Big Data and Economics

Regression Trees

Kyle Coombs
Bates College | ECON/DCS 368

Table of contents

Prologue

Prologue

- Last week we talked about the basics of machine learning
- You take a bunch of data, split it into training and testing sets, fit a model to the training data, then test it on the testing data
- Let's look at a particular type of machine learning model: decision trees
- Decision trees are a type of machine learning model that are easy to interpret
    - They allow for non-linear relationships between the dependent and independent variables
    - But the math is a lot more complicated than an OLS regression
    - The visualizations are more straight-forward
- Trees **stratify**, **segment**, or **partition** the data into subgroups
    - Each subgroup predicts a different value of the dependent variable
    - These lend themselves to flowcharts!

Questions

Hack-a-thon

- Fill out survey please
- Right now we have three participants

Attribution

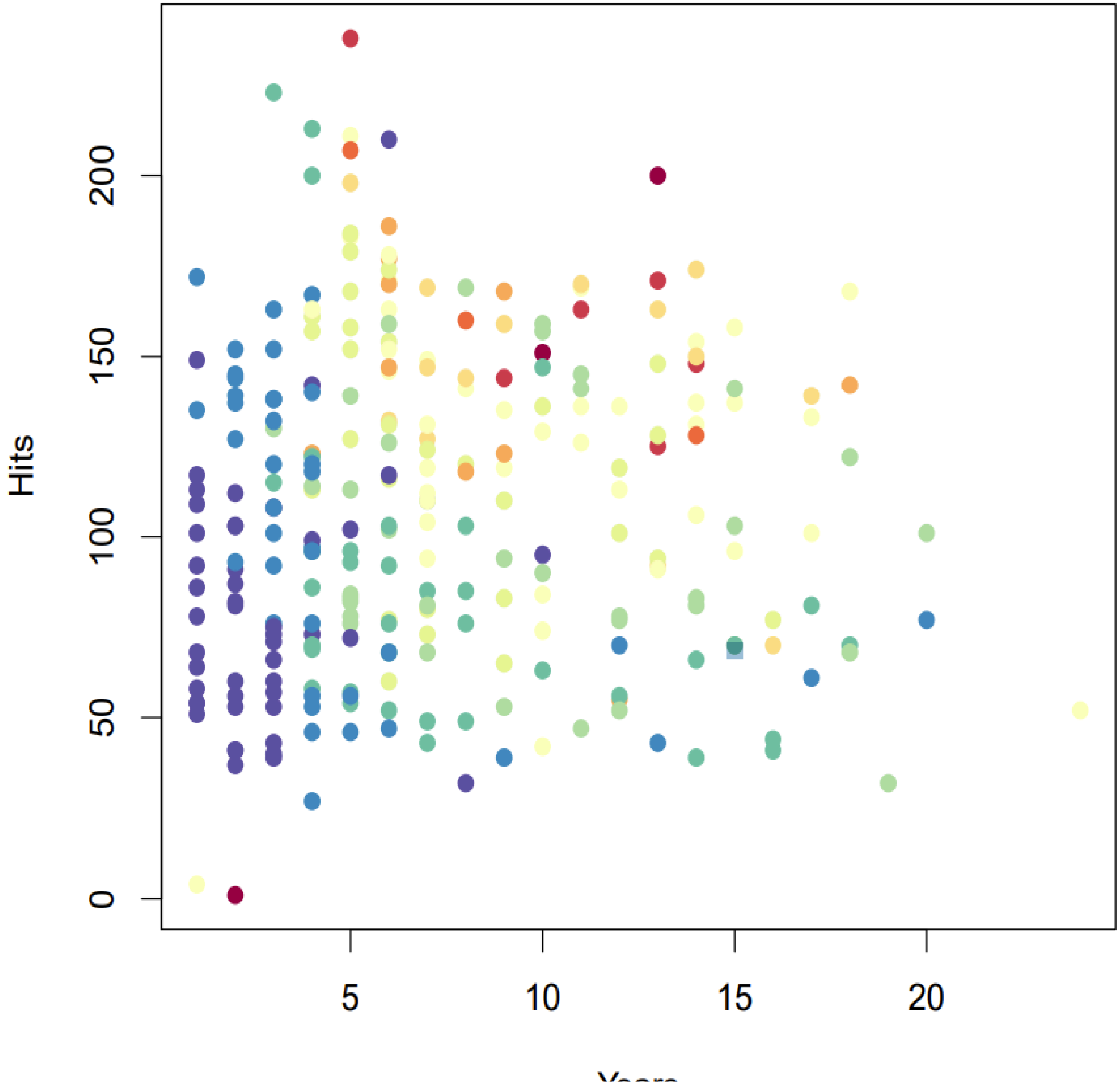
- This lecture is based on the following resources:
- Introduction to Statistical Learning, Chapter 8
- Tyler Ransom's lecture notes

