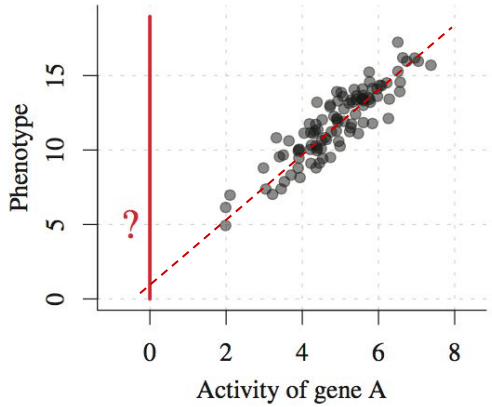
Machine Learning for Causal Inference

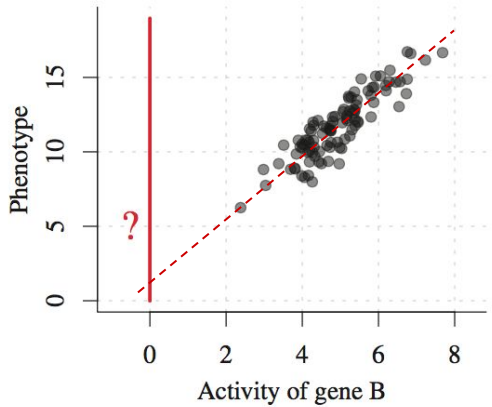
By Sorawit Saengkyongam, Data Scientist at Agoda and GDE in Machine Learning

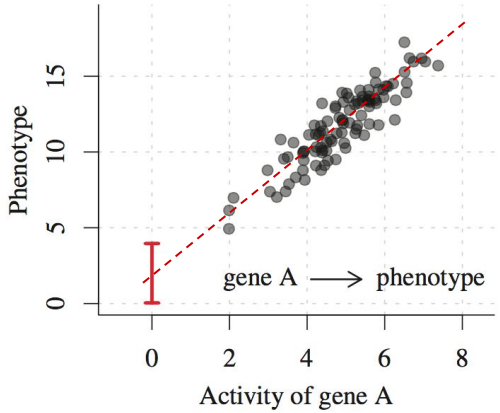
Talk outline

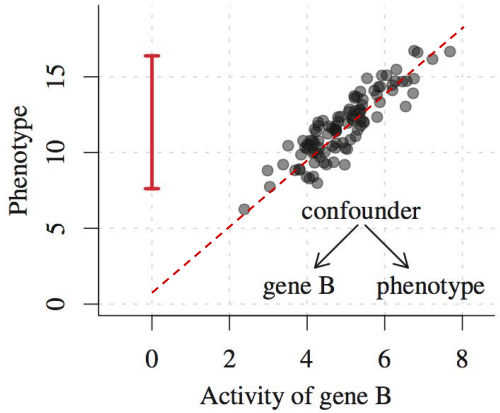
- Introduction to Causal Inference
- Machine Learning for Counterfactual Predictions
  - Bayesian Additive Regression Trees
  - Deep Balanced Neural Networks
  - Deep Instrumental Variable
- Challenges
- What else ?

Why do we care ?









Seeing != Doing

The best answer in this case
"**I don't know**"

Elements of Causal Inference Foundations and Learning Algorithms (Jonas Peter et al. 2017)

We often deal with causal problems

- Recommender systems
- Drug design
- Pricing
- Self-driving cars
- Lending systems

Causal questions as Counterfactual questions

- Will new recommendation algorithm bring more customers ?
  - Counterfactuals: old vs new algorithm

- Does this medication improve patients health
  - Counterfactuals: taking vs not taking

- Is driving off a cliff a good idea ?
  - Counterfactuals: ….

Potential Outcome Framework

- Each unit (patient, customer, student ..)
  has two potential outcomes: (y_i^0, y_i^1)
  - y_i^0: outcome of the i^{th} unit if the control is given "**control outcome**"
  - y_i^1: outcome of the i^{th} unit if the treatment is given "**treatment outcome**"

- Treatment effect for unit i
  $= y_i^1 - y_i^0$

- Often interested in Average Treatment Effect: $E[y_i^1 - y_i^0]$

Hypothetical Example - Effect of treatment on blood pressure

Unit	female	age	treatment	potential outcome y_i^0	potential outcome y_i^1	observed outcome y_i
Audrey	1	40	0	**140**		140
Anna	1	40	0	**140**		140
Bob	0	50	0	**150**		150
Bill	0	50	0	**150**		150
Caitlin	1	60	1		**155**	155
Cara	1	60	1		**155**	155
Dave	0	70	1		**160**	160
Doug	0	70	1		**160**	160

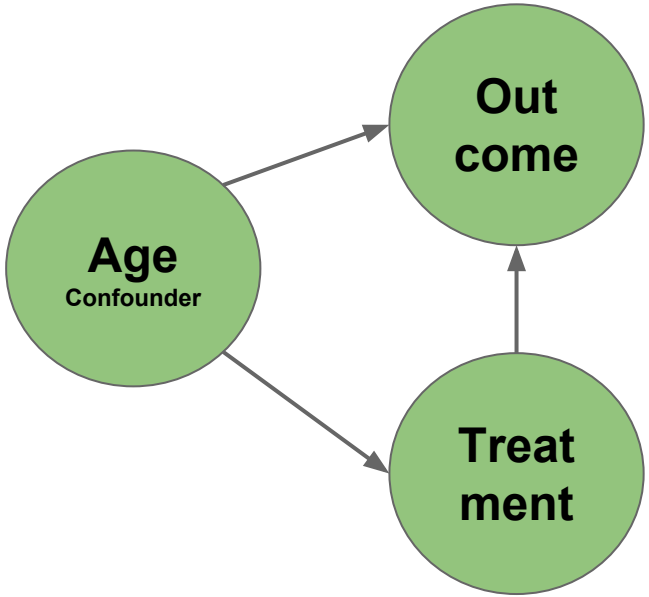
Source: Jennifer Hill

$\text{Mean}(y_i^1 - y_i^0) = -7.5$

$\text{Mean}((y_i \mid \text{treatment}=1) - (y_i \mid \text{treatment}=0)) = 12.5$

