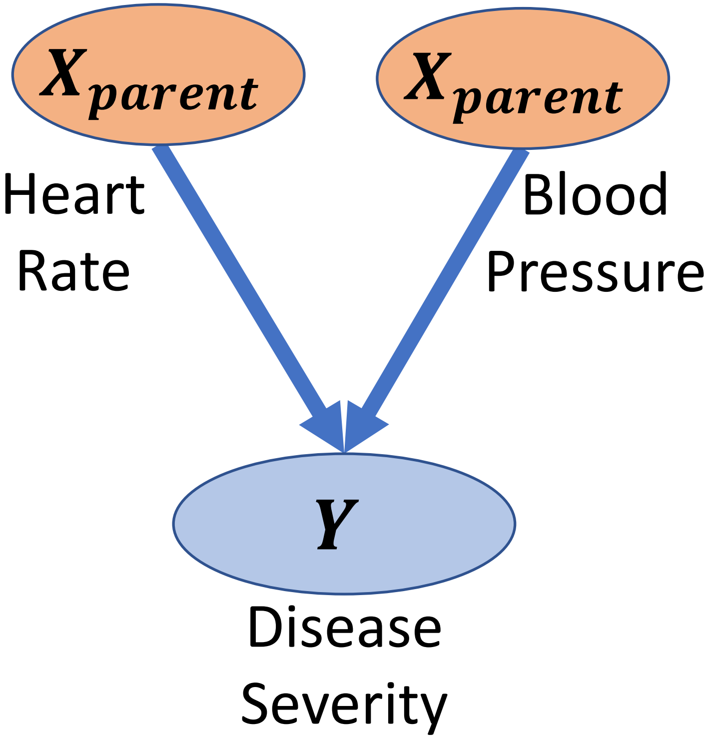
Summary: Causal predictive models offer better accuracy and privacy.

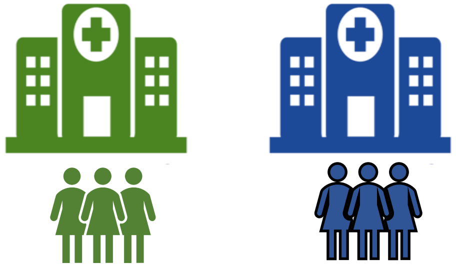
So why is everyone not using it?

Same problem as for causal inference: Rare to have an outcome variable where all parents are observed.

Can methods from causal inference also be used to solve it?

Where's the catch? Learning causal models





$h(x_C)$ will lead to similar accuracy on both domains.

Blood Pressure ⬆ => Disease Severity ⬆

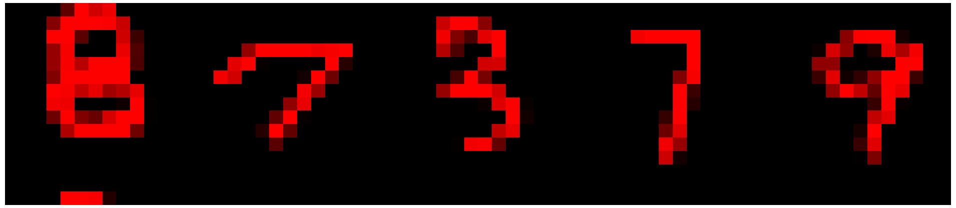
Structural Approach: Create a causal graph based on external knowledge.

Multiple Domains Approach: Find features whose effect stays invariant across many domains.

Constraints Approach: Identify the constraints that any causal model should satisfy.