Decision Trees

Motivating example

- Imagine you want to predict the income mobility of those raised in the 25th percentile for some county
  - **Remember we are not making causal inferences here, just predictions**
- You have tons of predictors: job growth, health, education, crime rates, share of each race, median HH income, college degrees, bankruptcies, religiosity, and more
- The correlations between these variables and mobility is likely non-linear
- The correrlations between mobility and median HH income could change depending on the rate of education
- Would a regression capture all of that?
- Probably not...

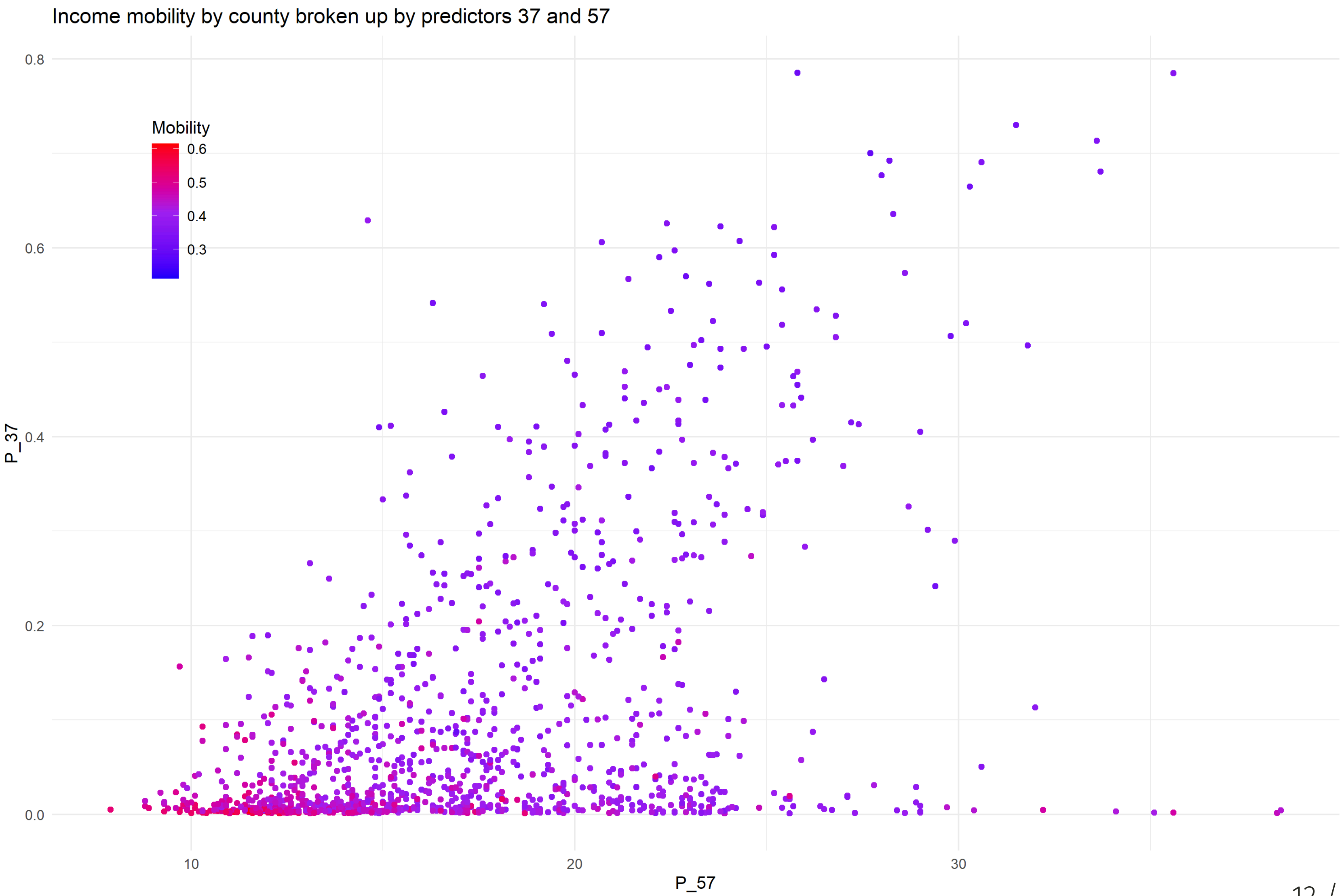
Stratifying baseball data



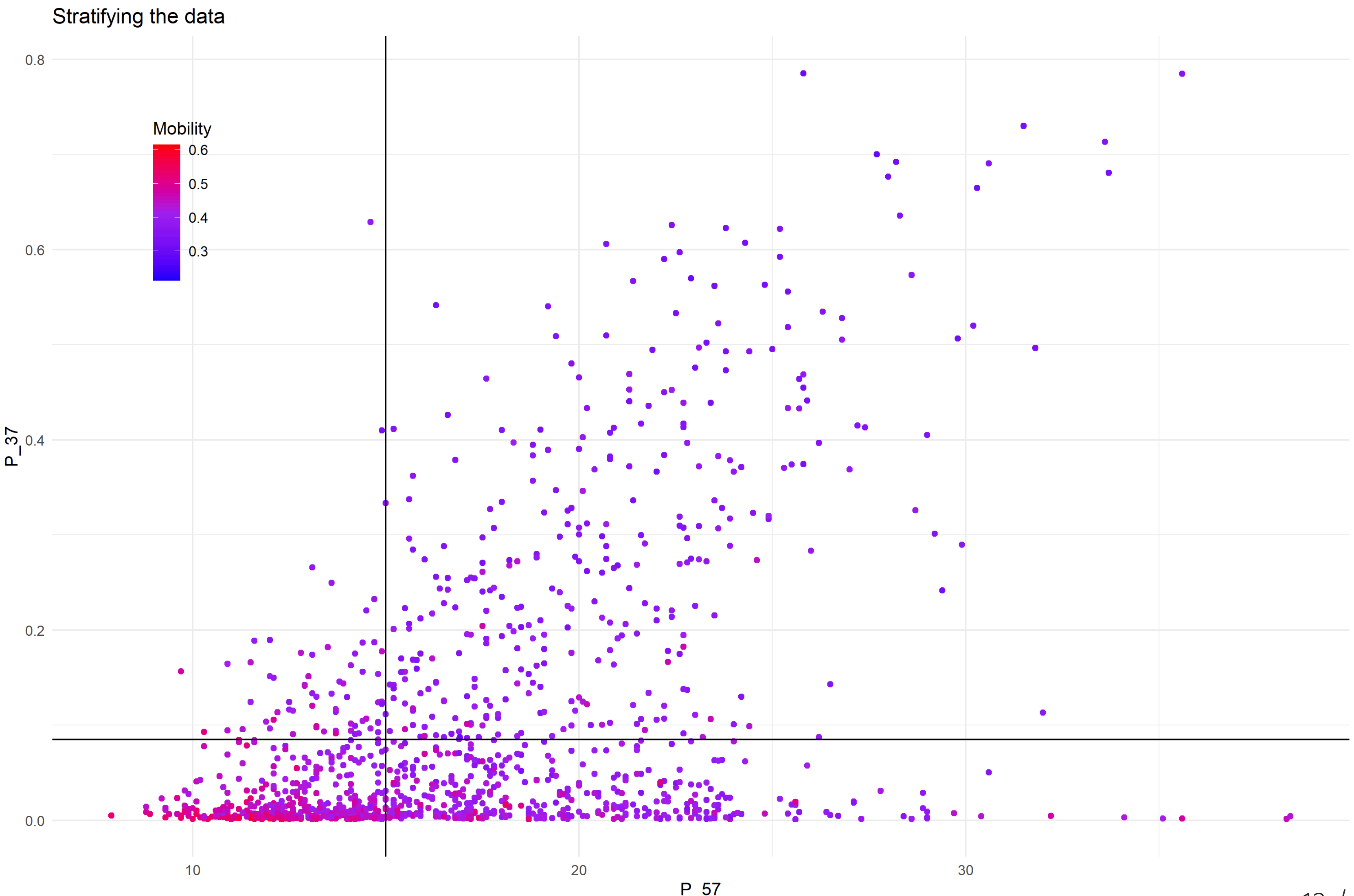
Opportunity Atlas data

Couldn't load plugin.

Stratify opp atlas data



Strata lines



What is a decision tree?

- A decision tree organizes variables into tree-like structure
  - It is essentially, a really fancy flowchart
- At each node, pick the variable that best meets a decision rule
- At node 1, the algorithm cycles through each X variable and finds the split in the data that best meets the decision rule
  - It picks the best X variable
  - It follows the branch down and creates nodes by looking at the remaining X's that best meet the decision rule
- When making a decision about an observation, follow the tree down the branches

Types of decision trees

Regression trees

- The decision rule is what variable X best predicts y when split at some cutoff point $\bar{X}$
  - Typically the predicted $\hat{y}$ is the average (could be mode) of y conditional on $X > \bar{X}$ or $X < \bar{X}$
- At the terminal node, the prediction $\hat{y}$ is for all observations in that node

Classification trees

- Your outcome is now a categorical variable and you predict the probability an observation fits the category