The fundamental problem of causal inference:

We only ever observe one of the two outcomes



- How to deal with the problem
  - Randomization -> very expensive and time consuming
  - Statistical Adjustment (with assumptions)

Statistical Adjustment

- Make some assumptions
  - Major one -> Ignorability: $Y^0, Y^1 \perp Z$ (treatment) | X (covariates)
- Under ignorability

$$E(Y1) - E(Y0) = E\{ E(Y|Z = 1, X) \} - E\{ E(Y|Z = 0, X) \}$$

$$= E\{ E(Y|Z = 1, X) - E(Y|Z = 0, X) \}$$

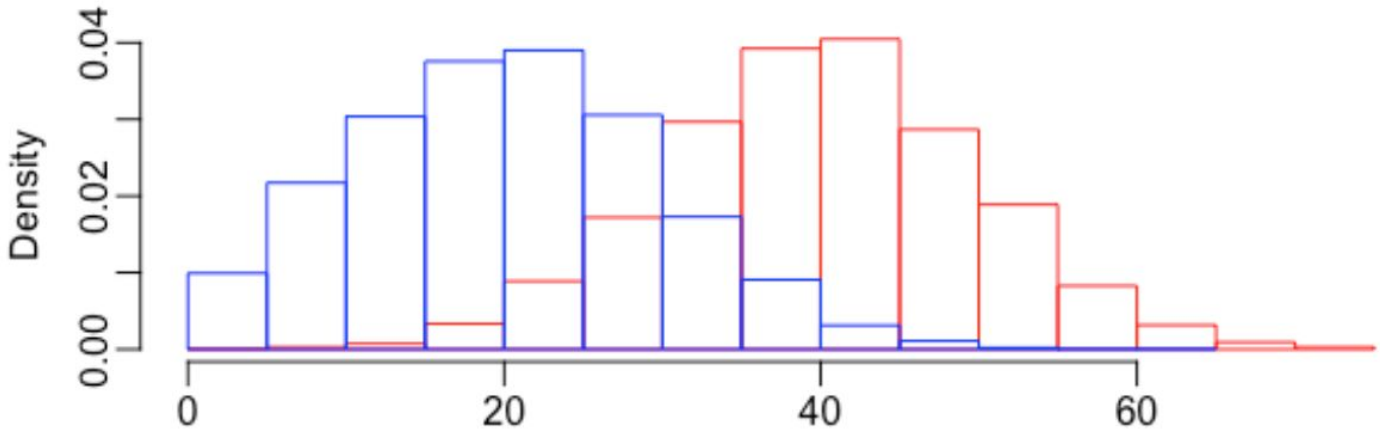
$$= E\{ f(1, x) - f(0, x) \}$$

- Estimate the outcome function f(z, x) using a model known as **Response Surface Modeling**

Consider a simple example

- Effect of an enrichment program on subsequent test scores
- Suppose that exposure to the program is
  - Determined based on one pre-test score and
  - Is probabilistic, as in:

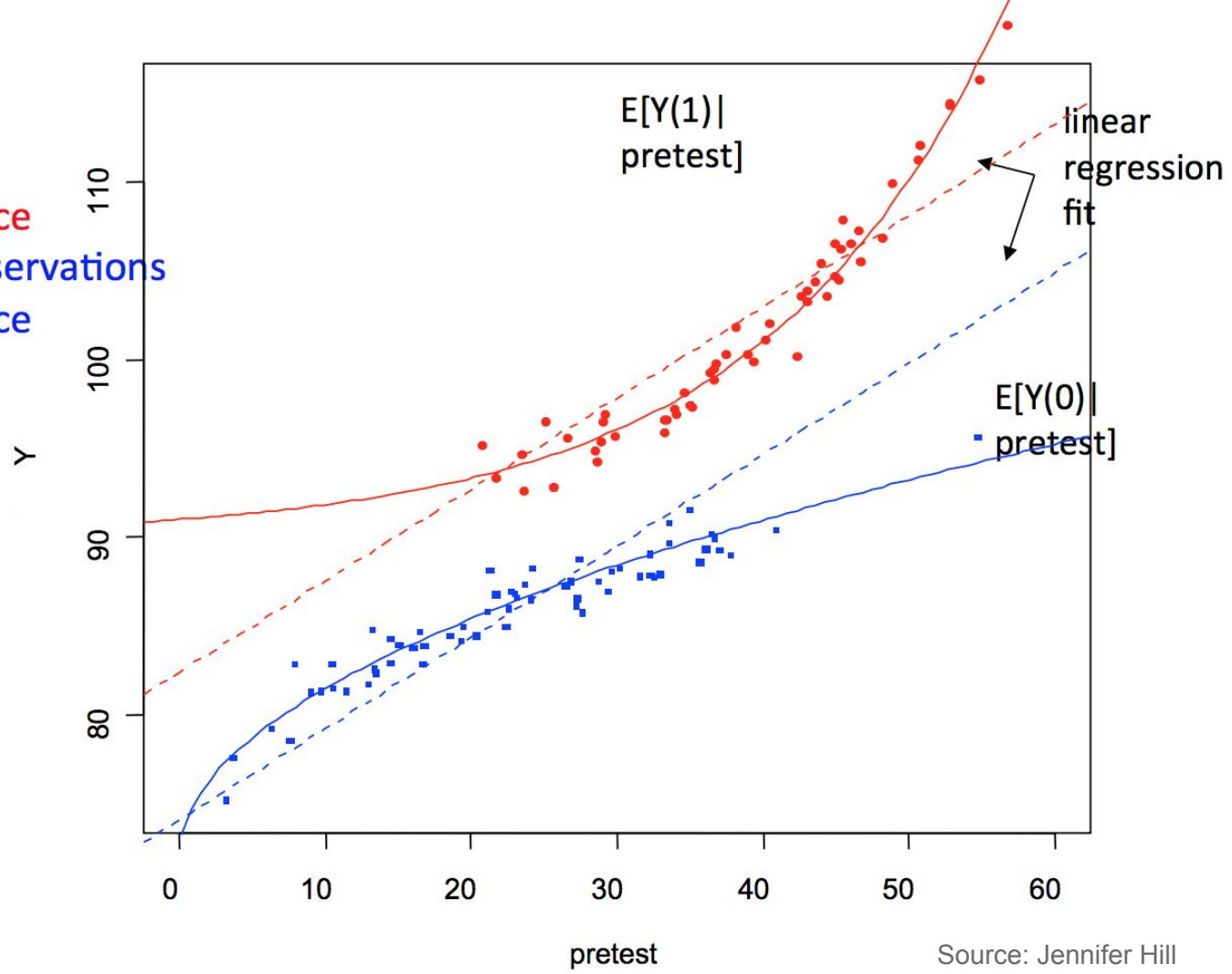
red for treated
blue for controls



Source: Jennifer Hill

red for treatment observations and response surface

blue for control observations and response surface



Source: Jennifer Hill

Machine Learning for Counterfactual Predictions

- We wish to model **f(1, x) = E(Y|Z=1,X)** and **f(0, x) = E(Y|Z=0,X)**

- In principle any regression method can work:
  use Z_i (treatment) as a feature, predict for both Z_i =0, Z_i=1

- Linear regression is far too weak for most problems of interest!

Bayesian Additive Regression Trees (BART)

Bayesian Nonparametric Modeling for Causal Inference, Jennifer L. Hill (2012)

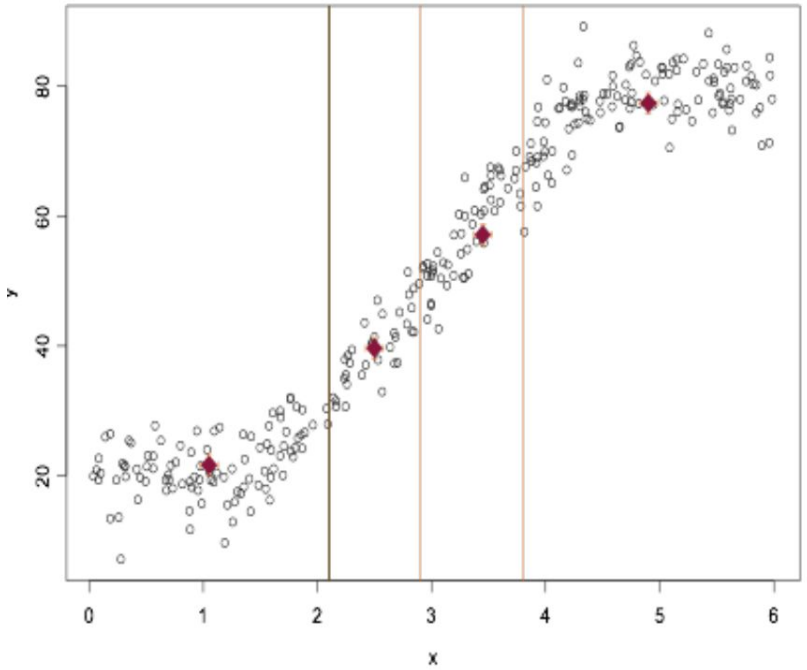
Bayesian Additive Regression Trees (BART)

- Goal: Estimate surface response using BART

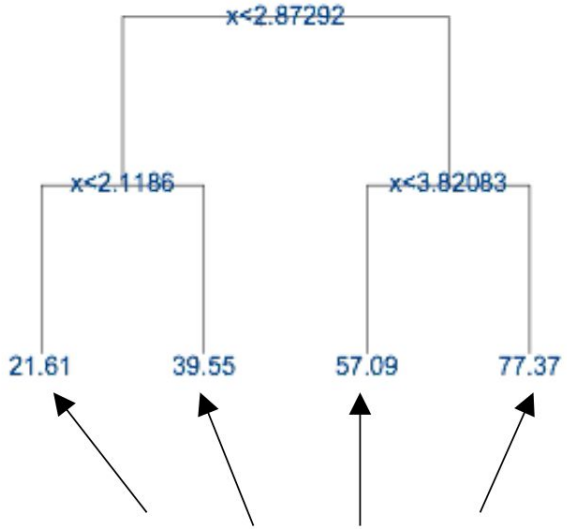
- BART is a Bayesian form of boosted regression trees

Will find interactions, non-linearities. Not the best for additive models.

Regression trees



Progressively splits the data into more and more homogenous subsets. Within each of these subsets the mean of y can be calculated



"terminal nodes"

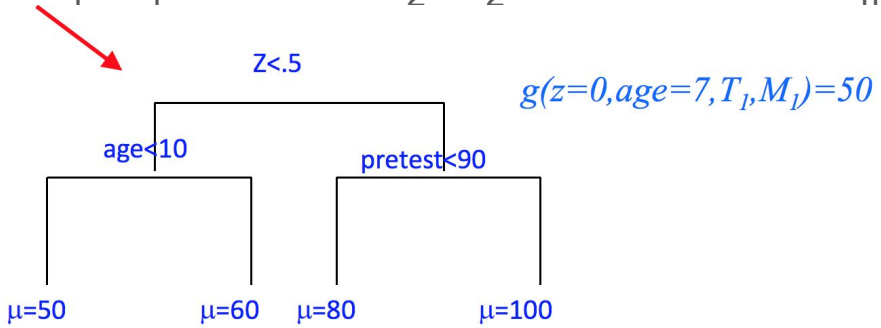
Source: Jennifer Hill

Boosted Regression Trees

- Builds on the idea of a treed model to create a “sum-of-trees” model

  Let $\{T_j, M_j\}$ j=1,…,m, be a set of tree models
  T_j denotes the j^{th} tree,
  M_j denotes the means from the terminal nodes from the j^{th} tree,

  $$f(z, x) = g(z,x,T_1,M_1) + g(z,x,T_2,M_2) + \ldots + g(z,x,T_m,M_m)$$



Boosted Regression Trees

Boosting is great for prediction but …

– Requires ad-hoc choice of tuning parameters to force trees to be weak learners (shrink each mean towards zero)

– How estimate uncertainty? Generally, people use bootstrapping which can be cumbersome and time consuming

