Machine Learning and Economic Modeling

Annie Liang

Northwestern

Machine Learning Algorithms and Economic Models

ML algorithms have demonstrated tremendous predictive success.

Machine Learning Algorithms and Economic Models

ML algorithms have demonstrated tremendous predictive success.

But for economic models, care about criteria besides predictiveness:

Machine Learning Algorithms and Economic Models

ML algorithms have demonstrated tremendous predictive success.

But for economic models, care about criteria besides predictiveness:

- are the model's parameters interpretable economically?

Machine Learning Algorithms and Economic Models

ML algorithms have demonstrated tremendous predictive success.

But for economic models, care about criteria besides predictiveness:

- are the model's parameters interpretable economically?
- is the model descriptive only of a specific behavior, or is it a good abstraction more broadly?

Machine Learning Algorithms and Economic Models

ML algorithms have demonstrated tremendous predictive success.

But for economic models, care about criteria besides predictiveness:

- are the model's parameters interpretable economically?
- is the model descriptive only of a specific behavior, or is it a good abstraction more broadly?
- does the model convey a useful narrative that shapes a perspective on a behavior or domain?

This Talk

Can black box techniques still be useful to the economic modeler?

This talk: **yes**.

This Talk

Can black box techniques still be useful to the economic modeler?

This talk: **yes**.

Slides will explore a series of methodologies and their applications to specific economic problems.

- Predicting and Understanding Initial Play, *AER*, 2019
  (with Drew Fudenberg)
- Measuring the Completeness of Economic Models, *JPE*, 2022
  (with Drew Fudenberg, Jon Kleinberg, and Sendhil Mullainathan)
- How Flexible is that Functional Form? Measuring the Restrictiveness of Theories, conditionally accepted at *REStat*, 2023
  (with Drew Fudenberg and Wayne Gao)
- The Transfer Performance of Economic Models, 2023
  (with Isaiah Andrews, Drew Fudenberg, Lihua Lei, Chaofeng Wu)

Outline

Part I:

Black Box Prediction as a Predictive Upper Bound

Measuring the Completeness of Economic Models, *JPE*, 2022
(Fudenberg, Liang, Kleinberg, and Mullainathan)

An Example Problem: Predicting Human Generation of Random Sequences

Of interest: human (mis)perception of randomness

Specific problem instance:

- suppose a human is asked to generate eight realizations of $\{H, T\}$ as if flipping a fair coin

An Example Problem: Predicting Human Generation of Random Sequences

Of interest: human (mis)perception of randomness

Specific problem instance:

- suppose a human is asked to generate eight realizations of $\{H, T\}$ as if flipping a fair coin
- you have access to the first seven flips

An Example Problem: Predicting Human Generation of Random Sequences

Of interest: human (mis)perception of randomness

Specific problem instance:

- suppose a human is asked to generate eight realizations of $\{H, T\}$ as if flipping a fair coin
- you have access to the first seven flips
- can you predict the eighth?

An Example Problem: Predicting Human Generation of Random Sequences

Of interest: human (mis)perception of randomness

Specific problem instance:

- suppose a human is asked to generate eight realizations of $\{H, T\}$ as if flipping a fair coin
- you have access to the first seven flips
- can you predict the eighth?

Prediction rule: Any function $f : \{H, T\}^7 \mapsto [0, 1]$ (probability final flip is H).

An Example Problem: Predicting Human Generation of Random Sequences

Of interest: human (mis)perception of randomness

Specific problem instance:

- suppose a human is asked to generate eight realizations of $\{H, T\}$ as if flipping a fair coin
- you have access to the first seven flips
- can you predict the eighth?

Prediction rule: Any function $f : \{H, T\}^7 \mapsto [0, 1]$ (probability final flip is H).

Loss function is mean-squared error: $(p - s_8)^2$ when prediction is $p \in [0, 1]$ and actual final flip is $s_8 \in \{0, 1\}$

Behavioral Model: Rabin and Vayanos (2010)

Behavioral model: humans think "random" means "negatively auto-correlated"

- If last flips were HHHH, then T "is due"

Rabin and Vayanos (2010):

$$\Pr(s_8 = H \mid s_1, \ldots, s_7) = \frac{1}{2} - \alpha \sum_{k=1}^{7} \delta^k \mathbb{1}(s_{8-k} = H).$$

Two parameters: α measures strength of negative auto-correlation, δ measures a recency effect.

How Predictive is the Behavioral Model?

- Data set: 22K human-generated strings on Mechanical Turk. Tenfold cross-validated error:

	Prediction Error
Rabin and Vayanos (2010)	0.2494 (0.0003)

How Predictive is the Behavioral Model?

- Data set: 22K human-generated strings on Mechanical Turk. Tenfold cross-validated error:

	Prediction Error
Rabin and Vayanos (2010)	0.2494 (0.0003)

- But what does this number mean?

Model is Predictive

Can get a sense of size by comparing against a naive model: guess 1/2 for every sequence.

	Prediction Error
Naive Baseline	0.25
Rabin and Vayanos (2010)	0.2494 (0.0003)

Model is Predictive

Can get a sense of size by comparing against a naive model: guess 1/2 for every sequence.

	Prediction Error
Naive Baseline	0.25
Rabin and Vayanos (2010)	0.2494 (0.0003)

- This shows us that the behavioral model improves on random guessing (the behavioral model is **predictive**).
- But it still doesn't help us to understand the margin of improvement.

Two Very Different Explanations for Error

Key confound: error can emerge from

- suboptimal combination of features

Two Very Different Explanations for Error

Key confound: error can emerge from

- suboptimal combination of features
  - can reduce prediction error by writing down a different model
  - negative auto-correlation isn't the full story

Two Very Different Explanations for Error

Key confound: error can emerge from

- suboptimal combination of features
  - can reduce prediction error by writing down a different model
  - negative auto-correlation isn't the full story
- basic limitations of the feature set (first seven flips)

Two Very Different Explanations for Error

Key confound: error can emerge from

- suboptimal combination of features
  - can reduce prediction error by writing down a different model
  - negative auto-correlation isn't the full story
- basic limitations of the feature set (first seven flips)
  - can't reduce predictive error with a new model based on the same features
  - instead need to acquire/measure new features

Table Lookup

Straightforward problem for an ML algorithm. Given enough data, solution is to use a Table Lookup algorithm.

s_1	s_2	s_3	s_4	s_5	s_6	s_7	probability that $s_8 = H$
H	H	H	H	H	H	H	p_1
H	H	H	H	H	H	T	p_2
H	H	H	H	H	T	H	p_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
T	T	T	T	T	T	T	p_{128}

This model has 2^7 free parameters, none economically meaningful.

Table Lookup

Straightforward problem for an ML algorithm. Given enough data, solution is to use a Table Lookup algorithm.

s_1	s_2	s_3	s_4	s_5	s_6	s_7	probability that $s_8 = H$
H	H	H	H	H	H	H	p_1
H	H	H	H	H	H	T	p_2
H	H	H	H	H	T	H	p_3
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
T	T	T	T	T	T	T	p_{128}

This model has 2^7 free parameters, none economically meaningful.

- trained on sufficient data, table lookup approximates **best achievable accuracy**

"Achievable" Predictive Accuracy

	Prediction Error
Naive Baseline	0.25
Rabin and Vayanos (2010)	0.2494 (0.0003)

"Achievable" Predictive Accuracy

	Prediction Error
Naive Baseline	0.25
Rabin and Vayanos (2010)	0.2494 (0.0003)
Table Lookup	0.2441 (0.002)

"Achievable" Predictive Accuracy

	Prediction Error
Naive Baseline	0.25
Rabin and Vayanos (2010)	0.2494 (0.0003)
Table Lookup	0.2441 (0.002)

- Black box performance is far worse than perfect prediction (large amount of irreducible noise)
- For the given features, 0.2441 represents "predictive limit."
- Naively comparing the model's performance against perfect zero grossly misrepresents performance!

Completeness

	Prediction Error	Completeness
Naive Baseline	0.25	0%
Rabin and Vayanos (2010)	0.2494 (0.0003)	
Table Lookup	0.2441 (0.002)	100%

Simple measure of completeness is ratio of achieved versus achievable improvement: $(0.25 - 0.2494)/(0.25 - 0.2441)$

Completeness

	Prediction Error	Completeness
Naive Baseline	0.25	0%
Rabin and Vayanos (2010)	0.2494 (0.0003)	10%
Table Lookup	0.2441 (0.002)	100%

Simple measure of completeness is ratio of achieved versus achievable improvement: $(0.25 - 0.2494)/(0.25 - 0.2441)$

Part II:

Use Black Box to Discover New Structure

Predicting and Understanding Initial Play, *AER*, 2019
(Fudenberg and Liang)

Problem Domain: Predicting Initial Play in Games

Problem:

	a_1	a_2	a_3
a_1	$25, 25$	$30, 40$	$100, 31$
a_2	$40, 30$	$45, 45$	$65, 0$
a_3	$31, 100$	$0, 65$	$40, 40$

- What is the modal action chosen by people in the role of the row player?

Prediction rule: Any map $f : \mathbb{R}^{18} \rightarrow \{a_1, a_2, a_3\}$ from payoff matrices into row player action.

Loss function: misclassification rate.

Models

Uniform Nash: Choose uniformly at random from actions consistent with (pure-strategy) Nash equilibrium.

Level 1: Predict the action that maximizes expected payoffs when the other player chooses uniformly at random.

Models

Uniform Nash: Choose uniformly at random from actions consistent with (pure-strategy) Nash equilibrium.

Level 1: Predict the action that maximizes expected payoffs when the other player chooses uniformly at random.

e.g. action a_1 is the Level 1 action here:

	a_1	a_2	a_3	Average Payoff
a_1	$25, 25$	$30, 40$	$100, 31$	51.6
a_2	$40, 30$	$45, 45$	$65, 0$	50
a_3	$31, 100$	$0, 65$	$40, 40$	23.6

Machine Learning Approach

- Identify each game with a feature vector describing various known strategic properties.

  e.g. for each action:
  - is it part of a pure-strategy Nash equilibrium?
  - is it part of a profile that maximizes sum of player payoffs?

- Train a decision tree ensemble to predict modal action given features.

Comparison of Prediction Accuracies

Tenfold cross-validated classification rate on meta-data set from six lab experiments (86 total games):

	Accuracy
Naive Baseline	0.33

Comparison of Prediction Accuracies

Tenfold cross-validated classification rate on meta-data set from six lab experiments (86 total games):

	Accuracy
Naive Baseline	0.33
Uniform Nash	0.42 (0.05)

Comparison of Prediction Accuracies

Tenfold cross-validated classification rate on meta-data set from six lab experiments (86 total games):

	Accuracy
Naive Baseline	0.33
Uniform Nash	0.42 (0.05)
Level 1	0.72 (0.04)

Comparison of Prediction Accuracies

Tenfold cross-validated classification rate on meta-data set from six lab experiments (86 total games):

	Accuracy
Naive Baseline	0.33
Uniform Nash	0.42 (0.05)
Level 1	0.72 (0.04)
Decision Tree Ensemble	0.77 (0.02)

Identifying Predictable Structure Beyond Level 1

- Look at games where the modal action is correctly predicted by our algorithm but not by Level 1.
  - opportunity for identifying new regularities that are missed by the behavioral model

Identifying Predictable Structure Beyond Level 1

- Look at games where the modal action is correctly predicted by our algorithm but not by Level 1.
  - opportunity for identifying new regularities that are missed by the behavioral model

- Example game:

	a_1	a_2	a_3	Average Payoff
a_1	$25, 25$	$30, 40$	$100, 31$	51.6
a_2	$40, 30$	$45, 45$	$65, 0$	50
a_3	$31, 100$	$0, 65$	$40, 40$	23.6

- Level 1 action is a_1, but a_2 played most frequently

Identifying Predictable Structure Beyond Level 1

- Look at games where the modal action is correctly predicted by our algorithm but not by Level 1.
  - opportunity for identifying new regularities that are missed by the behavioral model

- Example game:

	a_1	a_2	a_3	Average Payoff
a_1	$25, 25$	$30, 40$	$100, 31$	51.6
a_2	$40, 30$	$45, 45$	$65, 0$	50
a_3	$31, 100$	$0, 65$	$40, 40$	23.6

- Level 1 action is a_1, but a_2 played most frequently

- Modify Level 1 by giving participants utility functions $u(x) = x^{\alpha}$; this adds one parameter to the Level 1 model
  - $\alpha \in [0, 1)$ captures risk aversion

Extension of Level 1 Achieves Performance of Algorithm

	Accuracy
Level 1	0.72 (0.04)
Decision Tree Ensemble	0.77 (0.02)

Extension of Level 1 Achieves Performance of Algorithm

	Accuracy
Level 1	0.72 (0.04)
Decision Tree Ensemble	0.77 (0.02)
Level 1(α)	0.79 (0.04)

- Black box algorithm can lead to interpretable extensions of behavioral models

Takeaways from Lab Games

- Could stop here and conclude that Level 1(α) is an almost complete model of play
- But set of games in our data was small and special

Takeaways from Lab Games

- Could stop here and conclude that Level 1(α) is an almost complete model of play
- But set of games in our data was small and special
- Is the performance of Level 1(α) due to idiosyncratic properties of the data set?
- What regularities exist outside of this data?

Takeaways from Lab Games

- Could stop here and conclude that Level 1(α) is an almost complete model of play
- But set of games in our data was small and special
- Is the performance of Level 1(α) due to idiosyncratic properties of the data set?
- What regularities exist outside of this data?
- Space of payoff matrices $\mathbb{R}^{18}$ is large—how to populate this space in a useful way?

Part III:

Use Black Box to Find Cases That Break Our Best Model

Predicting and Understanding Initial Play, *AER*, 2019
(Fudenberg and Liang)

Algorithmic Experimental Design

Approach:

- Teach an algorithm to recognize games where Level-1(α) performs poorly.

Algorithmic Experimental Design

Approach:

- Teach an algorithm to recognize games where Level-1(α) performs poorly.
- Randomly generate games.

Algorithmic Experimental Design

Approach:

- Teach an algorithm to recognize games where Level-1(α) performs poorly.
- Randomly generate games.
- Use algorithm to predict performance of the model on these randomly generated games.

Algorithmic Experimental Design

Approach:

- Teach an algorithm to recognize games where Level-1(α) performs poorly.
- Randomly generate games.
- Use algorithm to predict performance of the model on these randomly generated games.
- Keep the games where the model is predicted to perform poorly.

Algorithmic Experimental Design

Approach:

- Teach an algorithm to recognize games where Level-1(α) performs poorly.
- Randomly generate games.
- Use algorithm to predict performance of the model on these randomly generated games.
- Keep the games where the model is predicted to perform poorly.
- Repeat.

