Machine Learning for Economists

Using ML for causal inference and economic theory

Jason DeBacker

October 2021

Outline

- ML vs Econ
  - → Goals
  - → Approaches
  - → Language
- In vs. out of sample fit
- Measuring out of sample fit
- Balancing training vs test accuracy
  - → Subset selection
  - → Model regularization
- Decision trees
- Neural nets
- ML in Economics Research

Roadmap

Economics vs Machine Learning

Supervised Learning

Examples of ML in Economics

ML vs Econ: Goals

Consider the model:

$$y = \beta X + \varepsilon$$

- ECON: want to get an unbiased estimate of β, $\hat{\beta}$
  - $\rightarrow$ Economists ask questions such as "what is the effect of X on y?"
- Traditional ML: want to get an accurate prediction of y, $\hat{y}$ (Mullainathan and Spiess (2017))
  - $\rightarrow$ Traditional users of ML ask questions such as "what will y be tomorrow?"
  - $\rightarrow$ Or "what will y be in some alternative scenario (where I don't care about why y has changed)"

ML vs Econ: Approaches

- ML is about *algorithms*
- Econometrics is built on *theory* about statistical processes
  - → e.g., What is the BLUE?
- Econometricians tend to avoid a methodology if there are not proofs about its asymptotic properties (e.g., normality, consistency)
- ML practitioners tend to focus on what's practical, what works well in many cases

ML vs Econ: Language

- ML leverages many concepts from econometrics that will be familiar to you
- But ML uses often uses different jargon to explain these concepts
- So before we set out, it'll be useful to layout this new jargon we will use – and to compare it to how economists talk where possible

ML vs Econ: Language

Statistics/Econometrics	**Machine Learning**
Data Point	Instance
Covariate	Feature
Parameters	Weights
Estimation/Fitting	Learning
Regression/Classification	Supervised learning
Clustering/Density Estimation	Unsupervised learning
Response	Label
Test set performance	Generalization
"Nonlinear model" $\Rightarrow$ nonlinear in parameters	"Nonlinear model" $\Rightarrow$ nonlinear covariates

Source: "Towards Data Science"

Roadmap

Economics vs Machine Learning

Supervised Learning

Examples of ML in Economics

Our focus: Supervised Learning

- Throughout this talk, we'll focus on "supervised learning"
  - → i.e., cases where we observe the outcome, y
- Often, I'll use continuous outcome variables (i.e., "regression" problems) – but the same (or similar) methods are used for discrete outcome variables ("classification" problems)
- There are cases where economics uses "unsupervised learning" (and we may talk about those)
- But many ideas generalize and we will want some focus as we learn

Overfitting Models

- The main tension in ML is about the fit of the model “in sample” (i.e., on the “training data”) vs “out of sample” (i.e., on “test data”)
- Economists typically not so worried about this, because we just want to measure the treatment effect in the context of our data.
- This tension is driven by the goal of prediction, which often means putting covariates in models without theoretical (from economic theory) point of view
- And the tension isn’t obvious.
- At first blush, it’s hard to see why “overfitting” is a problem.
  - → If I can predict accurately in my sample, why not in others?

Measuring model fit

Most often used measure (with continuous y): **Mean Squared Error**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

Where $\{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$ are the data over which $\hat{f}(x)$ is estimated

- This is called the *training* data

Improving model fit

- Q: How can we improve model fit?
- A: Make the model more flexible! i.e., add more covariates
- In fact, if the number of covariates in $f(x)$, p, equals the number of observations, n, we can fit the data exactly! $MSE = 0$
  - $\rightarrow$ This why you use an adjusted R^2, AIC, or similar criteria to compare models with different numbers of covariates

Measuring model accuracy

- Accuracy: How well can the model fit data *not used to estimate the model parameters*?
- We can measure accuracy using a metric like MSE, but on new data, $\{(x_0, y_0)\}$
  - → This is called the *test* data
- Thus, the test accuracy can be measured as:

$$MSE_{test} = avg(y_0 - \hat{f}(x_0))^2$$

where (y_0, x_0) are some previously unseen data, not used to fit the model

The fit vs accuracy tradeoff

- Let $E\left(y_0 - \hat{f}(x_0)\right)^2$ be the *expected test MSE*
  - $\rightarrow$ The expected test MSE comes from estimating $\hat{f}(x)$ over a large number of training data sets
  - $\rightarrow$ Then, for each estimate of $\hat{f}(x)$ we can find the test MSE by using $\hat{f}(x)$ with the test data
- It can be shown that the expected test MSE can be decomposed as follows:

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \underbrace{\text{Var}(\hat{f}(x_0))}_{\text{variance}} + \underbrace{\left[\text{Bias}(\hat{f}(x_0))\right]^2}_{\text{bias}} + \underbrace{\text{Var}(\varepsilon)}_{\text{irreducible error}}$$

The fit vs accuracy tradeoff

- Variance = how much does $\hat{f}(x)$ change if estimate it on a different training data set?
- Bias = error introduced by model not matching the true data generating process (i.e., model misspecification)
- More flexible methods lower bias, but increase variance

Overfitting: a simple example

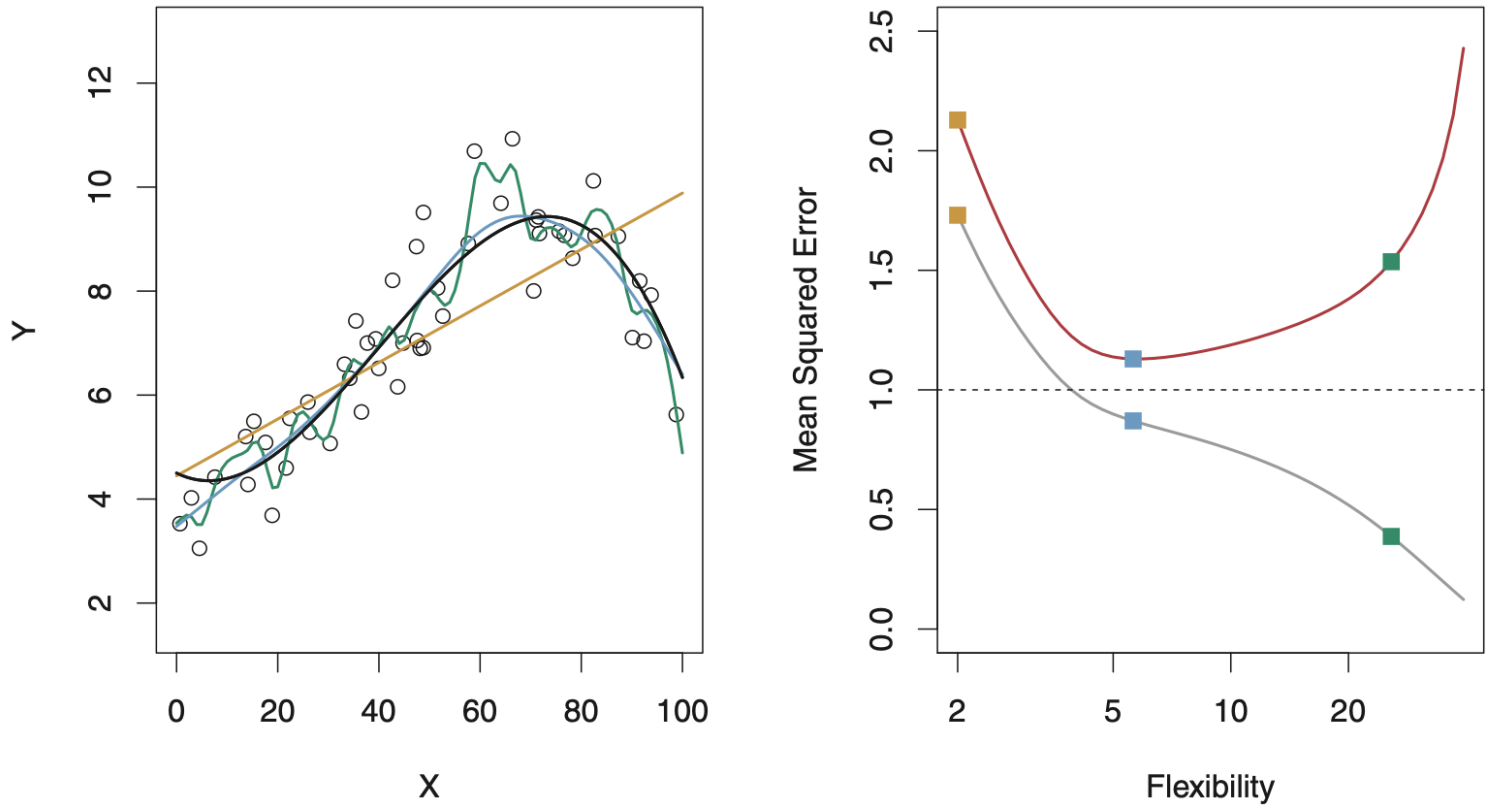


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

Measuring test accuracy: resampling

- To test a model, we need to reserve some data for testing.
  - → Data set aside are called *validation data*
- This limits the data we can use to estimate the model, reducing the precision of our estimates/limiting the number of covariates we can Use
- A solution to having to set aside data is to use *resampling methods*

Resampling methods

Some main resampling methods:

1. Leave one out cross-validation (LOOCV)
2. k-fold cross-validation
3. Bootstrapping

Leave One Out Cross-Validation

Algorithm:

- Drop one observation, estimate model on the remaining $n-1$ observations
- Then, compute the squared error from the prediction of the left out observation (e.g., if observation n is left out, $SE_n = (y_n - \hat{f}_n(x_n))^2$)
- Repeat this n times, dropping a different observation each time
- The test MSE is the average of the MSE from each of the n squared errors:

$$MSE_{test} = \frac{1}{n}\sum_{i=1}^{n} SE_n$$

k-Fold Cross-Validation

Algorithm:

- Randomly divide training data into k groups, set one of the k groups aside as a validation set
- Estimate the model on the $k-1$ remaining groups
- Find the MSE from predictions on the left out group:

$$MSE_k = \frac{1}{n/k} \sum_{j=1}^{n/k} (y_j - \hat{f}_k(x_j))^2$$

- Repeat this k times, leaving out a different group each time
- The test MSE this then:

$$MSE_{test} = \frac{1}{k} \sum_{i=1}^{k} MSE_i$$

Summary of validation methods

Method	Bias	Computational Cost
Validation data	High	Low
LOOCV	Low	High[1]
k-Fold CV	Medium	Medium

[1] May be avoided in model can be fit via least squares

Bootstrapping

Algorithm:

- Resample from training data, drawing n observations *with replacement*

Uses:

- This is not typically used for validation, but rather to find the variance in the estimates of $\hat{f}(x)$.
- e.g., the std error of parameters $\hat{\beta}$ can be estimated through B bootstrapped samples with:

$$SE_B(\hat{\beta}) = \sqrt{\frac{1}{B}\sum_{r=1}^{B}\left(\hat{\beta}^{*r} - \frac{1}{B}\sum_{r'=1}^{B}\hat{\beta}^{*r'}\right)^2}$$

Recap

- You all know how to estimate a model on training data
- We now understand how to measure test error via the test MSE (via validation data or resampling CV methods)
- Next, we'll talk about how to find the model that minimizes the test MSE

Model Selection and Regularization

To find the model that minimizes the test MSE, we have a few general approaches:

1. Subset selection: estimating a bunch of models and choosing the "best"
2. Model regularization/shrinkage: creating a penalty function to affects parameter estimates (i.e., don't just use least squares)
3. Dimension reduction methods

Subset Selection

A few of the most common subset selection methods are the following:

1. Best subset selection
2. Forward stepwise selection
3. Backward stepwise selection

Best subset selection

The algorithm for *best subset selection* is:

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. The model simply predicts the sample mean for each observation.
2. For $k = 1, 2, ..., p$:
   - 2.1 Fit all $\binom{p}{k}$ models that contain exactly p predictors
   - 2.2 Pick the best among these $\binom{p}{k}$ models and call it $\mathcal{M}_k$. Define best as the model with the lowest RSS, or equivalently the largest R^2.
3. Select the single best model from among $\mathcal{M}_0, ..., \mathcal{M}_k$ using cross-validation prediction error, C_p, AIC, BIC, or adjusted R^2.

Forward stepwise selection

The algorithm for *forward stepwise selection* is:

1. Let $\mathcal{M}_0$ denote the *null model*, which contains no predictors. The model simply predicts the sample mean for each observation.
2. For $k = 1, 2, ..., p-1$:
   2.1 Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one more predictor.
   2.2 Choose the best among these $p-k$ models and call it $\mathcal{M}_{k+1}$. Define best as the model with the lowest RSS, or equivalently the largest R^2.
3. Select the single best model from among $\mathcal{M}_0, ..., \mathcal{M}_p$ using cross-validation prediction error, C_p, AIC, BIC, or adjusted R^2.

Backward stepwise selection

The algorithm for *backward stepwise selection* is:

1. Let $\mathcal{M}_p$ denote the *full model*, which contains all p predictors.
2. For $k = p, p-1, ..., 1$:
   - 2.1 Consider all k models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
   - 2.2 Choose the best among these k models and call it $\mathcal{M}_{k-1}$. Define best as the model with the lowest RSS, or equivalently the largest R^2.
3. Select the single best model from among $\mathcal{M}_0, ..., \mathcal{M}_p$ using cross-validation prediction error, C_p, AIC, BIC, or adjusted R^2.

Summary of tradeoffs among selection methods

Selection method	Number of models to estimate	Likelihood of finding the "best" model?[1]
Best subset	2^p	100%
Forward stepwise	$1 + p(p+1)/2$	Not 100%
Backward stepwise	$1 + p(p+1)/2$	Not 100%

[1] Among linear models

Model Regularization/Shrinkage

- The subset selection methods above essentially use brute force to find the best performing model
- This may be computationally intensive
- As an alternative, we can estimate a model with all p covariates and add a "penalty" that will help "shrink" the parameters back towards zero
  - $\rightarrow$ Shrinking the parameters towards zero has the effect of reducing the variance in the fitted model, $\hat{f}(x)$

Shrinkage Methods

The two most common methods are:

1. Ridge Regression:

$$\hat{\beta}^R = argmin \underbrace{\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta x_{i,j}) \right)^2}_{\text{RSS}} + \lambda \sum_{j=1}^{p} \beta_j^2$$

2. Lasso Regression:

$$\hat{\beta}^L = argmin \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta x_{i,j}) \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

Shrinkage Methods

- λ is a "tuning" parameter (set higher if want to shrink parameters more)
  - $\rightarrow$ Can do a grid search over λ values to find which gives best test MSE (via cross-validation)
- Ridge Regression
  - $\rightarrow$ All $\hat{\beta}_j^R$ will be non-zero
- Lasso Regression
  - $\rightarrow$ Some $\hat{\beta}_j^L$ will be zero
- Can generalize penalty functions:

$$\hat{\beta}^Q = argmin \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} \beta x_{i,j}) \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^q$$

Dimension Reduction Methods

Dimension reduction methods transform our original predictors in $\mathbf{X}$ into new predictors, $\mathbf{Z}$.