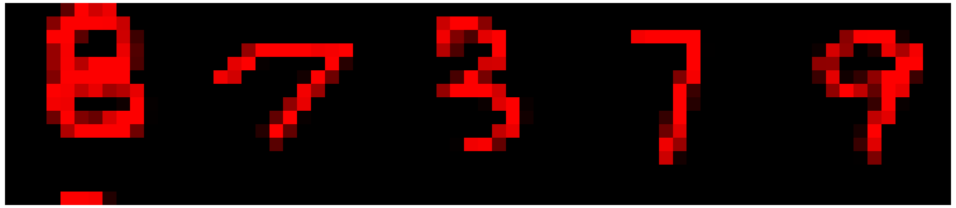
Leveraging data from multiple domains

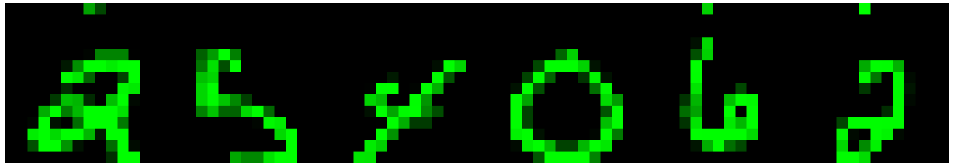
TRAIN DATASET



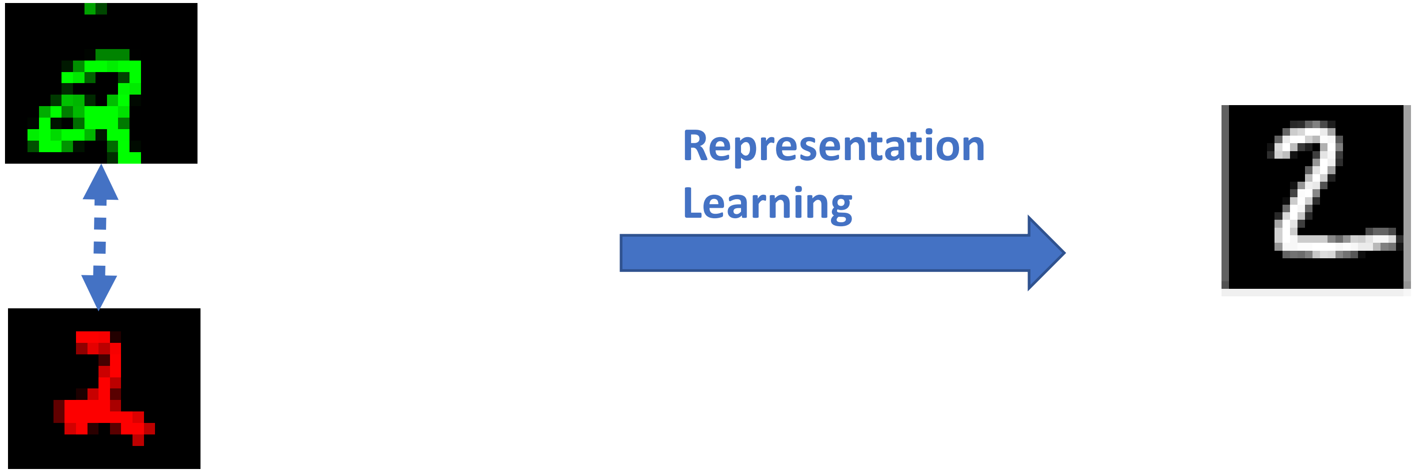
TEST DATASET

TRAIN DATASET





TEST DATASET

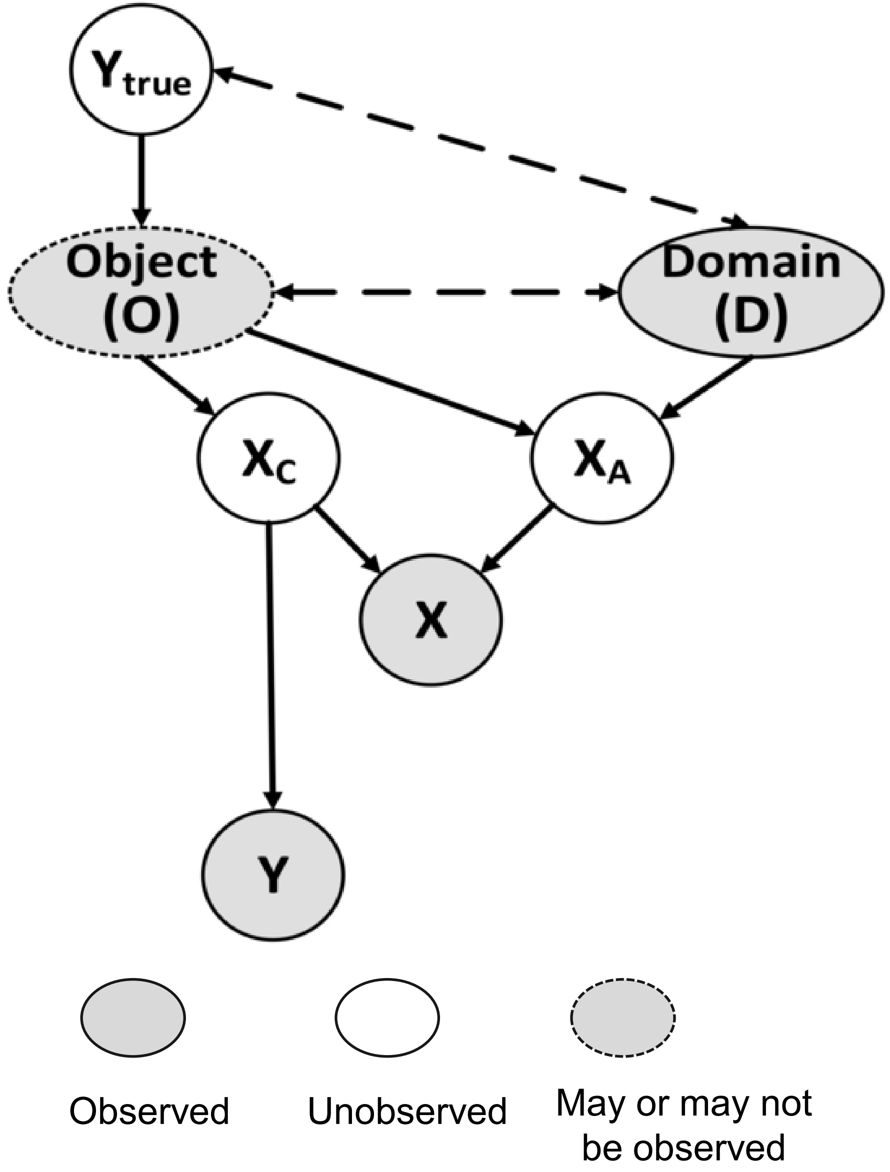


Need to ensure that pair of images exactly match on shape features, but vary on color (i.e., confounder)

Difference from causal inference: Matching for same causal features, rather than same confounders

Data augmentation in ML

How it works?



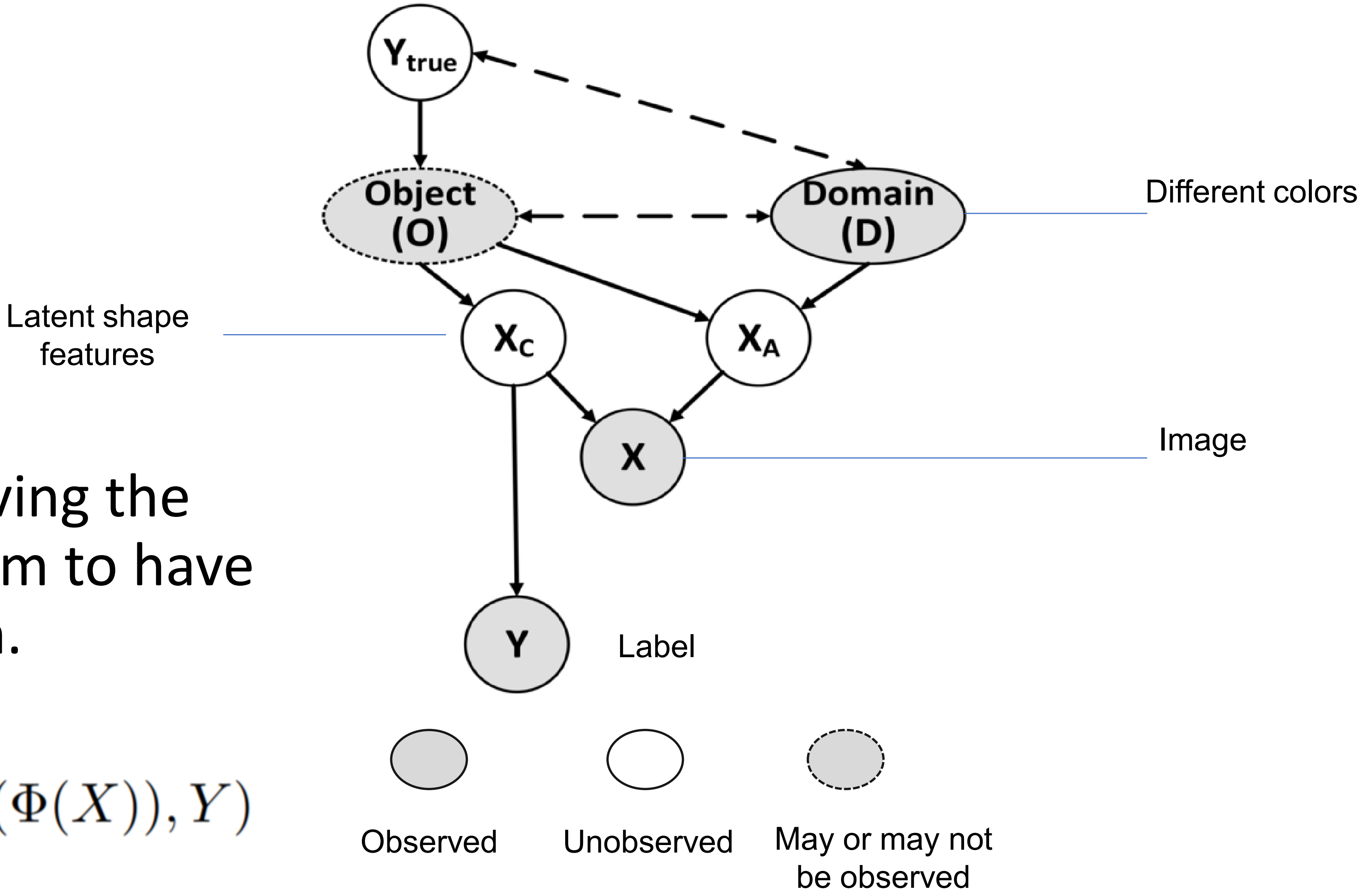
Y_{true}
Object
(O)
Domain
(D)
X_C
X_A
X
Y
Observed
Unobserved
May or may not
be observed

How it works?

Goal: Learn E$(Y|X_c)$

But X_c is not observed.