Step 1: Train Algorithm to Predict Freq of Level 1 Play

	a_1	a_2	a_3
a_1	$40, 40$	$10, 20$	$70, 30$
a_2	$20, 10$	$80, 80$	$0, 100$
a_3	$30, 70$	$100, 0$	$60, 60$

freq. of Level 1(α) action: 73%

	a_1	a_2	a_3
a_1	$20, 20$	$0, 60$	$100, 0$
a_2	$60, 0$	$20, 20$	$0, 60$
a_3	$0, 100$	$60, 0$	$40, 40$

freq. of Level 1(α) action: 65%

	a_1	a_2	a_3
a_1	$20, 20$	$30, 40$	$100, 30$
a_2	$40, 30$	$40, 40$	$60, 0$
a_3	$30, 100$	$0, 60$	$40, 40$

freq. of Level 1(α) action: 35%

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Step 2: Generate New Games, Predict Freq of Level 1 Play

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Step 2: Generate New Games, Predict Freq of Level 1 Play

	a_1	a_2	a_3
a_1	$90, 90$	$30, 80$	$45, 30$
a_2	$80, 30$	$55, 55$	$37, 5$
a_3	$30, 45$	$5, 37$	$70, 70$

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Step 2: Generate New Games, Predict Freq of Level 1 Play

	a_1	a_2	a_3
a_1	$90, 90$	$30, 80$	$45, 30$
a_2	$80, 30$	$55, 55$	$37, 5$
a_3	$30, 45$	$5, 37$	$70, 70$

	a_1	a_2	a_3
a_1	$70, 70$	$45, 30$	$40, 35$
a_2	$30, 45$	$53, 53$	$93, 31$
a_3	$35, 40$	$31, 93$	$10, 10$

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Step 2: Generate New Games, Predict Freq of Level 1 Play

	a_1	a_2	a_3
a_1	$90, 90$	$30, 80$	$45, 30$
a_2	$80, 30$	$55, 55$	$37, 5$
a_3	$30, 45$	$5, 37$	$70, 70$

	a_1	a_2	a_3
a_1	$70, 70$	$45, 30$	$40, 35$
a_2	$30, 45$	$53, 53$	$93, 31$
a_3	$35, 40$	$31, 93$	$10, 10$

	a_1	a_2	a_3
a_1	$60, 60$	$40, 40$	$51, 40$
a_2	$40, 40$	$80, 80$	$35, 10$
a_3	$40, 51$	$10, 35$	$100, 100$

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Step 2: Generate New Games, Predict Freq of Level 1 Play

	a_1	a_2	a_3
a_1	$90, 90$	$30, 80$	$45, 30$
a_2	$80, 30$	$55, 55$	$37, 5$
a_3	$30, 45$	$5, 37$	$70, 70$

predicted frequency: 48%

	a_1	a_2	a_3
a_1	$70, 70$	$45, 30$	$40, 35$
a_2	$30, 45$	$53, 53$	$93, 31$
a_3	$35, 40$	$31, 93$	$10, 10$

predicted frequency: 56%

	a_1	a_2	a_3
a_1	$60, 60$	$40, 40$	$51, 40$
a_2	$40, 40$	$80, 80$	$35, 10$
a_3	$40, 51$	$10, 35$	$100, 100$

predicted frequency: 46%

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Step 3: Redraw Games with High Predicted Frequencies

	a_1	a_2	a_3
a_1	$90, 90$	$30, 80$	$45, 30$
a_2	$80, 30$	$55, 55$	$37, 5$
a_3	$30, 45$	$5, 37$	$70, 70$

predicted frequency: 48%

	a_1	a_2	a_3
a_1	$60, 60$	$40, 40$	$51, 40$
a_2	$40, 40$	$80, 80$	$35, 10$
a_3	$40, 51$	$10, 35$	$100, 100$

predicted frequency: 46%

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Step 3: Redraw Games with High Predicted Frequencies

	a_1	a_2	a_3
a_1	$90, 90$	$30, 80$	$45, 30$
a_2	$80, 30$	$55, 55$	$37, 5$
a_3	$30, 45$	$5, 37$	$70, 70$

predicted frequency: 48%

	a_1	a_2	a_3
a_1	$50, 50$	$70, 40$	$50, 15$
a_2	$40, 70$	$15, 15$	$88, 100$
a_3	$15, 50$	$100, 88$	$58, 58$

	a_1	a_2	a_3
a_1	$60, 60$	$40, 40$	$51, 40$
a_2	$40, 40$	$80, 80$	$35, 10$
a_3	$40, 51$	$10, 35$	$100, 100$

predicted frequency: 46%

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Step 3: Redraw Games with High Predicted Frequencies

	a_1	a_2	a_3
a_1	$90, 90$	$30, 80$	$45, 30$
a_2	$80, 30$	$55, 55$	$37, 5$
a_3	$30, 45$	$5, 37$	$70, 70$

predicted frequency: 48%

	a_1	a_2	a_3
a_1	$50, 50$	$70, 40$	$50, 15$
a_2	$40, 70$	$15, 15$	$88, 100$
a_3	$15, 50$	$100, 88$	$58, 58$

predicted frequency: 47%

	a_1	a_2	a_3
a_1	$60, 60$	$40, 40$	$51, 40$
a_2	$40, 40$	$80, 80$	$35, 10$
a_3	$40, 51$	$10, 35$	$100, 100$

predicted frequency: 46%

learn a map from **payoff matrices** into **prediction of frequency of play of Level 1(α) action**

Performance on New Games

	Accuracy
Guess at random	0.33
Level 1	0.36
	(0.01)
Level 1(α)	0.41
	(0.05)

- Algorithmically designed games succeed in being poor matches for Level 1 and Level 1(α).

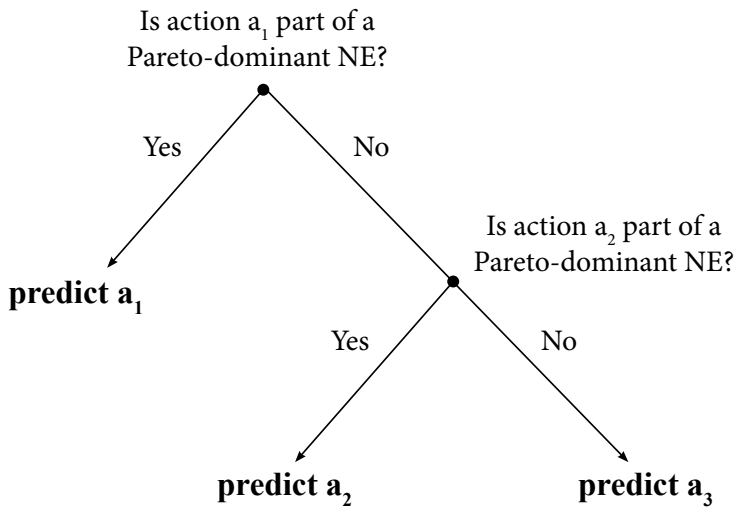
Performance on New Games

	Accuracy
Guess at random	0.33
Level 1	0.36
	(0.01)
Level 1(α)	0.41
	(0.05)
Decision Tree Ensemble	0.73
	(0.02)

- Algorithmically designed games succeed in being poor matches for Level 1 and Level 1(α).

Best 2-split Decision Tree

Decision tree ensemble is hard to interpret, but best 2-split decision tree is not:



motivates:

Pareto-Dominant NE (PDNE): predict uniformly at random from actions consistent with PDNE, otherwise predict at random.

Example of Pareto-Dominant Nash Equilibrium

(a_2, a_2) is a Pareto-Dominant Nash equilibrium.

	a_1	a_2	a_3
a_1	$10, 10$	$0, 0$	$0, 0$
a_2	$0, 0$	$30, 30$	$0, 0$
a_3	$0, 0$	$0, 0$	$5, 5$

Performance of PDNE

	Accuracy
Guess at random	0.33
Level 1	0.36
	(0.01)
Level 1(α)	0.41
	(0.05)
Decision Tree Ensemble	0.73
	(0.02)

Performance of PDNE

	Accuracy
Guess at random	0.33
Level 1	0.36
	(0.01)
Level 1(α)	0.41
	(0.05)
PDNE	0.65
	(0.02)
Decision Tree Ensemble	0.73
	(0.02)

- PDNE performs very well on this data set (substantially outperforms Level 1(α))

Part IV:

Use Synthetic Data to Evaluate the Restrictiveness of Models

How Flexible is that Functional Form? Quantifying the Restrictiveness of Theories, *conditionally accepted at REStat*, 2023 (Fudenberg, Gao, and Liang)

The Basic Problem

When a model is very complete, we'd like to interpret this as evidence that the model is a good one.

But another possibility is that the model is simply so flexible it would have fit any data.

- At an extreme: the model may not be falsifiable.

The Basic Problem

When a model is very complete, we'd like to interpret this as evidence that the model is a good one.

But another possibility is that the model is simply so flexible it would have fit any data.

- At an extreme: the model may not be falsifiable.

We'd like to distinguish between when a model is **precisely tailored to capture real regularities** versus when it is simply **unrestrictive**.

Stylized Example

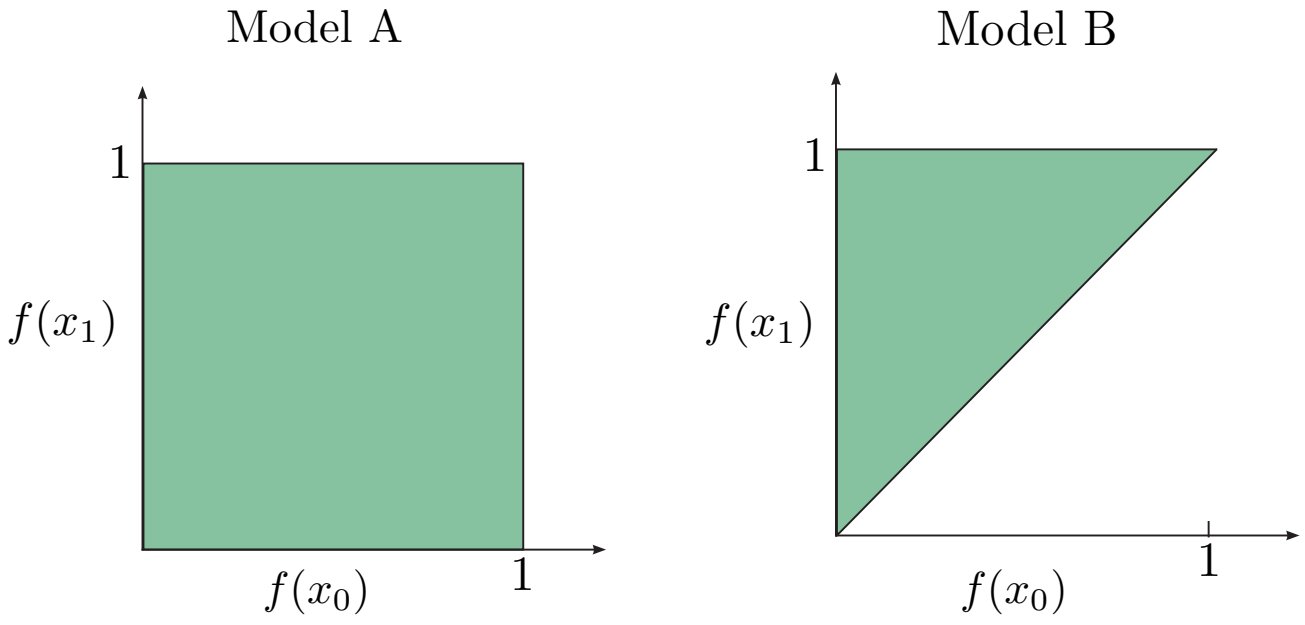
- There is a single (binary) covariate $x \in \{x_0, x_1\}$ and outcome variable $y \in [0, 1]$

Stylized Example

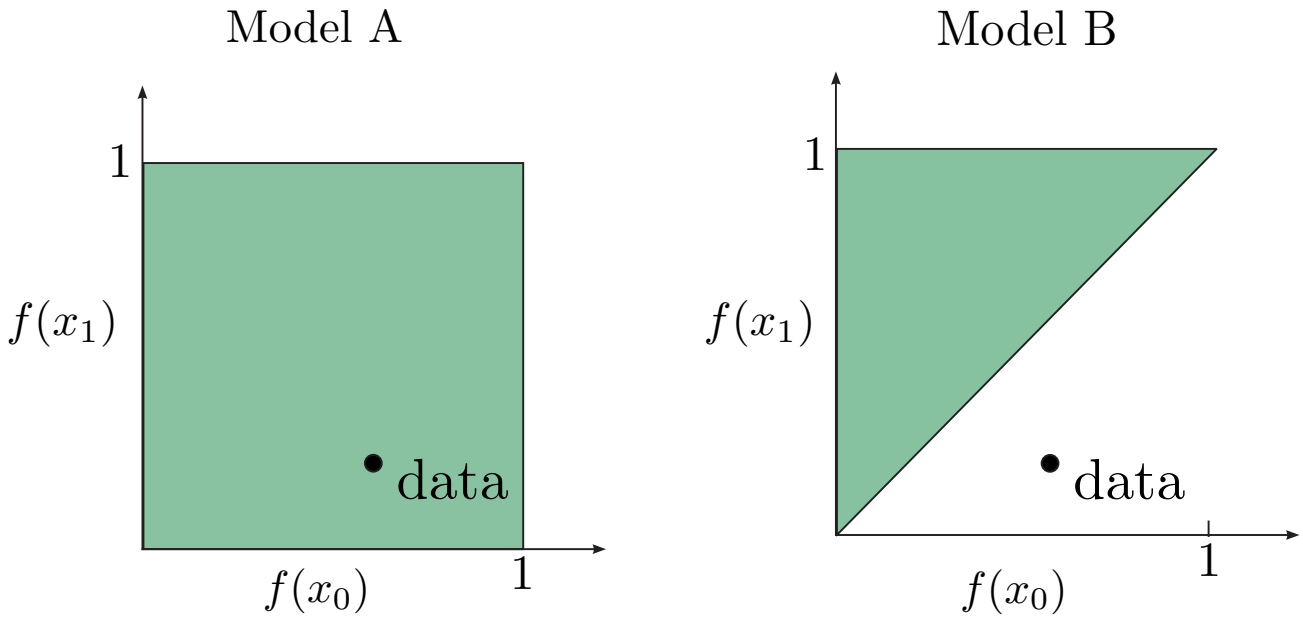
- There is a single (binary) covariate $x \in \{x_0, x_1\}$ and outcome variable $y \in [0, 1]$
- A prediction rule is any mapping $f : \{x_0, x_1\} \to [0, 1]$

Stylized Example

- There is a single (binary) covariate $x \in \{x_0, x_1\}$ and outcome variable $y \in [0, 1]$
- A prediction rule is any mapping $f : \{x_0, x_1\} \to [0, 1]$
- A model is a set of prediction rules, e.g.,

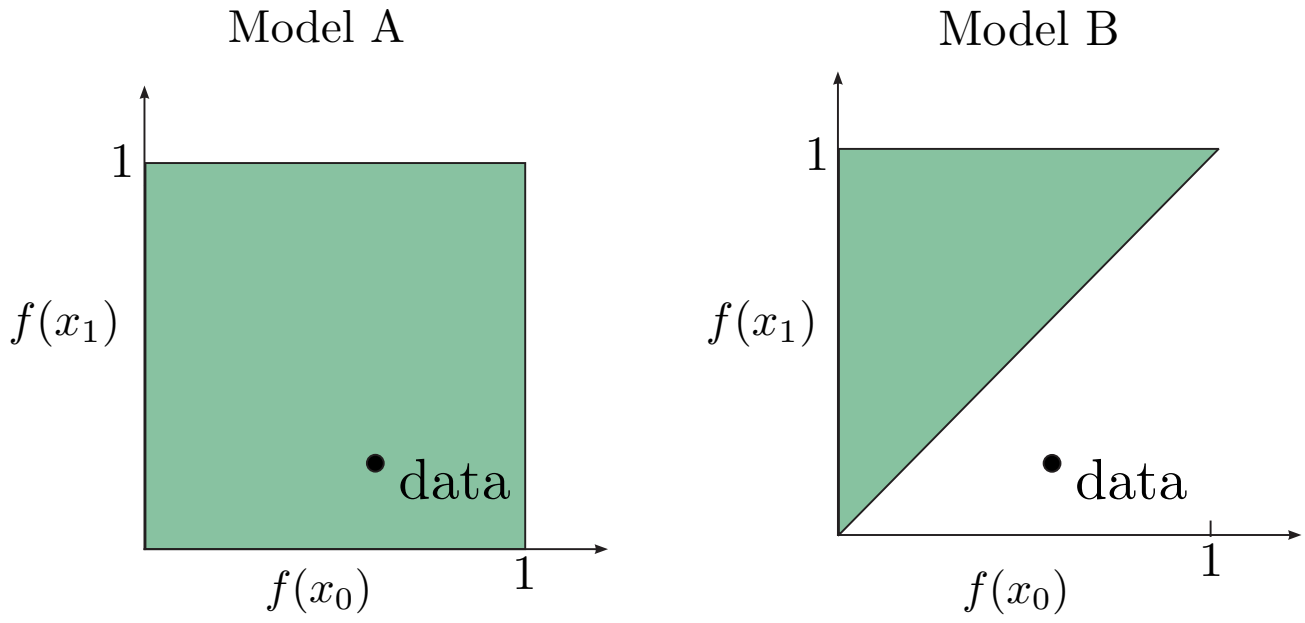


Stylized Example



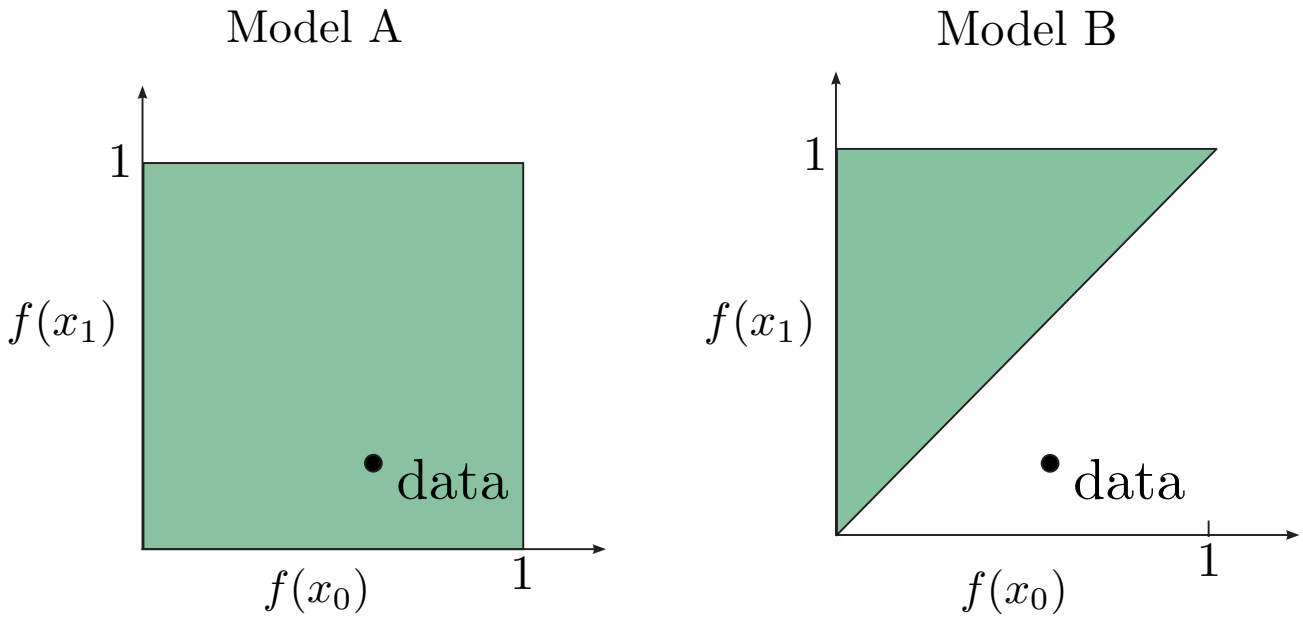
- Data is an observed outcome for each covariate value (a point in $[0, 1] \times [0, 1]$)