Most commonly, these transformations are linear:

$$Z_m = \sum_{j=1}^{p} \phi_{j,m} X_j$$

The model is then fit on these transformed variables:

$$y_i = \theta_0 + \sum_{m=1}^{M} \theta_m z_{i,m} + \varepsilon_i, \;\; i = 1, ..., n$$

Dimension Reduction Methods

The transformation of $\mathbf{X}$ to $\mathbf{Z}$, reduces the dimension of the problem from $p+1$ to $M+1$ M is a tuning parameter that is chosen.

Main dimension reduction methods:

1. Principal component analysis: find linear combinations of covariates that explain the most variation in $\mathbf{X}$, then most of what's left (i.e., the residuals), and the most of what's left after that...M times
   $\rightarrow$ This is *unsupervised* since it only uses $\mathbf{X}$ and not y
2. Partial least squares: find linear combinations of covariates that explain the most variation in y, then most of what's left (i.e., the residuals from a regression on y), and the most of what's left after that... M times.
   $\rightarrow$ This is *supervised* since it uses $\mathbf{X}$ and y

Principal componenet analysis

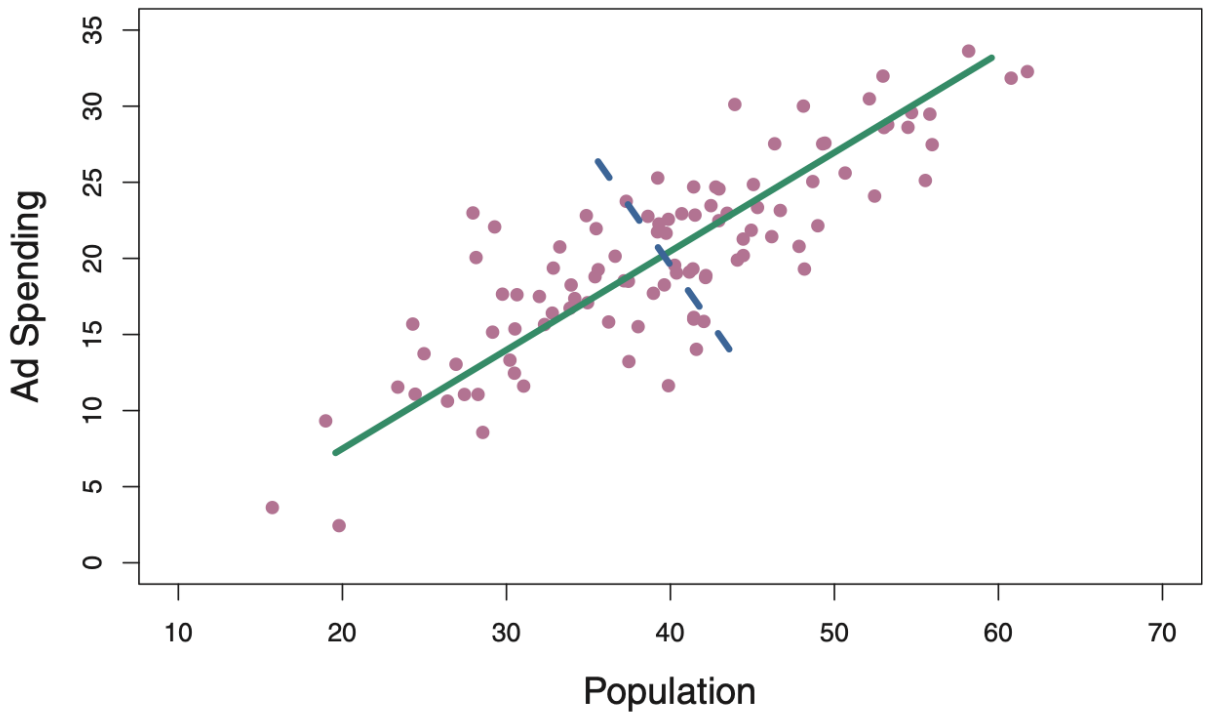


FIGURE 6.14. *The population size (*pop*) and ad spending (*ad*) for* 100 *different cities are shown as purple circles. The green solid line indicates the first principal component, and the blue dashed line indicates the second principal component.*

Limitations of Model Selection

- Interpretability
  - $\rightarrow$ The algorithm put X_j in the regression – what does economic theory say about this?
  - $\rightarrow$ What is the economic interpretation of Z_m?
- High Dimensions
  - $\rightarrow$ If $p > n$ (or close to it), then issues with increased noise, multicollinearity, and interpretability

“Nonlinear” models

Thus far, we have just considered models with linear covariates, but we can extend $f(x)$ to include non-linear functions of covariates (but still linear in parameters).

Some common approaches:

1. Polynomials
2. Step functions
3. Regression splines
4. Smoothing splines
5. Local regression
6. Generalized additive models

Polynomial Regression

Include polynomials of the covariates:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + ... + \beta_d x_i^d + \varepsilon_i$$

Polynomial Functions

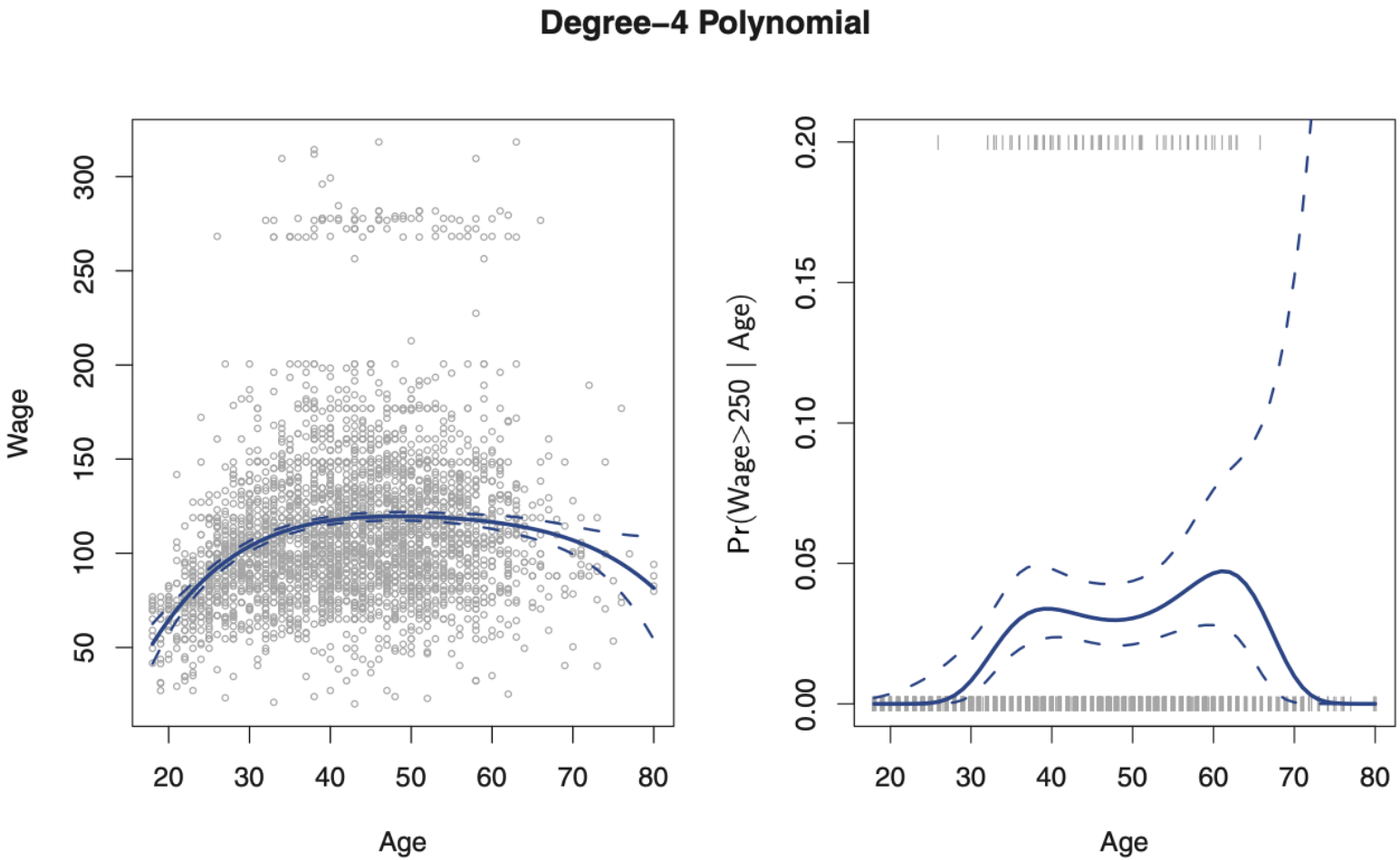


FIGURE 7.1. *The* `Wage` *data.* Left: *The solid blue curve is a degree-4 polynomial of* `wage` *(in thousands of dollars) as a function of* `age`*, fit by least squares. The dotted curves indicate an estimated 95 % confidence interval.* Right: *We model the binary event* `wage>250` *using logistic regression, again with a degree-4 polynomial. The fitted posterior probability of* `wage` *exceeding* $250,000 *is shown in blue, along*

Step functions

Divide X into K bins and fit a different constant to each bin.

Define bins with indicator functions for x_i being in a certain range.

e.g., $C_k(X) = I(c_{k-1} \leq X < c_k)$

The Model:

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \beta_3 C_3(x_i) + ... + \beta_K C_K(x_i) + \varepsilon_i$$

Will exclude $C_0(X) = I(x < c_1)$ – so all coeffs are change in outcome relative to moving from 0th bin to kth bin.

Step Functions

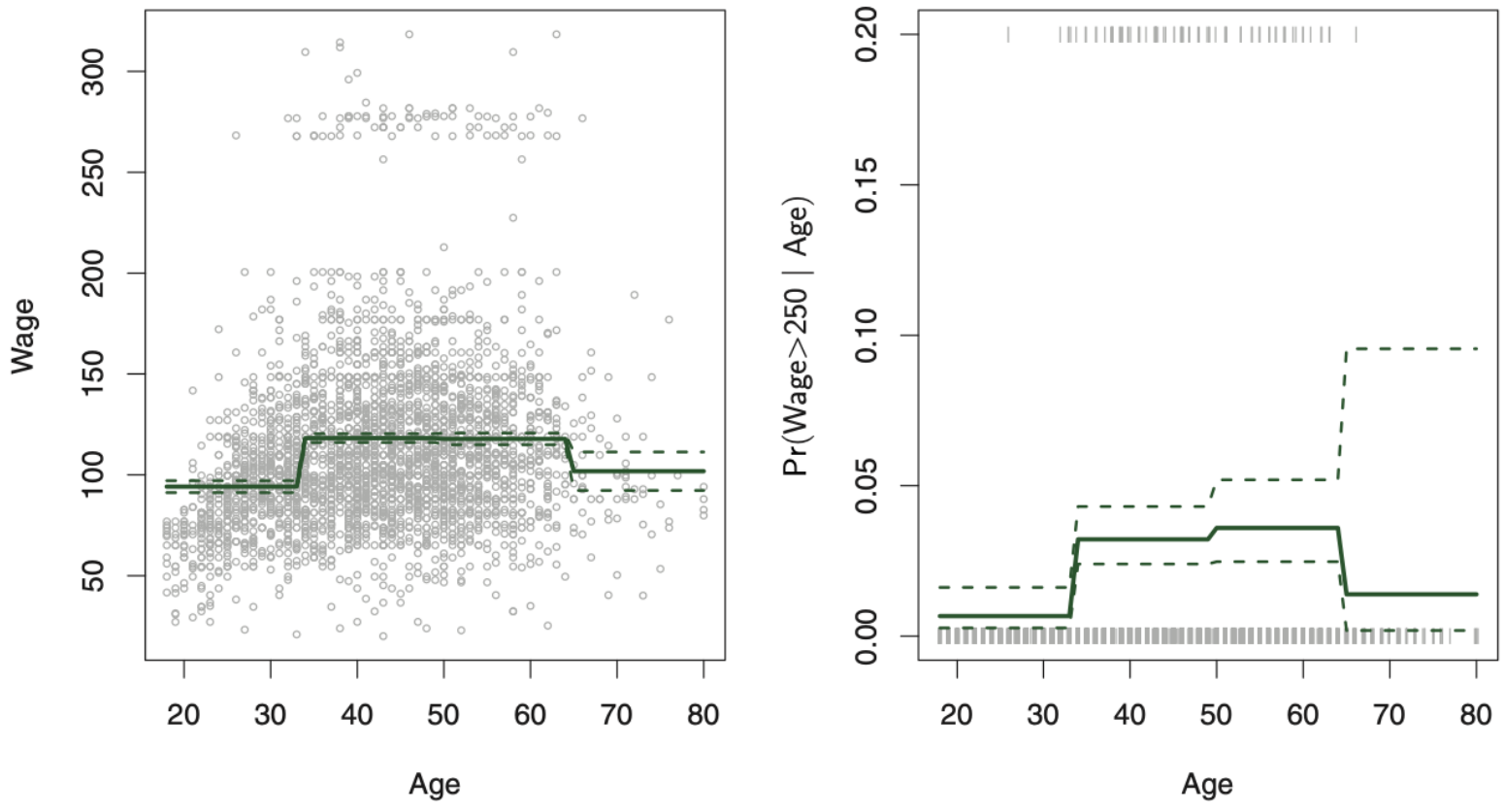


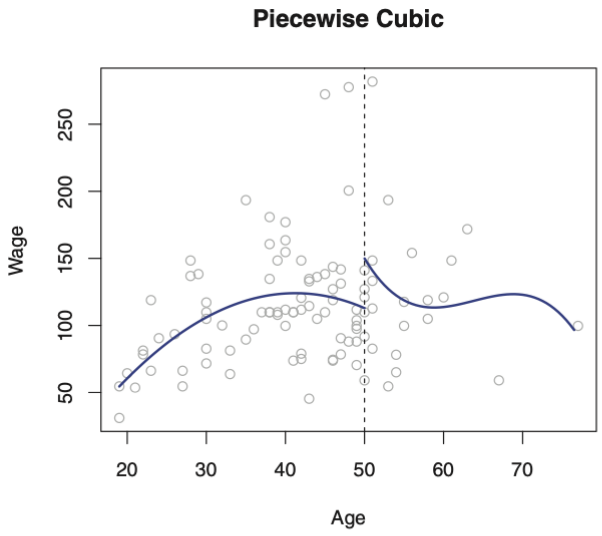
FIGURE 7.2. *The* `Wage` *data.* Left: *The solid curve displays the fitted value from a least squares regression of* `wage` *(in thousands of dollars) using step functions of* `age`*. The dotted curves indicate an estimated 95 % confidence interval.* Right: *We model the binary event* `wage>250` *using logistic regression, again using step functions of* `age`*. The fitted posterior probability of* `wage` *exceeding* $250,000 *is shown, along with an estimated 95 % confidence interval.*