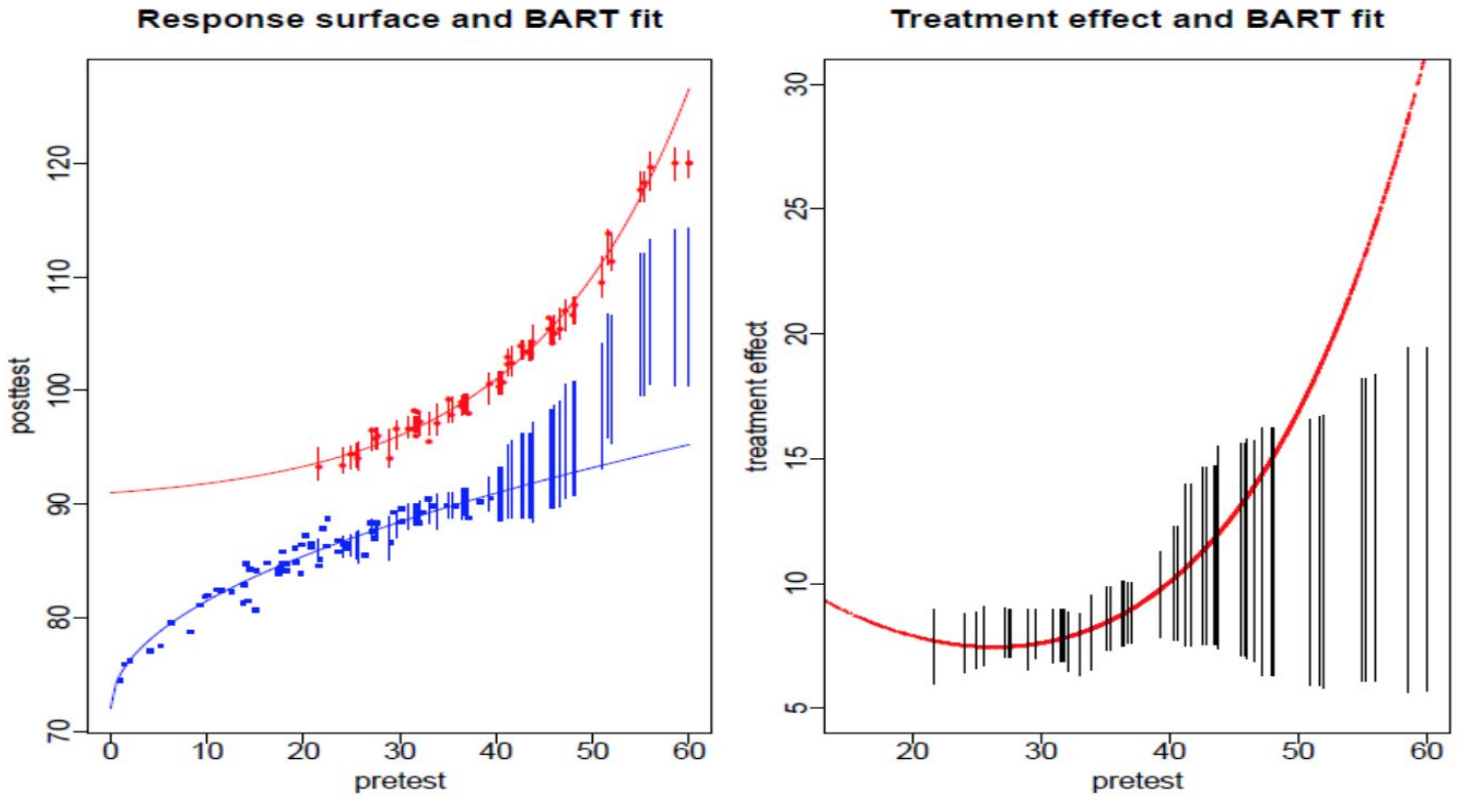
How BART differs from boosting

BART can be thought of as a stochastic alternative to boosting.

It differs because:

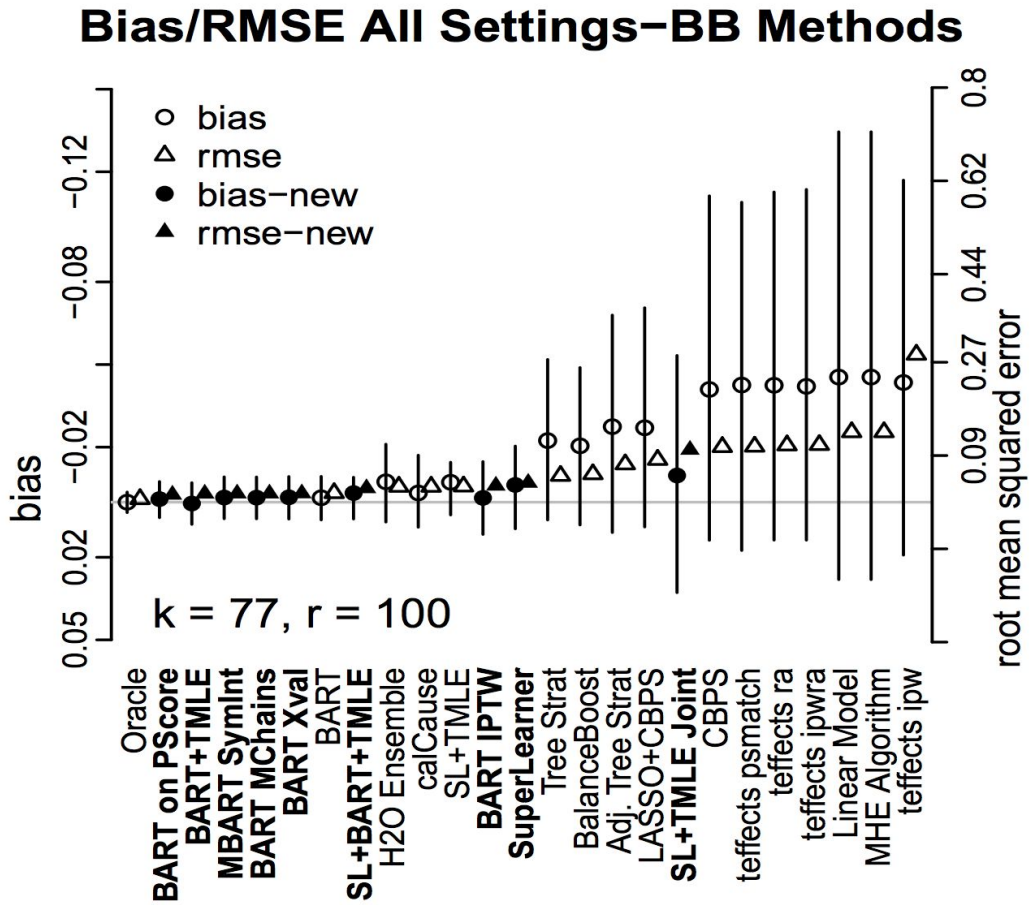
- **f(z,x)** is a random variable
- Using an MCMC algorithm, we sample **f(z,x)** it from a posterior
  - Allows for **uncertainty** in our model
- Avoids **overfitting** by the prior specification that shrinks towards a simple fit:
  - Priors tend towards small number of trees (“weak learners”)
  - Each tree is pruned using priors