Stylized Example



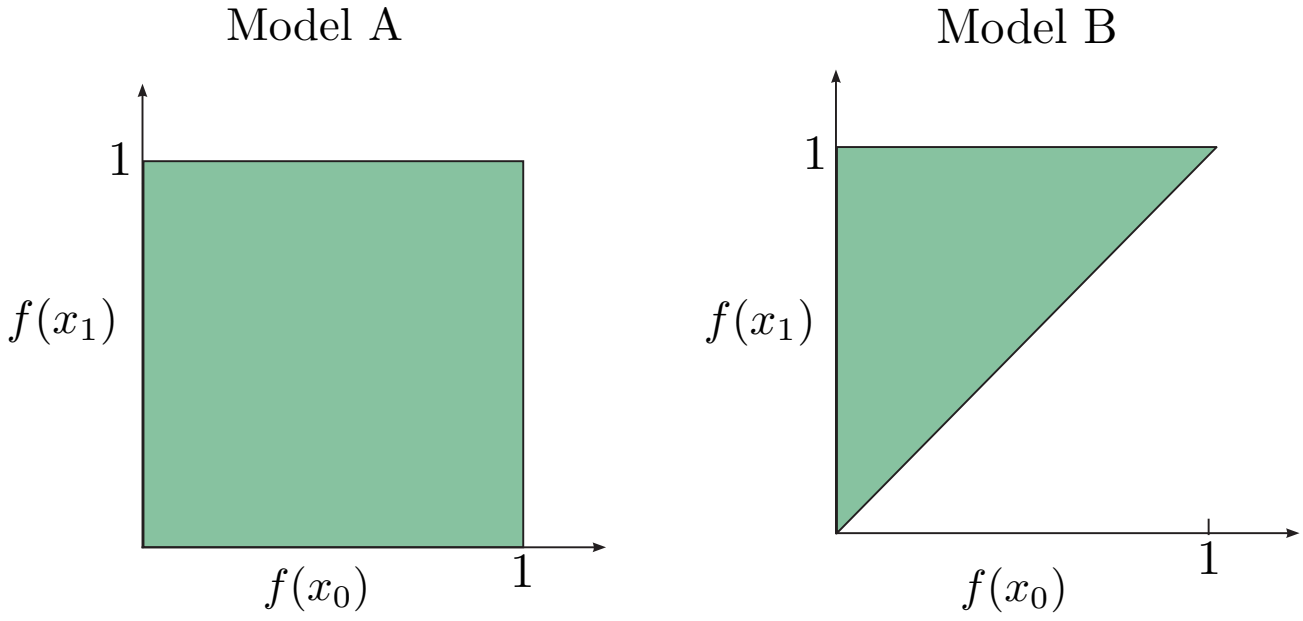
- Data is an observed outcome for each covariate value (a point in $[0, 1] \times [0, 1]$)
- Model A is consistent with all possible data (**unrestrictive**)

Stylized Example



- Data is an observed outcome for each covariate value (a point in $[0, 1] \times [0, 1]$)
- Model A is consistent with all possible data (**unrestrictive**)
- Model B imposes the restriction that $f(x_1) > f(x_0)$, will imperfectly fit some data sets

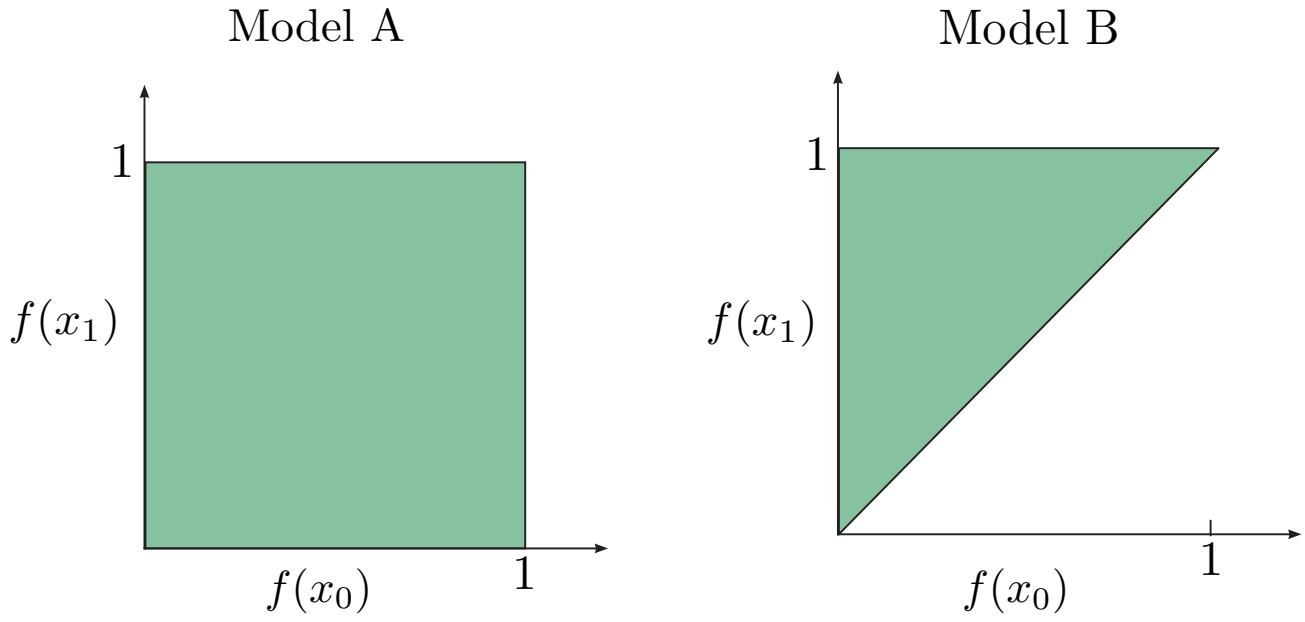
Selten's Measure



One way of evaluating the restrictiveness of these models is the fraction of data that they can fit exactly (Selten, 1991).

- Model A can exactly fit all possible data
- Model B can only fit 1/2 of it

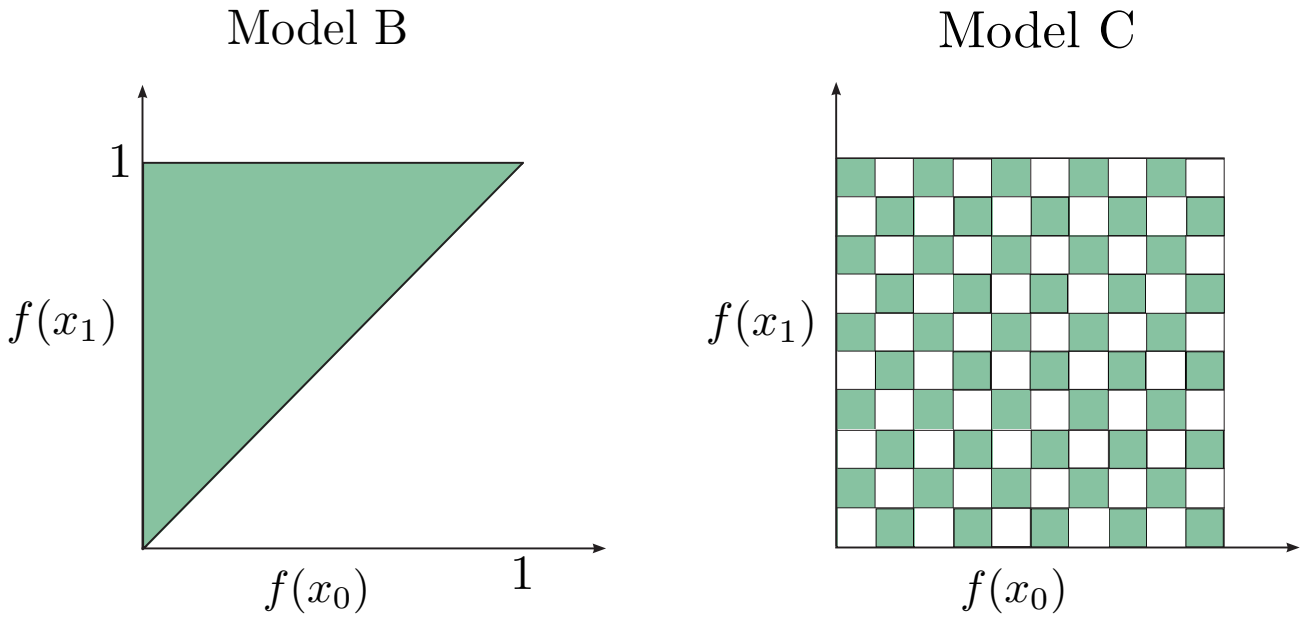
How to Measure Restrictiveness?



Two drawbacks:

- Easy to determine in this example, but can be difficult to determine in general without analytical results.

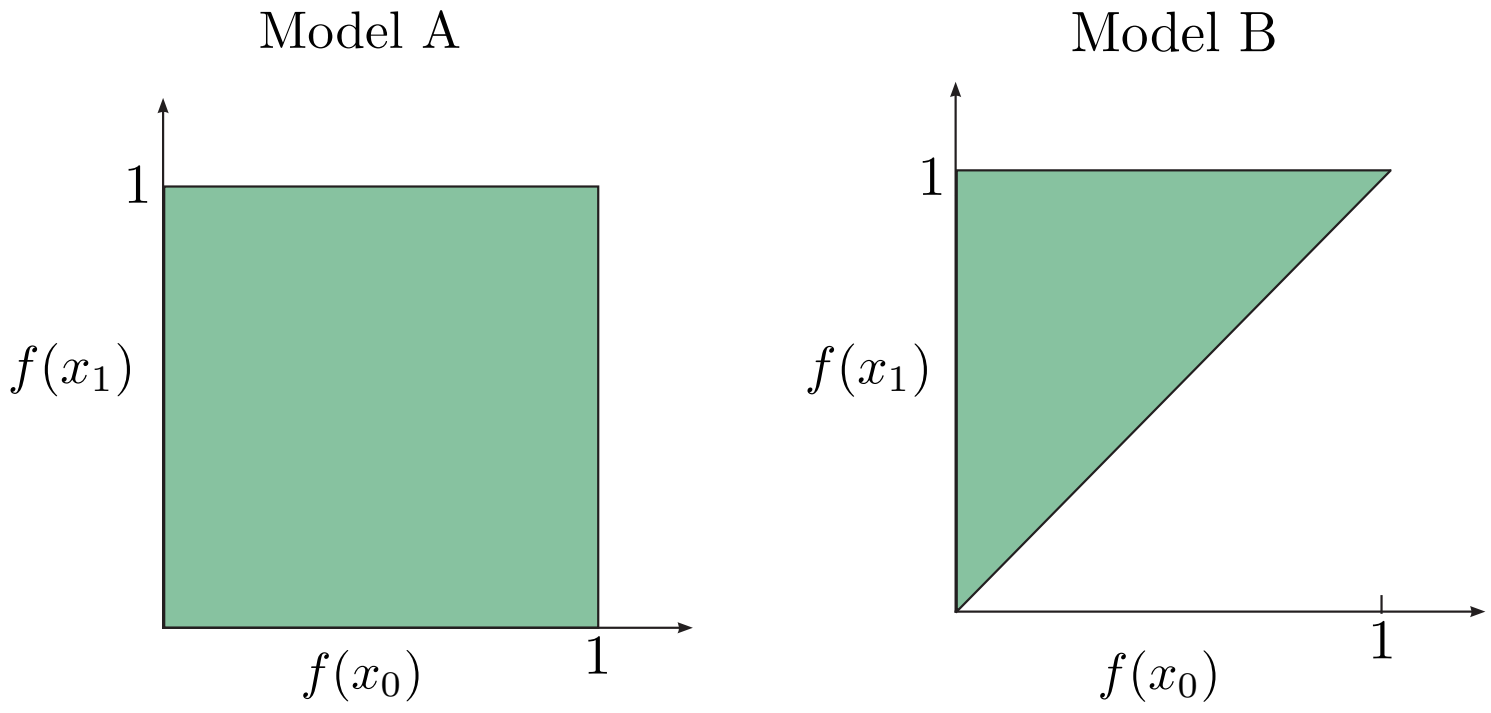
How to Measure Restrictiveness?



Two drawbacks:

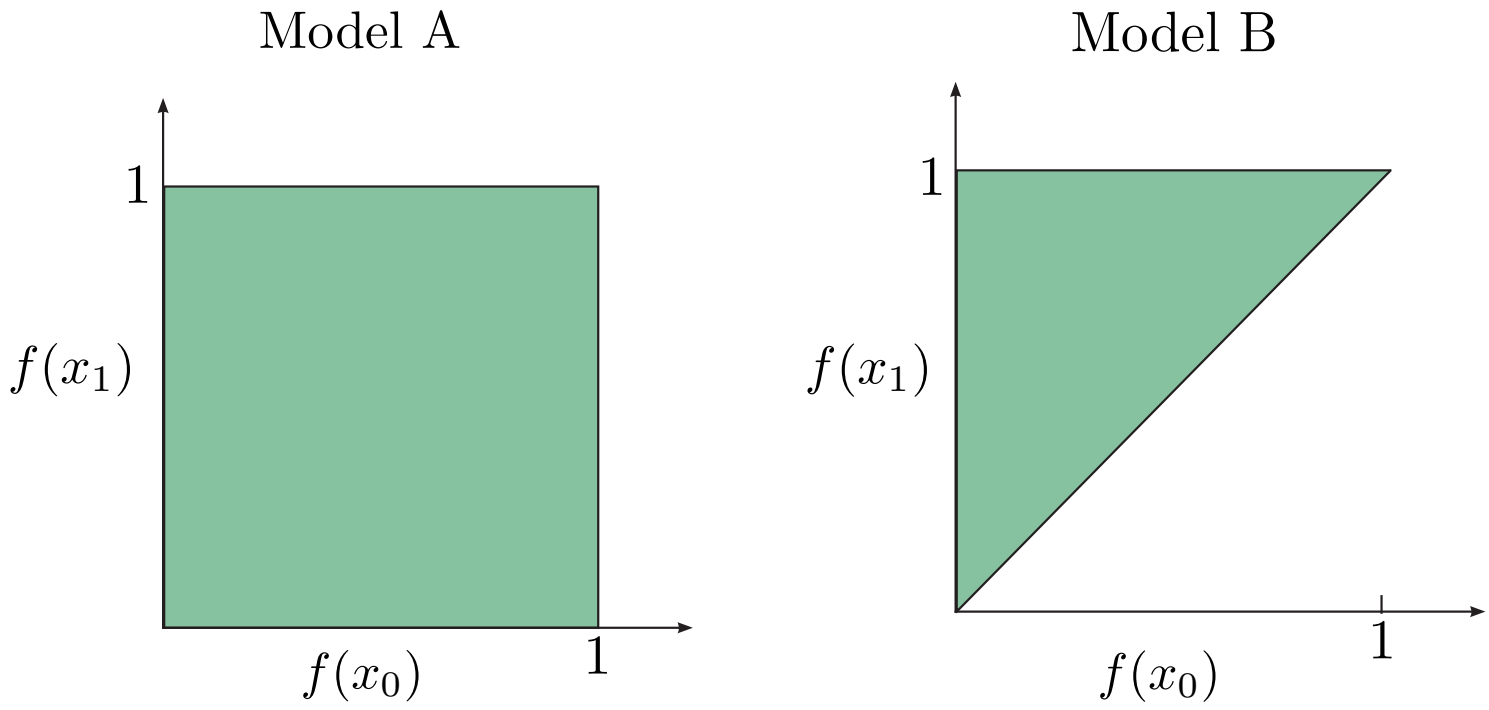
- Easy to determine in this example, but can be difficult to determine in general without analytical results.
- Obscures important differences between models such as between Models B and C