So match two images having the same X_c and enforce them to have the same representation.



$$f_{\texttt{perfectmatch}} = \arg\min_{h,\Phi} \sum_{d=1}^{m} L_d(h(\Phi(X)), Y)$$

$$\texttt{s.t.} \sum_{\Omega(j,k)=1; d\neq d'} \text{dist}(\Phi(\mathbf{x}_j^{(d)}), \Phi(\mathbf{x}_k^{(d')})) = 0$$

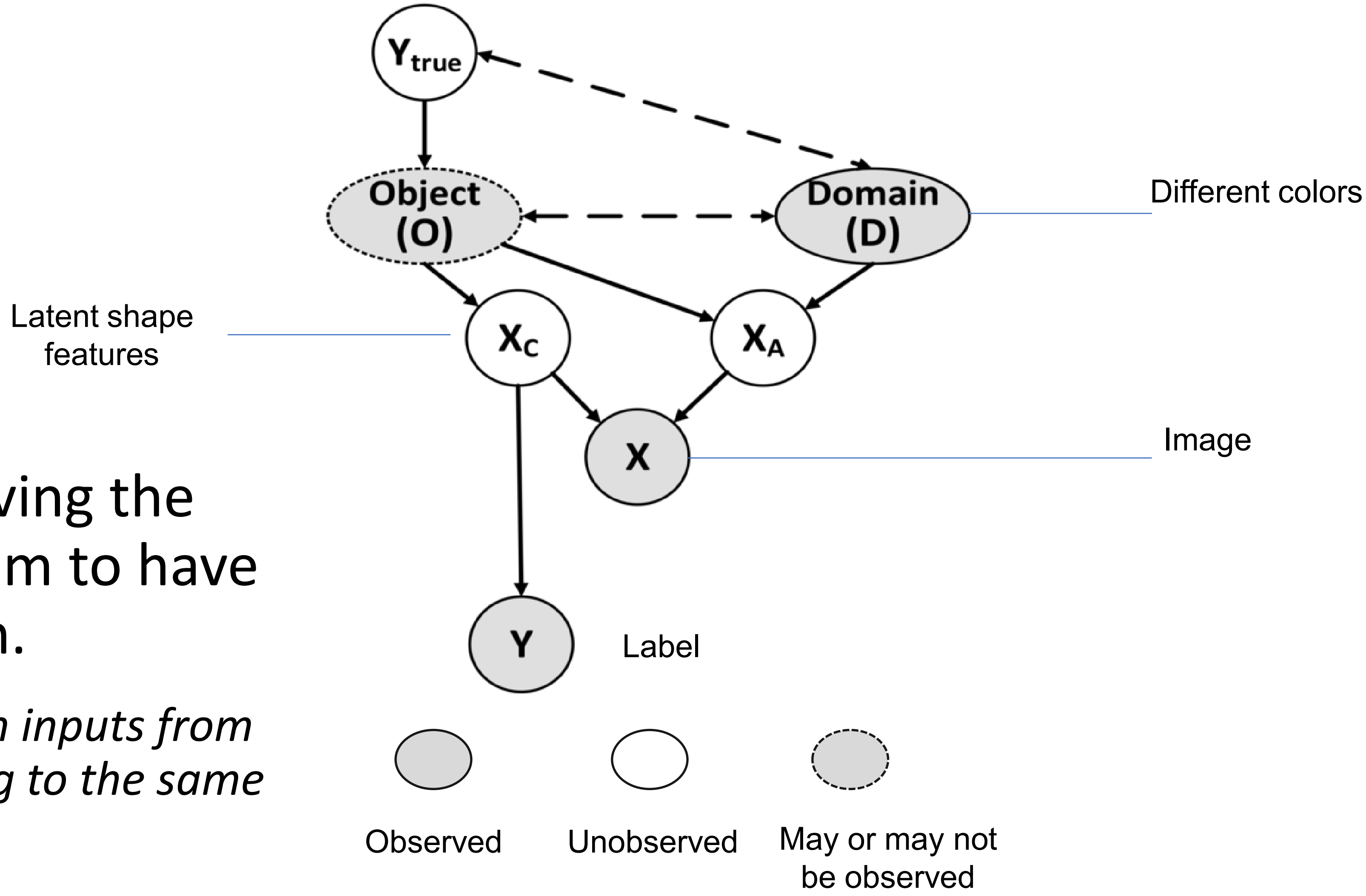
How it works?

Goal: Learn E$(Y|X_c)$

But X_c is not observed.

So match two images having the same X_c and enforce them to have the same representation.

$\boldsymbol{X_C \perp\!\!\!\perp Domain|Object}$ **=>** *match inputs from the different domains that belong to the same object.*



Aside: Many prior works on domain generalization optimize for the incorrect objective

- "Domain Invariant Representation" proposes $X_C \perp\!\!\!\perp Domain$
- "Class-conditional Domain Invariant Representation" proposes $X_C \perp\!\!\!\perp Domain|Y$

Leveraging multiple domains: Can also work if data augmentations are not available

- If objects are not known, iteratively learn matched pairs of inputs from different domains.
- **Assumption:** Same-class inputs are closer in causal features to inputs from different classes.
    - Start with matching random inputs from the same class.
    - **Minimize intra-match distance:** Find a feature representation that minimizes the distance within matches.
    - **Estimate matches:** Update matches based on the new representation and repeat.

Algorithm 1: MatchDG

Input: Dataset $(d_i, x_i, y_i)_{i=1}^{n}$ from m domains, τ, t

Output: Function $f : \mathcal{X} \rightarrow \mathcal{Y}$

Create random match pairs Ω_Y.

Build a $n * m$ data matrix $\mathcal{M}$.

Phase I. **while** *notconverged* **do**

    for *batch* $\sim \mathcal{M}$ **do**

        Minimize contrastive loss (6).

    if *epoch % t ==0* **then**

        Update match pairs using Φ_{epoch}.

Phase 2. Compute matching based on Φ. Minimize the loss (5) to obtain f.

IV. Empirical results: Causal models are more accurate on unseen rotations of MNIST digits

Dataset	Source	ERM	ERM-PerfMatch
Rotated MNIST	15, 30, 45, 60, 75	96.5 (0.15)	98.5 (0.08)
	30, 45, 60	80.6 (2.9)	93.6 (0.53)
	30, 45	64.0 (2.28)	84.2 (2.33)
Rotated Fashion MNIST	15, 30, 45, 60, 75	78.5 (1.15)	85.1 (0.97)
	30, 45, 60	33.9 (1.04)	61.04 (1.33)
	30, 45	21.85 (0.93)	42.0 (2.42)

Test Domain: Images rotated by 0 and 90 degrees.

This method also achieves state-of-the-art accuracy on PACS , the most popular domain generalization benchmark.

IV. Empirical results: Causal models are more accurate on unseen rotations of MNIST digits

Dataset	Source	ERM	MASF	CSD	ERM-RandMatch	MatchDG	ERM-PerfMatch
Rotated MNIST	15, 30, 45, 60, 75	96.5 (0.15)	93 (0.2)	94.7 (0.2)	97.5 (0.17)	**97.5** (0.36)	98.5 (0.08)
	30, 45, 60	80.6 (2.9)	69.4 (1.32)	**89.1 (0.004)**	82.8 (2.3)	88.9 (2.01)	93.6 (0.53)
	30, 45	64.0 (2.28)	60.8 (1.53)	77.2 (0.04)	69.7 (2.93)	**79.3** (4.2)	84.2 (2.33)
Rotated Fashion MNIST	15, 30, 45, 60, 75	78.5 (1.15)	72.4 (2.9)	78.9 (0.7)	80.5 (0.97)	**83.5** (1.16)	85.1 (0.97)
	30, 45, 60	33.9 (1.04)	25.7 (1.73)	27.8 (0.01)	35.5 (1.07)	**51.7** (2.08)	61.04 (1.33)
	30, 45	21.85 (0.93)	20.8 (1.26)	20.2 (0.01)	23.9 (0.93)	**36.6** (2.17)	42.0 (2.42)

Test Domain: Images rotated by 0 and 90 degrees.

This method also achieves state-of-the-art accuracy on PACS , the most popular domain generalization benchmark.

2. ML Explanation is a causal problem.

Explaining Machine Learning Classifiers through Counterfactual Examples. FaccT 2020.

