Causal Machine Learning: Necessary Ingredient for building generalizable models

Intro to decision-making using DoWhy

Amit Sharma
Microsoft Research India
Twitter: @amt_shrma

www.amitsharma.in
(A) **Cow: 0.99**, Pasture: 0.99, Grass: 0.99, No Person: 0.98, Mammal: 0.98

(B) No Person: 0.99, Water: 0.98, Beach: 0.97, Outdoors: 0.97
Machine learning has a correlation problem

ML models should have captured the **causal** features (e.g., cow’s pixels, stop sign)

**Failure Reason:** Independent and identically distributed (IID) assumption.

![Diagram](image.png)
Learnt correlations become a bigger problem for decision-making

**Prediction:** If we obtain a new input, what will be the outcome?  
E.g., *what will be the heart attack risk for a new person?*

**Decision-making:** If we change a feature for a given input, how will that impact the outcome?  
E.g., *if a person starts exercising, how much does it change the heart attack risk?*

<table>
<thead>
<tr>
<th>Training Data</th>
<th>$P(X, Y)$</th>
<th>$f(x)$</th>
<th>Validation Data</th>
<th>$P(X, Y)$</th>
<th>Decision</th>
<th>Test Data</th>
<th>$P^*(X, Y)$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IID</strong> (in distribution)</td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
<td><img src="image" alt="Diagram" /></td>
</tr>
<tr>
<td><strong>OOD</strong> (out of distribution)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Today’s session

PART I:
• Out-of-distribution: A key problem for machine learning
• Why causality is necessary for OOD generalization?
• Causal prediction in practice
  • (Conditional) independence regularization
  • Counterfactual augmentations
  • Domain knowledge regularization

PART II:
• Decision-making: A classic causal inference problem
• Important to explicitly state and validate assumptions
• Four steps of causal inference: Model, Identify, Estimate, Refute
  • Code demo using DoWhy
Part I: Causal reasoning is necessary for out-of-distribution generalization


State-of-the-art for OOD generalization

**Domain generalization**

**Multiple domains:** Assume access to data from multiple distributions

- Learn invariant patterns across the different sources
  - Invariant Risk Minimization (Arjovsky et al., 2019)

**Group generalization**

**Single domain:** Assume access to group attributes for each input

- Equalize accuracy across groups/maximize worst-group accuracy
  - Group-DRO (Sagawa et al., 2020), (Ahmed et al. 2021)
Sobering state of SoTA algorithms
Sobering state of SoTA algorithms

Ye et al., OoD-Bench, CVPR 2022
Sobering state of SoTA algorithms

Rotated MNIST

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>15°</td>
<td>60°</td>
<td>90°</td>
</tr>
<tr>
<td>Y=0</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>Y=1</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

Colored MNIST

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.9</td>
<td>0.8</td>
</tr>
<tr>
<td>Y=0</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>Y=1</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

Ye et al., OoD-Bench, CVPR 2022
Sobering state of SoTA algorithms

Ye et al., OoD-Bench, CVPR 2022
Sobering state of SoTA algorithms

No method can surpass ERM on all kinds of shifts!

Ye et al., OoD-Bench, CVPR 2022
Sobering state of SoTA algorithms

IID

Spurious correlation b/w category and lighting

Unseen data shift unseen azimuth values

Best methods are not consistent over different datasets and shifts

Wiles et al., ICLR 2022
What if different distribution shifts co-exist?

<table>
<thead>
<tr>
<th>Satellite Image (x)</th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year / Region (d)</td>
<td>2002 / Americas</td>
<td>2009 / Africa</td>
</tr>
<tr>
<td>Building / Landtype (y)</td>
<td>shopping mall</td>
<td>multi-unit residential</td>
</tr>
</tbody>
</table>

Koh et al., WILDS, ICML 2021
What if different distribution shifts co-exist?