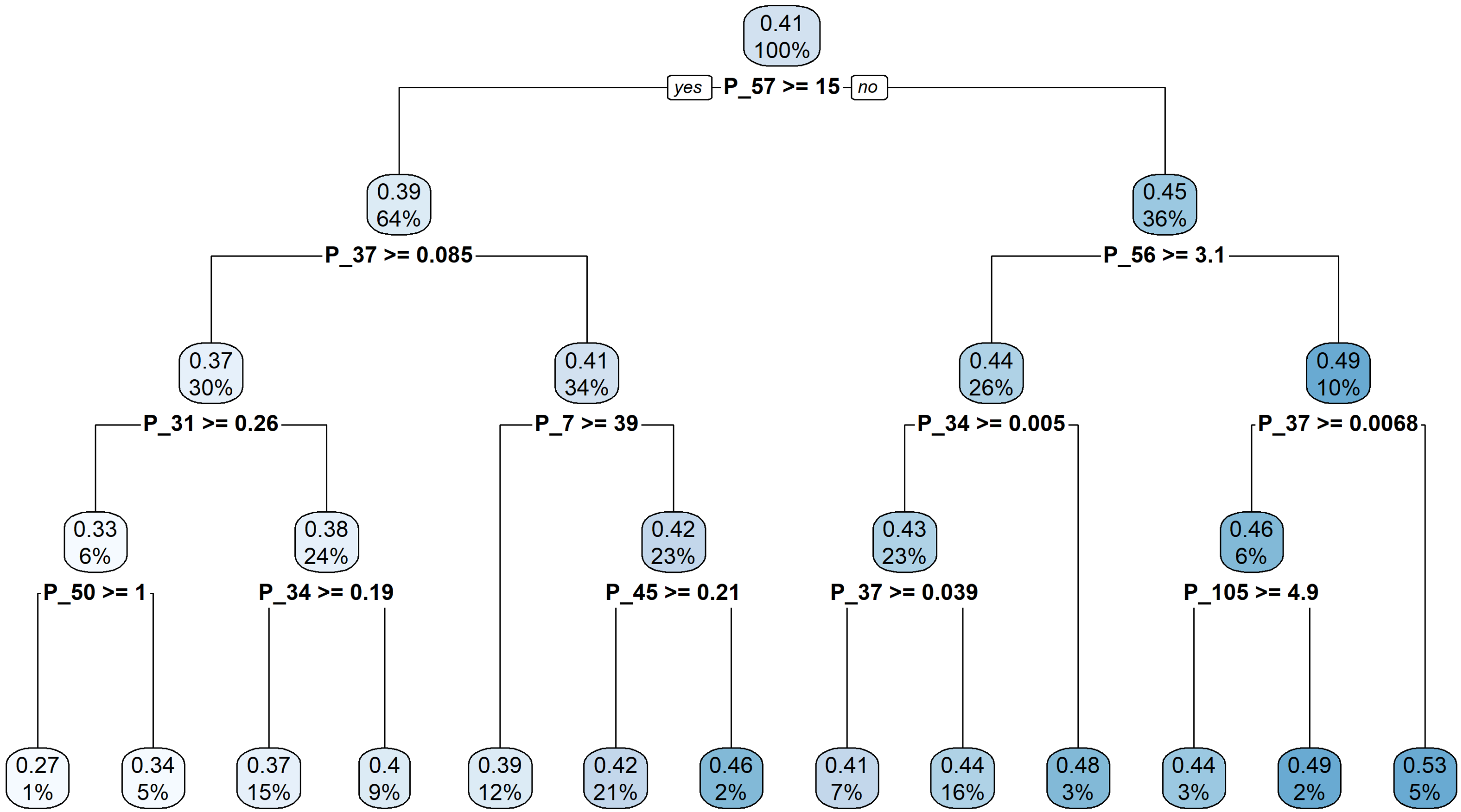
Causal Trees

- Instead of splitting based on prediction, split to maximize the difference in the average treatment effect (ATE) between the two branches
- At each node, the X covariate that maximizes the difference in the ATE is selected
- The goal is to see how varied the treatment effect is across different subgroups of the population
  - This is the heterogeneity of treatment effect (HTE)
  - We want to find the subgroups that have the largest HTE

- The result gives a conditional average treatment effect (CATE) for each subgroup from a variety of

Regression Tree of income mobility

- Each node shows the share of observations and the average income mobility for thse observations
- Each branch shows the decision rule as a cutoff in the variable that minimizes the RSS



Random Forests

Many trees make a forest

- Decision trees are fairly easily to interpret once you make one
- But one drawback is that they are very sensitive to the data
  - Too many nodes and you could overfit
  - Too few nodes and you'll just have noise
- So what if we made many trees and averaged the predictions?
  - Technically this is just called "bagging" (bootstrap aggregating)
  - Random forests also randomize the variables available to split the nodes
  - See more at Introduction to Statistical Learning, Chapter 8.2
- But won't we just repeat the same tree over and over?

Pull yourself up by your bootstraps

- How could we use bootstrapping?

Pull yourself up by your bootstraps

- How could we use bootstrapping?