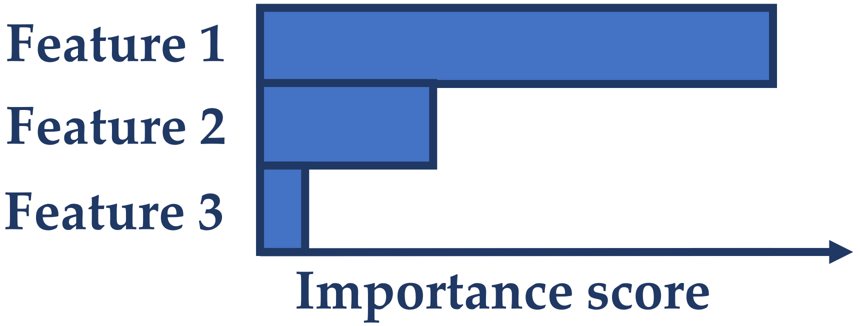
Towards Unifying Feature Attribution and Counterfactual Explanations: Different Means to the Same End. AIES 2021.

Explaining machine learning predictions

Techniques to explain machine predictions

LIME (Ribeiro et al., 2016); **Local Rule-based** (Guidotti et al., 2018);
SHAP (Lundberg et al., 2017); **Intelligible Models** (Lou et al., 2012); …..

Feature importance-based methods are widely used in many practical applications



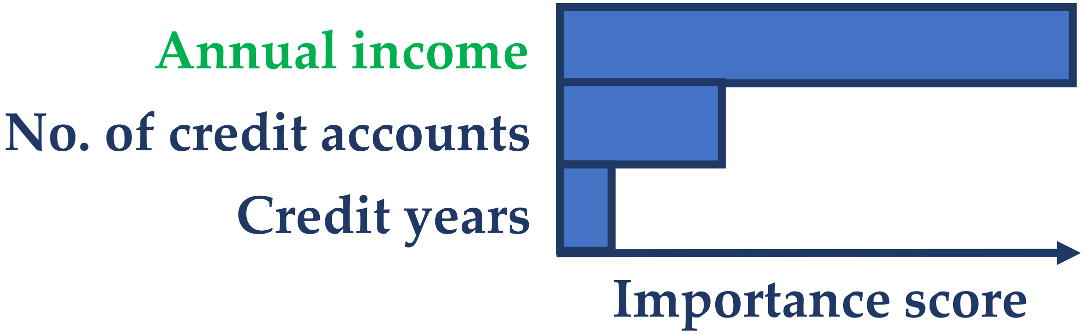
In many cases, feature importance is not enough

Bank → Loan distribution algorithm → Loan granted / Loan denied

Suppose a person does not get the loan.

Person: What should I do to get the loan in the future?

Feature importance-based explanations



Counterfactual explanations (CF)

("what-if" scenarios) (Wachter et al., 2017)

You would have got the loan if your annual income had been 100,000

Many explanation scenarios are actually asking "what-if" or causal questions

"If I change the most important feature according to explanation, will it change the predicted outcome?"

"What if we change the second most important feature?"

Statistical summaries are not enough.

Require different kind of reasoning -> **Causal reasoning**

Individual treatment effect of different features

Causal reasoning for explaining machine learning

Bank → Loan distribution algorithm → Loan granted / Loan denied

Counterfactual explanations (CF)
("what-if" scenarios) (Wachter et al., 2017)

You would have got the loan if your annual income had been 100,000

What feature value caused the prediction?

How to provide a feature ordering?

What does it mean to explain an event?

Event = ML model predicts 1.

[Halpern 2016] A feature is an ideal causal explanation iff:

- Necessity: Changing the feature changes model's prediction.
- Sufficiency: If the feature stays the same, cannot change the model's prediction.

- **Ideal explanations are rare.**

$$f\ (x1\ \ , x2, x3) = I(0.4x1 + 0.1x2 + 0.1x3 >= 0.5)$$

Given f(1,1,1) = 1,

x1 is necessary.

No feature is sufficient.

But we can quantify degree of necessity or sufficiency

(x, f(x))

- Necessity = P (f(x) changes | feature is changed)
- Sufficiency = P(f(x) is unchanged | feature is unchanged)

Where these probabilities are over all plausible values of the features.

In practice, approximate by neighborhood of the point **x**.

Simple algorithm

Necessity: Given $(x, f(x))$, find necessity of feature x_i

- Sample point x' such that x_i is changed while keeping every other feature constant.
- Calculate $P(f(x') \,!= f(x))$ over all such x'

Sufficiency: Given $(x, f(x))$, find sufficiency of feature x_i

- Sample point x' such that x_i is constant while changing all other features.
- Calculate $P(f(x') == f(x))$ over all such x'

A more efficient approximate algorithm

Necessity: Given $(x, f(x))$,

- Find the smallest changes to the input x that change the outcome.
- Necessity is proportional to the number of times a feature is changed to lead to a different outcome.

Sufficiency: Given $(x, f(x))$,

- Find the smallest changes to the input x that change the outcome, without changing x_i.
- Sufficiency is inversely proportional to the number of times a valid change is found.

More generally, counterfactual explanations involve an optimization

Diverse counterfactual explanations

Loss to get desirable outcome

Loss to ensure proximity to original input

Loss to provide diverse explanations

$$C(x) = \arg\min_{c_1,\ldots,c_k} \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{yloss}(f(c_i), y) + \frac{\lambda_1}{k}\boldsymbol{dist}(c_i, x) - \lambda_2 \boldsymbol{dpp\_diversity}(c_1, \ldots, c_k)$$

k – no. of counterfactuals
λ_1 and λ_2 – loss-balancing hyperparameters

$$\boldsymbol{dpp\_diversity} = \boldsymbol{det(K)},$$
$$K = \frac{1}{1 + dist(\mathbf{c}_i, \mathbf{c}_j)}$$

Practical considerations

$$C(x) = \underset{c_1,\ldots,c_k}{\arg\min} \frac{1}{k}\sum_{i=1}^{k} \boldsymbol{yloss}(f(c_i), y) + \frac{\lambda_1}{k}\boldsymbol{dist}(c_i, x) - \lambda_2 \boldsymbol{dpp\_diversity}(c_1, \ldots, c_k)$$

- Incorporate additional feasibility properties
  a) Sparsity
  b) User constraints
- Choice of yloss – hinge loss
- Separate categorical and continuous distance functions
- Relative scale of mixed features

Python library
DiCE
(**Di**verse **C**ounterfactual **E**xplanations)
https://github.com/interpretml/DiCE

```python
# Using sklearn backend
m = dice_ml.Model(model=model, backend="sklearn")
# Using method=random for generating CFs
exp = dice_ml.Dice(d, m, method="random")
```

```python
e1 = exp.generate_counterfactuals(x_train[0:1], total_CFs=2, desired_class="opposite")
e1.visualize_as_dataframe(show_only_changes=True)
```

Query instance (original outcome : 0)

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	38	Private	HS-grad	Married	Blue-Collar	White	Male	44	0

Diverse Counterfactual set (new outcome: 1.0)

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	67.0	-	Masters	-	-	Other	-	-	1
1	66.0	-	Prof-school	-	-	Other	-	-	1

```python
# Restricting age to be between [20,30] and Education to be either {'Doctorate', 'Prof-school'}.
e3 = exp.generate_counterfactuals(x_train[0:1],
                                  total_CFs=2,
                                  desired_class="opposite",
                                  permitted_range={'age':[20,30], 'education':['Doctorate', 'Prof-school']})
e3.visualize_as_dataframe(show_only_changes=True)
```

```
Query instance (original outcome : 0)
```

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	38	Private	HS-grad	Married	Blue-Collar	White	Male	44	0

```
Diverse Counterfactual set (new outcome: 1.0)
```

	age	workclass	education	marital_status	occupation	race	gender	hours_per_week	income
0	28.0	Self-Employed	Doctorate	-	Professional	-	Female	21.0	1
1	27.0	Self-Employed	Doctorate	-	Professional	-	Female	50.0	1

3. Evaluating fairness is a causal problem.

The Importance of Modeling Data Missingness in Algorithmic Fairness: A Causal Perspective. AAAI 2021

Slides Credit: Naman Goel.