Accuracy decreases further for all algorithms.
I. Causal reasoning can explain this failure

[single shift] Explain results from causal perspective

• Different distribution shifts arise due to differences in data-generating process (DGP)
  • Leading to different independence constraints
• No single independence constraint can work for all shifts
II. Causal reasoning can provide a better algorithm

[single shift] Explain results from causal perspective
- Different distribution shifts arise due to differences in data-generating process (DGP)
  - Leading to different independence constraints
  - No single independence constraint can work for all shifts

[multi-shift] Can we develop an algorithm that generalizes to individual as well as multi-attribute shifts?
- We propose Causally Adaptive Constraint Minimization (CACM) to model the causal relationships in DGP
Representation of shifts using causal graph

Causal DAG to specify multi-attribute shifts
Causal DAG to specify multi-attribute shifts

Representation of shifts using causal graph

Observed variables $X, Y$

Causal DAG to specify multi-attribute shifts
Representation of shifts using causal graph

Observed variables $X, Y$
Causal features $X_c$

Causal DAG to specify multi-attribute shifts
Representation of shifts using causal graph

Causal DAG to specify multi-attribute shifts

Observed variables $X, Y$

Causal features $X_c$

Attributes $A_{ind}, A_{ind}' \cup E$ s.t. $A_{ind} \cup A_{ind}' \cup \{E\} = A$
Representation of shifts using causal graph

Causal DAG to specify multi-attribute shifts

Observed variables $X, Y$
Causal features $X_c$
Attributes $A_{ind}, A_{ind}', E$ st $A_{ind} \cup A_{ind}' \cup \{E\} = A$

- independent of label
- correlated with label
- domain attribute
Representation of shifts using causal graph

Causal DAG to specify multi-attribute shifts

Different $Y - \overline{A_{ind}}$ relationships
Representation of shifts using causal graph

Causal DAG to specify multi-attribute shifts

Different $Y \rightarrow A_{ind}$ relationships
Representation of shifts using causal graph

Causal DAG to specify multi-attribute shifts

Different $Y \rightarrow A_{ind}$ relationships
Causal DAG to specify multi-attribute shifts

Different $Y \rightarrow A_{ind}$ relationships

Representation of shifts using causal graph

Causal

Confounded

Selected
Back to the MNIST example

### Rotation

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y=0</td>
<td>15°</td>
<td></td>
</tr>
<tr>
<td>Y=1</td>
<td>60°</td>
<td></td>
</tr>
<tr>
<td>Y=0</td>
<td>90°</td>
<td></td>
</tr>
<tr>
<td>Y=1</td>
<td>60°</td>
<td></td>
</tr>
</tbody>
</table>

### Color

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y=0</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>Y=1</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>Y=0</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>Y=1</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

### Causal + Independent

\[ \mathcal{A}_{\text{cause}} \cup \mathcal{A}_{\text{ind}} \]

### Col+Rot

<table>
<thead>
<tr>
<th></th>
<th>(0.9,15°)</th>
<th>(0.8,60°)</th>
<th>(0.1,90°)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y=0</td>
<td>[ \mathcal{A}_{\text{cause}} ]</td>
<td>[ \mathcal{A}_{\text{cause}} ]</td>
<td>[ \mathcal{A}_{\text{cause}} ]</td>
</tr>
<tr>
<td>Y=1</td>
<td>[ \mathcal{A}_{\text{cause}} ]</td>
<td>[ \mathcal{A}_{\text{cause}} ]</td>
<td>[ \mathcal{A}_{\text{cause}} ]</td>
</tr>
</tbody>
</table>
Generalization to multi-attribute shifts

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Color</th>
<th>Rotation</th>
<th>Col+Rot</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>30.9 ± 1.6</td>
<td>61.9 ± 0.5</td>
<td>25.2 ± 1.3</td>
</tr>
<tr>
<td>IRM</td>
<td>50.0 ± 0.1</td>
<td>61.2 ± 0.3</td>
<td>39.6 ± 6.7</td>
</tr>
<tr>
<td>MMD</td>
<td>29.7 ± 1.8</td>
<td>62.2 ± 0.5</td>
<td>24.1 ± 0.6</td>
</tr>
<tr>
<td>C-MMD</td>
<td>29.4 ± 0.2</td>
<td>62.3 ± 0.4</td>
<td>32.2 ± 7.0</td>
</tr>
<tr>
<td><strong>CACM</strong></td>
<td><strong>70.4 ± 0.5</strong></td>
<td><strong>62.4 ± 0.4</strong></td>
<td><strong>54.1 ± 0.3</strong></td>
</tr>
</tbody>
</table>

**CACM** outperforms on individual as well as combination of shifts
The CACM Approach

Identifying the correct regularizer under multi-attribute shifts
The CACM Approach