Our Restrictiveness Measure



(1) Decide on any **background constraints** based on existing knowledge about structure of data (e.g., $y \in [0, 1]$)

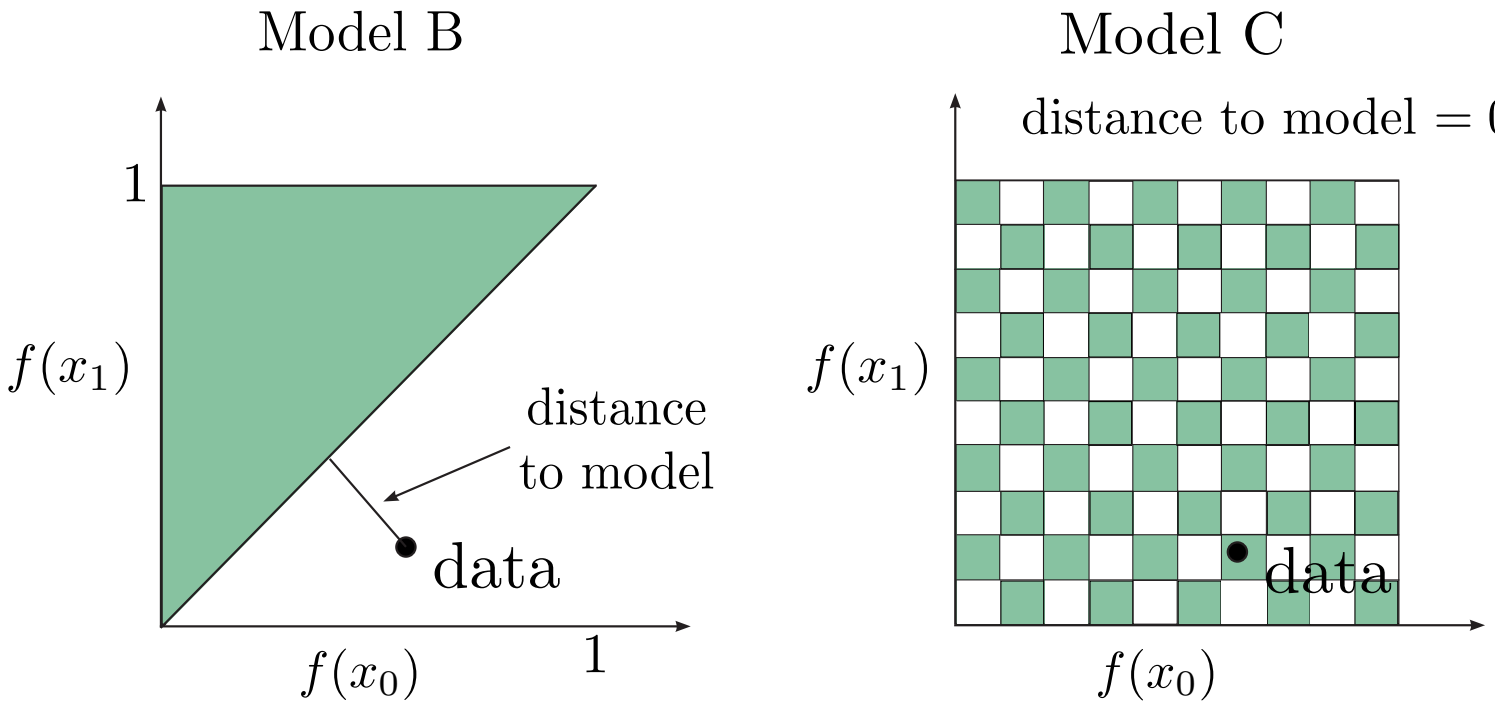
Our Restrictiveness Measure



(1) Decide on any **background constraints** based on existing knowledge about structure of data (e.g., $y \in [0, 1]$)

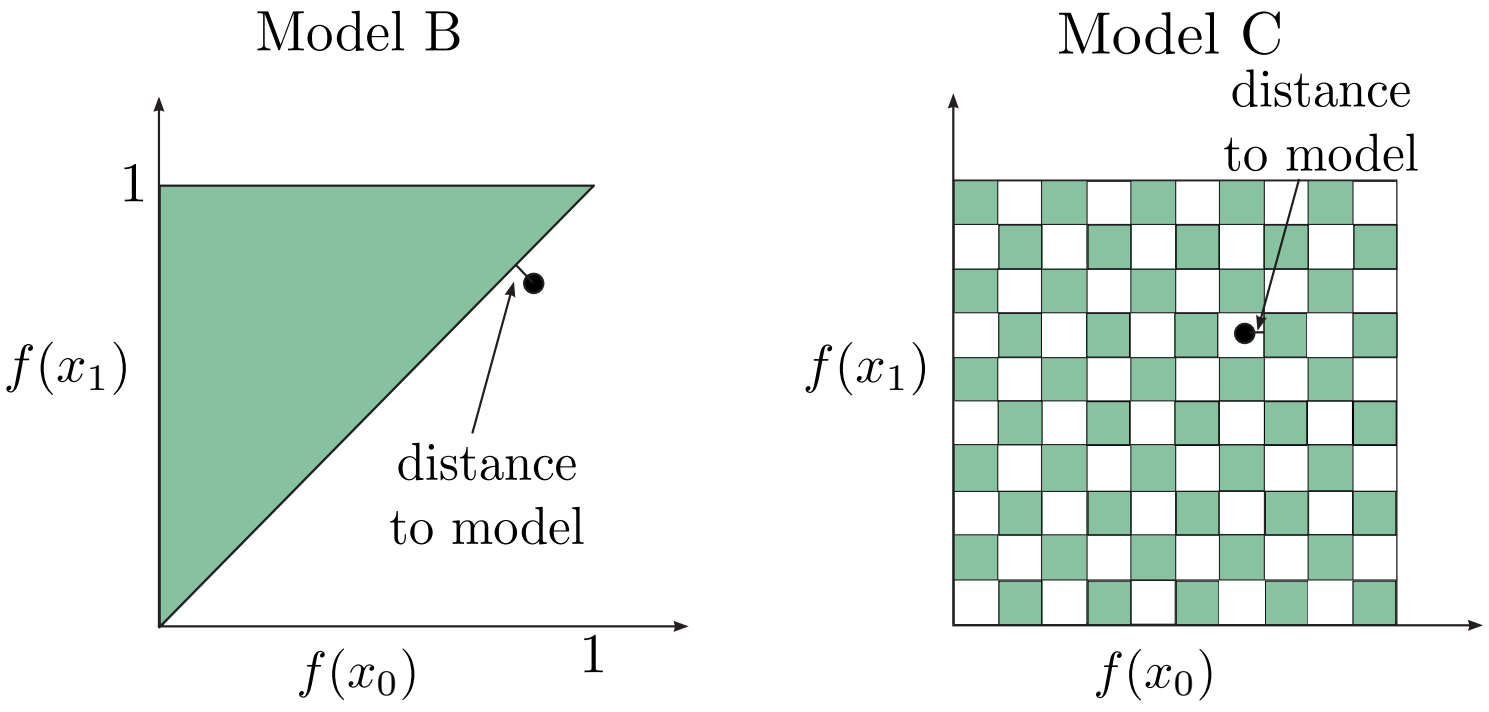
(2) Uniformly sample over all possible data satisfying the background constraints **(synthetic data)**

Our Restrictiveness Measure



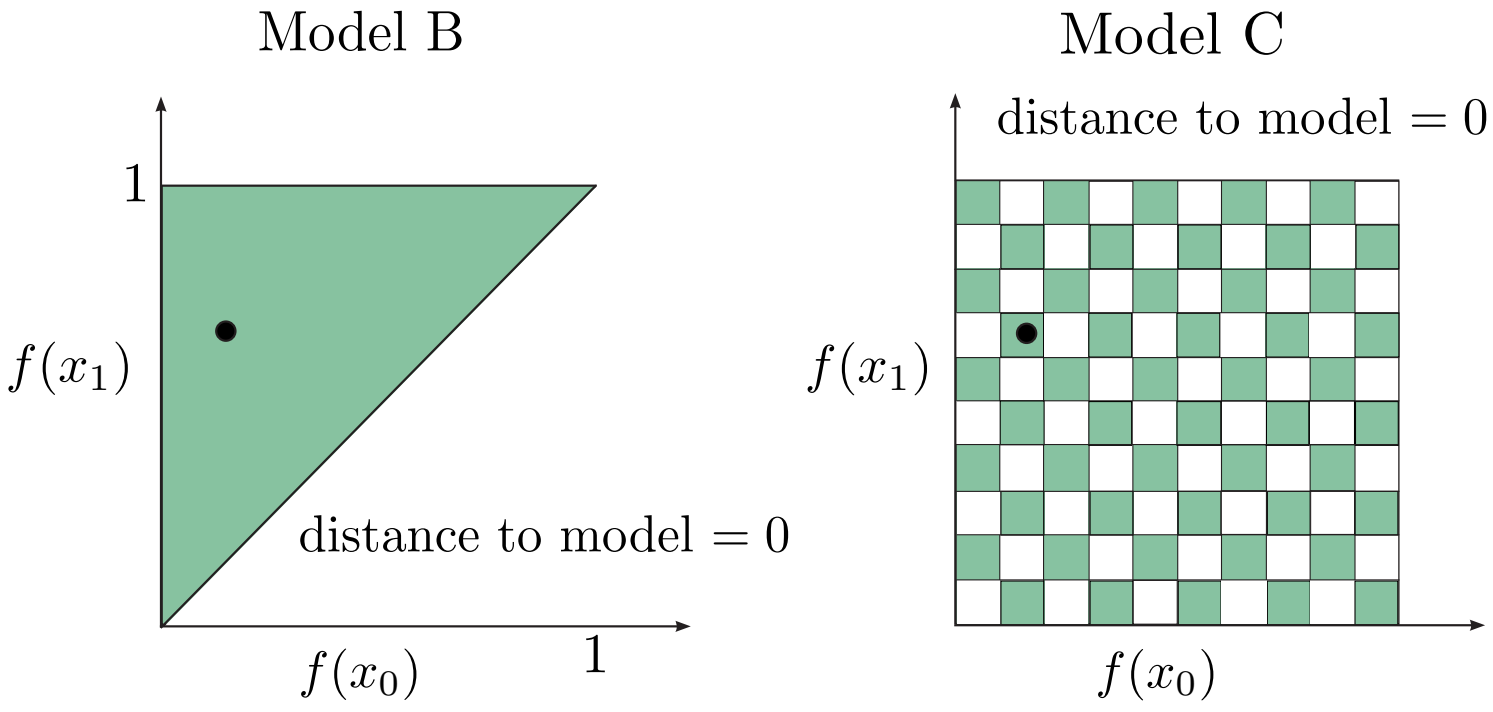
(3) Evaluate the average distance between the model and the realized data **(approximation error)**

Our Restrictiveness Measure



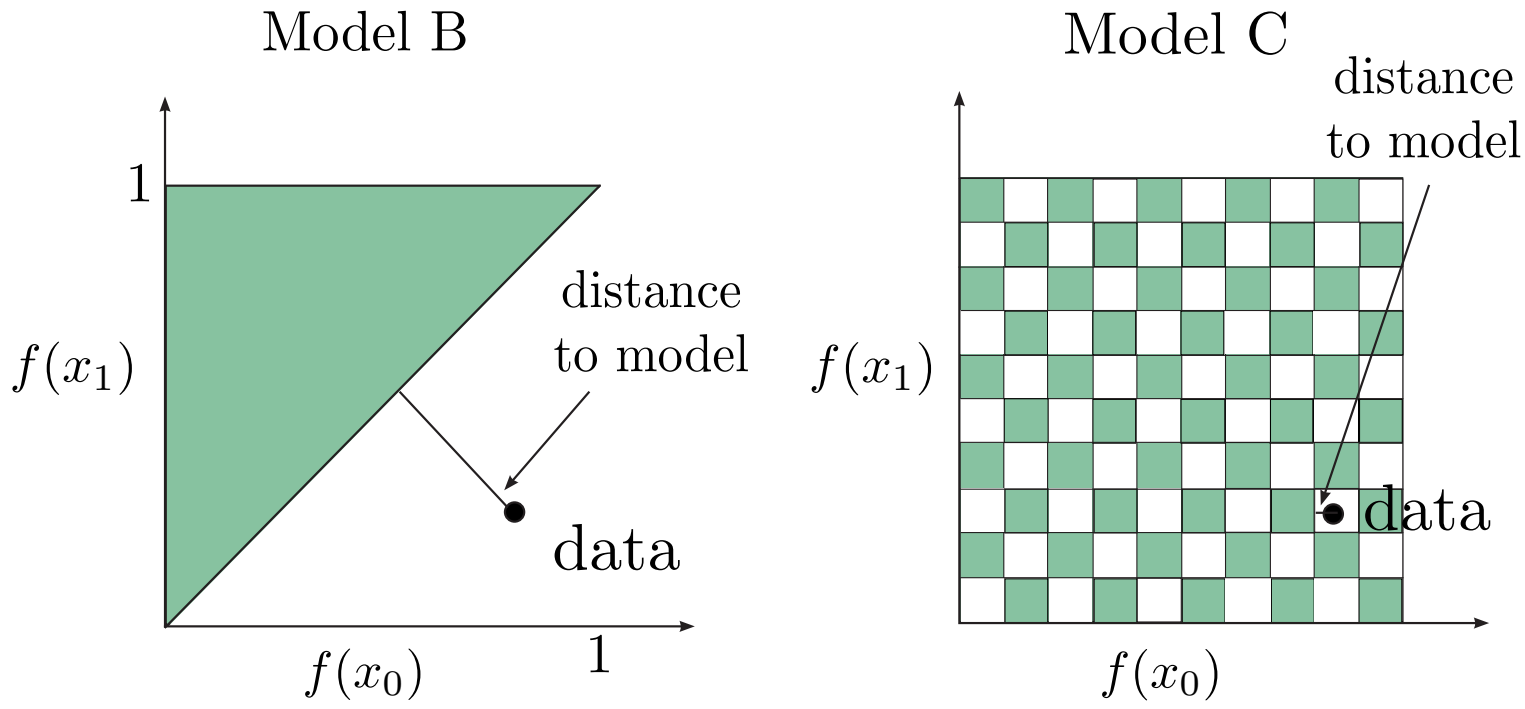
(3) Evaluate the average distance between the model and the realized data **(approximation error)**

Our Restrictiveness Measure



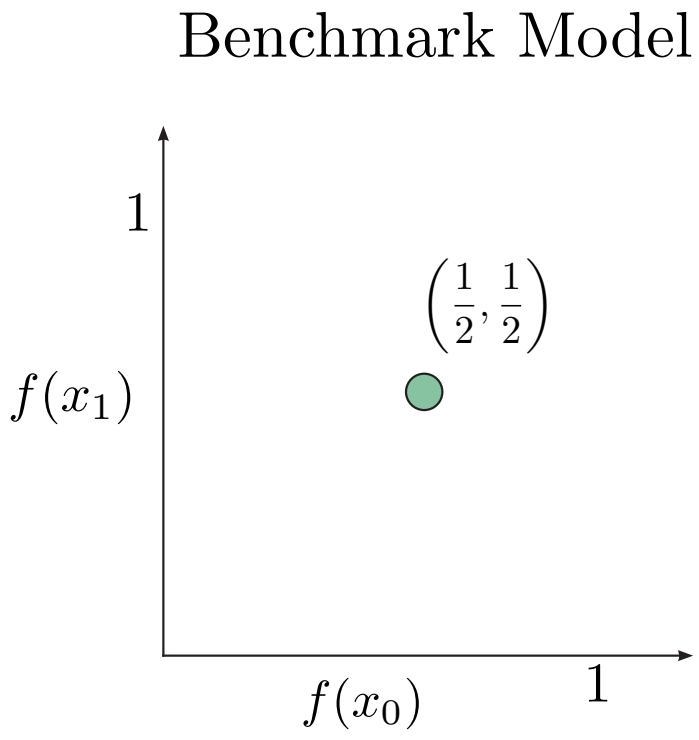
(3) Evaluate the average distance between the model and the realized data **(approximation error)**