Fair Machine Learning

Common approach:

1. Get a big training dataset, different rows containing observed outcomes for different feature values.

Fair Machine Learning

Common approach:

1. Ge rved ou

4	12	4	21	1	4	3	3	1	49	3	1	2	1	1	0	0	1	0	0	1	0	1	0	1
1	42	2	79	1	4	3	4	2	45	3	1	2	1	1	0	0	0	0	0	0	0	0	1	1
1	24	3	49	1	3	3	4	4	53	3	2	2	1	1	1	0	1	0	0	0	0	0	1	2
4	36	2	91	5	3	3	4	4	35	3	1	2	2	1	0	0	1	0	0	0	0	1	0	1
4	24	2	28	3	5	3	4	2	53	3	1	1	1	1	0	0	1	0	0	1	0	0	1	1
2	36	2	69	1	3	3	2	3	35	3	1	1	2	1	0	1	1	0	1	0	0	0	0	1
4	12	2	31	4	4	1	4	1	61	3	1	1	1	1	0	0	1	0	0	1	0	1	0	1
2	30	4	52	1	1	4	2	3	28	3	2	1	1	1	1	0	1	0	0	1	0	0	0	2
2	12	2	13	1	2	2	1	3	25	3	1	1	1	1	1	0	1	0	1	0	0	0	1	2
1	48	2	43	1	2	2	4	2	24	3	1	1	1	1	0	0	1	0	1	0	0	0	1	2
2	12	2	16	1	3	2	1	3	22	3	1	1	2	1	0	0	1	0	0	1	0	0	1	1
1	24	4	12	1	5	3	4	3	60	3	2	1	1	1	1	0	1	0	0	1	0	1	0	2
1	15	2	14	1	3	2	4	3	28	3	1	1	1	1	1	0	1	0	1	0	0	0	1	1
1	24	2	13	2	3	2	2	3	32	3	1	1	1	1	0	0	1	0	0	1	0	1	0	2

$\underbrace{\hspace{20em}}_{\text{Features for different individuals}}$

Outcome
(e.g. loan paid back or not)

Fair Machine Learning

Common approach:

1. Get a big training dataset, different rows containing observed outcomes for different feature values.
2. Select an appropriate fairness metric (e.g. equal error rates).
3. Apply state-of-the-art algorithm on this dataset to train a classifier with fairness constraints.
4. Deploy the trained classifier to make future decisions.

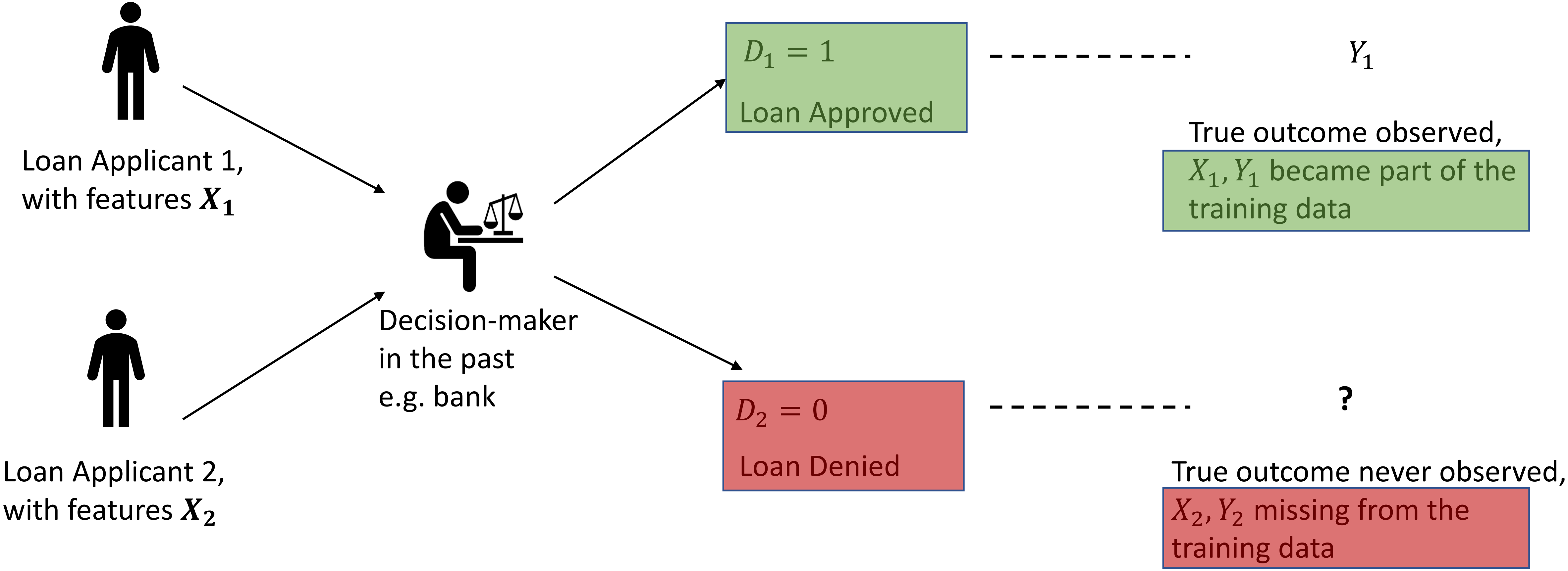
Fair Machine Learning

This common approach, proposed and shown to work on *benchmark datasets* in fair machine learning papers, doesn't work in practice unfortunately.

There is no guarantee that the supposedly fair classifer will actually take fair decisions in the real-world.

Reason: *Missingness in Training Data*.

Missingness in Training Data



Missingness in Training Data

- Training data, even if it contains objective ground truth outcome and infinitely many samples, is *one-sided* due to systematic *censoring by past decisions.*

Empirical Implications

	(Pleiss et al. 2017) with FPR Constraints	
COMPAS Dataset	Train FPRD −0.00155	Test FPRD 0.061

Difference in Test and Train Fairness of Fair ML Algorithms *under Training Data with Missingness*

All implementations and respective hyper-parameters settings taken from examples provided in IBM AI Fairness 360 Library

Empirical Implications

	(Pleiss et al. 2017) with FPR Constraints	
COMPAS Dataset	Train FPRD −0.00155	Test FPRD 0.061
ADULT Dataset	Train FPRD −0.00724	Test FPRD 0.0725

Difference in Test and Train Fairness of Fair ML Algorithms *under Training Data with Missingness*

All implementations and respective hyper-parameters settings taken from examples provided in IBM AI Fairness 360 Library

Empirical Implications

	(Pleiss et al. 2017) with FPR Constraints		(Pleiss et al. 2017) with FNR Constraints	
COMPAS Dataset	Train FPRD −0.00155	Test FPRD 0.061	Train FNRD 0.0056	Test FNRD 0.099
ADULT Dataset	Train FPRD −0.00724	Test FPRD 0.0725	Train FNRD 0.00295	Test FNRD 0.0377

Difference in Test and Train Fairness of Fair ML Algorithms *under Training Data with Missingness*

All implementations and respective hyper-parameters settings taken from examples provided in IBM AI Fairness 360 Library

Empirical Implications

	(Pleiss et al. 2017) with FPR Constraints		(Pleiss et al. 2017) with FNR Constraints	
COMPAS Dataset	Train FPRD −0.00155	Test FPRD 0.061	Train FNRD 0.0056	Test FNRD 0.099
ADULT Dataset	Train FPRD −0.00724	Test FPRD 0.0725	Train FNRD 0.00295	Test FNRD 0.0377

Kallus and Zhou (2018) made similar observations for Hardt et al (2016)'s algorithm on NYPD SQF dataset.