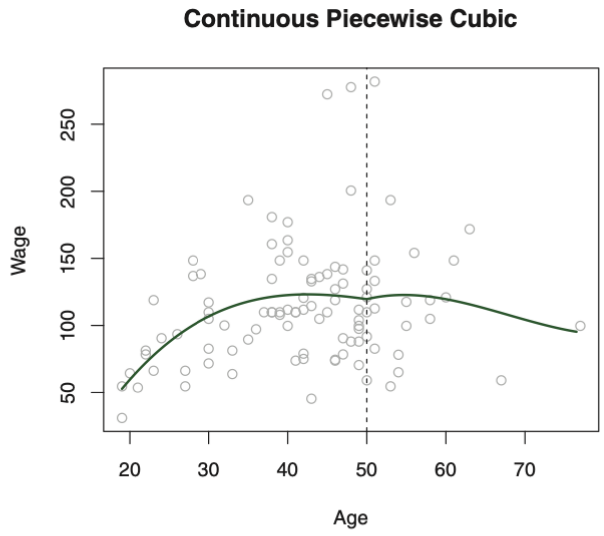
Spline Methods

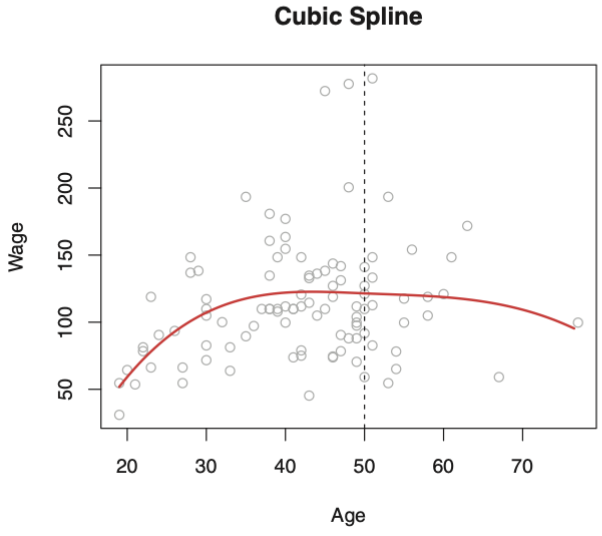
A number of approaches:

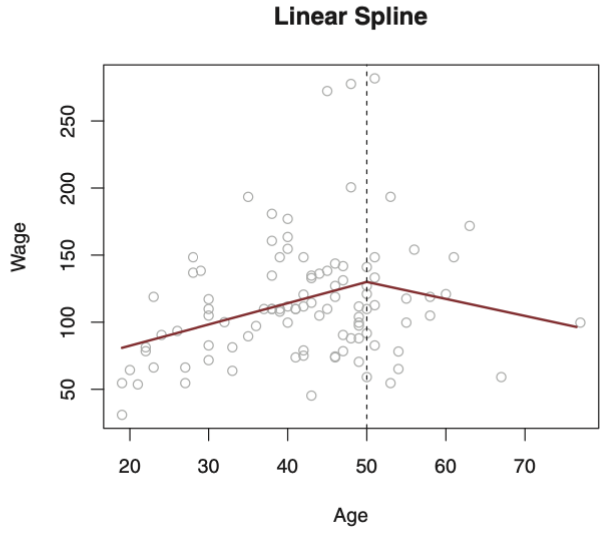
- Regression splines: fit different polynomials to different ranges of data
  - → Usually constraints that smooth through the *knots*
- Smoothing splines: allow knots at all data points, but enforce smoothness through a smoothing parameter
- Local regression: fit a linear model around each data point using WLS (where weights higher for closer data points)

Splines


Piecewise Cubic
Wage
Age
50
100
150
200
250
20
30
40
50
60
70


Continuous Piecewise Cubic
Wage
Age
50
100
150
200
250
20
30
40
50
60
70


Cubic Spline
Wage
Age
50
100
150
200
250
20
30
40
50
60
70


Linear Spline
Wage
Age
50
100
150
200
250
20
30
40
50
60
70

Generalized Additive Models

These allow for *different* transformations of each X_j, $f_j(\cdot)$:

$$y_i = \beta_0 + \sum_{j=1}^{p} f_j(x_{i,j}) + \varepsilon_i$$

But still maintain a model that is linear in parameters (since each covariate-transformation is additive).

Basis Functions

Polynomials, step functions, regression splines, are all examples of *basis functions*:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + ...\beta_k b_k(x_i) + \varepsilon_i$$

where $b(X)$ can be any transformation of X

Time to check-in

Where are we?

- Want to predict y, but battle between fitting and overfitting a model
- Cross-validation important for model selection
- We first saw ways to select a linear model
- Now we've seen the many ways we might transform the covariates to provide a better fit
- We still need to do cross-validation to select best!
- It's just that now we've increased the space of possible models to search over - to include a wide class of non-linear transformations of the covariates!
  - $\rightarrow$ Even with huge computational resources, we can't try out every model.
  - $\rightarrow$ Typically we'll limit to one class of models (e.g., find the best regression spline model)

Multivariate Adaptive Regression Spline

Multivariate Adaptive Regression Spline (MARS)

- MARS is an algorithm to find the optimal (from the point of view of test error rates) basis function + feature set to model the outcome variable
- MARS is patented, so in open source software, this algorithm will be called “Earth”
- Algorithm (general):
  1. Forward pass: add basis functions to model that improve fit (tends to overfit)
  2. Backward pass: use generalized CV to remove terms from the model (comparing test error + penalty for more terms in model)
  3. Can include constraints in (1) and (2) (e.g., max number of terms allowed)

Regression Trees

- As we'll see, tree-based methods are commonly used in economics applications of machine learning
- Tree-based methods can be used for classification and regression problems – this discussion will focus on their use in regression problems.

Regression Tree, Basics

- Regression trees split the data into groups based on covariates
  - → These splits happen at what are called *internal nodes*
- The $\hat{y}$ is then the average value of the outcome variable within each final grouping
  - → These final groupings are called *termianl nodes* or *leaves* of the tree

Building a Regression Tree

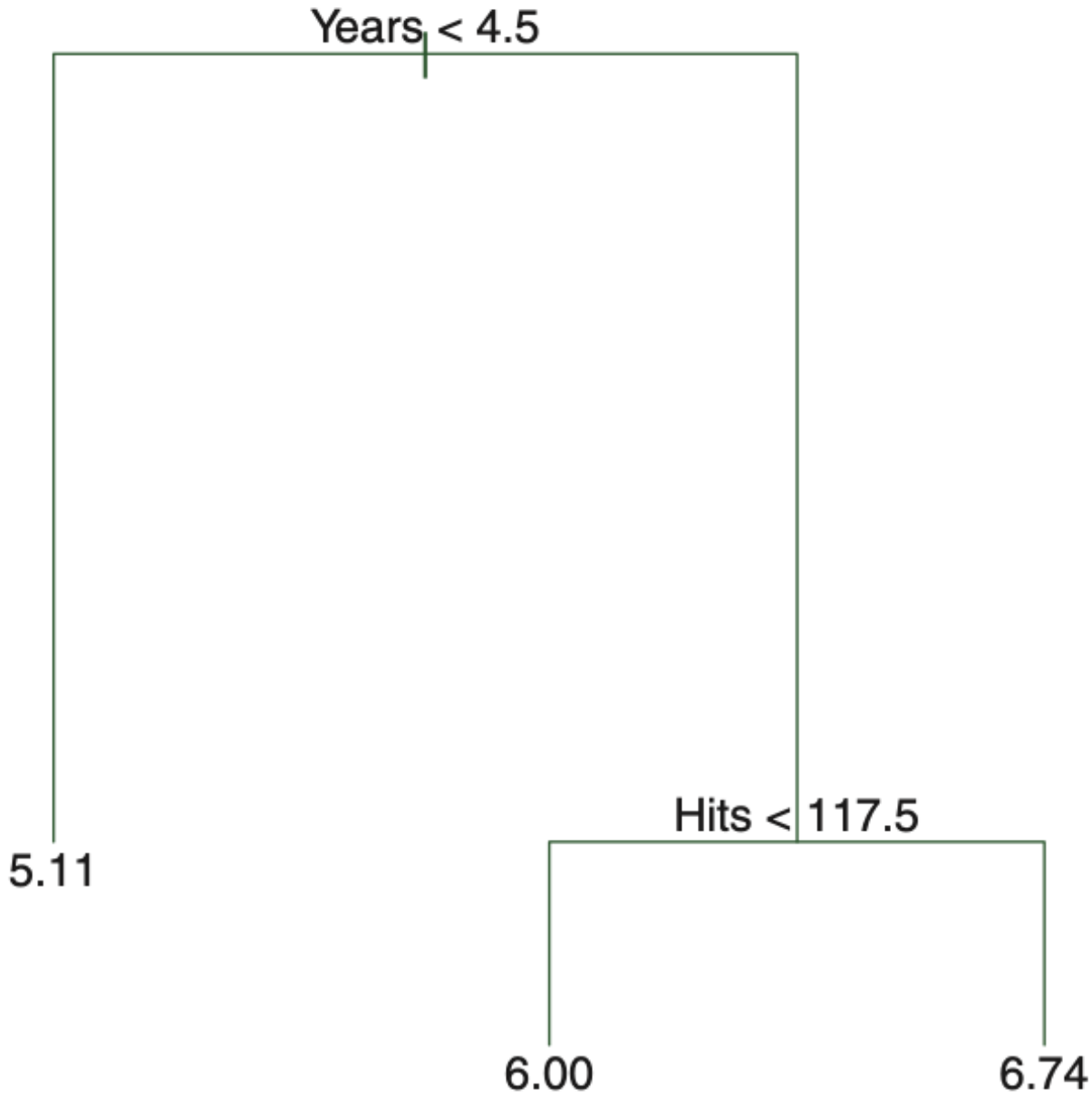
1. Divide the space of covariates $(X_1, X_2, ..., X_p)$ into J distinct and non-overlapping regions, $R_1, R_2, ..., R_J$.
2. For every observation that falls into the region R_j, we make the same prediction, which is the mean of the outcome variable for the observations in R_j
3. The objective is to choose the R_j such that we minimize:

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2$$

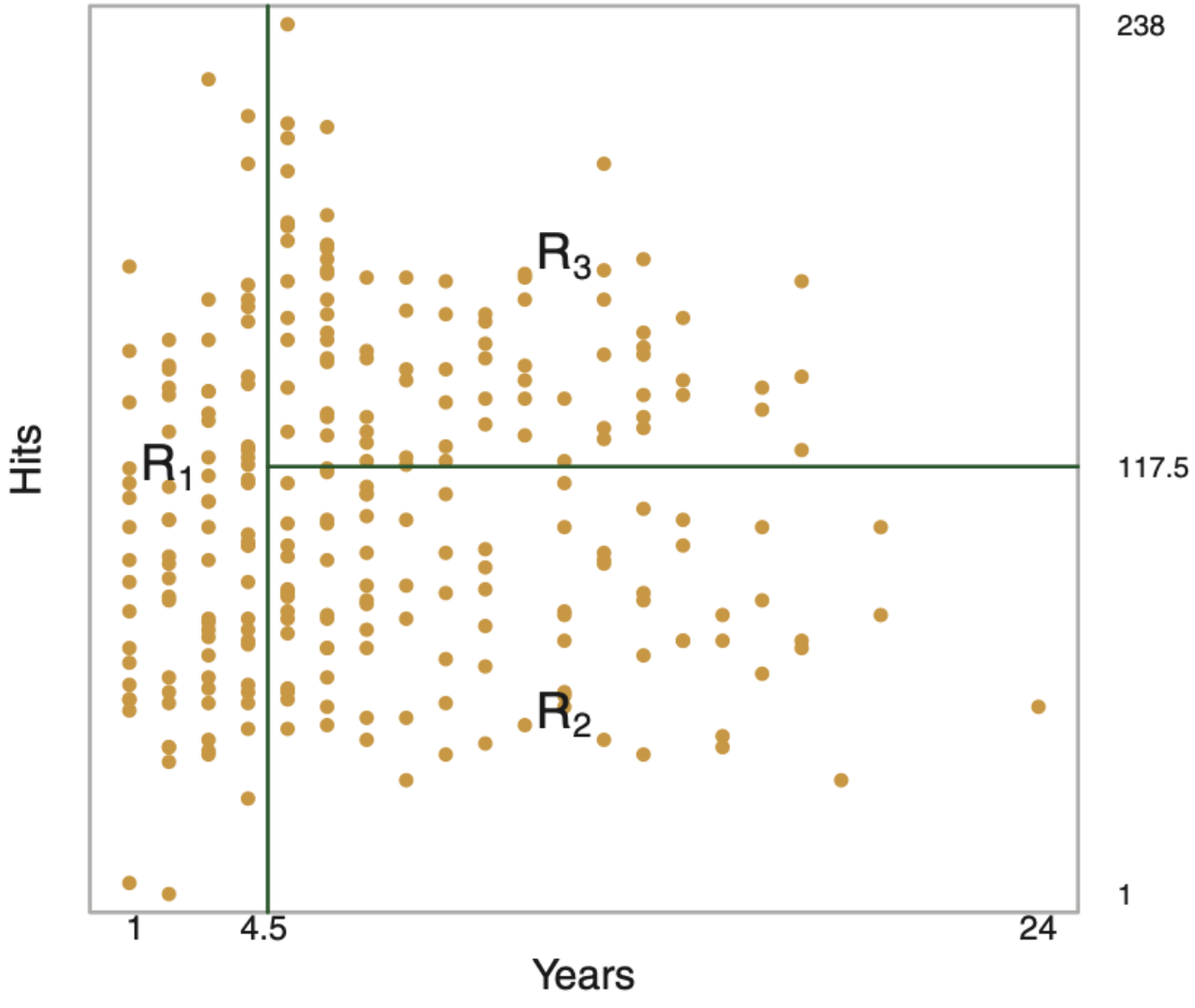
How to split the tree?