Our Restrictiveness Measure



(3) Evaluate the average distance between the model and the realized data **(approximation error)**

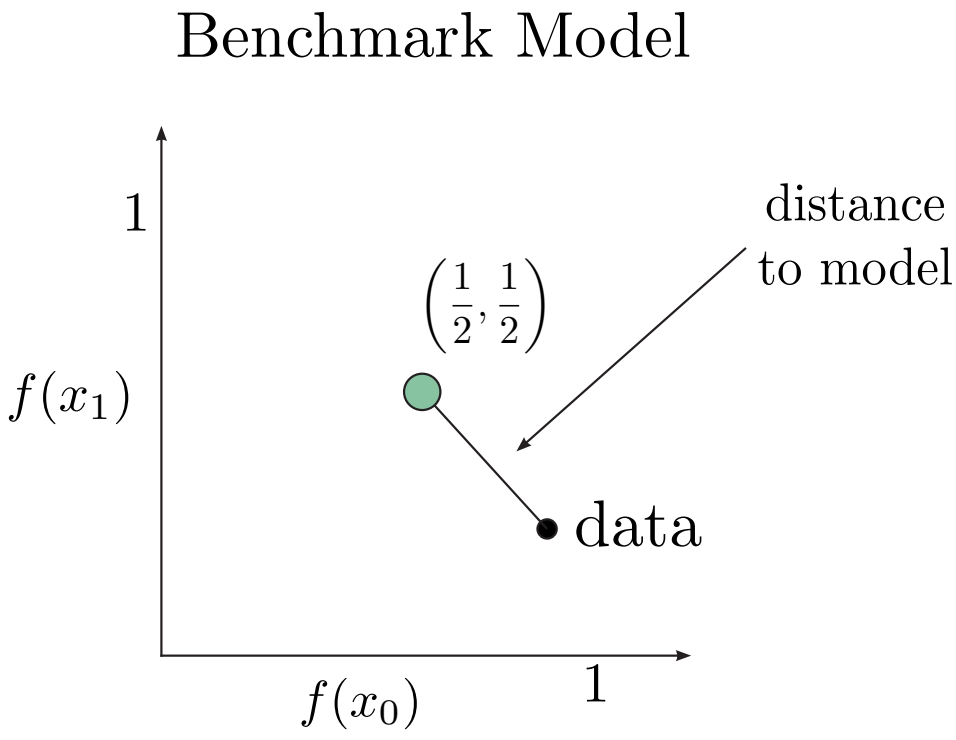
Our Restrictiveness Measure



(3) Evaluate the average distance between the model and the realized data **(approximation error)**

(4) Repeat (3) for some constant benchmark model **(normalization)**, e.g., the model $\{(1/2, 1/2)\}$

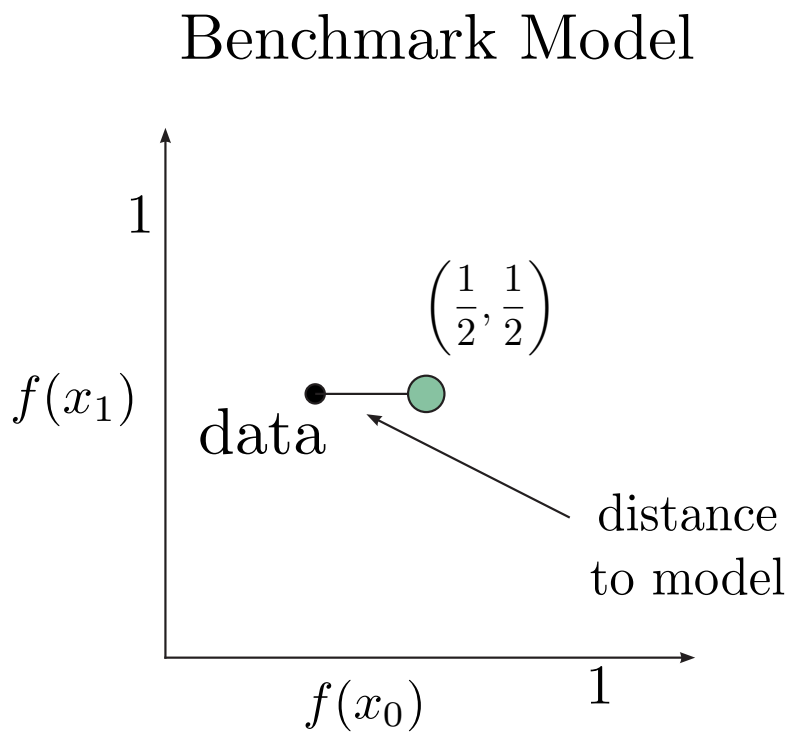
Our Restrictiveness Measure



(3) Evaluate the average distance between the model and the realized data **(approximation error)**

(4) Repeat (3) for some constant benchmark model **(normalization)**, e.g., the model $\{(1/2, 1/2)\}$

Our Restrictiveness Measure



(3) Evaluate the average distance between the model and the realized data **(approximation error)**

(4) Repeat (3) for some constant benchmark model **(normalization)**, e.g., the model $\{(1/2, 1/2)\}$

Our Measure of Restrictiveness

(3) Evaluate the average distance between the model and the realized data **(approximation error)**

(4) Repeat (3) for some constant benchmark model **(normalization)**, e.g., the model $\{(1/2, 1/2)\}$

The restrictiveness of the model is the ratio of (3) to (4).

- ranges from zero (completely unrestrictive) to 1 (as restrictive as the benchmark)

Restrictiveness and Completeness

Restrictiveness:

- Ranges from zero to 1
- Computed from synthetic data
- Larger values mean that the model imposes more restrictions.

Completeness:

- Ranges from zero to 1
- Computed from real data
- Larger values implies a model that predicts real data better.

Prefer models that have high completeness (good fit to real data) and high restrictiveness (poor fit to synthetic data).

Economic Application

Prediction problem:

- $x = (\overline{z}, p; \underline{z}, 1-p)$ is a binary lottery
- y is a subject's certainty equivalent for that lottery
  - subject is indifferent between receiving y dollars for sure versus the random outcome of the lottery
- evaluate mean-squared error $(\hat{y} - y)^2$ given prediction $\hat{y}$

The real data:

- 25 binary lotteries $(\overline{z}, p; \underline{z}, 1-p)$ from Bruhin et. al (2010)
- 179 reported certainty equivalents per lottery

the 25 lotteries from the Bruhin et al. (2010) dataset

$\overline{z}$	p	$\underline{z}$	$1-p$	$f(\overline{z}, p; \underline{z}, 1-p)$
20	0.25	0	0.75	
40	0.95	10	0.05	
⋮	⋮	⋮		
150	0.05	50	0.95	

generate an average certainty equivalent
for each of these lotteries

$\overline{z}$	p	$\underline{z}$	$1-p$	$f(\overline{z}, p; \underline{z}, 1-p)$
20	0.25	0	0.75	
40	0.95	10	0.05	
⋮	⋮	⋮		
150	0.05	50	0.95	

background constraints:

1. if one lottery first-order stochastically dominates another, then its average certainty equivalent is higher (people like more money)
2. certainty equivalents fall within the range of outcomes

$\overline{z}$	p	$\underline{z}$	$1-p$	$f(\overline{z}, p; \underline{z}, 1-p)$
20	0.25	0	0.75	
40	0.95	10	0.05	
⋮	⋮	⋮		
150	0.05	50	0.95	

background constraints:

1. if one lottery first-order stochastically dominates another, then its average certainty equivalent is higher (people like more money)
2. certainty equivalents fall within the range of outcomes

$\overline{z}$	p	$\underline{z}$	$1-p$	$f(\overline{z}, p; \underline{z}, 1-p)$
20	0.25	0	0.75	15.96
40	0.95	10	0.05	18.58
⋮	⋮	⋮	⋮	⋮
150	0.05	50	0.95	83.71

background constraints:

1. if one lottery first-order stochastically dominates another, then its average certainty equivalent is higher (people like more money)
2. certainty equivalents fall within the range of outcomes

$\overline{z}$	p	$\underline{z}$	$1-p$	$f(\overline{z}, p; \underline{z}, 1-p)$
20	0.25	0	0.75	17.04
40	0.95	10	0.05	39.45
⋮	⋮	⋮	⋮	⋮
150	0.05	50	0.95	73.99

background constraints:

1. if one lottery first-order stochastically dominates another, then its average certainty equivalent is higher (people like more money)
2. certainty equivalents fall within the range of outcomes

$\overline{z}$	p	$\underline{z}$	$1-p$	$f(\overline{z}, p; \underline{z}, 1-p)$
20	0.25	0	0.75	17.04
40	0.95	10	0.05	39.45
⋮	⋮	⋮	⋮	⋮
150	0.05	50	0.95	73.99

Sample uniformly over all vectors satisfying these background constraints.

Two Economic Models

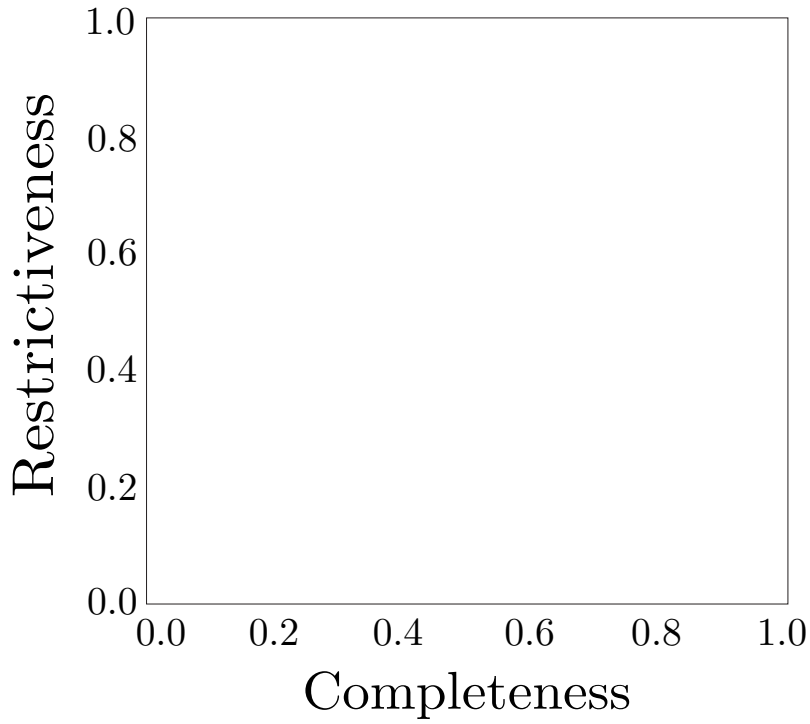
The predicted utility for lottery $(\overline{z}, p; \underline{z}, 1-p)$ is

$$w(p) \times \overline{z}^{\alpha} + (1 - w(p)) \times \underline{z}^{\alpha}$$

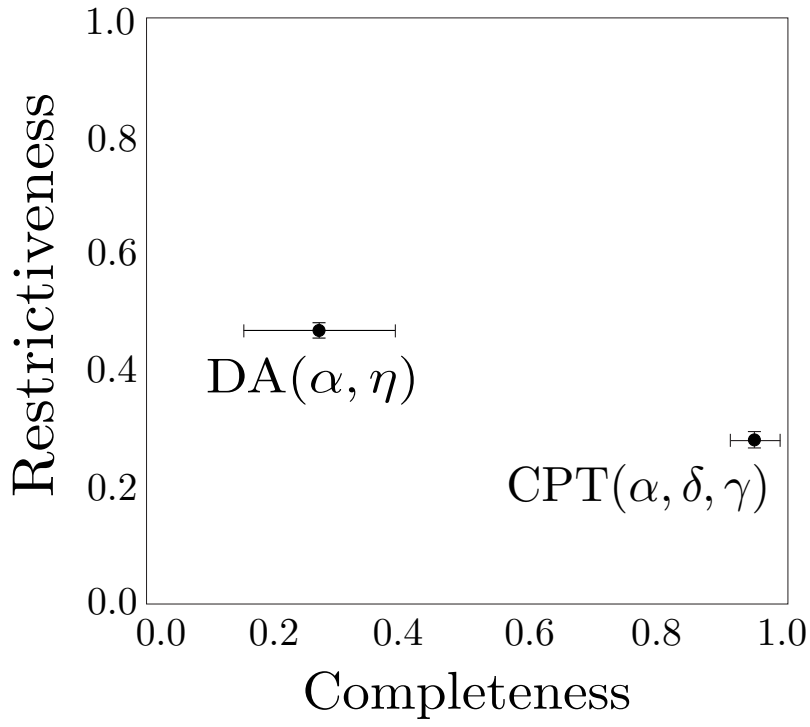
where $w(p)$ is a probability weighting function satisfying either:

- $w(p) = \frac{\delta p^{\gamma}}{\delta p^{\gamma} + (1-p)^{\gamma}}$ **(Cumulative Prospect Theory)**, or
- $w(p) = \frac{p}{1+(1-p)\eta}$ **(Disappointment Aversion)**

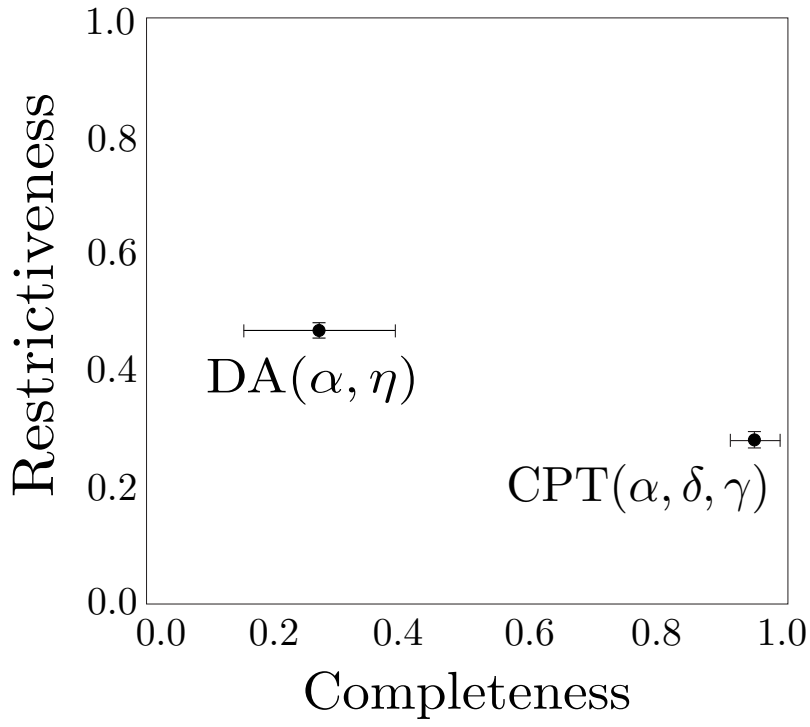
Comparison of Models



Comparison of Models

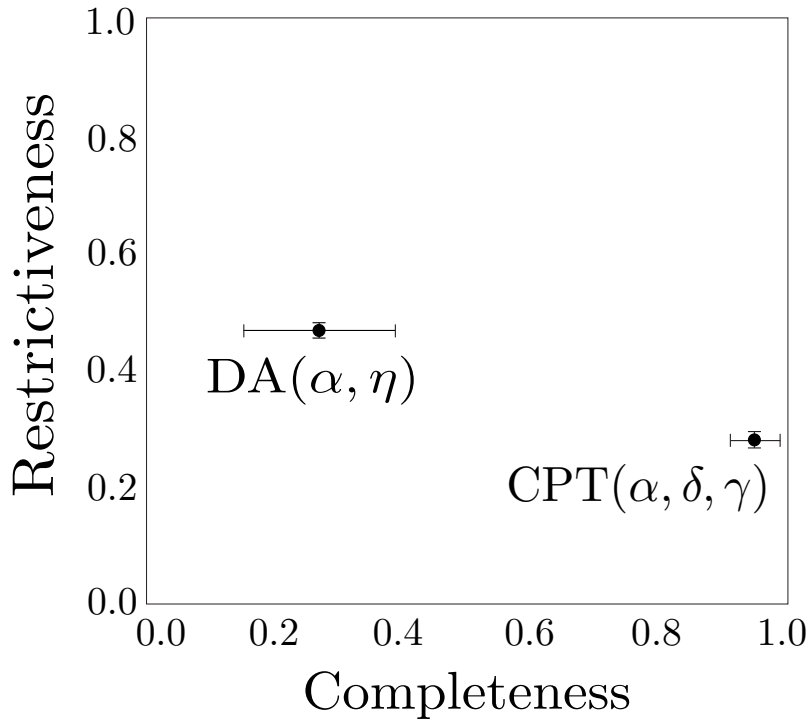


Comparison of Models



- CPT(α, δ, γ) is nearly complete but not very restrictive.