Identifying the correct regularizer under multi-attribute shifts

I. Derive correct independence constraints for $\mathbf{x}_c$ based on causal graph

II. Apply the constraints as regularizer to standard ERM loss.
Step I: Deriving independence constraints

Predictor $g(x) = g_1(\phi(x))$

Representation $\phi$ should follow same conditional independence constraints as $X_c$
Step I: Deriving independence constraints

Predictor \( g(x) = g_1(\phi(x)) \)
Representation \( \phi \) should follow same conditional independence constraints as \( X_c \)

**Proposition 3.1.** Given a dataset \((x_i, a_i, y_i)_{i=1}^n\) and a causal DAG over \( \langle X_c, X, A, Y \rangle \) such that \( X_c \) is the only variable (or set of variables) that causes \( Y \) and is not independent of \( X \), then the conditional independence constraints satisfied by \( X_c \) are necessary for a risk-invariant predictor.
Step I: Deriving independence constraints

Different $Y - A_{\text{ind}}$ relationships lead to different constraints
Step I: Deriving independence constraints

**Causal**

\[ X_c \perp\!\!\!\!\!\!\perp A_{cause} \mid Y, E \quad \checkmark \]
\[ X_c \perp\!\!\!\!\!\!\perp A_{cause} \mid E \quad \xmark \]

**Confounded**

\[ X_c \perp\!\!\!\!\!\!\perp A_{conf} \mid Y, E \quad \xmark \]
\[ X_c \perp\!\!\!\!\!\!\perp A_{conf} \mid E \quad \checkmark \]
Step I: Deriving independence constraints

**Theorem 3.1.**

1. *Independent:* $X_c \perp \perp A_{ind}$; $X_c \perp \perp E$; $X_c \perp \perp A_{ind}|Y$; $X_c \perp \perp A_{ind}|E$; $X_c \perp \perp A_{ind}|Y, E$

2. *Causal:* $X_c \perp \perp A_{cause}|Y$; $X_c \perp \perp E$; $X_c \perp \perp A_{cause}|Y, E$

3. *Confounded:* $X_c \perp \perp A_{conf}$; $X_c \perp \perp E$; $X_c \perp \perp A_{conf}|E$

4. *Selected:* $X_c \perp \perp A_{sel}|Y$; $X_c \perp \perp A_{sel}|Y, E$
Step I: Deriving independence constraints

**Theorem 3.1.**

1. **Independent:** $X_c \perp \perp A_{ind}; \ X_c \perp \perp E; \ X_c \perp \perp A_{ind}|Y; \ X_c \perp \perp A_{ind}|E; X_c \perp \perp A_{ind}|Y, E$

2. **Causal:** $X_c \perp \perp A_{cause}|Y; \ X_c \perp \perp E; X_c \perp \perp A_{cause}|Y, E$

3. **Confounded:** $X_c \perp \perp A_{conf}; \ X_c \perp \perp E; X_c \perp \perp A_{conf}|E$

4. **Selected:** $X_c \perp \perp A_{sel}|Y; \ X_c \perp \perp A_{sel}|Y, E$

No (conditional) independence constraint valid for all shifts
Theoretical evidence for past work: A fixed conditional independence constraint cannot work for all datasets

Ye et al., OoD-Bench, CVPR 2022; Wiles et al., ICLR 2022
Theoretical evidence for previous results: A fixed conditional independence constraint cannot work for all datasets

**Theorem 3.2.** For any predictor algorithm for $Y$ that uses a single type of (conditional) independence constraint, there exists a realized graph $\mathcal{G}$ and a corresponding training dataset such that the learned predictor cannot be a risk-invariant predictor across distributions in $\mathcal{P}_\mathcal{G}$. 