Causal Inference using BART



Source: Jennifer Hill

Causal Inference using BART



Automated versus do-it-yourself methods
for causal inference:
Lessons learned from a data analysis competition
Vincent Dorie et al. (2018)

Handle imbalance problem

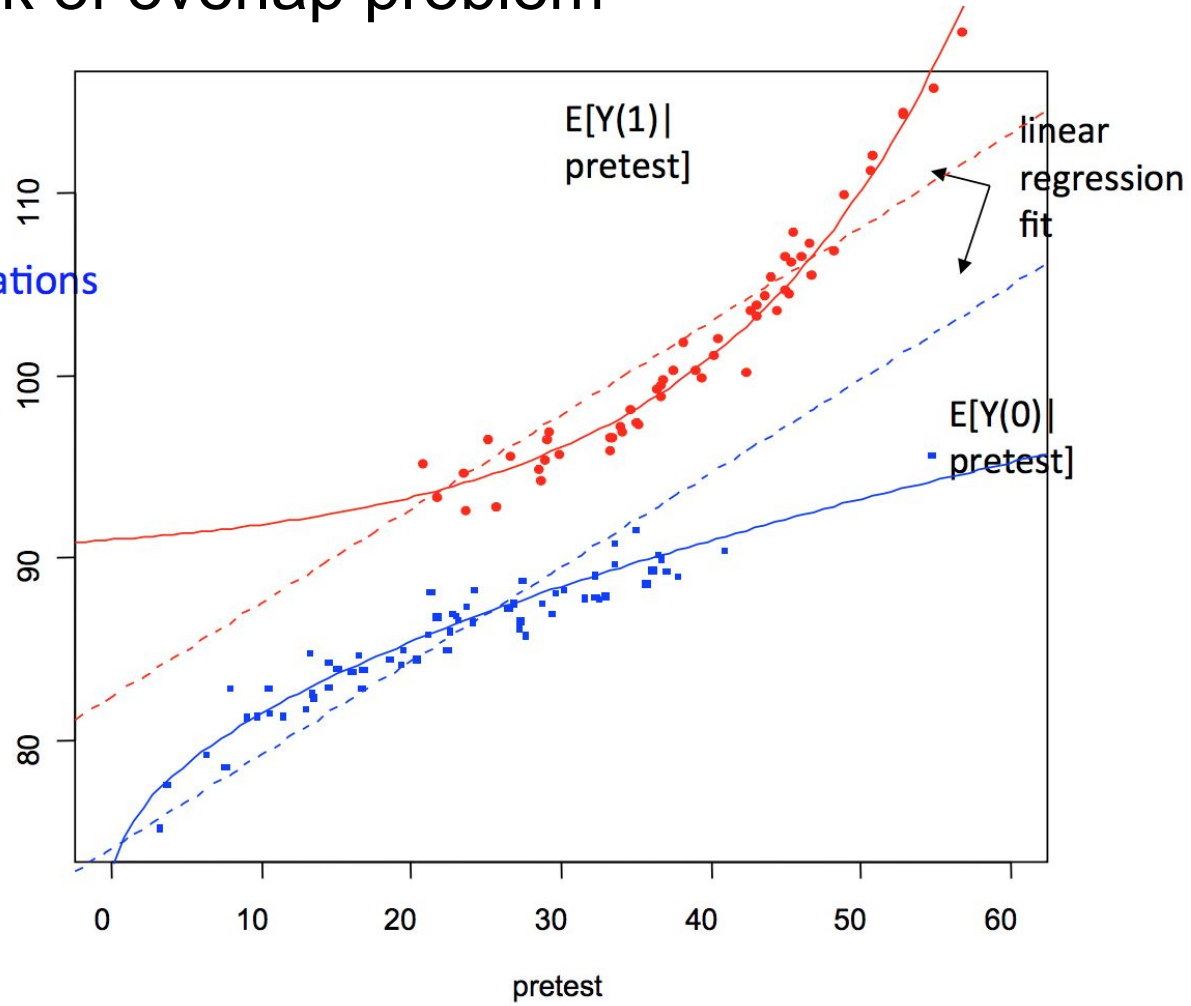
Imbalance and lack of overlap problem

red for treatment observations and response surface

blue for control observations and response surface

Linear regression (dotted lines) is not an appropriate model here

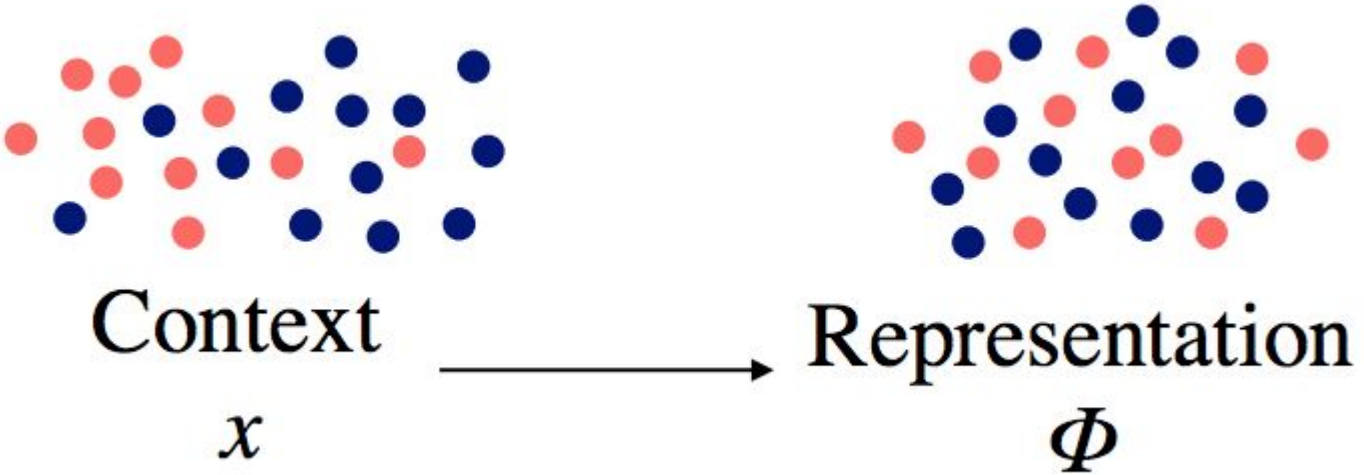
Lack of overlap in pretest scores exacerbates the problem by forcing model extrapolation



Counterfactual Regression with Neural Network

Estimating individual treatment effect: generalization bounds and algorithms, Uri Shalit et al. (2017)