- There are way too many potential ways to split the data to look at all possible divisions
- Thus, method often used is *recursive binary splitting*
  1. Start at the top of the tree (i.e., with all data)
  2. Split into two groups, choosing them by minimizing the the RSS *at that step*
  3. Then repeat at these two nodes and repeat...

A simple regression tree



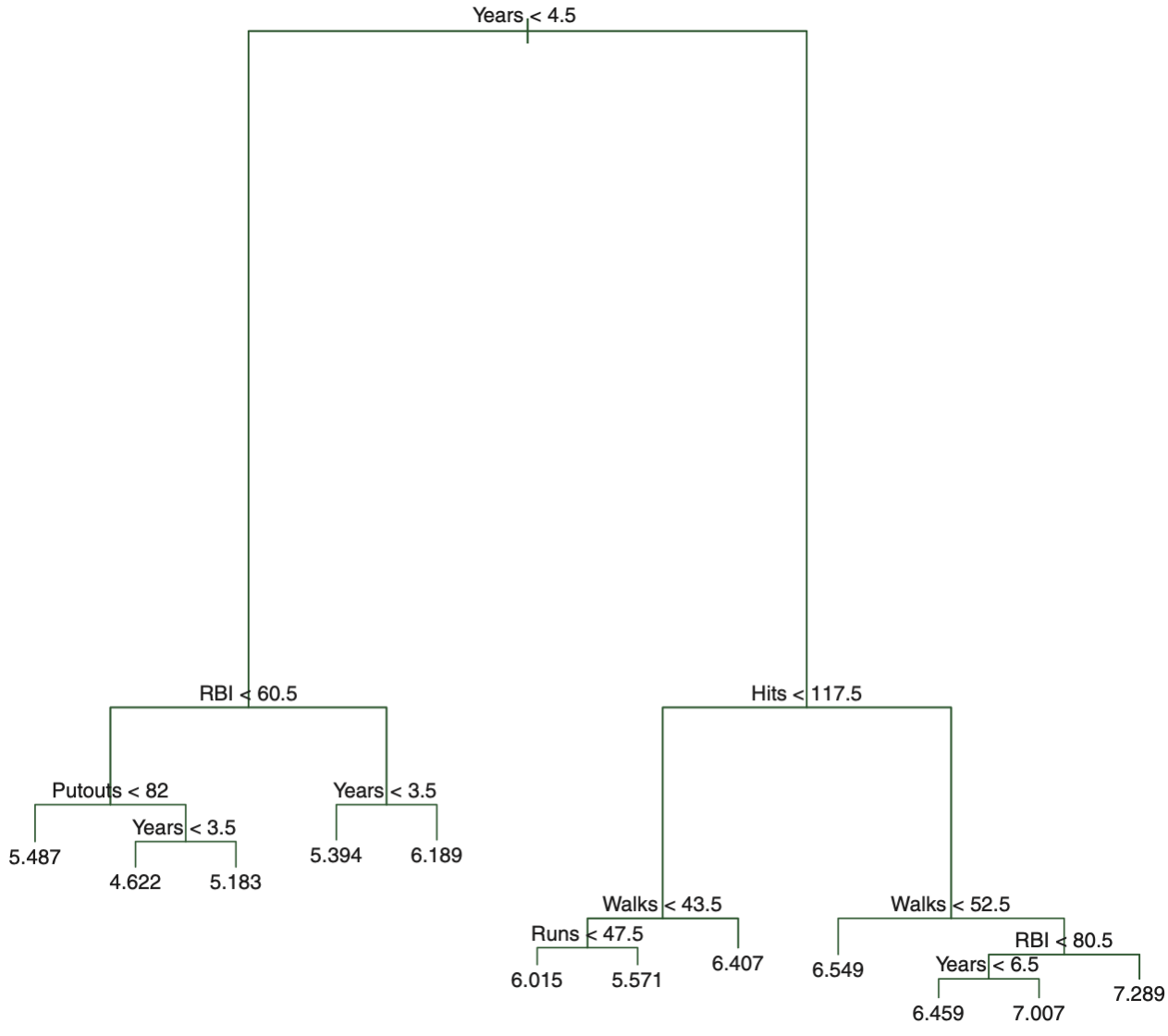
Visualizing a tree in 2D



Fitting Regression Trees

- As with a linear regression model, we can increase the complexity of the tree to improve the model fit
- In fact, you can fit the data perfect, splitting the tree until each leaf has just one observation!
- But just like with linear regression models, there's the risk of overfitting – the tradeoff between model fit and testing accuracy.

A bigger tree



Tree size and model fit

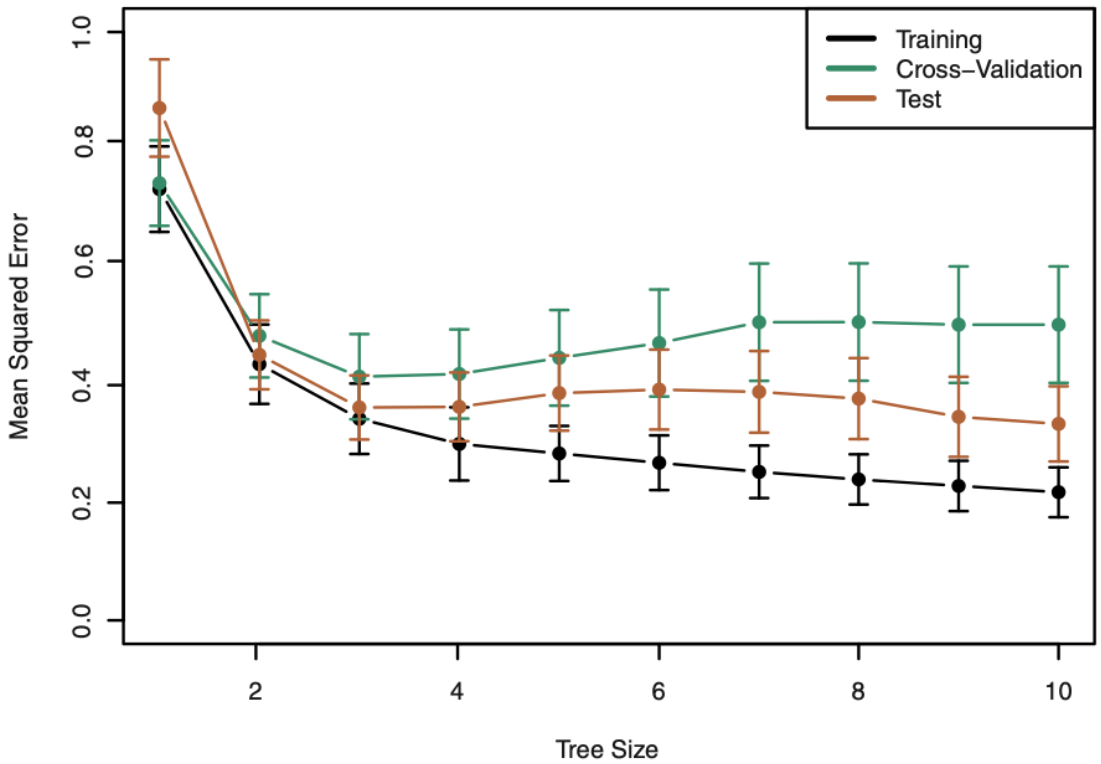


FIGURE 8.5. *Regression tree analysis for the* `Hitters` *data. The training, cross-validation, and test MSE are shown as a function of the number of terminal nodes in the pruned tree. Standard error bands are displayed. The minimum cross-validation error occurs at a tree size of three.*

Tree Pruning

- To avoid overfitting, we can *prune* the tree, but cutting off some of the leaves
- Question: how to do this systematically?
- (One) Answer: *cost complexity pruning* (aka, weakest link pruning)

Cost Complexity Pruning

Cost complexity pruning starts by setting up an objective function that incorporates a constraint on the number of terminal nodes:

$$\sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

where α is a tuning parameter that will determine the number of terminal nodes and R_m is the predictor space at the mth terminal node

Cost Complexity Pruning Algorithm

1. Use recursive binary splitting to grow a large tree on the training data, stopping only when each terminal node has fewer than some minimum number of observations.
2. Apply cost complexity pruning to the large tree in order to obtain a sequence of best subtrees as a function of α.
3. Use K-fold cross-validation to choose α. That is, divide the training observations into K folds. For each $k = 1, ..., K$:
   - 3.1 Repeat steps 1 and 2 on all but the kth fold of the training data.
   - 3.2 Evaluate the mean squared prediction error on the data in the left-out kth fold, as a function of α.
4. Average the results (across the K folds) for each value of α, and pick the α to minimize the average error.
5. Return the subtree from Step 2 that corresponds to the chosen value of α.

Methods to Make Trees More Robust

1. Bagging
2. Random forests
3. Boosting

Bagging

1. Bagging is akin to bootstrapping: draw random samples from the training data and construct a regression tree for each of these samples.
2. Average the predicted values for each observation over these different estimates.
3. i.e., if $f^{*b}(x)$ is the estimated function for the predicted values from the bth sample, then the bagged estimator is given by:

$$f_{bag} = \frac{1}{B}\sum_{b=1}^{B} f^{*b}(x)$$

4. Problem of this approach: lose interpretability – averaging over different trees so terminal notes have different groupings – so don't know what variables predict outcomes.

Random Forests

- Another issue with bagging is that the trees estimated for each bootstrapped sample will often be quite similar to one another
  - $\rightarrow$ This is because if one predictor is very strong in one sample, it will likely be strong in another
- *Random forests* solve this by tweaking the construction of trees
- Random forests only allow a random subset, m, of the p predictors to be available for splits at each node
- This randomization means that trees are less correlated across samples and so the average across samples will likely have lower test error

Boosting

- *Boosting* involves repeatedly fitting small trees to the data
- At each step of the repeated process, the residuals of the fitted values are calculated and the next tree is fit to those residuals
- There are 3 tuning parameters in this process:
  1. B, the number of trees (too many will lead to overfitting)
  2. λ, parameter saying how much tree is updated at each step
  3. d, the number of splits in each tree

Boosting Algorithm

1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training data
2. For $b = 1, 2, ..., B$, repeat:
   - 2.1 Fit a tree, $\hat{f}^b$ with d splits (d+1 terminal nodes) to the training data (X, r).
   - 2.2 Update $\hat{f}$ by adding a shrunken version of the new tree:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

   - 2.3 Update the residuals: $r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$
3. Output the boosted model,

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

Neural Networks

- Neural networks are a common ML tool for what is called “deep learning”
- Neural networks derive their name from an early use of these algorithms: trying to model the human brain
- You can think of deep learning as the ultimate in data mining: allow for many, many combinations of the covariates in the prediction of the dependent variable
  - → Can be used in regression of classification problems
  - → Can have many “layers” (levels of transformation of features in the data)
  - → “Deep learning” refers to neural networks with more than one layer

Neural Network Structure

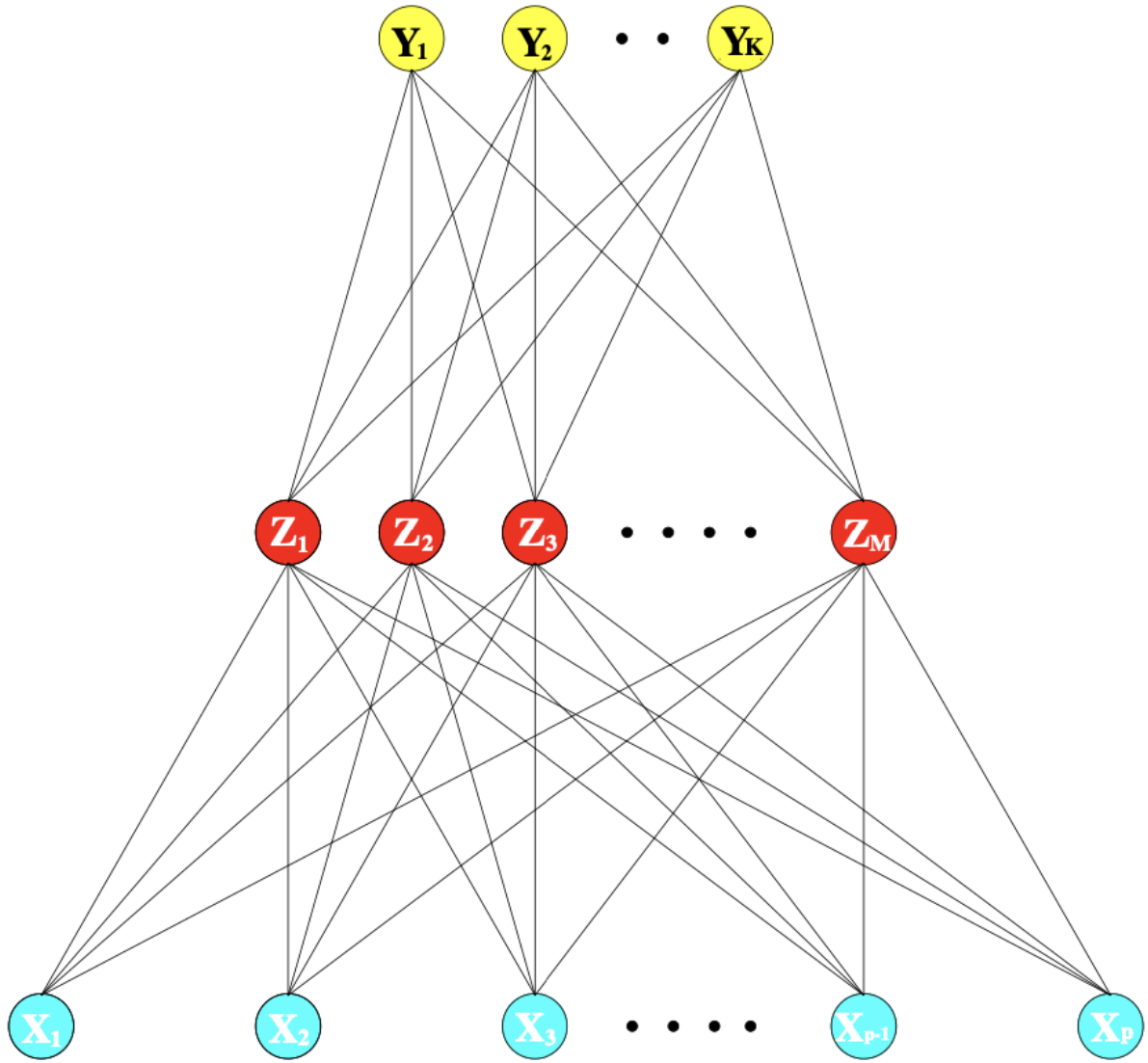


FIGURE 11.2. *Schematic of a single hidden layer, feed-forward neural network.*

Neural Network Structure

- "Hidden Layer(s)"