Difference in Test and Train Fairness of Fair ML Algorithms *under Training Data with Missingness*

All implementations and respective hyper-parameters settings taken from examples provided in IBM AI Fairness 360 Library

Empirical Implications

	Postprocessing Approach			
	(Pleiss et al. 2017) with FPR Constraints		(Pleiss et al. 2017) with FNR Constraints	
COMPAS Dataset	Train FPRD −0.00155	Test FPRD 0.061	Train FNRD 0.0056	Test FNRD 0.099
ADULT Dataset	Train FPRD −0.00724	Test FPRD 0.0725	Train FNRD 0.00295	Test FNRD 0.0377

Kallus and Zhou (2018) made similar observations for Hardt et al (2016)'s algorithm on NYPD SQF dataset.

Difference in Test and Train Fairness of Fair ML Algorithms *under Training Data with Missingness*

All implementations and respective hyper-parameters settings taken from examples provided in IBM AI Fairness 360 Library

Empirical Implications

	Postprocessing Approach				Inprocessing Approach	
	(Pleiss et al. 2017) with FPR Constraints		**(Pleiss et al. 2017) with FNR Constraints**		**(Kamiran et al. 2012) with SP Constraints**	
COMPAS Dataset	Train FPRD −0.00155	Test FPRD 0.061	Train FNRD 0.0056	Test FNRD 0.099	Train SPD 0.0229	Test SPD 0.2651
ADULT Dataset	Train FPRD −0.00724	Test FPRD 0.0725	Train FNRD 0.00295	Test FNRD 0.0377	Train SPD −0.0390	Test SPD −0.1137

Kallus and Zhou (2018) made similar observations for Hardt et al (2016)'s algorithm on NYPD SQF dataset.

* Similar observations for Celis et al (2019)' algorithm.

Difference in Test and Train Fairness of Fair ML Algorithms *under Training Data with Missingness*

All implementations and respective hyper-parameters settings taken from examples provided in IBM AI Fairness 360 Library

Empirical Implications

	Postprocessing Approach				Inprocessing Approach		Preprocessing Approach	
	(Pleiss et al. 2017) with FPR Constraints		**(Pleiss et al. 2017) with FNR Constraints**		**(Kamiran et al. 2012) with SP Constraints**		**(Kamiran and Calders 2012)**	
COMPAS Dataset	Train FPRD −0.00155	Test FPRD 0.061	Train FNRD 0.0056	Test FNRD 0.099	Train SPD 0.0229	Test SPD 0.2651	Train EOD 0.0111	Test EOD −0.2266
ADULT Dataset	Train FPRD −0.00724	Test FPRD 0.0725	Train FNRD 0.00295	Test FNRD 0.0377	Train SPD −0.0390	Test SPD −0.1137	Train EOD 0.0293	Test EOD −0.1327

Kallus and Zhou (2018) made similar observations for Hardt et al (2016)'s algorithm on NYPD SQF dataset.

* Similar observations for Celis et al (2019)' algorithm.

Difference in Test and Train Fairness of Fair ML Algorithms *under Training Data with Missingness*

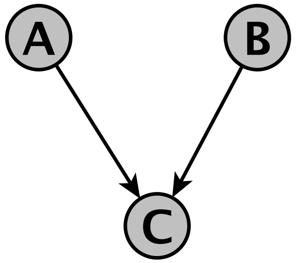
All implementations and respective hyper-parameters settings taken from examples provided in IBM AI Fairness 360 Library

Related Work

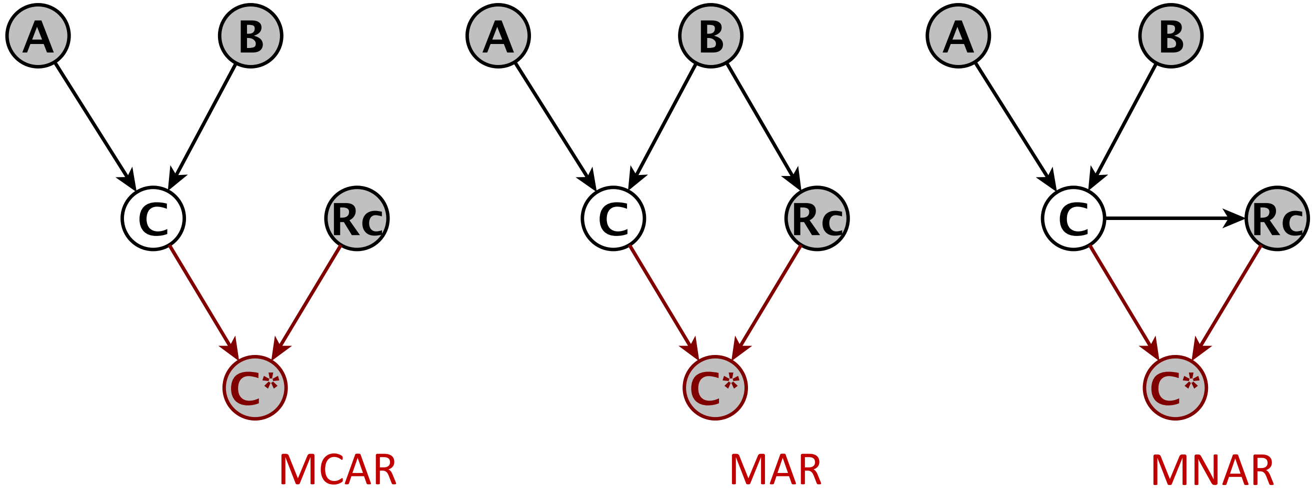
Missingness in Variables	D affects Y?	Related Work
Only Y is missing in the rows corresponding to $D = 0$	No	(Lakkaraju et al. 2017)
Entire rows (X, Y, Z) corresponding to $D = 0$ are missing.	No	This Paper + (Kallus and Zhou 2018; Ensign et al. 2018; Kilbertus et al. 2020)
Only Y^0 is not observed, X, Y, Z have no missingness.	Yes	(Jung et al. 2018; Coston et al. 2020; Kallus and Zhou 2019)

Our focus is on general identifiability and implications for fair machine learning

Causal Graphs for Data Missingness
(Karthika Mohan and Judea Pearl, 2019)



#	A	B	C	R_C
1	A_1	B_1	C_1	OFF
2	A_2	B_2	C_2	OFF
3	A_3	B_3		ON
4	A_4	B_4		ON
5	A_5	B_5	C_5	OFF
6	A_5	B_6		ON
7	A_6	B_7	C_7	OFF



R_C : Missigness mechanism variable for variable C

$$C^* = \text{C} \quad \text{if } \mathrm{R_C} = \mathrm{OFF}$$
$$C^* = \text{missing} \quad \text{if } \mathrm{R_C} = \mathrm{ON}$$

Notation

- X – Non-sensitive Features
- Z – Sensitive Attribute
- D – Past Binary Decision
- Y – Outcome