Learning Representations for Counterfactual Inference, Fredrik D. Johansson et al. (2016)

Balanced-Representation Learning



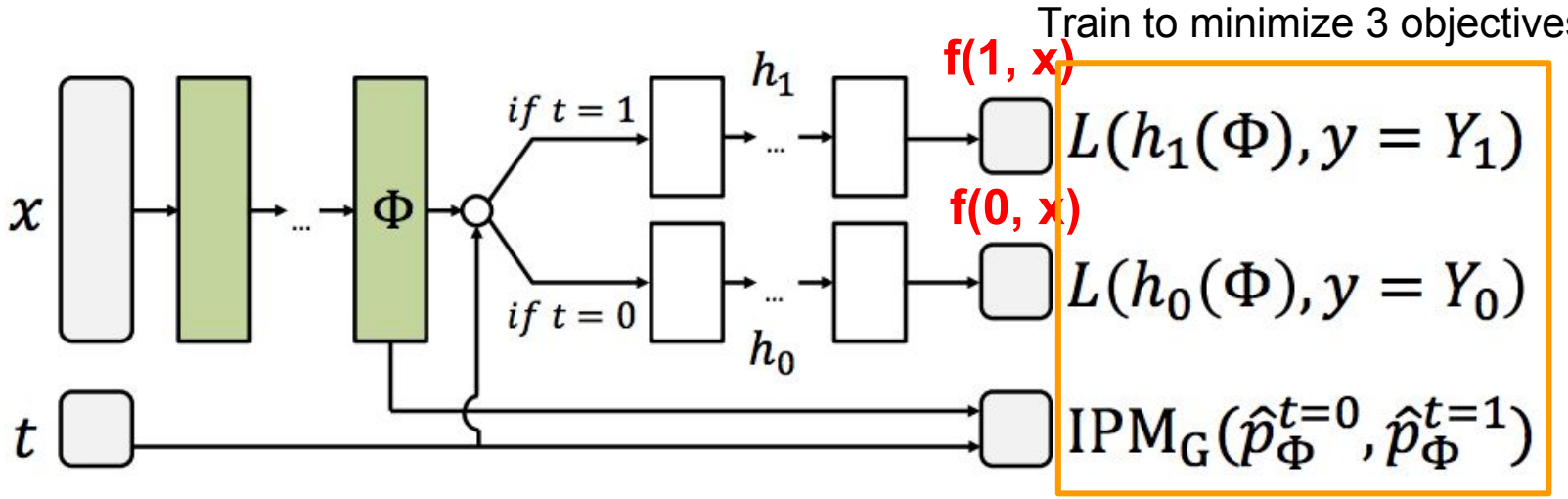
Counterfactual regression with Neural Network

Neural net based representation learning algorithm with explicit regularization for counterfactual estimation

Goal: Estimate **f(z, x)** using neural networks

- Add explicit regularization to balance feature representation in treated and controlled groups

Counterfactual regression with Neural Network



Integral Probability Metric (IPM) measures distance between two distributions

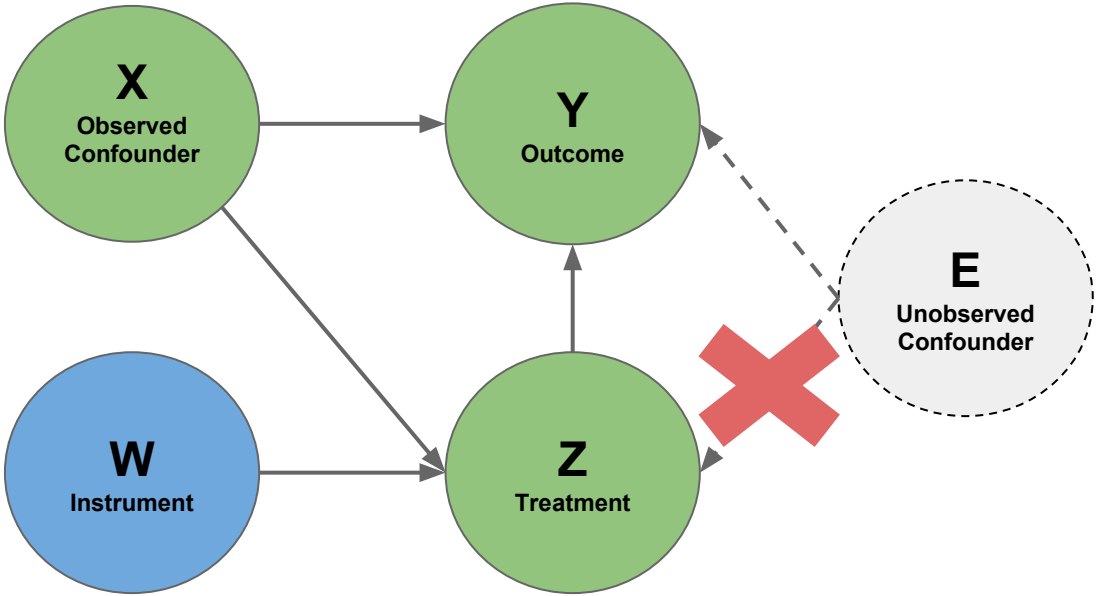
- Such as Wasserstein and Maximum Mean Discrepancy (MMD) distances