$$Z_m = \sigma\left(\alpha_{0m} + \alpha_m^T X\right), m = 1, ..., M$$

- Target:

$$T_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$$

- Outcome:

$$f_k(X) = g_k(T), k = 1, ..., K$$

Neural Network Structure

- With a regression problem, typically $g(\cdot)$ is the identity matrix, so we have:

$$y_k = \beta_{0k} + \beta_k^T Z, k = 1, ..., K$$

- σ is an "activation" function
  - $\rightarrow$ Usually chosen to be the sigmoid $\sigma(v) = 1/(1 + e^{-v})$
  - $\rightarrow$ If σ is the identity matrix, then we have case of model linear in inputs
  - $\rightarrow$ Generally, σ generalizes the model to a non-linear one
- Can have multiple hidden layers (i.e., inputs predict on layer, that layer another, and so on until a final transformation is used to predict the output variable)

Training Neural Nets

Training can be hard because prone to over fitting and optimization can be difficult (e.g., because instability in objective function, local minima)

Some issues (and solutions to them):

- Starting values: start small, but not zero
- Overfitting: use *weight decay*, a penalty function like ridge regression to regularized coefficients
- Scale inputs: normalize each so mean zero and std dev 1
- Hidden units: more typically better than less (can let regularization shrink coefficients as necessary to avoid overfitting)
- Multiple minima: can use bagging to average over multiple training datasets to avoid influence of starting values on minima

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲ *= good,* ◆ *=fair, and* ▼ *=poor.*

Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

For more details

The best two books on ML, both with a very applied focus, are:

1. James et al. (2013)
2. Hastie, Tibshirani, and Friedman (2001)

Roadmap

Economics vs Machine Learning

Supervised Learning

Examples of ML in Economics

How economists use ML

Broadly speaking, the most successful application of ML in economics are in the following areas:

1. **Causal inference**: Estimating heterogeneous treatment effects
2. **Data creation**: Creating new data sources (e.g., from combing through text)
   - → Often unsupervised learning approaches
3. **Quantitative Econ theory**: Computing models with high degrees of heterogeneity
4. **Econ theory/computational methods**: Helping to break the curse of dimensionality in dynamic programming problems

Heterogeneous treatment effects: Athey and Imbens (2016)

Athey and Imbens (2016), ”Recursive partitioning for heterogeneous causal effects”

- Idea: Rather than estimate a population wide average treatment effect, estimate the treatment effects for different groups (partitions) in the data
- Q: How define partitions?
- Q: How estimate treatment in each partition?

Heterogeneous treatment effects: Athey and Imbens (2016)

Methodology:

- Want to use ML tree-based methods to partition a sample based on *treatment effects* (not covariates)
- Challenge: if traditional ML methods were used, the results would not provide proper measures of uncertainty in the true parameter values because you will partition the sample by treatment effect, putting all those with similarly measured treatment effects in one group.
- Solution: They separate the partitioning of the sample from the estimation of the treatment effects

Heterogeneous treatment effects: Athey and Imbens (2016)

- Athey and Imbens call this approach *honest* estimation
  - → As opposed to the *adaptive* estimation, which would result if traditional machine learning CART algorithms were used and the same data were used to partition the data and estimate treatment effects
- Costs of the honest approach
  - → Sample size: reserving part of the data for partitioning and part for model estimation means that the models estimated will not fit as well out of sample (i.e., higher MSE)
- Benefits of the honest approach
  - → It avoids spurious extreme values of Y_i being placed into the same leaf as other extreme values by the algorithm
  - → This results in the poor coverage properties of confidence intervals for adaptive estimation methods relative to the honest methods

Heterogeneous treatment effects: Athey and Imbens (2016)

Punchline:

> *A potentially important application of the techniques is to "data mining" in randomized experiments. Our method can be used to explore any previously conducted randomized controlled trial, for example, medical studies or field experiments in development economics. Our methods can discover subpopulations with lower-than-average or higher-than-average treatment effects while producing confidence intervals for these estimates with nominal coverage, despite having searched over many possible subpopulations.*

Extended to random forests in Wager and Athey (2018)

Heterogeneous treatment effects: Cengiz et al. (2021)

Cengiz et al. (2021)

- Question: What are the the effects of minimum wage laws on employment, unemployment, and labor force participation?
- Why this is hard to answer: endogeneity of min wage laws, *heterogeneous treatment effect*
  - → Hedge fund managers – and teachers too – unlikely affected by the minimum wage.

Heterogeneous treatment effects: Cengiz et al. (2021)

- Solution to issues of heterogeneous effects: Use machine learning to classify workers are likely impacted by the minimum wage
  - → Assume that if likely to make below 125% of the minimum wage, that worker is likely impacted
- They use ML, in particular, treat based methods, to classify workers in this way

Heterogeneous treatment effects: Cengiz et al. (2021)

- Solution to issues of heterogeneous effects: Use machine learning to classify workers are likely impacted by the minimum wage
  - → Assume that if likely to make below 125% of the minimum wage, that worker is likely impacted
- They use ML, in particular, treat based methods, to classify workers in this way

Figure 1: Minimum Wage Workers According to Pruned Trees

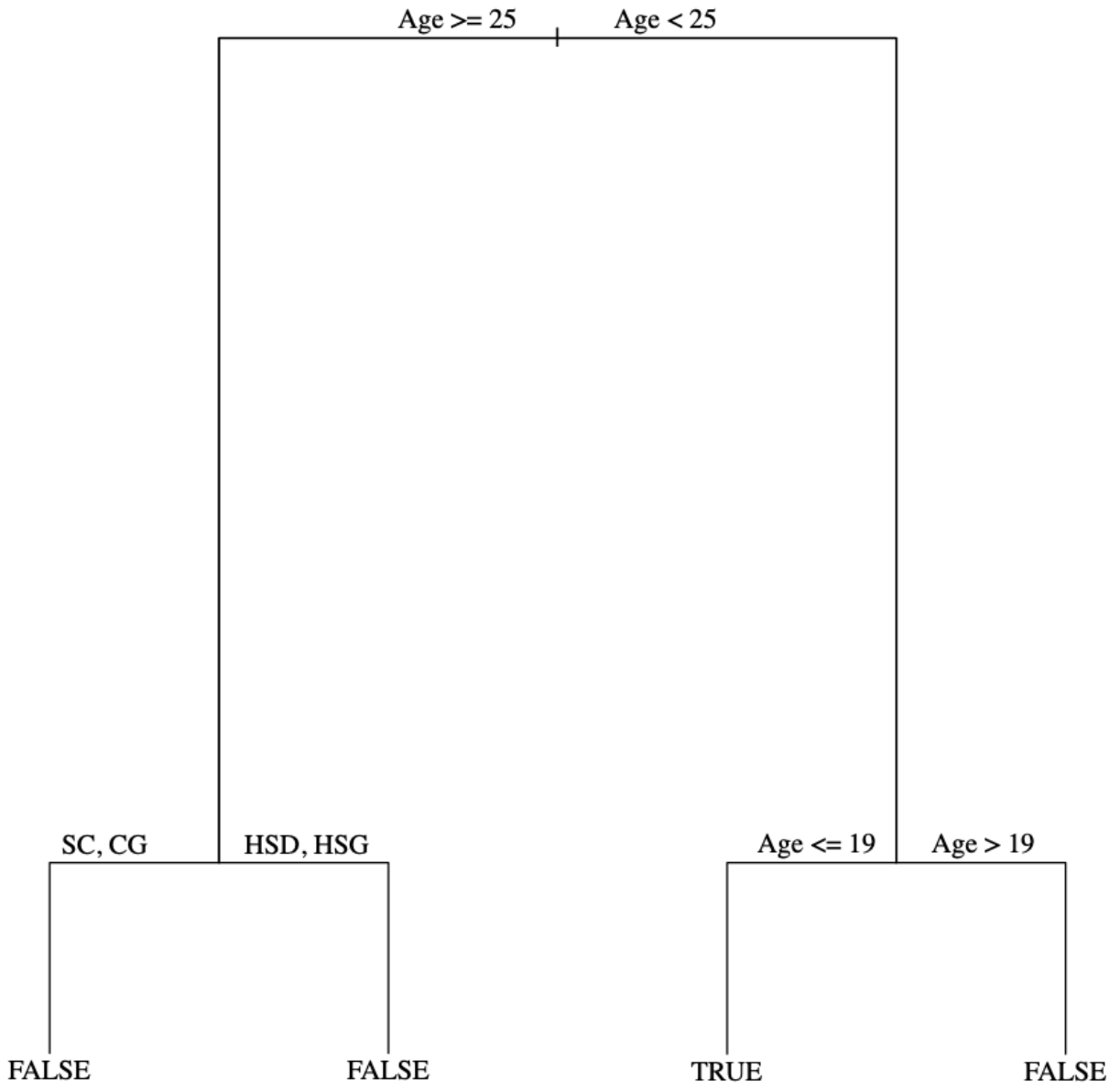


Figure 4: Relative Influences of the Predictors in the Boosted Tree Prediction Model

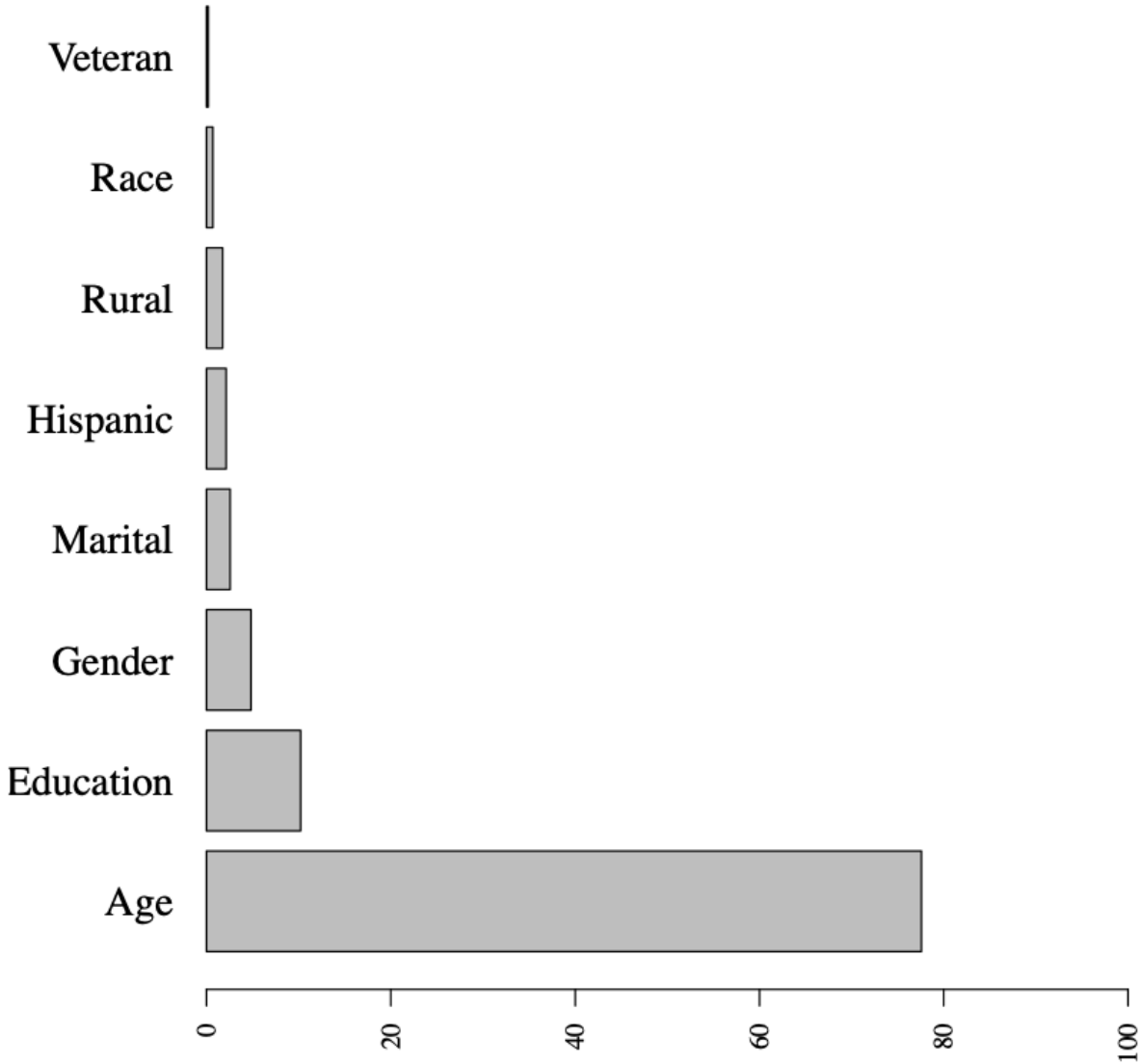
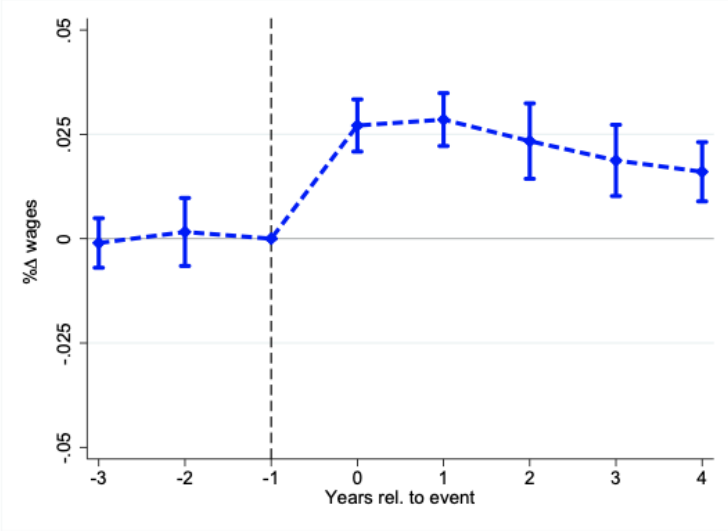
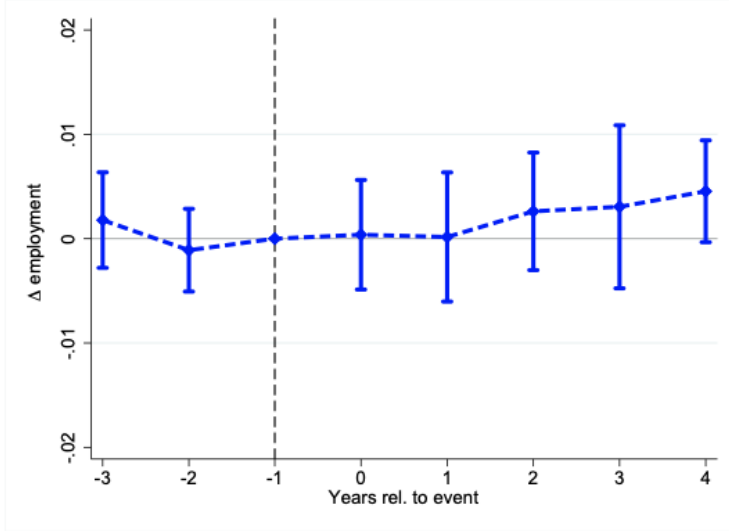