- U – Unobserved features

- $\hat{Y}$ – Classifier Prediction

Fairness

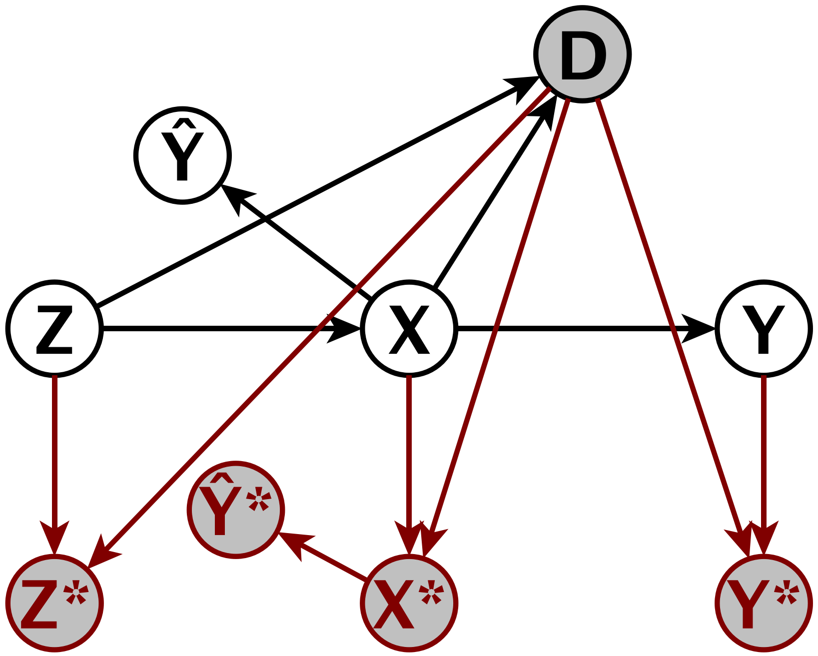
- Demographic Parity (DP)

$$P(\hat{Y} = 1 \mid Z = b) = P(\hat{Y} = 1 \mid Z = w)$$

- Equality of Opportunity (EOP)

$$P(\hat{Y} = 1 \mid Y = 1, Z = b) = P(\hat{Y} = 1 \mid Y = 1, Z = w)$$

Estimating Equality of Opportunity Fairness constraint with Incomplete Data



d-separation (Pearl 1988)

What fairness algorithms actually estimate from incomplete data

$$P(\hat{Y}^* | Y^*, Z^*)$$

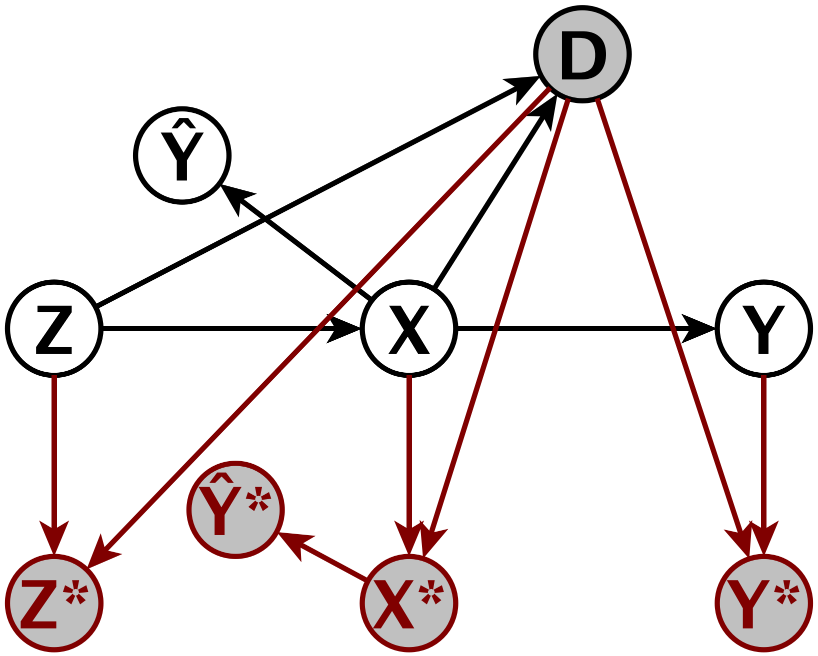
$$= P(\hat{Y} | Y, Z, D = 1)$$

$$\neq P(\hat{Y} | Y, Z)$$

Quantity we need

because $\hat{Y} \not\perp\!\!\!\perp D \mid Y, Z$

Estimating Demographic Parity Constraints with Incomplete Data



$$P(\hat{Y}^*|Z^*) \neq P(\hat{Y}|Z)$$

More general results

Fairness Algorithms: Demographic parity, equality of opportunity

$P(Y|X), P(Y|X,Z)$, and/or $P(X), P\ (X,Z)$.

Missingness Mechanisms:

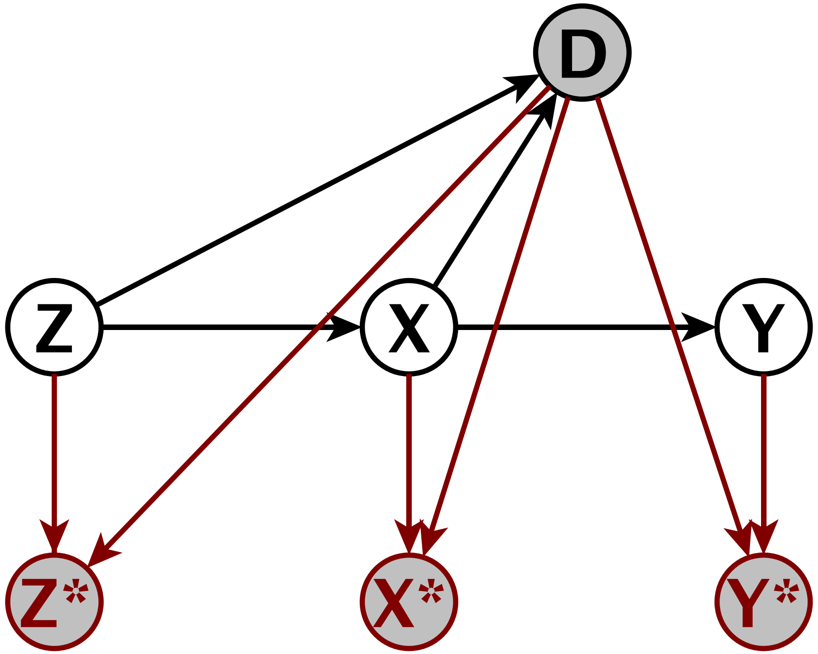
Fully-Automated, Human, Machine-Aided Decision Making

Recovering joint distribution of features is impossible in almost all cases of missingness caused by past decisions. Conditional distributions (risk scores) may be recoverable in some cases, depending on the causal graph.

Missingness caused by human (or machine aided) decision making is more challenging than that caused by fully automated decision making.

Importance of Modeling Data Missingness in Algorithmic Fairness: A Causal Perspective. AAAI 2021

Censoring due to Fully Automated Decisions



$P(Y|X)$ ✔

$P(Y|X,Z)$ ✔

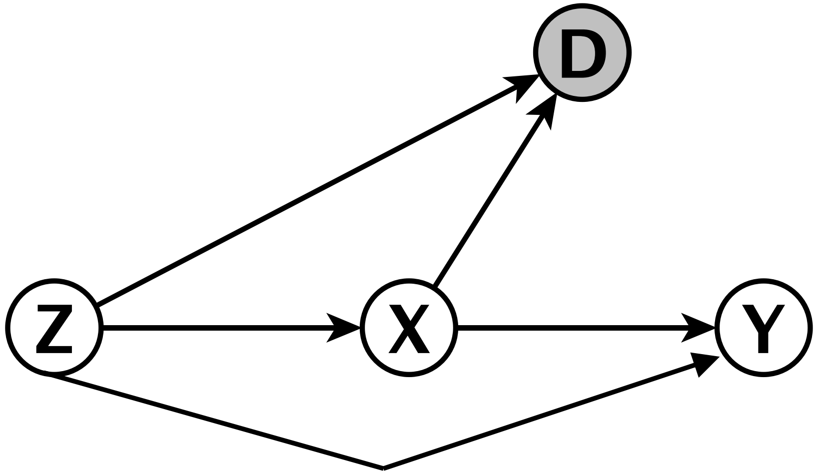
$P(X,Z)$

$P(X)$

✘ **and non-recoverable**

Non-recoverable: No matter how many data samples are provided, there exists no estimator to get the correct probability distribution.