Estimating individual treatment effect: generalization bounds and algorithms, Uri Shalit et al. (2017)

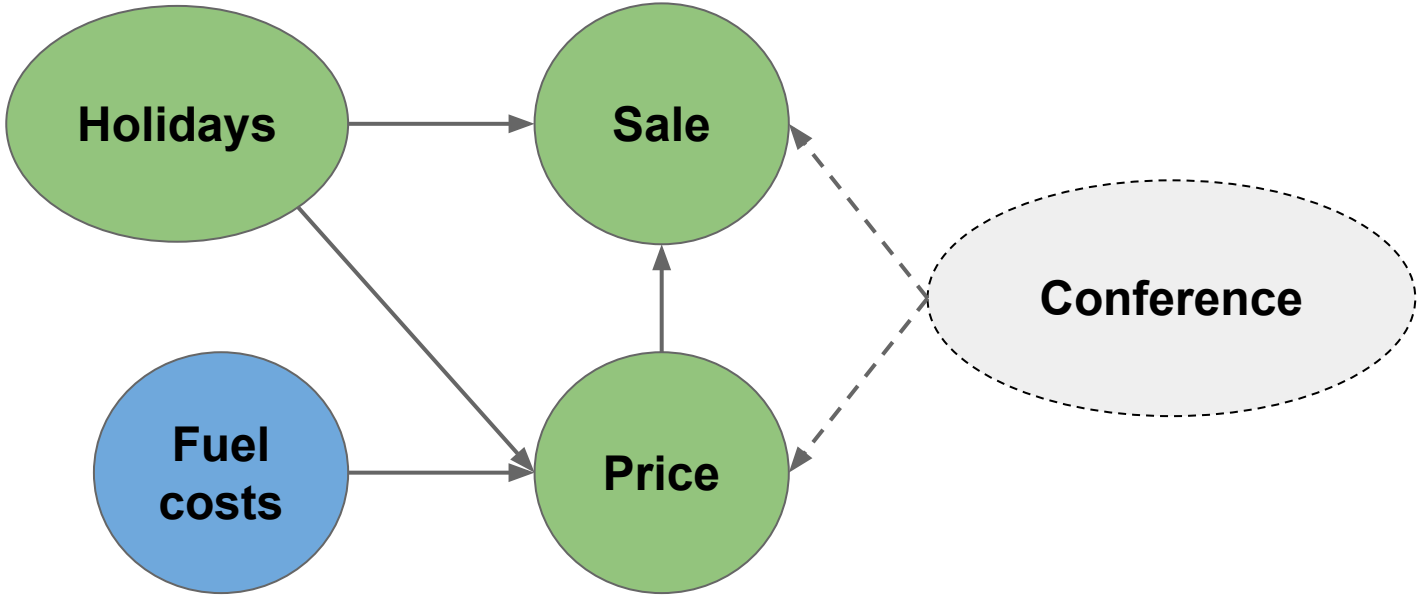
Handle unobserved confounders

Instrumental Variable

Instrumental Variable



Airline Price Example

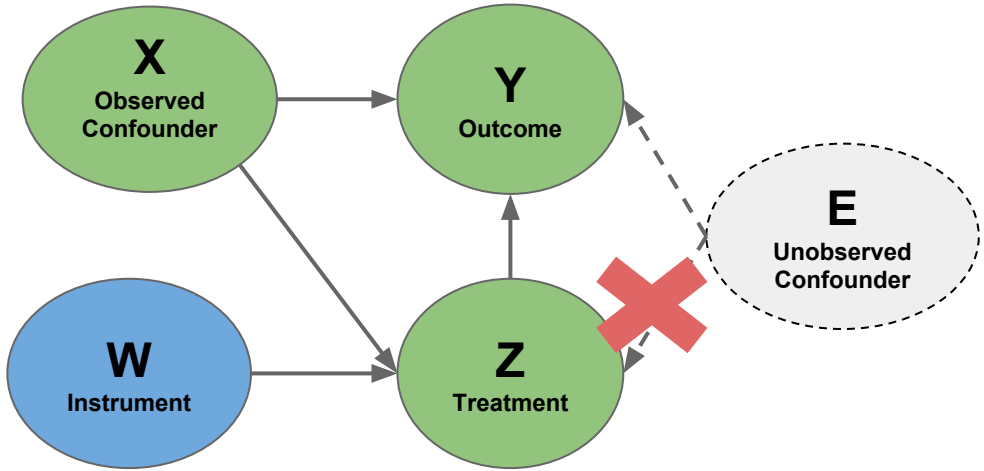


Deep IV: A Flexible Approach for Counterfactual Prediction, J Hartford et al. (2017)

Instrumental Variable

Two main assumptions:

1. Relevance: F(z|x,w), the distribution of z given x and w, is not constant in w.
2. Exclusion: w ⊥ y | (x, z, e).



Instrumental Variable

We assume additive error assumption

$$y = g(z, x) + e$$

Taking the expectation of both sides conditional on [x, w] and applying the assumptions establishes the relationship

$$E[y|x, w] = E[g(z, w)|x, w] + E[e|x]$$

$$= \int g(z, x) dF(z|x, w)$$

Instrumental Variable

$$E[y|x,w] = \int g(z,x) dF(z|x,w)$$

We can recover g(z,x) by solving implied inverse problem

Closed-form solution exists if we posit linearity assumption in g(z, w) and F(z|x, w): Two-stage least square

Very inflexible!