Ye et al., OoD-Bench, CVPR 2022; Wiles et al., ICLR 2022
Step II: Applying regularization penalty

Constraint: \( X_c \perp\!\!\!\!\perp A_{\text{cause}} \mid Y, E \)  \[\text{[Causal shift]}\]

\[
\text{RegPenalty}_{A_{\text{cause}}} = \sum_{|E|} \sum_{y \in Y} \sum_{i=1}^{A_{\text{cause}}} \sum_{j>i} \text{MMD} \left( P(g_1(\phi(x))|a_{i,\text{cause}},y), P(g_1(\phi(x))|a_{j,\text{cause}},y) \right)
\]

\[
g_1, \phi = \arg\min_{g_1, \phi} \quad L(g_1(\phi(x)), y) + \lambda^* (\text{RegPenalty}_{A_{\text{cause}}})
\]
Finally, CACM Algorithm for general graphs

**Phase I:** Derive correct independence constraints

1. For every observed variable \( A \in \mathcal{A} \) in the graph, check whether \((X_c, A)\) are d-separated.
   
   \[ X_c \perp\!\!\!\perp A \text{ is a valid constraint} \]

2. If not, check whether \((X_c, A)\) are d-separated conditioned on any subset \( \mathcal{A}_s \) of the remaining observed variables in \( \mathcal{A} \setminus \{A\} \).
   
   \[ X_c \perp\!\!\!\perp A | \mathcal{A}_s \text{ is a valid constraint} \]
Finally, CACM Algorithm for general graphs

**Phase II:** Apply regularization penalty using constraints derived

If \( X_c \perp A \)

\[
\text{Reg Penalty}_A = \sum_{|E|} \sum_{i=1}^{\frac{|A|}{2}} \sum_{j>i} \text{MMD} \left( P(g_1(\phi(x)|A_i), P(g_1(\phi(x)|A_j) \right)
\]

If \( X_c \perp A | A_s \)

\[
\text{Reg Penalty}_A = \sum_{|E|} \sum_{a \in A_s} \sum_{i=1}^{\frac{|A|}{2}} \sum_{j>i} \text{MMD} \left( P(g_1(\phi(x)|A_i, a), P(g_1(\phi(x)|A_j, a) \right)
\]

\[
\text{Reg Penalty} = \sum_{A \in A} \text{Penalty}_A
\]

\[
g_1, \phi = \arg\min_{g_1, \phi} L(g_1(\phi(x), y) + \lambda^{*}(\text{Reg Penalty})
\]
Empirical evaluation

Spurious correlation b/w category and lighting ($A_{cause}$)

Unseen data shift unseen azimuth values ($A_{ind}$)

small NORB dataset

- Multi-class (5 classes)
- Multi-valued attributes
- Real objects
Correct constraint derived from CG matters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>lighting $A_{cause}$</th>
<th>azimuth $A_{ind}$</th>
<th>lighting+azimuth $A_{cause} \cup A_{ind}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>65.5 ± 0.7</td>
<td>78.6 ± 0.7</td>
<td>64.0 ± 1.2</td>
</tr>
<tr>
<td>IRM</td>
<td>66.7 ± 1.5</td>
<td>75.7 ± 0.4</td>
<td>61.7 ± 1.5</td>
</tr>
<tr>
<td>VREx</td>
<td>64.7 ± 1.0</td>
<td>77.6 ± 0.5</td>
<td>62.5 ± 1.6</td>
</tr>
<tr>
<td>MMD</td>
<td>66.6 ± 1.6</td>
<td>76.7 ± 1.1</td>
<td>62.5 ± 0.3</td>
</tr>
<tr>
<td>CORAL</td>
<td>64.7 ± 1.5</td>
<td>77.2 ± 0.7</td>
<td>62.9 ± 0.3</td>
</tr>
<tr>
<td>DANN</td>
<td>64.6 ± 1.4</td>
<td><strong>78.6 ± 0.7</strong></td>
<td>60.8 ± 0.7</td>
</tr>
<tr>
<td>C-MMD</td>
<td>65.8 ± 0.8</td>
<td>76.9 ± 1.0</td>
<td>61.0 ± 0.9</td>
</tr>
<tr>
<td>CDANN</td>
<td>64.9 ± 0.5</td>
<td>77.3 ± 0.3</td>
<td>60.8 ± 0.9</td>
</tr>
</tbody>
</table>