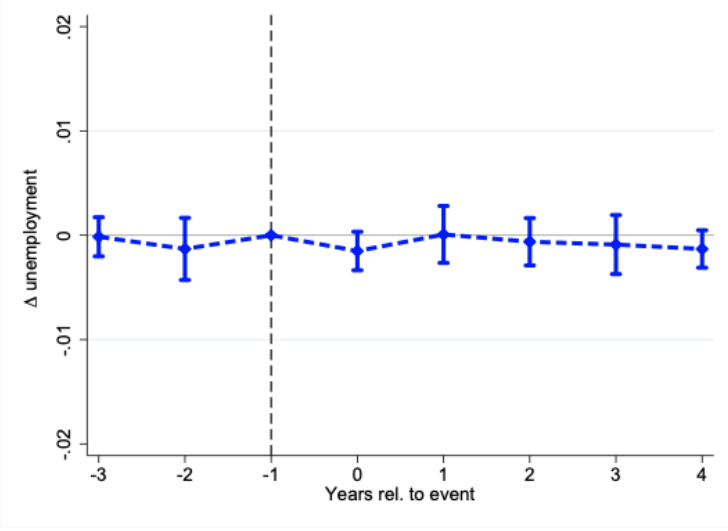
Figure 7: Impact of the Minimum Wage Over Time, High-Probability Group



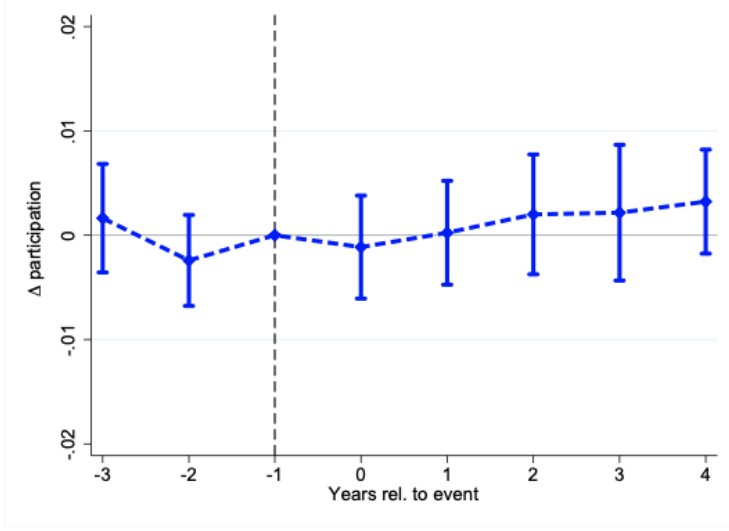
(a) Wage



(b) Employment

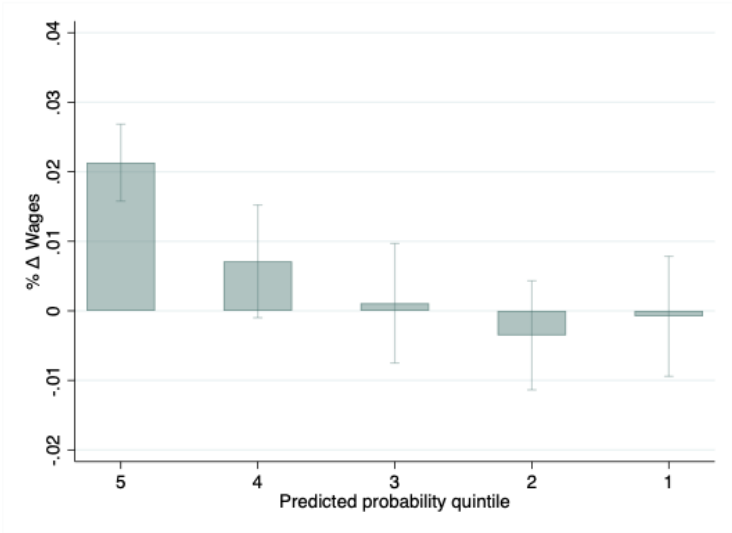


(c) Unemployment

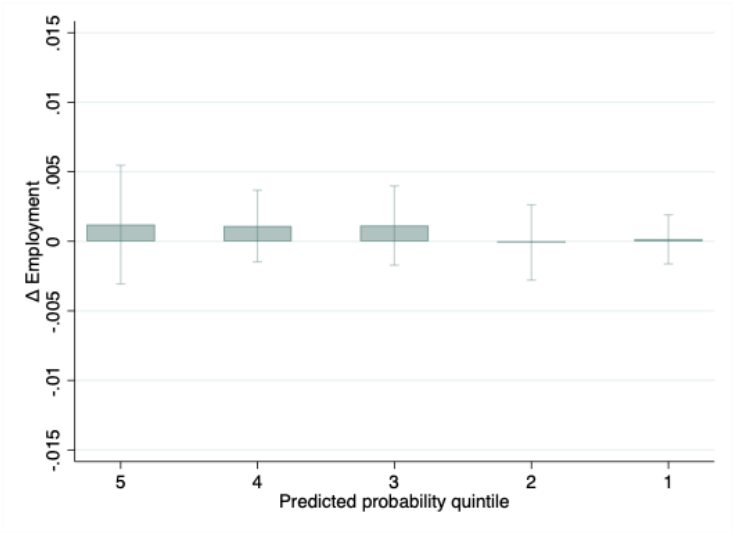


(d) Participation

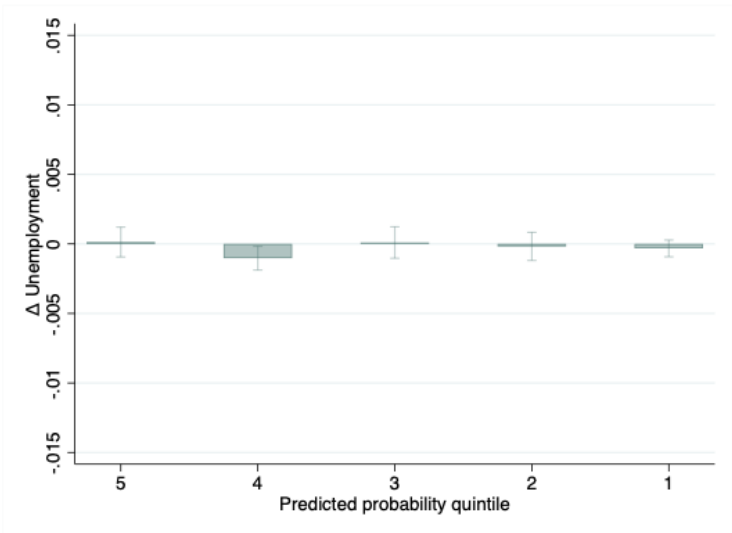
Figure 8: Impact of the Minimum Wage by Predicted Probability Quintiles



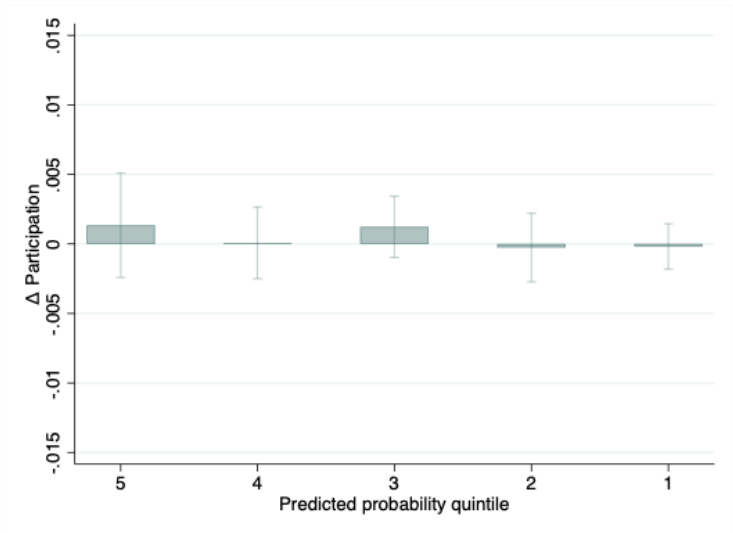
(a) Wage



(b) Employment



(c) Unemployment



(d) Participation

Heterogeneous treatment effects: Knittel and Stolper (2019)

- Knittel and Stolper (2019) consider the question of targeting treatment
- They run an RCT looking at nudges towards reduced electricity consumption
- They estimate the average treatment effect
- But a question: what if you can target treatment? Can you get better than average results?

Heterogeneous treatment effects: Knittel and Stolper (2019)

- Methods: Apply the generalized random forest methods of Wager and Athey (2018) to a large (900,000 obs) RCT
- They estimate conditional average treatment effects (CATEs) for partitions of the sample
- They then illustrate the value of forest-derived CATEs, by measuring "the potential welfare gains from selective targeting of treatment to maximize, alternatively, social and private (i.e., electric utility) objective functions"