Deep Instrumental Variable

Deep IV: A Flexible Approach for Counterfactual Prediction, J Hartford et al. (2017)

Deep Instrumental Variable

$$E[y|x,w] = \int g(z,x) dF(z|x,w)$$

We can recover g(z,x) by solving implied inverse problem

$$\min_{g \in G} \sum_{t=1}^{n} (y_t - \int g(z,x_t) dF(z|x,w))^2$$

Deep IV: A Flexible Approach for Counterfactual Prediction, J Hartford et al. (2017)

Deep Instrumental Variable

DeepIV procedure has two stages:

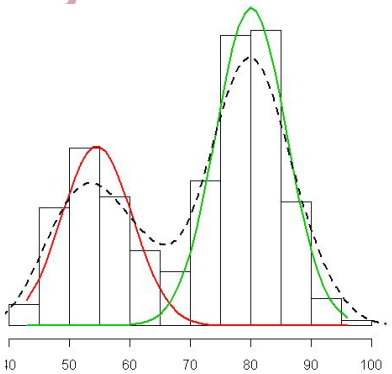
- Estimate density $\hat{F}(z|x,w)$

- Optimize the loss function

Deep Instrumental Variable

$$\min_{g \in G} \sum_{t=1}^{n} (y_t - \int g(z, x_t) dF(z|x, w))^2$$

Stage 1: fit $\hat{F}(z|x, w)$
Using the model of choice
The author uses **Mixture Density Networks**

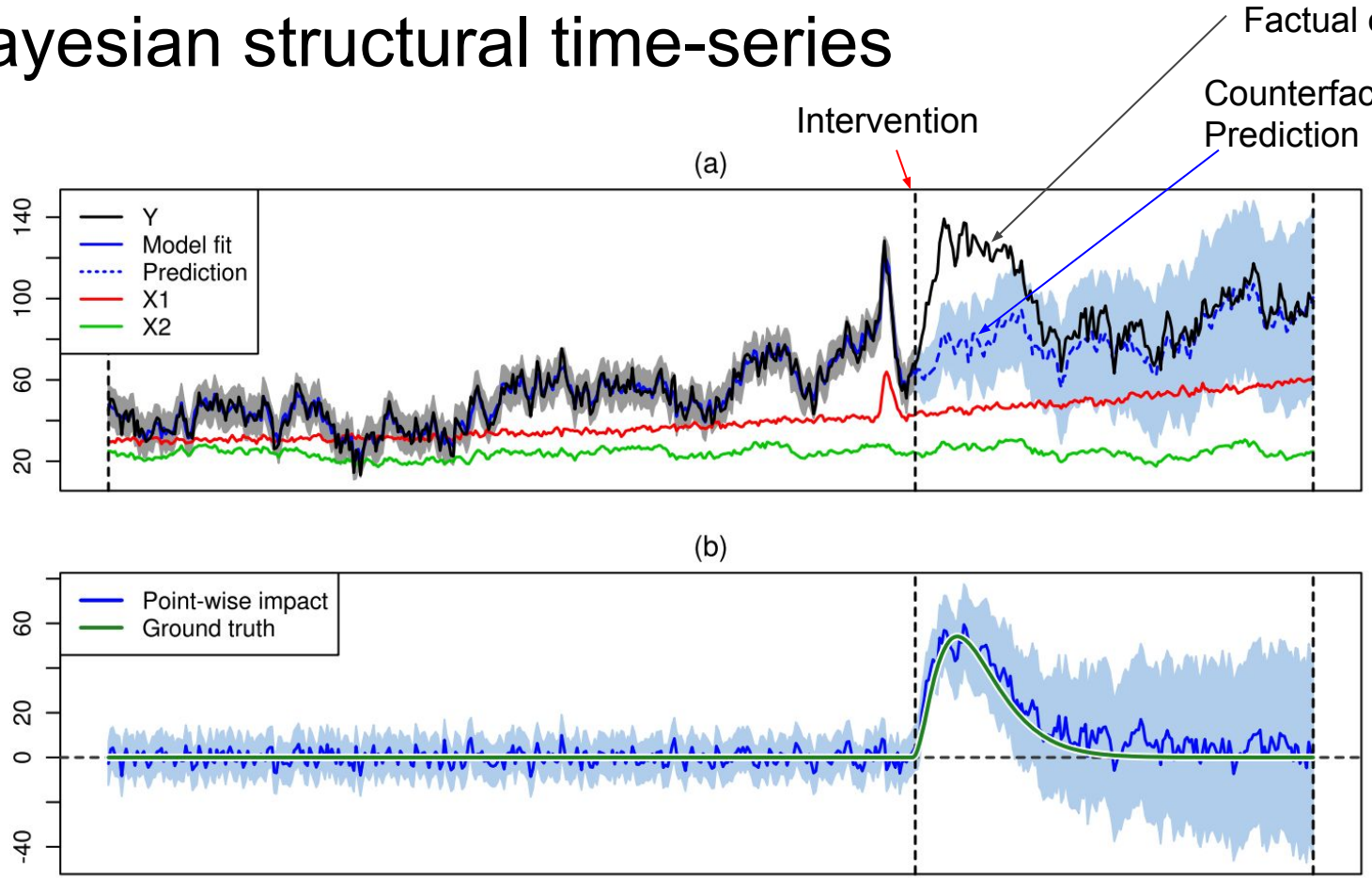


Stage 2: train network $\hat{g}_\theta$
using stochastic gradient descent
with monte carlo integration

Deep IV: A Flexible Approach for Counterfactual Prediction, J Hartford et al. (2017)

Time Series Data

Bayesian structural time-series



Inferring causal impact using Bayesian structural time-series models, Kay H. Brodersen (2015)

Some other stuff

- ML for Propensity Score
- ML for matching (e.g. kernel-based matching)
- ML for variable selection (e.g. LASSO)

Challenge

- Regularization bias
- Cannot perform conventional cross validation because of the fundamental problem of causal inference
  - How to perform model selection and hyper-parameter tuning
- Very few benchmark dataset available

What else ?

What else?

- Causal discoveries: The next big thing!!
- Combining observational and interventional data
- Relationship with reinforcement learning