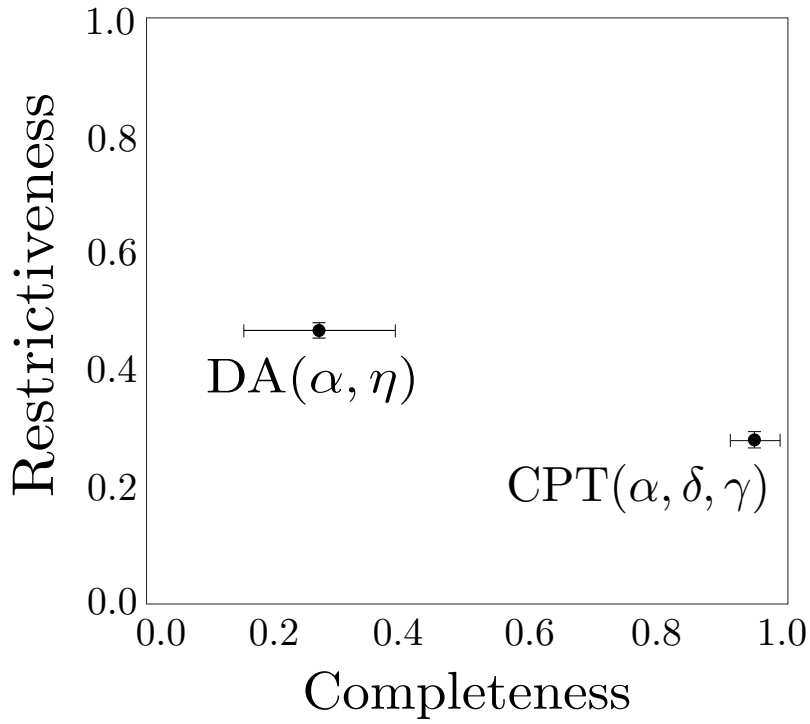
Comparison of Models



- CPT(α, δ, γ) is nearly complete but not very restrictive.

  This flexibility is not revealed by a simple count of the number of free parameters!

Comparison of Models

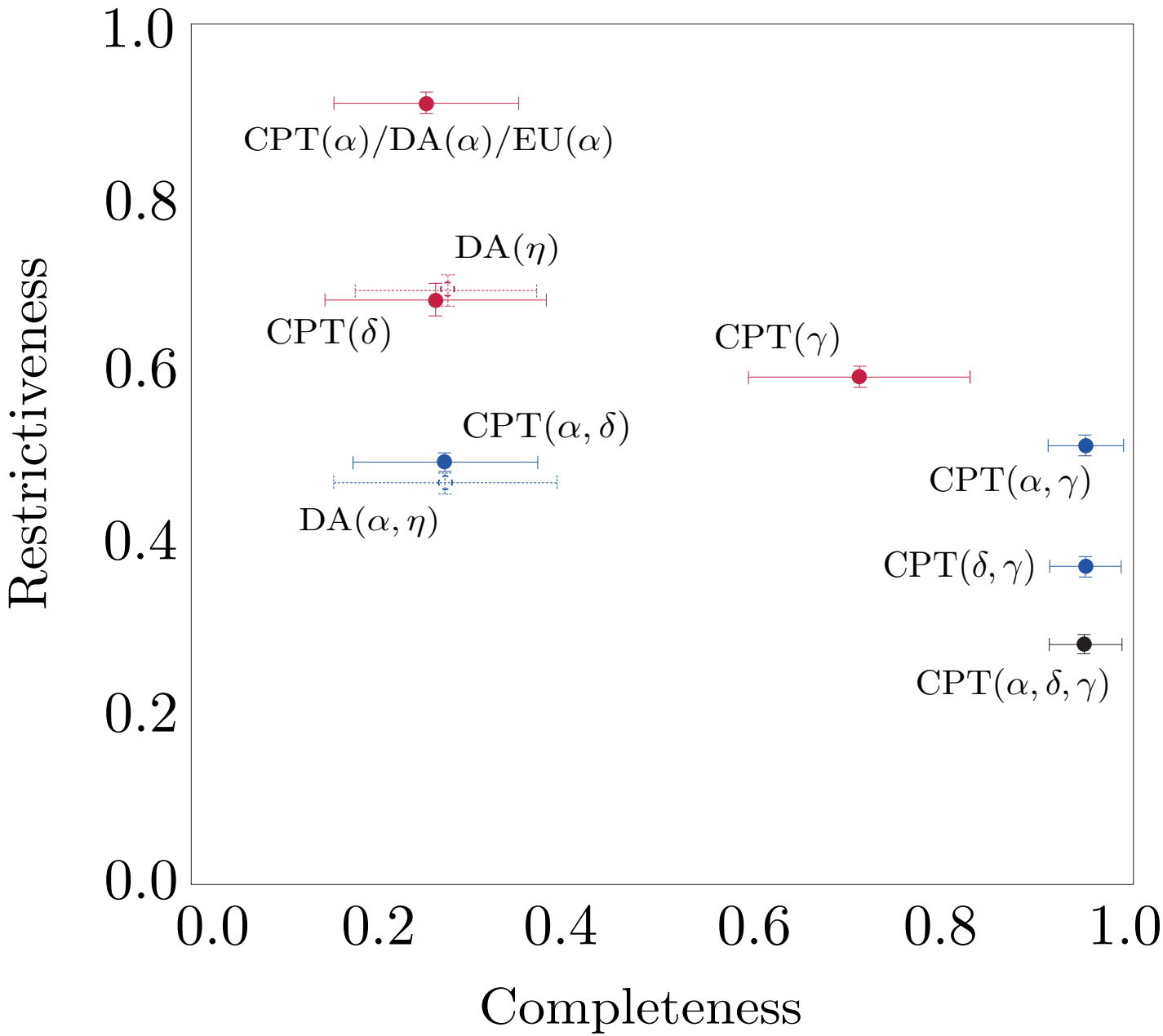


- CPT(α, δ, γ) is nearly complete but not very restrictive.

  This flexibility is not revealed by a simple count of the number of free parameters!

- DA(α, η) is more restrictive than CPT(α, δ, γ), but substantially less predictive of the real data.

Completeness-Restrictiveness Pareto Frontier



Part V:
Transfer Performance

The Transfer Performance of Economic Models, 2023
(Andrews, Fudenberg, Lei, Liang, and Wu)

Generalizability

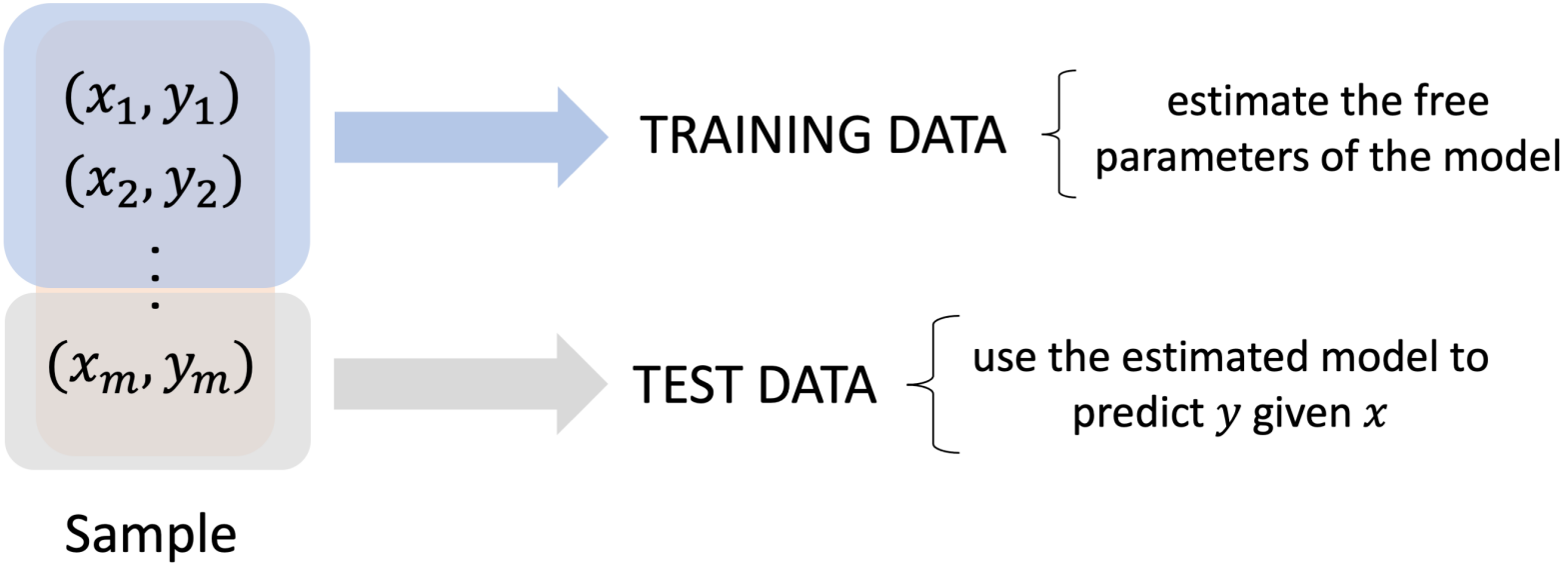
- Previous analyses have all been "within domain"
  - train and test on data drawn from the same economic context
- But economic models are meant to capture structure that is shared across contexts
- When a model is complete on data from a given context, will it also perform well on data from another?

Out of Sample Testing

So far have considered the standard "out of sample" test for a model:

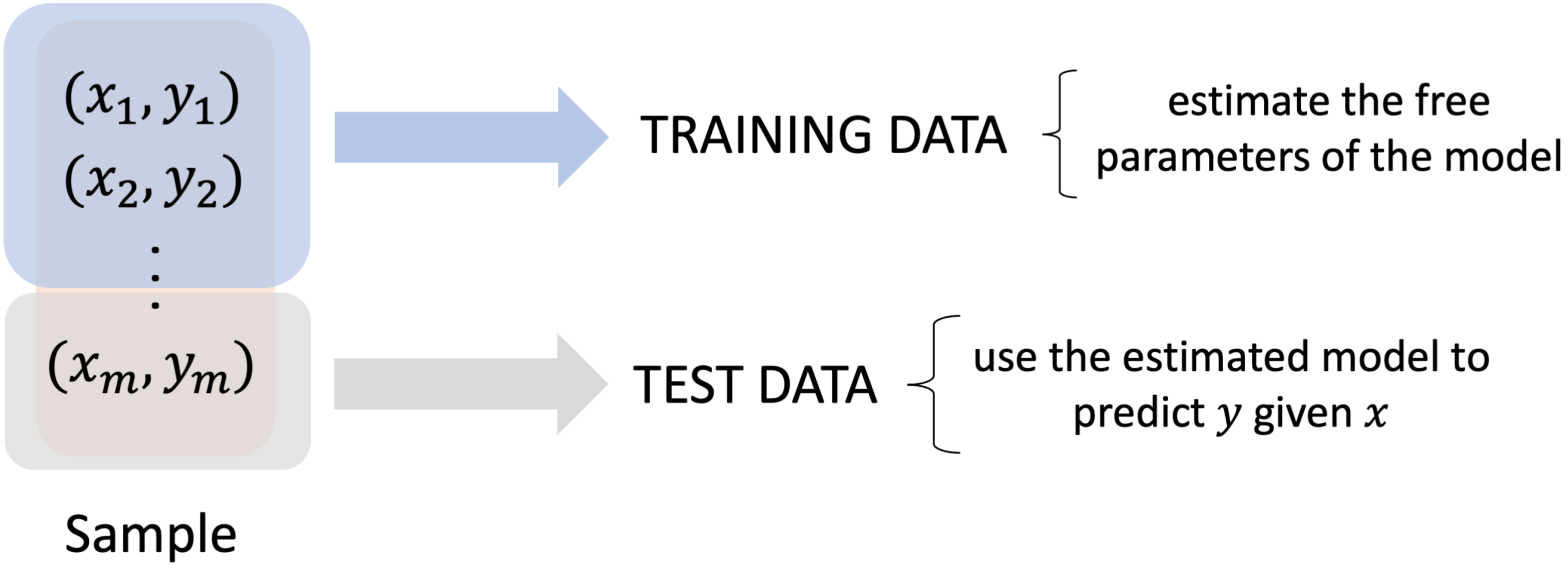
Out of Sample Testing

So far have considered the standard "out of sample" test for a model:



Out of Sample Testing

So far have considered the standard "out of sample" test for a model:



Performance on test data informative about performance on new unseen sample of observations **drawn from the same distribution**.

Out of Domain Testing

But sometimes what we're interested in instead is how well the estimated model will perform on data from a **different** distribution:

$$(x_1, y_1)$$
$$(x_2, y_2)$$
$$\vdots$$
$$(x_{m_1}, y_{m_1})$$

U.S. data

$$(x_i, y_i) \sim P_1$$

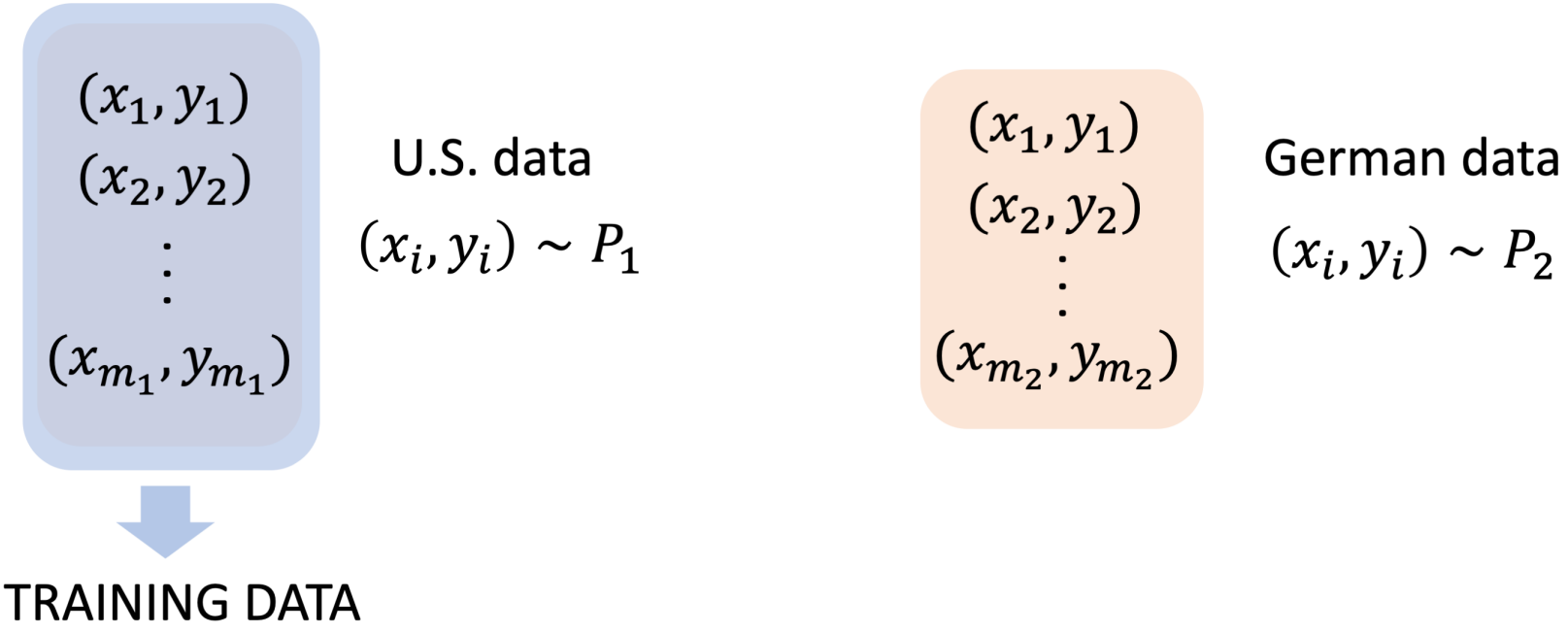
$$(x_1, y_1)$$
$$(x_2, y_2)$$
$$\vdots$$
$$(x_{m_2}, y_{m_2})$$