ERM outperforms all DG algorithms!
Correct constraint derived from CG matters

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>lighting $A_{cause}$</th>
<th>azimuth $A_{ind}$</th>
<th>lighting+azimuth $A_{cause} \cup A_{ind}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>65.5 ± 0.7</td>
<td>78.6 ± 0.7</td>
<td>64.0 ± 1.2</td>
</tr>
<tr>
<td>IRM</td>
<td>66.7 ± 1.5</td>
<td>75.7 ± 0.4</td>
<td>61.7 ± 1.5</td>
</tr>
<tr>
<td>VREx</td>
<td>64.7 ± 1.0</td>
<td>77.6 ± 0.5</td>
<td>62.5 ± 1.6</td>
</tr>
<tr>
<td>MMD</td>
<td>66.6 ± 1.6</td>
<td>76.7 ± 1.1</td>
<td>62.5 ± 0.3</td>
</tr>
<tr>
<td>CORAL</td>
<td>64.7 ± 1.5</td>
<td>77.2 ± 0.7</td>
<td>62.9 ± 0.3</td>
</tr>
<tr>
<td>DANN</td>
<td>64.6 ± 1.4</td>
<td>78.6 ± 0.7</td>
<td>60.8 ± 0.7</td>
</tr>
<tr>
<td>C-MMD</td>
<td>65.8 ± 0.8</td>
<td>76.9 ± 1.0</td>
<td>61.0 ± 0.9</td>
</tr>
<tr>
<td>CDANN</td>
<td>64.9 ± 0.5</td>
<td>77.3 ± 0.3</td>
<td>60.8 ± 0.9</td>
</tr>
<tr>
<td><strong>CACM</strong></td>
<td><strong>85.4 ± 0.5</strong></td>
<td><strong>80.5 ± 0.6</strong></td>
<td><strong>69.6 ± 1.6</strong></td>
</tr>
</tbody>
</table>

**CACM** provides upto 20% improvement
Incorrect constraints hurt generalization!

$X_c \perp\!\!\!\!\!\!\perp A_{cause} \ | \ E \quad \times$

$X_c \perp\!\!\!\!\!\!\perp A_{cause} \ | \ Y, E \quad \checkmark$

Diagram: Object (Obj) → $X_c$ → $Y$ → $A_{cause}$

Nodes labeled: $X_c$, $X$, $Y$, $E$, $A_{cause}$

Causal relationships:
- $X_c \perp\!\!\!\!\!\!\perp A_{cause} \ | \ E$
- $X_c \perp\!\!\!\!\!\!\perp A_{cause} \ | \ Y, E$
Incorrect constraints hurt generalization!

\[ X_c \perp \perp A_{\text{cause}} \mid E \] \quad ❌

\[ X_c \perp \perp A_{\text{cause}} \mid Y, E \] \quad ✔️

OOD Accuracy of incorrect constraint decreases as regularization penalty is increased.
Incorrect constraints hurt generalization!

Causal

Confounded
Incorrect constraints hurt generalization!

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Causal</th>
<th>Confounded</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_c \perp \perp A \mid E$</td>
<td>$29.7 \pm 3.8$</td>
<td>$62.4 \pm 1.9$</td>
</tr>
<tr>
<td>$X_c \perp \perp A \mid Y, E$</td>
<td>$94.1 \pm 0.5$</td>
<td>$56.0 \pm 1.0$</td>
</tr>
</tbody>
</table>
Incorrect constraints hurt generalization!

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Causal</th>
<th>Confounded</th>
</tr>
</thead>
</table>
| $X_c \perp\!
\perp A_{cause} \mid E$ | 29.7 ± 3.8 | 62.4 ± 1.9 |
| $X_c \perp\!
\perp A_{cause} \mid Y, E$ | 94.1 ± 0.5 | 56.0 ± 1.0 |
Takeaways

• Necessary to model causal relationships in the data-generating process for OOD generalization
  • Algorithms based on single, fixed constraint fail to generalize

• Do not need full causal graph
  • Only the attributes and their relationship with outcome variable

• Algorithm with causally adaptive constraints outperforms existing OOD algorithms
  • Works equally well on single dataset, datasets with multiple domains, etc.
Beyond CACM: Counterfactual data augmentation

• Generate synthetic data with different attributes that breaks the correlation

• What if we change only the spurious attribute while keeping the rest of input identical?