Figure 4: A sample causal tree

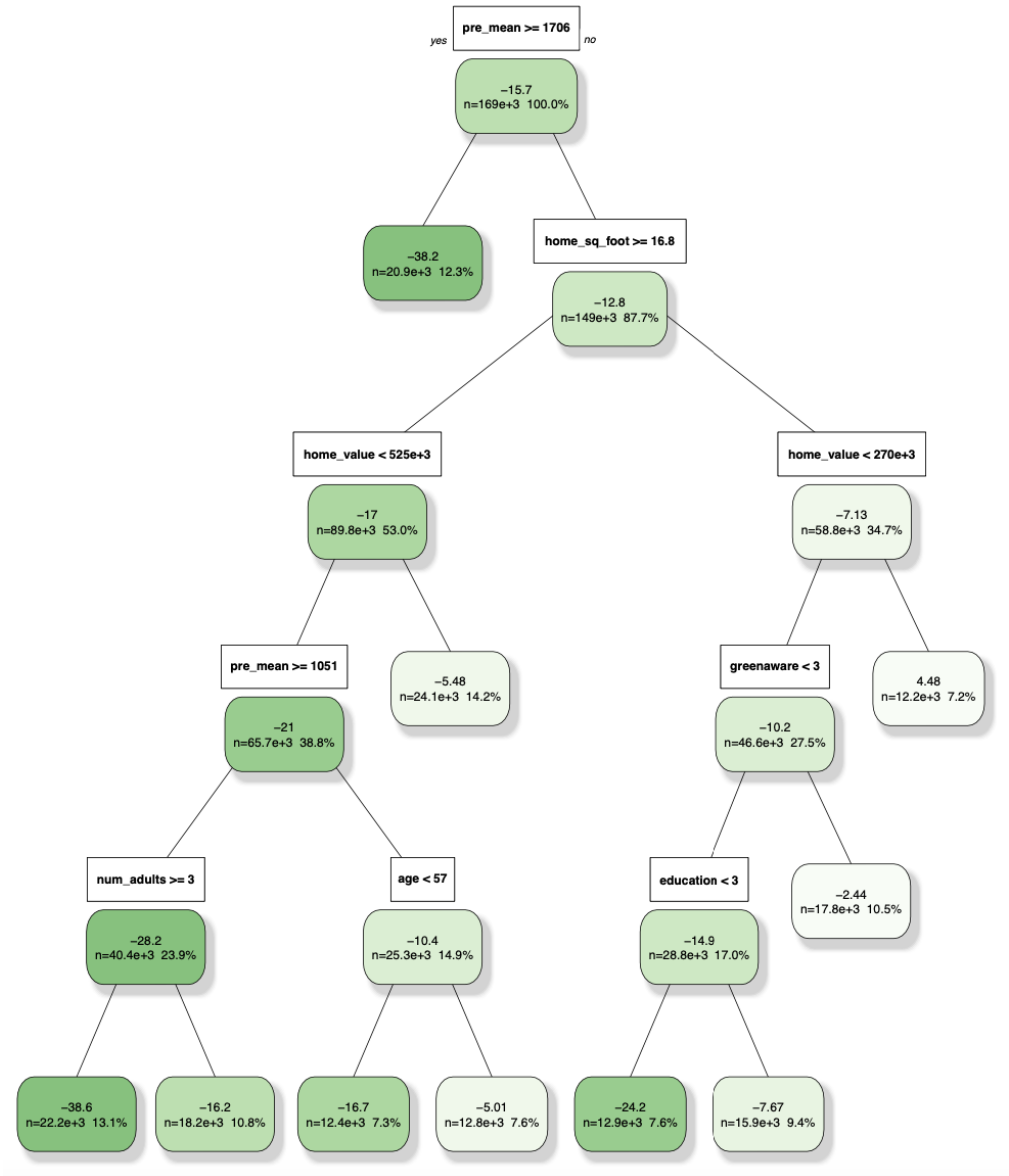
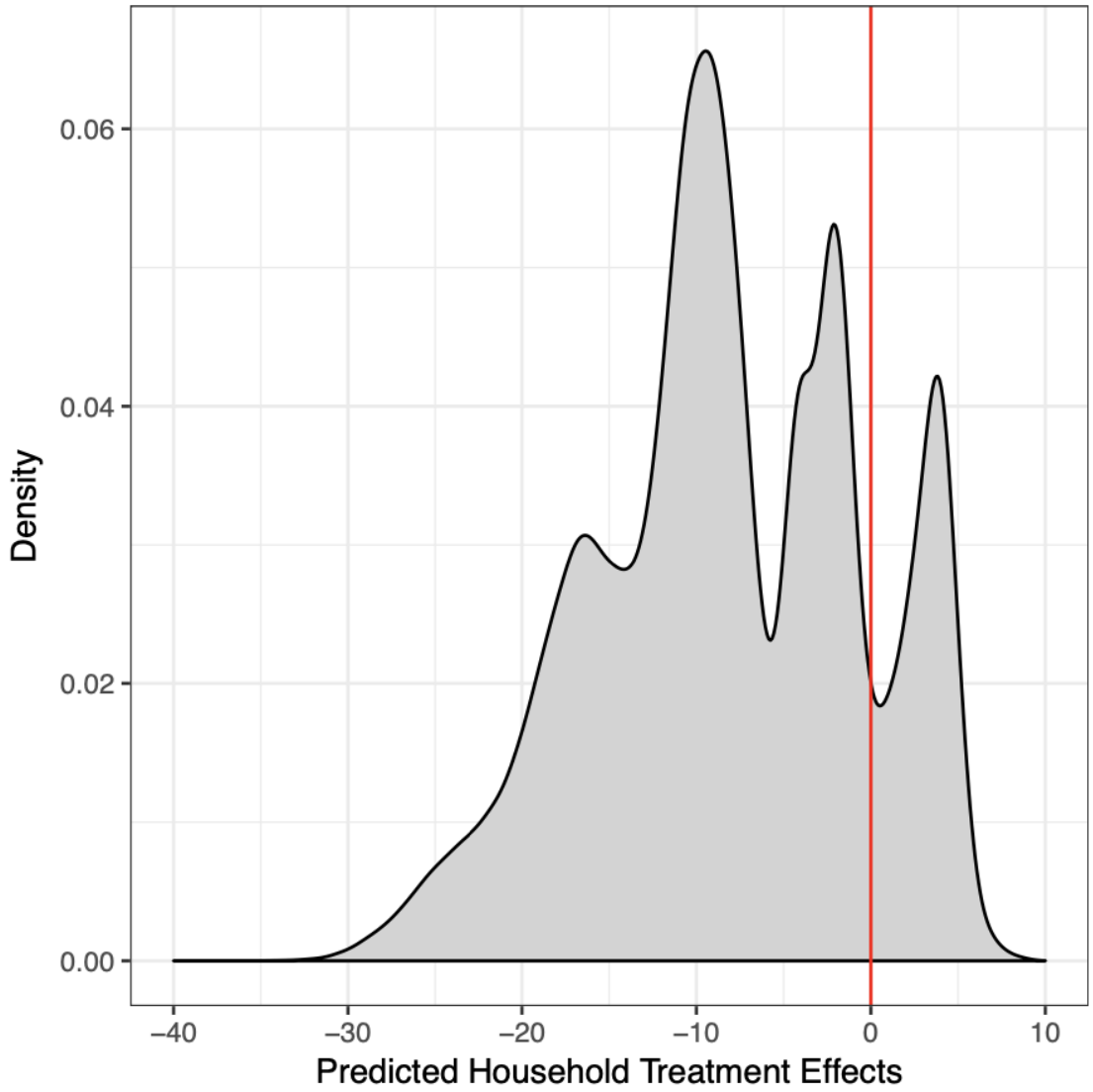


Figure 1: Distribution of Predicted Treatment Effects: 3-Year Average



Heterogeneous treatment effects: Knittel and Stolper (2019)

Punchline:

- Significant heterogeneity in responses to treatment
- If could target the nudges, welfare gains between 14% and 60+% (depends on outcome, year of program)

Data Creation: Gentzkow and Shapiro (2010)

- Gentzkow and Shapiro (2010) ask: What drives the ideological slant in news?
- Is it: consumers? media owners? reporters? pressure from politicians?
- Complications: what do consumers demand? How is slant supplied? **How to measure slant?**

Data Creation: Gentzkow and Shapiro (2010)

To measure ideological slant of a newspaper, GS:

- Analyze all the text in the 2005 Congressional Record, relating 2 and 3 word phrases to members of Congress
- They know the ideology of those members of congress (measured by share of conservative voters in their district/state)
- They use this information together to assign a “conservativeness” score to 2 and 3 word phrases
- Finally, they look for the frequency of these 2 and 3 word phrases in newspapers, to assign a conservativeness score to the newspaper

Data Creation: Gentzkow and Shapiro (2010)

Gentzkow and Shapiro (2010) are applying:

- Unsupervised learning: identifying the 2 and 3 word phrases most commonly used by different members of Congress
- Supervised learning: identifying the relation between 2 and 3 word phrases and ideology
  - → Remember, this is just done for members of Congress, where ideology is known

TABLE I

MOST PARTISAN PHRASES FROM THE 2005 *CONGRESSIONAL RECORD*[a]

Panel A: Phrases Used More Often by Democrats

Two-Word Phrases		
private accounts	Rosa Parks	workers rights
trade agreement	President budget	poor people
American people	Republican party	Republican leader
tax breaks	change the rules	Arctic refuge
trade deficit	minimum wage	cut funding
oil companies	budget deficit	American workers
credit card	Republican senators	living in poverty
nuclear option	privatization plan	Senate Republicans
war in Iraq	wildlife refuge	fuel efficiency
middle class	card companies	national wildlife
Three-Word Phrases		
veterans health care	corporation for public broadcasting	cut health care
congressional black caucus	additional tax cuts	civil rights movement
VA health care	pay for tax cuts	cuts to child support
billion in tax cuts	tax cuts for people	drilling in the Arctic National
credit card companies	oil and gas companies	victims of gun violence
security trust fund	prescription drug bill	solvency of social security
social security trust	caliber sniper rifles	Voting Rights Act
privatize social security	increase in the minimum wage	war in Iraq and Afghanistan
American free trade	system of checks and balances	civil rights protections
central American free	middle class families	credit card debt

TABLE I—*Continued*

Panel B: Phrases Used More Often by Republicans

Two-Word Phrases		
stem cell	personal accounts	retirement accounts
natural gas	Saddam Hussein	government spending
death tax	pass the bill	national forest
illegal aliens	private property	minority leader
class action	border security	urge support
war on terror	President announces	cell lines
embryonic stem	human life	cord blood
tax relief	Chief Justice	action lawsuits
illegal immigration	human embryos	economic growth
date the time	increase taxes	food program
Three-Word Phrases		
embryonic stem cell	Circuit Court of Appeals	Tongass national forest
hate crimes legislation	death tax repeal	pluripotent stem cells
adult stem cells	housing and urban affairs	Supreme Court of Texas
oil for food program	million jobs created	Justice Priscilla Owen
personal retirement accounts	national flood insurance	Justice Janice Rogers
energy and natural resources	oil for food scandal	American Bar Association
global war on terror	private property rights	growth and job creation
hate crimes law	temporary worker program	natural gas natural
change hearts and minds	class action reform	Grand Ole Opry
global war on terrorism	Chief Justice Rehnquist	reform social security

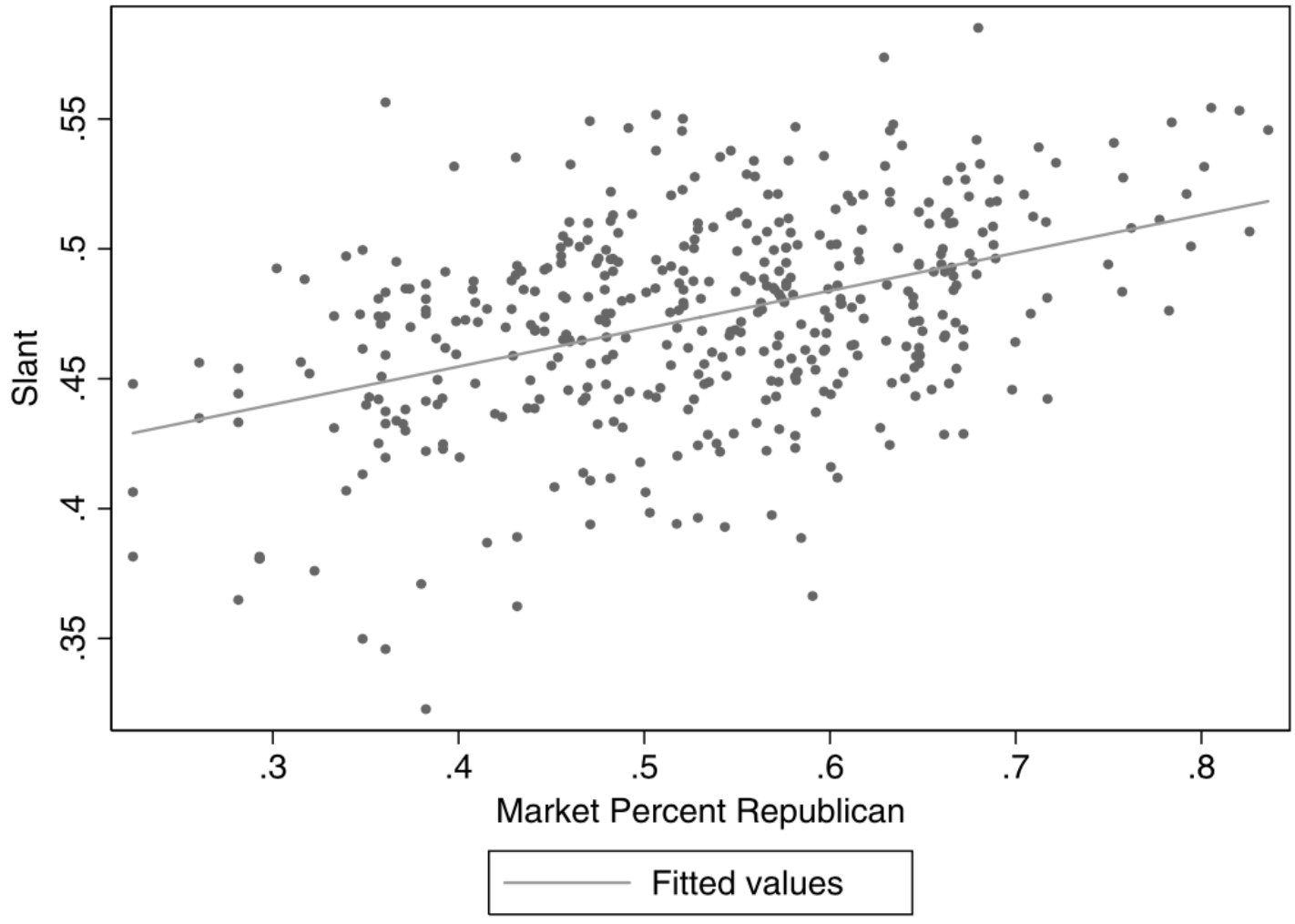


FIGURE 4.—Newspaper slant and consumer ideology. The newspaper slant index against Bush's share of the two-party vote in 2004 in the newspaper's market is shown.

TABLE IV

ECONOMIC INTERPRETATION OF MODEL PARAMETERS[a]

Quantity	Estimate
Actual slant of average newspaper	0.4734
	(0.0020)
Profit-maximizing slant of average newspaper	0.4600
	(0.0047)
Percent loss in variable profit to average newspaper	0.1809
from moving 1 SD away from profit-maximizing slant	(0.1025)
Share of within-state variance in slant from consumer ideology	0.2226
	(0.0406)
Share of within-state variance in slant from owner ideology	0.0380
	(0.0458)

Data Creation: Gentzkow and Shapiro (2010)

Punchline:

- Newspapers slant towards their customers! They demand it!
- It does not appear that newspaper owners or reporters are driving the slant
  - → Since GS estimate demand functions, they can run the counter factual – given this demand, what kind of slant would newspapers have if they wanted to maximize profits (i.e., no bias)
  - → Answer – pretty much what we see them doing now!

Data Creation: Bandiera et al. (2020)

Bandiera et al. (2020), “CEO Behavior and Firm Performance”

- Question: How does CEO behavior (e.g., do they meet with suppliers or C-suite execs) affect firm productivity?
- Challenges: How measure CEO behavior? How to define it?