German data

$$(x_i, y_i) \sim P_2$$

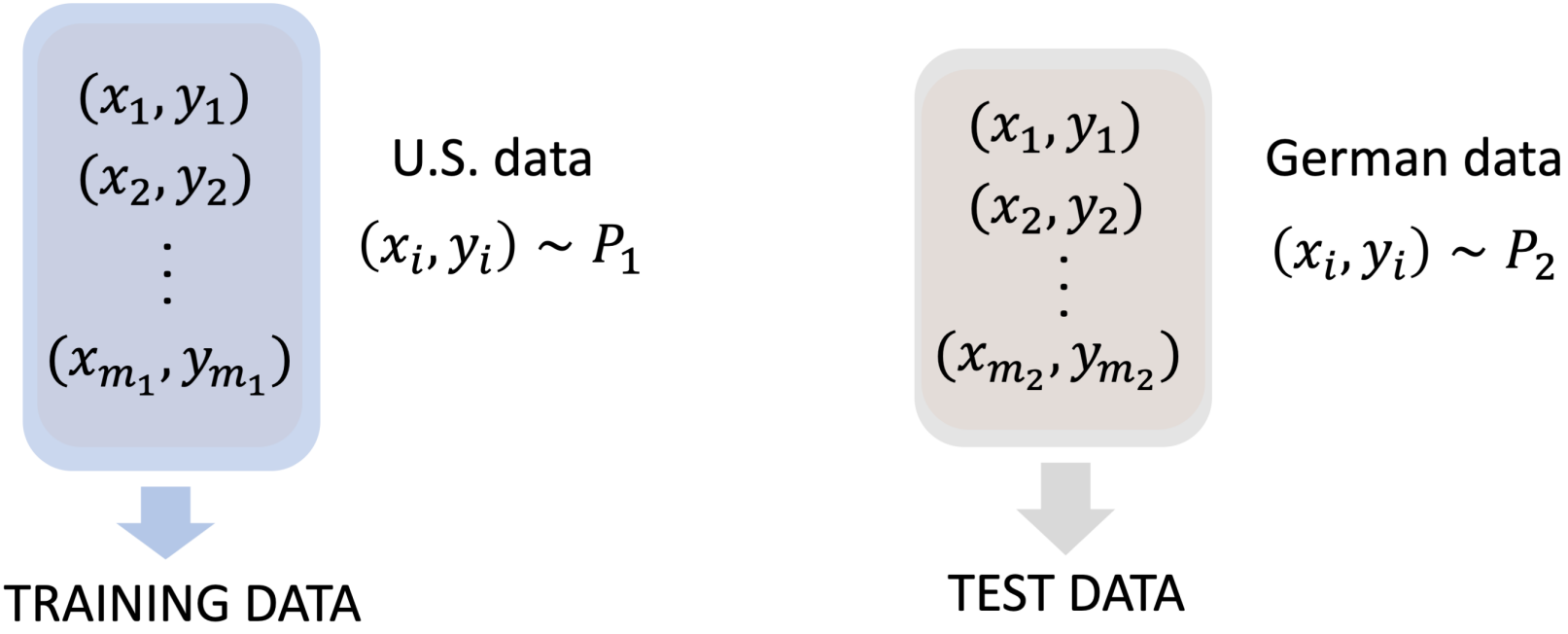
Out of Domain Testing

But sometimes what we're interested in instead is how well the estimated model will perform on data from a **different** distribution:



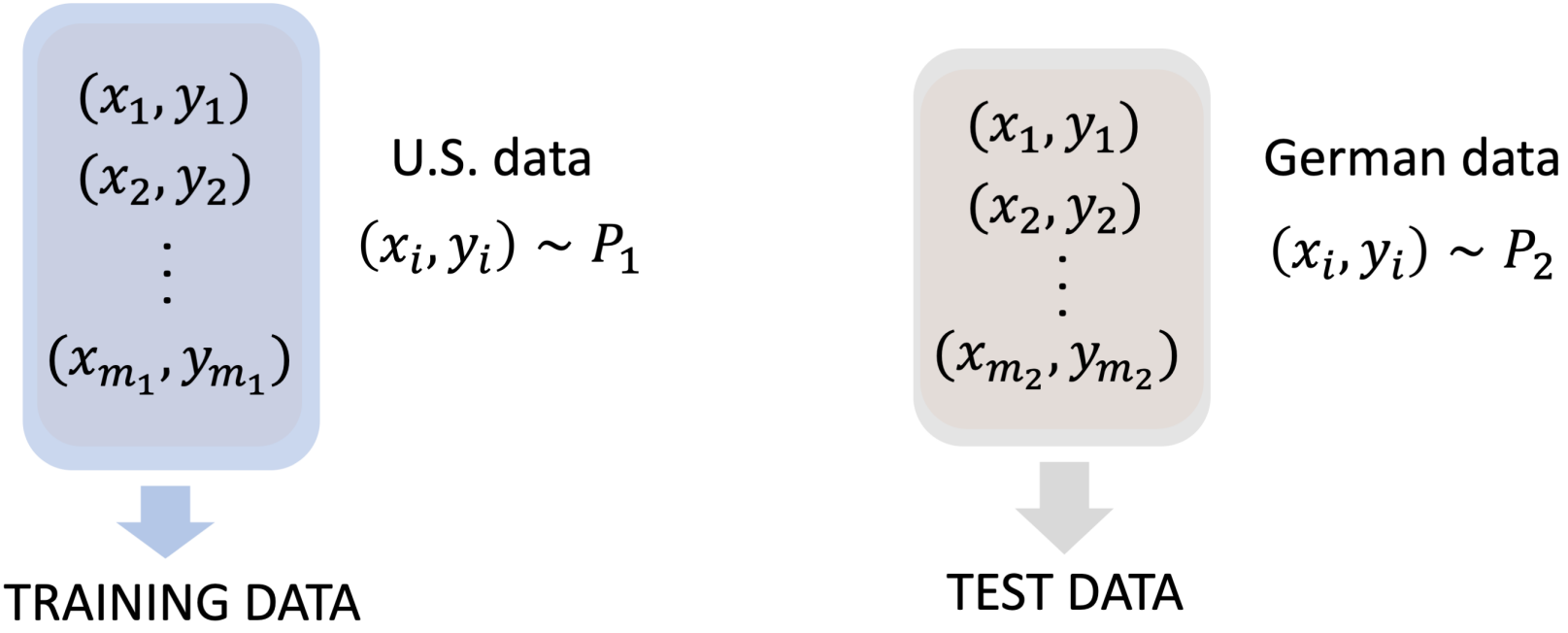
Out of Domain Testing

But sometimes what we're interested in instead is how well the estimated model will perform on data from a **different** distribution:



Out of Domain Testing

But sometimes what we're interested in instead is how well the estimated model will perform on data from a **different** distribution:



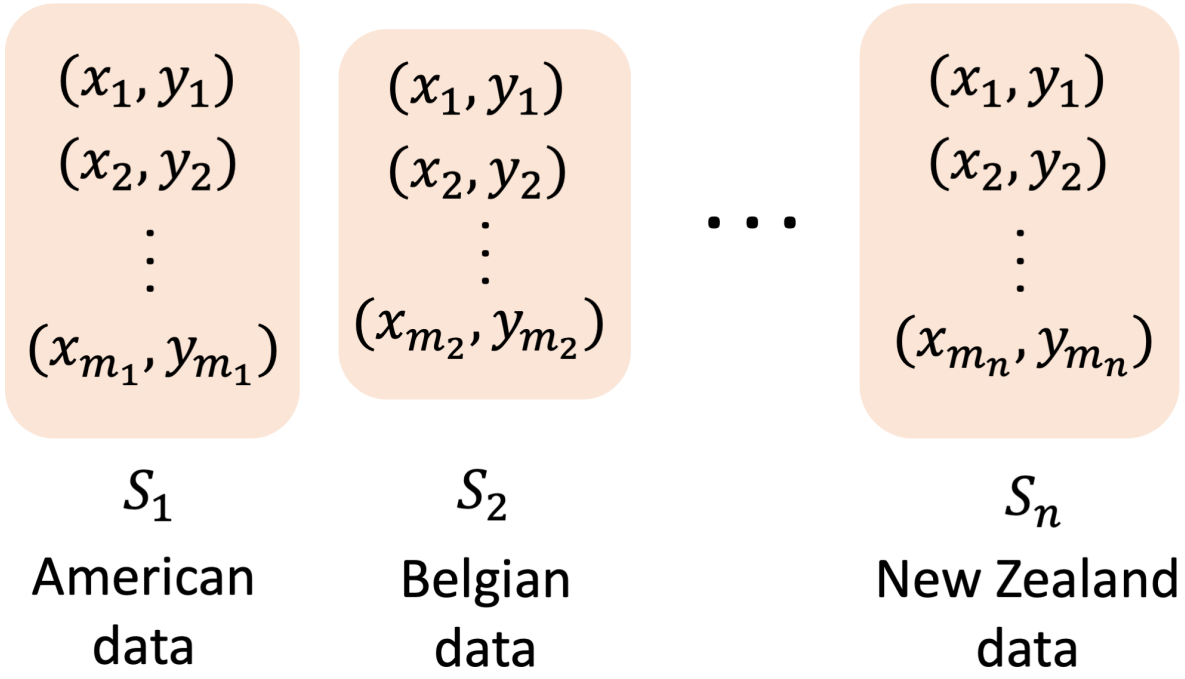
Can we use "out of domain" testing to learn about the estimated model's performance on a sample from a new domain?

Underlying Statistical Model

- Suppose each sample S_i consists of observations (x, y) drawn iid from some distribution P_i
- The distributions P_i vary across contexts but are themselves drawn iid from some underlying distribution
- The quantity that we're interested in is the transfer error for a model estimated on one sample and tested on another

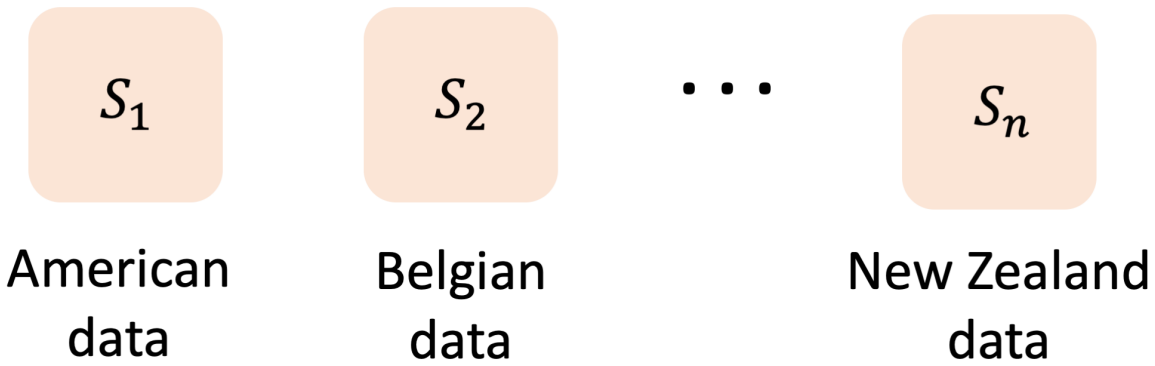
Analyst's Metadata

The analyst has access to *metadata* $\mathbf{M} = (S_1, \ldots, S_n)$ consisting of n samples independently generated in this way.

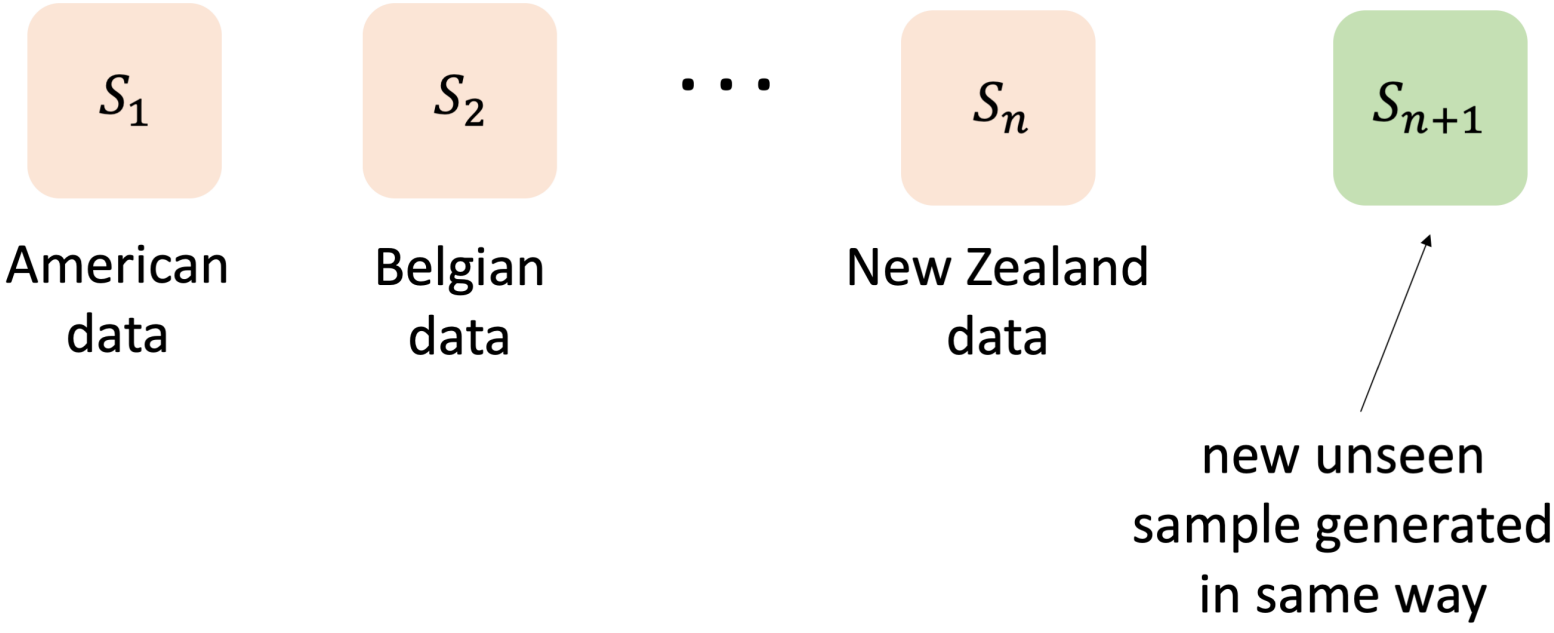


Analyst's Metadata

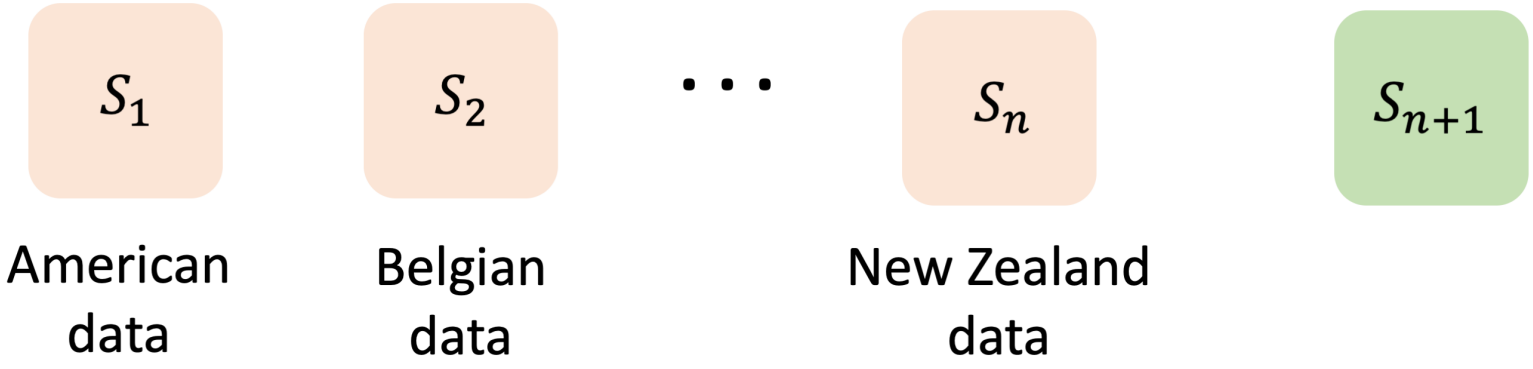
The analyst has access to *metadata* $\mathbf{M} = (S_1, \ldots, S_n)$ consisting of n samples independently generated in this way.