• Theoretically consistent with recovering $X_C$

• In practice, use GANs/Adversarially learnt inference to build generative model


Beyond CACM: Counterfactual data augmentation

- Generate synthetic data with different attributes that breaks the correlation
- What if we change only the spurious attribute while keeping the rest of input identical?
- Theoretically consistent with recovering $X_c$
- In practice, use GANs/Adversarially learnt inference to build generative model

\[ g_1, \phi = \arg\min_{g_1, \phi} L(g_1(\phi(x)), y) + \lambda^* \sum_{x, x'} (\phi(x) - \phi(x')^2) \]

**ICML 2021.** Domain generalization using causal matching. Mahajan, Tople, Sharma.

Beyond CACM: Using causal domain knowledge

In addition to structure, people may know the *shape* of causal effect function (causal prior).

**Shape:** diminishing return, U-shaped, Z-shaped, etc.

**Type:** direct causal effect, indirect effect, total effect

Beyond CACM: Using causal domain knowledge

In addition to structure, people may know the shape of causal effect function (causal prior).

**Shape:** diminishing return, U-shaped, Z-shaped, etc.

**Type:** direct causal effect, indirect effect, total effect

Can enforce it by,

1. Measuring causal effect of a feature on the model’s prediction
2. Matching the model’s gradient to provided causal prior’s gradient

PART II: Practical causal inference with DoWhy

- DoWhy Library: https://github.com/py-why/dowhy
From prediction to decision-making

**Decision-making:** Acting/intervening on a feature
- Interventions break correlations used by supervised ML
  - Special kind of OOD generalization
- The feature with the highest importance score in a prediction model,
  - Need not be the best feature to act on
  - May not even affect the outcome at all!

For decision-making, need to find the features that **cause the outcome** & estimate how the outcome would change if the features are changed.
Observed distribution

\[ P(Y|A) \]

Interventional Distribution

\[ P(Y|do(A = 1)) \]
Real World

Counterfactual World

User Interests → User Fatigue → Past Clicks → Rec. Algo → Y

User Interests → User Fatigue → Past Clicks → Rec. Algo → Y
Two Fundamental Challenges for Causal Inference

1. Assumptions
   - Multiple causal graphs can fit the same data distribution. **Do we have the right graph?**

2. Evaluation
   - Target distribution is unobserved. **No easy “cross-validation”**.
We built DoWhy library to make assumptions front-and-center of any causal analysis.
- Transparent declaration of assumptions
- Evaluation of those assumptions, to the extent possible

One of the most popular causal libraries on GitHub
(>1.3M downloads, 5K stars, 690+ forks)

Taught in third-party tutorials and courses: O’Reilly, PyData, Northeastern, ...
Used by many companies and researchers.

Maintained by independent org py-why with >50 contributors

An end-to-end platform for doing causal inference
Formulate correct estimand based on causal assumptions?

Estimate causal effect

Check robustness?

EconML, CausalML, CausalImpact, tmle,...
1. **Model causal mechanisms**
   - Construct a causal graph based on domain knowledge

2. **Identify the target estimand**
   - Formulate correct estimand based on the causal model

3. **Estimate causal effect**
   - Use a suitable method to estimate effect

4. **Refute estimate**
   - Check robustness of estimate to assumption violations

**Input Data**

- <action, outcome, other variables>

**Domain Knowledge**

- DoWhy

**Causal effect**
DoWhy provides a general API for the four steps of causal inference

1. **Modeling**: Create a causal graph to encode assumptions.

2. **Identification**: Formulate what to estimate.

3. **Estimation**: Compute the estimate.

4. **Refutation**: Validate the assumptions.

We’ll discuss the four steps and show a code example using DoWhy.
I. Model the assumptions using a causal graph

Convert domain knowledge to a formal model of causal assumptions

- $A \rightarrow B$ or $B \rightarrow A$?

- Causal graph implies conditional statistical independences
  - E.g., $A \perp C$, $D \perp A \mid B$, ...
  - Identified by $d$-separation rules [Pearl 2009]

- These assumptions significantly impact the causal estimate we’ll obtain.
Example Graph

Assumption 1: User fatigue does not affect user interests

Assumption 2: Past clicks do not directly affect outcome

Assumption 3: Treatment does not affect user fatigue.

..and so on.
Intervention is represented by a new graph

Interventional graph:
All edges to Treatment $T$ removed, *keeping everything else the same*.