Censoring due to Fully Automated Decisions



$P(Y|X)$ ✗

$P(Y|X,Z)$ ✓

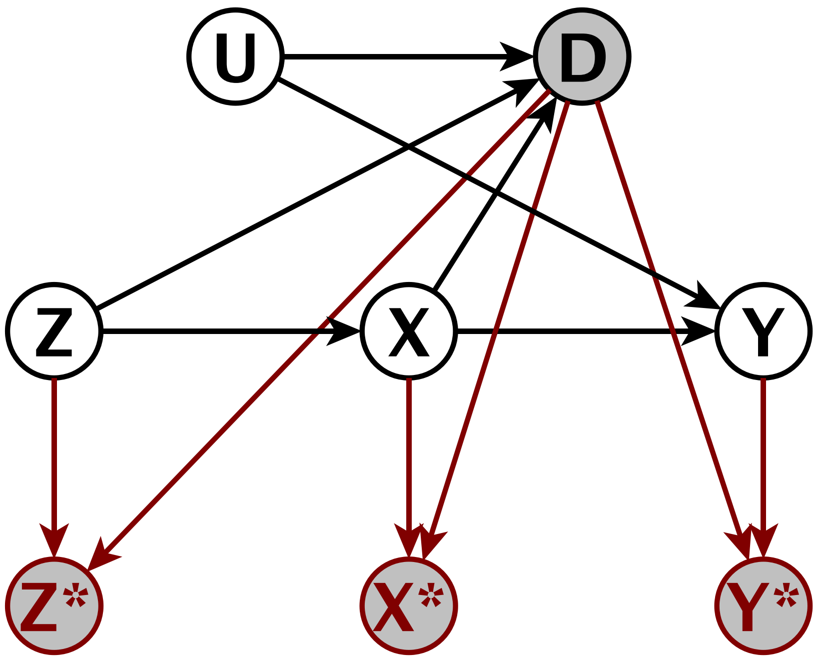
$P(X)$ ✗ **and non-recoverable**

Censoring due to Human Decisions

Distinguishing characteristic:

Use of unobserved features in decision making.

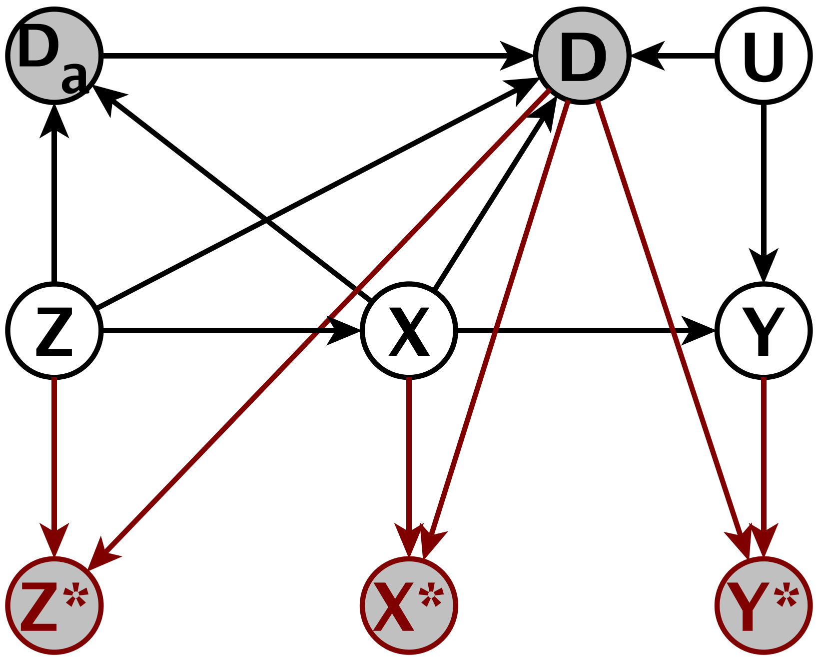
Censoring due to Human Decisions



$P(Y|X,Z)$ ✖ **and non-recoverable**

$P(X)$ ✖ **and non-recoverable**

Censoring due to Machine-Aided Decisions



$P(Y|X,Z)$ ✖ **and non-recoverable**

$P(X)$ ✖ **and non-recoverable**

Summary

- Recovering joint distribution of features is impossible in almost all cases of missingness caused by past decisions.

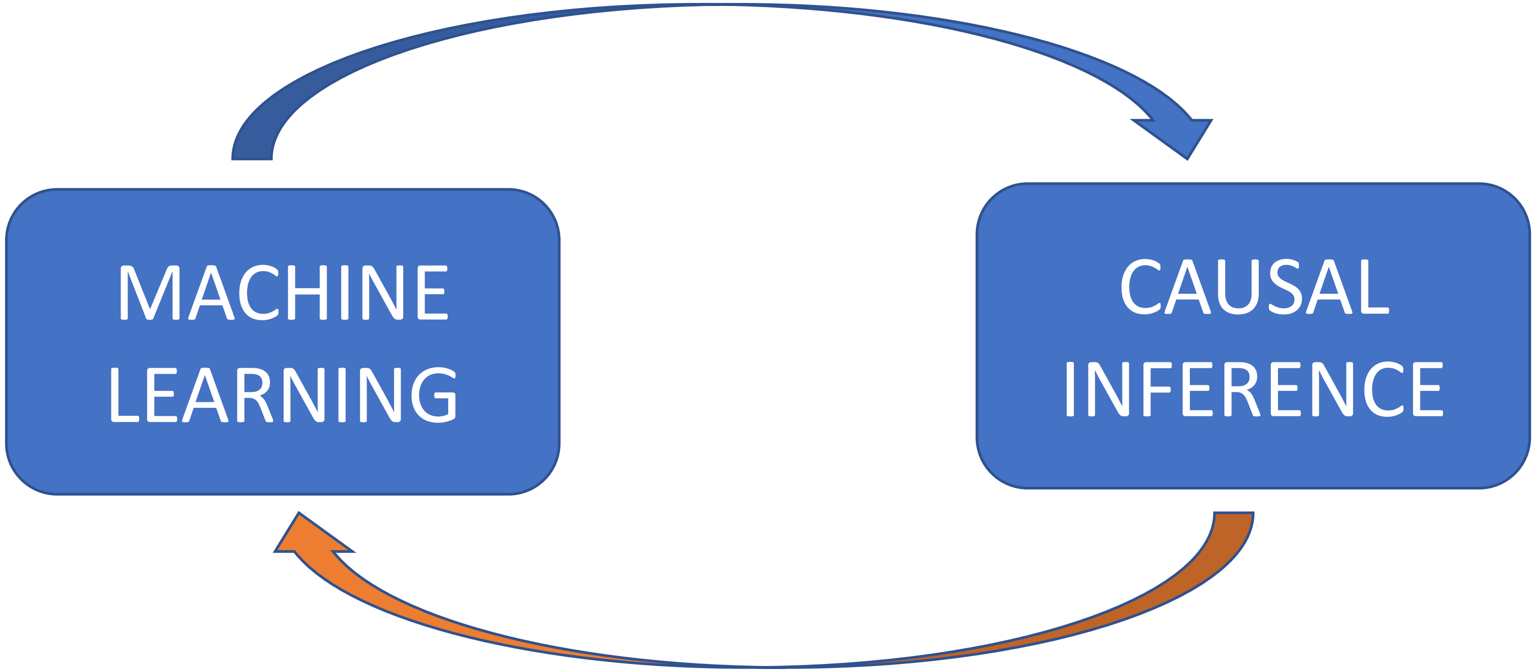
- Conditional distributions (risk scores) may be recoverable in some cases, depending on the causal graph for missingness.

Both conditional and joint distributions are used in several state of the art fairness algorithms.

- Missingness caused by human (or machine aided) decision making is more challenging than that caused by fully automated decision making.

- Small change in causal structure may lead to very different conclusions.

Better Estimation and
Refutation of Causal Effect
github.com/microsoft/dowhy



- Better Generalization & Robustness of ML models
- Principled framework for Fairness and Explanation

- **Amit Sharma**
- Microsoft Research India
- **@amt_shrma**
- www.amitsharma.in