The Transfer Prediction Problem



The Transfer Prediction Problem



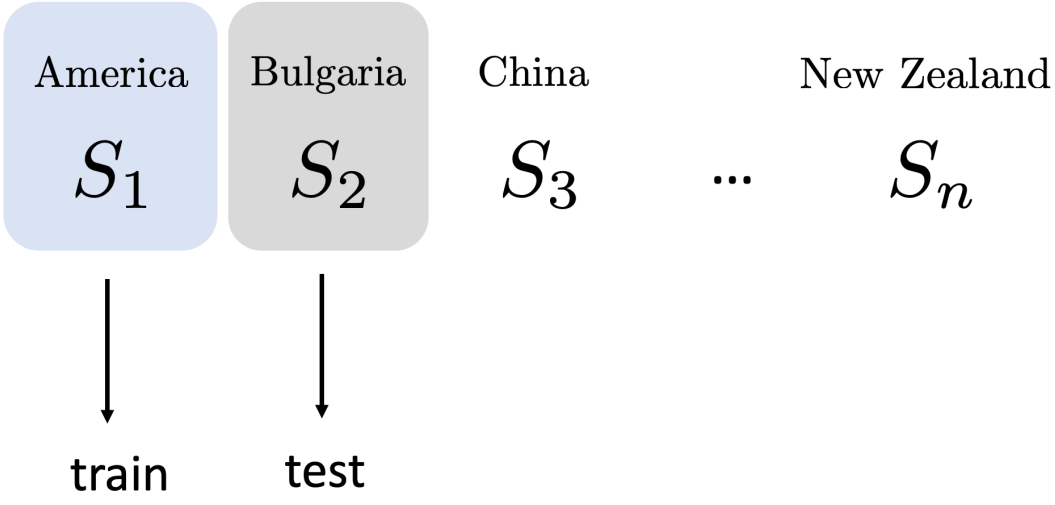
Construction of Confidence Interval

America Bulgaria China ... New Zealand

S_1 S_2 S_3 ... S_n

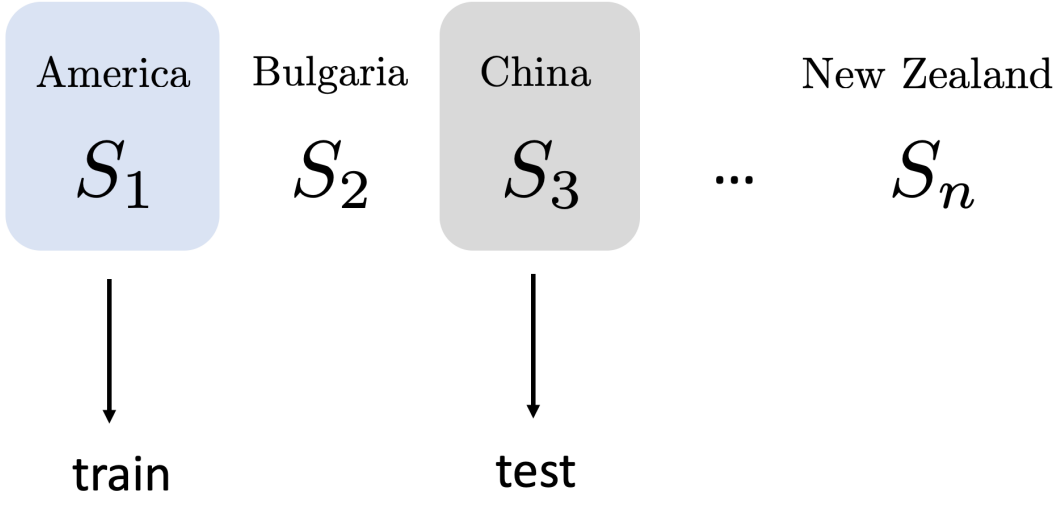
Consider all possible choices of one training sample and one testing sample.

Construction of Confidence Interval



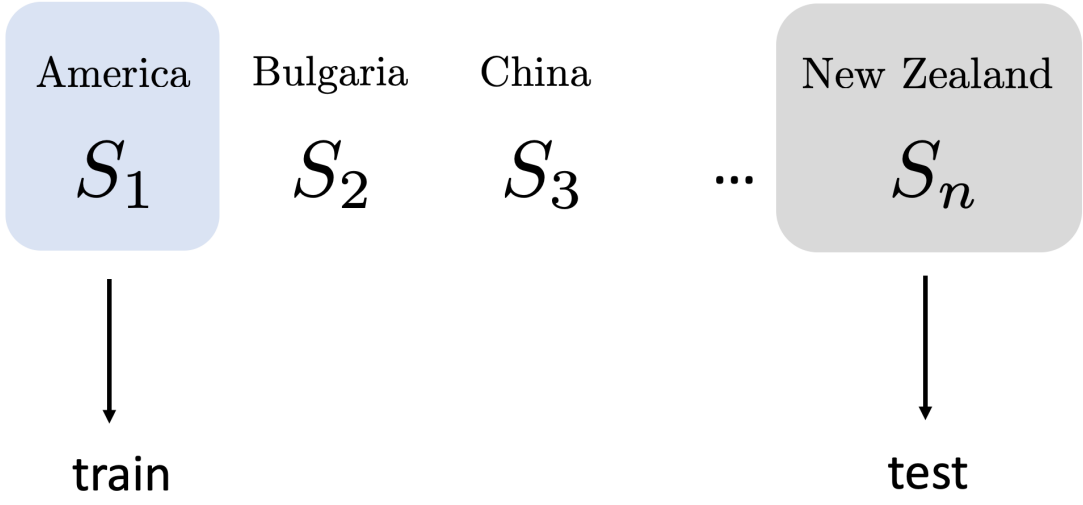
Consider all possible choices of one training sample and one testing sample.

Construction of Confidence Interval



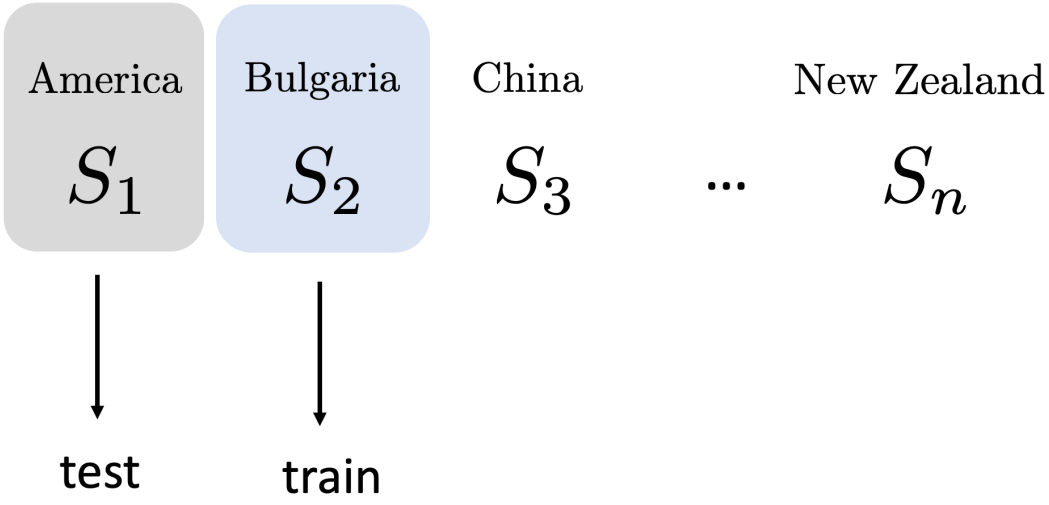
Consider all possible choices of one training sample and one testing sample.

Construction of Confidence Interval



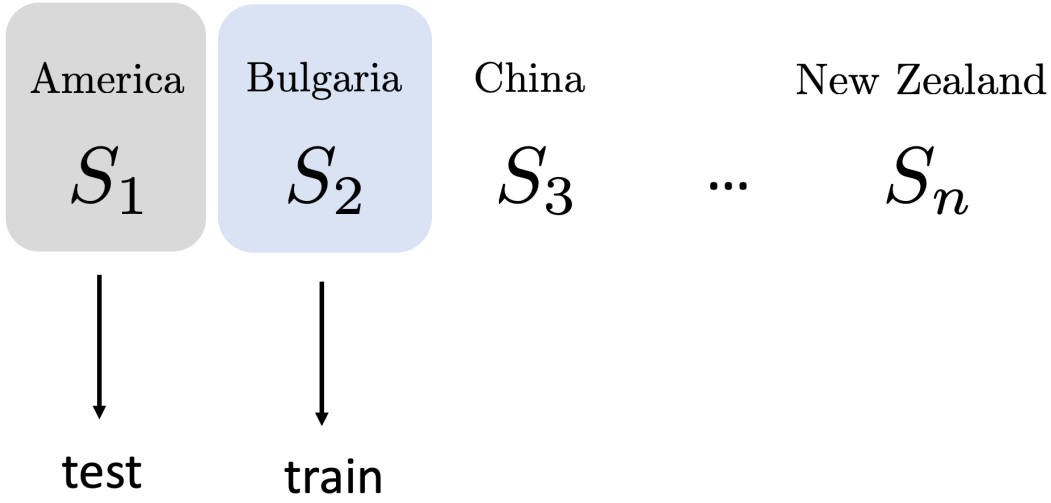
Consider all possible choices of one training sample and one testing sample.

Construction of Confidence Interval



Consider all possible choices of one training sample and one testing sample.

Construction of Confidence Interval



Let $e_{t,d}$ denote the error when we estimate the model's parameters on S_t and test on S_d

Construction of Confidence Interval

train \ test	1	2	$\dots$	$n-1$	n
1	$-$	$e_{1,2}$	$\dots$	$e_{1,n-1}$	$e_{1,n}$
2	$e_{2,1}$	$-$	$\ddots$		$\vdots$
$\vdots$	$\vdots$	$\ddots$	$-$	$\ddots$	$\vdots$
$n-1$	$\vdots$		$\ddots$	$-$	$e_{n-1,n}$
n	$e_{n,1}$	$\dots$	$\dots$	$e_{n,n-1}$	$-$

Construction of Confidence Interval

train \ test	1	2	$\dots$	$n-1$	n
1	$-$	$e_{1,2}$	$\dots$	$e_{1,n-1}$	$e_{1,n}$
2	$e_{2,1}$	$-$	$\ddots$		$\vdots$
$\vdots$	$\vdots$	$\ddots$	$-$	$\ddots$	$\vdots$
$n-1$	$\vdots$		$\ddots$	$-$	$e_{n-1,n}$
n	$e_{n,1}$	$\dots$	$\dots$	$e_{n,n-1}$	$-$

Definition: Let e_τ denote the τth quantile of the pooled sample of transfer errors.

Main Result

For any $\tau \in (0, 1)$,

$$[e_{1-\tau}, e_{\tau}]$$

is a level-$\left(2\tau\left(\frac{n-1}{n+1}\right) - 1\right)$ two-sided confidence interval for the transfer error of the model on a new sample

Example Application of Results

Prediction problem:

- $x = (\overline{z}, p; \underline{z}, 1 - p)$ is a binary lottery
- y is a subject's certainty equivalent for that lottery
- evaluate mean-squared error $(\hat{y} - y)^2$ given prediction $\hat{y}$

Example Application of Results

Prediction problem:

- $x = (\overline{z}, p; \underline{z}, 1-p)$ is a binary lottery
- y is a subject's certainty equivalent for that lottery
- evaluate mean-squared error $(\hat{y} - y)^2$ given prediction $\hat{y}$

Meta-data:

- 44 samples of reported certainty equivalents across different subject pools (from 14 papers)
- on average, 2752 observations per domain

Example Application of Results

Prediction problem:

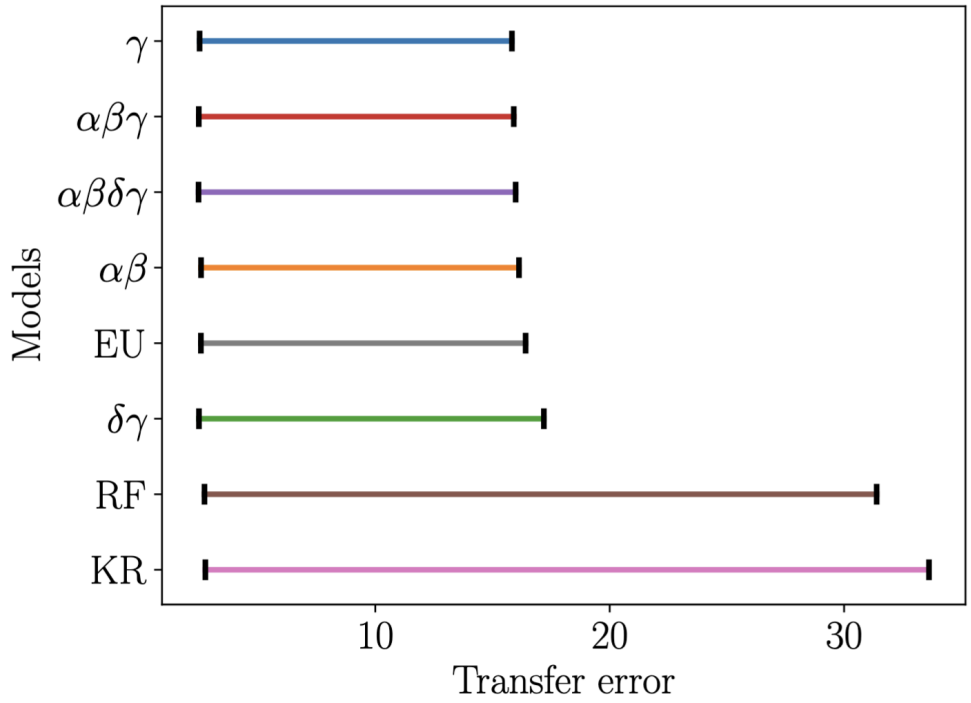
- $x = (\overline{z}, p; \underline{z}, 1-p)$ is a binary lottery
- y is a subject's certainty equivalent for that lottery
- evaluate mean-squared error $(\hat{y} - y)^2$ given prediction $\hat{y}$

Meta-data:

- 44 samples of reported certainty equivalents across different subject pools (from 14 papers)
- on average, 2752 observations per domain

Apply the previous result to compare generalizability of **two economic models** (EU and CPT) and **two black box methods** (random forest and kernel regression)

Comparing the Generalizability of Economic Models and Black Box Algorithms



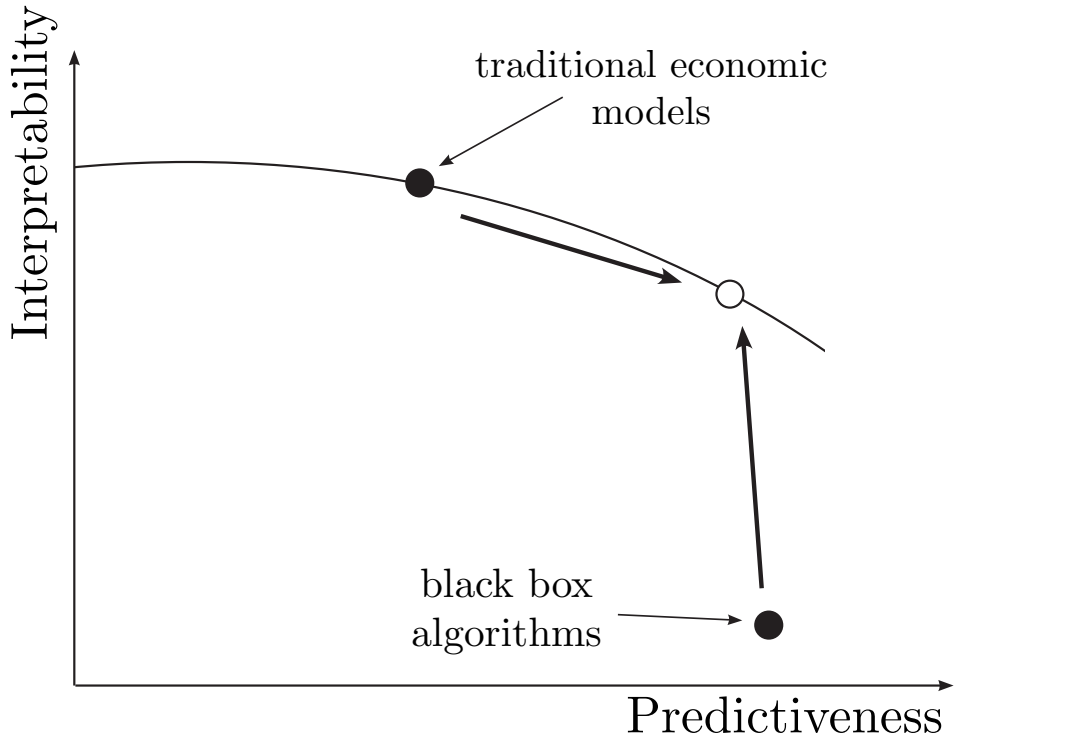
The economic models generalize substantially better!

- even though the black box algorithms perform slightly better on within domain tests

Summary of Methodologies

- black box algorithm can help to identify the degree of **irreducible noise** in the problem
- black box algorithm can help the modeler to **identify new (interpretable) structure** to add back to the model
- black box algorithms can help the modeler to "algorithmically generate" new test cases to **break the model**
- computationally simulate synthetic data to evaluate the **restrictiveness** of economic models
- compare the **transfer performance** of economic models and black box methods

Conclusion


Interpretability
traditional economic models
black box algorithms
Predictiveness