Represents new data distribution, referred as $do(T)$

Causal effect: $P(Y|do(T))$
II. Identification: Formulate desired quantity and check if it is estimable from given data

Observed data generated by this graph

Want to answer questions about data that will be generated by intervention graph

How to represent quantities from right hand graph (e.g., $P(Y|do(T))$) using only statistical observations from data generated from left hand graph?
Randomized Experiments and Backdoor criterion

• Observed graph is same as intervention graph in randomized experiment!
  • Treatment $T$ is already generated independent of all other features
  • $\Rightarrow P(Y|do(T)) = P(Y|T)$

• **Backdoor Intuition:** Generalize by simulating randomized experiment
  • When treatment $T$ is caused by other features, $Z$, adjust for their influence to simulate a randomized experiment

*Backdoor Adjustment formula*

$$p(Y|do(T)) = \sum_Z p(Y|T, Z)p(Z)$$
Many kinds of identification methods

**Graphical constraint-based methods**
- Randomized and natural experiments
- Adjustment Sets
  - Backdoor, “towards necessity”
- Front-door criterion
- Mediation formula

**Identification under additional non-graphical constraints**
- Instrumental variables
- Regression discontinuity
- Difference-in-differences

Many of these methods can be used through DoWhy.
III. Estimation: Compute the causal effect

Estimation uses observed data to compute the target probability expression from the Identification step.

For common identification strategies using adjustment sets,
\[
E[Y|do(T = t), W = w] = E[Y|T = t, W = w]
\]
assuming W is a valid adjustment set.

• For binary treatment,

  Causal Effect = \( E[Y|T = 1, W = w] - E[Y|T = 0, W = w] \)

**Goal:** Estimating conditional probability \( Y|T=t \) when all confounders \( W \) are kept constant.
**Simple Matching:** Match data points with the same confounders and then compare their outcomes

Control

Treatment (Cycling)
Simple Matching: Match data points with the same confounders and then compare their outcomes

Identify pairs of treated \((j)\) and untreated individuals \((k)\) who are similar or identical to each other.

\[
\text{Match} := \text{Distance}(W_j, W_k) < \epsilon
\]

- Paired individuals have almost the same confounders.

Causal Effect =

\[
\sum_{(j,k)\in Match} (y_j - y_k)
\]
Challenges of building a good estimator

- **Variance**: If we have a stringent matching criterion, we may obtain very few matches and the estimate will be unreliable.

- **Bias**: If we relax the matching criterion, we obtain many more matches but now the estimate does not capture the target estimand.

- **Uneven treatment assignment**: If very few people have treatment, leads to both high bias and variance.

Need better methods to navigate the bias-variance tradeoff.
Depending on the dataset properties, different estimation methods can be used:

<table>
<thead>
<tr>
<th>Simple Conditioning</th>
<th>Outcome-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Matching</td>
<td>• Double ML</td>
</tr>
<tr>
<td>• Stratification</td>
<td>• T-learner</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Propensity Score-Based</th>
<th>Loss-Based</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Rubin 1983]</td>
<td>• X-learner</td>
</tr>
<tr>
<td>• Propensity Matching</td>
<td>• R-learner</td>
</tr>
<tr>
<td>• Inverse Propensity Weighting</td>
<td>[Kunzel et al. 2017]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Synthetic Control</th>
<th>Threshold-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Abadie et al.]</td>
<td>• Difference-in-differences</td>
</tr>
</tbody>
</table>

All these methods can be called through DoWhy. *(directly or through the Microsoft EconML library)*
IV. Robustness Checks: Test robustness of obtained estimate to violation of assumptions

Obtained estimate depends on many (untestable) assumptions.

**Model:**
- Did we miss any unobserved variables in the assumed graph?
- Did we miss any edge between two variables in the assumed graph?

**Identify:**
- Did we make any parametric assumption for deriving the estimand?

**Estimate:**
- Is the assumed functional form sufficient for capturing the variation in data?
- Do the estimator assumptions lead to high variance?
Best practice: Do refutation/robustness tests for as many assumptions as possible

UNIT TESTS
Model:
• Conditional Independence Test
Identify:
• D-separation Test
Estimate:
• Bootstrap Refuter
• Data Subset Refuter

INTEGRATION TESTS
Test all steps at once.
• Placebo Treatment Refuter
• Dummy Outcome Refuter
• Random Common Cause Refuter
• Sensitivity Analysis
• Simulated Outcome Refuter /Synth-validation [Schuler et al. 2017]

All these refutation methods are implemented in Do Why.
Caveat: They can refute a given analysis, but cannot prove its correctness.
Example 1: Conditional Independence Refuter