Data Creation: Bandiera et al. (2020)

To answer their question, Bandiera et al. (2020) have CEOs or their PAs keep diaries:

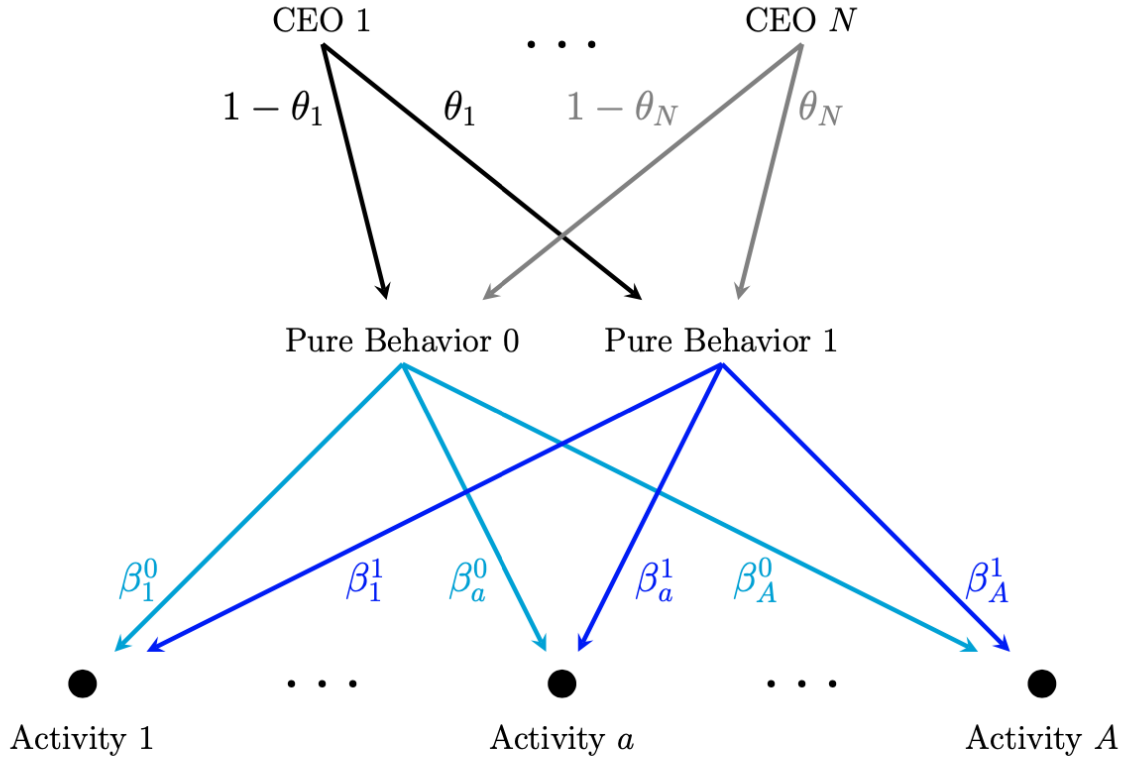
- 1,114 CEOs of manufacturing firms (randomly selected firms)
- Across six countries at different stages of development: Brazil, France, Germany, India, UK and the US.
- Overall, they collect data on 42,233 activities covering an average of 50 working hours per CEO

Data Creation: Bandiera et al. (2020)

Ok - they got detailed diaries. Now what? So many different activities (over 42K)

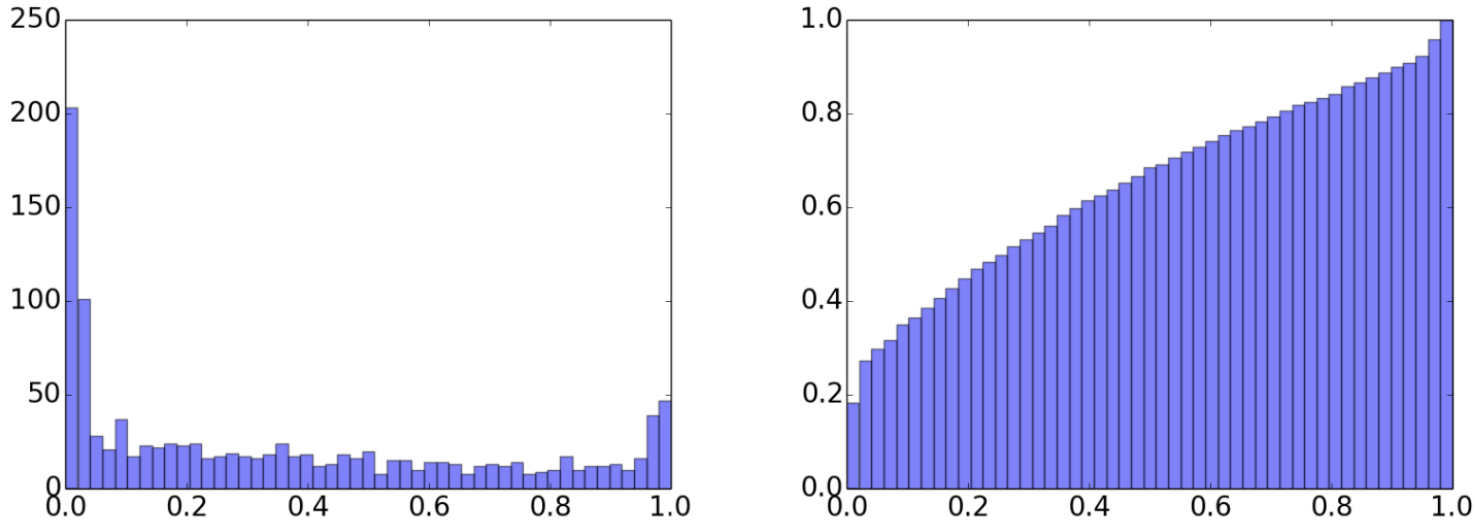
- Break down into 5 major, relevant activities where CEOS differ
- Find that behavior among these activities are correlated with one another
- Use Latent Dirichlet Allocation (LDA): unsupervised learning tool for identifying clusters of similar topics in data (often used for text analysis)
  - $\rightarrow$ This is a dimension reduction tool
  - $\rightarrow$ Ideal in this context with a high dimensional problem ($p >> n$)

Figure 3: Data Generating Process for Activities with Two Pure Behaviors



Notes: This figure provides a graphical representation of the data-generating process for the time-use data. First, CEO i chooses – independently for each individual unit of his time – one of the two pure behaviors according to a Bernoulli distribution with parameter θ_i. The observed activity for a unit of time is then drawn from the distribution over activities that the pure behavior defines.

Figure 5: CEO Behavior and Index Distribution



(a) Frequency Distribution **(b)** Cumulative Distribution

Notes: The left-hand side plot displays the number of CEOs with behavioral indices in each of 50 bins that divide the space [0,1] evenly. The right-hand side plot displays the cumulative percentage of CEOs with behavioral indices lying in these bins.

Data Creation: Bandiera et al. (2020)

But how know that CEO behavior impacts firm performance?

1. Set up theoretical model to show mechanisms
2. Estimate panel, fixed effects models:
   - → Firms who appoint CEOs high and low on index have similar productivity growth trends before the appointment
   - → But after the appointment, the firms who appoint "good behavior" CEOs do increase productivity more
   - → This effect is heterogeneous – strongest in poor regions of poor countries

Data Creation: Bandiera et al. (2020)

Finally,

- They derive empirical model from their theoretical model
- Allows for counterfactuals to be simulated: e.g., can see how much of productivity gap between firms in developing and developed countries is due to poor CEO choices
- Find that this is about 13% of the gap

Quantitative Theory: Kasy (2018)

Kasy (2018), "Optimal taxation and insurance using machine learning – sufficient statistics and beyond"

- Chetty (2009) summarizes what's called the "sufficient statistics approach"
  - → This maps empirical estimates of behavior parameters (e.g., elasticities) to theoretical models so one can use those empirical estimates for welfare analysis
  - → Aside: This is an important paper and anyone working in labor or public economics should read it.
- Kasy (2018) notes that this mapping requires some strong assumptions (e.g., about elasticities being constant over income)
- Others have noted this, but what can you do?
  - → The empirical estimates often didn't allow for the identification of functional forms like this

Quantitative Theory: Kasy (2018)

- Question: How are the results of optimal taxation models affected if we relax some of the functional form assumptions common to the sufficient statistics approach?
- Also: What is we model the uncertainty of policy makers over the behavioral responses as they try to maximize [expected] social welfare?

Quantitative Theory: Kasy (2018)

- Application: optimal health co-insurance rates
- Related to optimal taxation since the problem is an equity-efficiency trade-off
  - → Lower coinsurance rates redistribute to those who are sick
  - → But they raise the overall costs of insurance
  - → And just like change taxes, changes to coinsurance have behavioral effects
- Plus – there are good, randomized data available from which one can estimate behavioral parameters relevant to the policy choice

Quantitative Theory: Kasy (2018)

Traditional SWF

$$t^*(\hat{\theta}) = argmax_t u(t, \hat{\theta})$$

Bayesian SWF

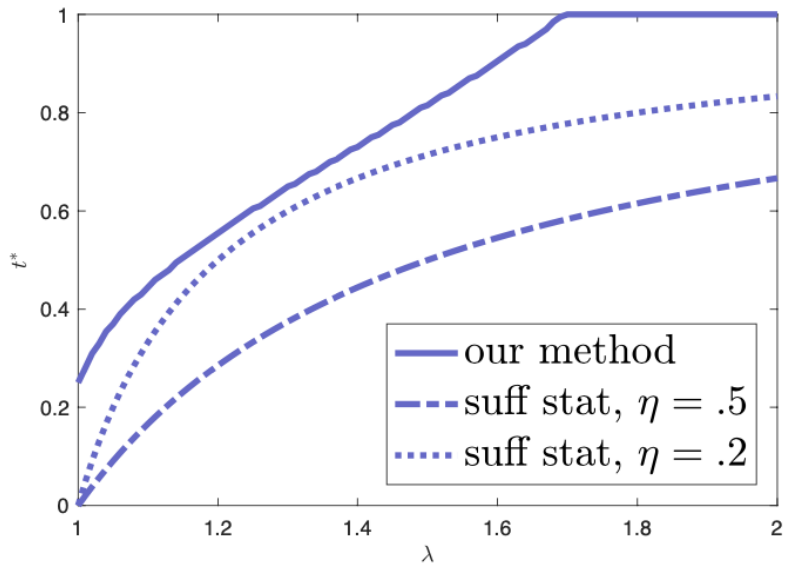
$$\hat{t}^* = argmax_t \int u(t, \theta) d\pi(\theta|X)$$

where $\pi(\theta|X)$ is the posterior distribution of X

Because of the curvature of $u(\cdot)$, in general:

$$E\left[u(t, \theta)|X\right] \neq u(t, E[\theta|X])$$

Figure 2: The optimal policy t^* as a function of λ



Notes: This graph plots the optimal policy t^* as a function of distributive preference λ, estimated using our proposed method, using the sufficient statistics approach with the Aron-Dine et al. (2013) estimate of $\widehat{\eta} = 0.5$, and using the RAND investigators' estimate of $\widehat{\eta} = 0.2$.

Quantitative Theory: Kasy (2018)

Punchline:

- Maximizing expected SWF gives much different results
  - → In health insurance example, optimal coinsurance rate drops from about 50% to about 18% (with traditional parameterization of preferences)
- Taking a Bayesian approach and non-parametric methods of estimation from machine learning, it's possible to estimate sufficient statistics non-parametrically.
  - → This too, is often quantitatively important

Computational economics: The work of Simon Scheidegger

See, for example:

- "Machine learning for high-dimensional dynamic stochastic economies" (*Journal of Computational Science*)
- "Deep Structural Estimation: With an Application to Option Pricing"
- "Machine Learning for Dynamic Incentive Problems"
- "Deep Equilibrium Nets"
- His lectures on "Deep Equilibrium Nets and Deep Structural Estimation"

Summary of ML and Econ

- Definitely something to be aware of
- It's "sexy"
- But be careful: just because you have a hammer, not everything is a nail
- ML was not designed for the problems of economics and often works against what out comparative advantages are
  - → Our comparative advantage is economic theory – being able to think through causal relationships
  - → ML algorithms tend to push one away from interpretable relationships and have little reliance on theory

Summary of ML and Econ

Places I think ML is most likely to be successful in helping [research] economists:

- Identifying heterogeneous treatment effects in large RCTs
- Helping construct original data sources (especially from "qualitative" information)
- Improving the ability to compute high-dimensional problems in quantitative theory

References I

Athey, Susan and Guido Imbens (July 2016). "Recursive Partitioning for Heterogeneous Causal Effects". *Proceedings of the National Academy of Science* 113.27, pp. 7353–7360. DOI: 10.1016/j.jpubeco.2018.09. URL: https://ideas.repec.org/a/eee/pubeco/v167y2018icp205-219.html.

Bandiera, Oriana et al. (2020). "CEO Behavior and Firm Performance". *Journal of Political Economy* 128.4, pp. 1325–1369. DOI: 10.1086/705331. URL: https://ideas.repec.org/a/ucp/jpolec/doi10.1086-705331.html.

Cengiz, Doruk et al. (Jan. 2021). *Seeing Beyond the Trees: Using Machine Learning to Estimate the Impact of Minimum Wages on Labor Market Outcomes*. NBER Working Papers 28399. National Bureau of Economic Research, Inc. URL: https://ideas.repec.org/p/nbr/nberwo/28399.html.

References II

Chetty, Raj (May 2009). "Sufficient Statistics for Welfare Analysis: A Bridge Between Structural and Reduced-Form Methods". *Annual Review of Economics* 1.1, pp. 451–488. URL: https://ideas.repec.org/a/anr/reveco/v1y2009p451-488.html.

Gentzkow, Matthew and Jesse M. Shapiro (Jan. 2010). "What Drives Media Slant? Evidence From U.S. Daily Newspapers". *Econometrica* 78.1, pp. 35–71. URL: https://ideas.repec.org/a/ecm/emetrp/v78y2010i1p35-71.html.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

James, Gareth et al. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer. URL: https://faculty.marshall.usc.edu/gareth-james/ISL/.

References III

Kasy, Maximilian (2018). "Optimal taxation and insurance using machine learning — Sufficient statistics and beyond". *Journal of Public Economics* 167.C, pp. 205–219. DOI: 10.1016/j.jpubeco.2018.09. URL: https://ideas.repec.org/a/eee/pubeco/v167y2018icp205-219.html.

Knittel, Christopher R. and Samuel Stolper (Dec. 2019). *Using Machine Learning to Target Treatment: The Case of Household Energy Use*. NBER Working Papers 26531. National Bureau of Economic Research, Inc. URL: https://ideas.repec.org/p/nbr/nberwo/26531.html.

Mullainathan, Sendhil and Jann Spiess (Spring 2017). "Machine Learning: An Applied Econometric Approach". *Journal of Economic Perspectives* 31.2, pp. 87–106. URL: https://ideas.repec.org/a/aea/jecper/v31y2017i2p87-106.html.

References IV

Wager, Stefan and Susan Athey (2018). “Estimation and Inference of Heterogeneous Treatment Effects using Random Forests”. *Journal of the American Statistical Association* 113, pp. 1228–1242.