Through its edges, each causal graph implies certain conditional independence constraints on its nodes. \([d\text{-separation}, \text{Pearl 2009}]\)

**Model refutation:** Check if the observed data satisfies the assumed model’s independence constraints.

- Use an appropriate statistical test for independence \([\text{Heinze-Demel et al. 2018}]\).
- If not, the model is incorrect.

\[
\begin{align*}
A \perp B \\
A \perp T | W \\
B \perp T | W
\end{align*}
\]
Example 2: Placebo Treatment ("A/A") Refuter

Q: What if we can generate a dataset where the treatment does not cause the outcome? Then a correct causal inference method should return an estimate of zero.

Placebo Treatment Refuter:
Replace treatment variable $T$ by a randomly generated variable (e.g., Gaussian).

• Rerun the causal inference analysis.
• If the estimate is significantly away from zero, then analysis is incorrect.
Example 3: Add Unobserved Confounder to check sensitivity of an estimate

Q: What if there was an unobserved confounder that was not included in the causal model? Check how sensitive the obtained estimate is after introducing a new confounder.

Unobserved Confounder Refuter:

• Simulate a confounder based on a given correlation $\rho$ with both treatment and outcome.
  • Maximum Correlation $\rho$ is based on the maximum correlation of any observed confounder.
• Re-run the analysis and check if the sign/direction of estimate flips.
Walk-through of the 4 steps using the DoWhy Python library
Problem: Estimating the effect of a customer loyalty rewards program

What is the impact of offering the customer loyalty program on total sales?

If the current members had not signed up for the program, how much less would they have spent?

ATT: Average treatment effect on the treated (customers who signed up for the program)

You can try out this example on Github: 
github.com/microsoft/dowhy/blob/master/docs/source/example_notebooks/dowhy_example_effect_of_memberrewards_program.ipynb
Step 1: Modeling. Create causal graph to encode assumptions.

```
model = dowhy.CausalModel(data=df_i_signupmonth,
    graph=causal_graph.replace("\n", " "),
    treatment="treatment",
    outcome="post_spends")
```
Step 2: Identification. Formulate what to estimate

```python
identified_estimand = model.identify_effect(proceed_when_unidentifiable=True)
print(identified_estimand)
```
Step 3: Estimation. Compute the estimate

```python
estimate = model.estimate_effect(identified_estimand,
                               method_name="backdoor.propensity_score_matching",
                               target_units="att")
print(estimate)
```
Step 4: Refutation. Validate the assumptions

```python
refutation = model.refute_estimate(identified_estimand, estimate, method_name="placebo_treatment_refuter", placebo_type="permute", num_simulations=20)
print(refutation)
```

Refute: Use a Placebo Treatment
Estimated effect: 0.03963044006804
New effect: 0.6054947726720156
p value: 0.24154316295878647
**Future:** Extending the four-step API to other causal tasks

- A unified, extensible API for causal inference that allows external implementations for the 4 steps
  - Supports invoking estimation methods from external libraries such as EconML and CausalML.

```python
dml_estimate = model.estimate_effect(identified_estimand,
  method_name="backdoor.econml.dml.DMLCateEstimator",
  target_units = lambda df: df["X0"]>1,
  confidence_intervals=True,
```

- Extend the same 4-step API for,
  - Graphical causal model inference
  - Learning a causal graph from data (experimental)
  - Causal prediction models (coming soon!)
Summary: DoWhy, a library that focuses on causal assumptions and their validation

**Goal:** A unified API for causal tasks, just like PyTorch or Tensorflow for predictive ML.

Growing open-source community: > 50 contributors
- Roadmap: More powerful refutation tests, counterfactual prediction.
- Please contribute! Join the community on Discord or Github.

**Resources**
- DoWhy Library: [https://github.com/py-why/dowhy](https://github.com/py-why/dowhy)
Conclusion: Causal reasoning is necessary for both prediction and decision-making

- Causal models require assumptions, but not the full graph
- Can achieve superior results by simple, standard assumptions
  - CACM: attributes and their correlation type
  - DoWhy: confounders based on time order

- Big open question: Evaluation of causal models
  - Important to track progress in the field, for widespread adoption

thank you– Amit Sharma (@amt_shrma)