- If you bootstrap the data B times, you create B new samples of the data indexed b

1. For each bootstrap sample b, create a decision tree T_b using the bootstrap sample b
2. For each observation i in the original sample, predict the outcome y_i using all B trees
3. Average the predictions as $\hat{y}_i = \frac{1}{B}\sum_{b=1}^{B} T_b(X_i)$

- This is called bagging (bootstrap aggregating)

- **Intuition**: With many trees, you can average out the noise and get a better prediction

- Random forests add a twist to bagging by randomly selecting a subset of X variables to split the nodes in the tree

    - This ensures the trees are uncorrelated with each other
    - Minimizes variance

Intuition: By randomizing the X variables available to a tree, they are less likely to only use the same variables to split the nodes in the tree. As a result, the algorithm evaluates other variables in the data.

Use cases of random forests

- Random forests are a very popular machine learning technique
- They are used for prediction, classification, and causal inference
- Kleinberg et al. (2018) use random forests to predict the judicial bail decisions in NYC

What next?

- Get your hands dirty!
- Navigate to the Generalized Random Forest vignette

```r
#install.packages('grf')
library(grf)
```

- This will walk you through how to use the **grf** package to estimate causal forests
- Once you finish, try the **grf** guided tour
    - I recommend you try the application to school program evaluation example
- This package is full of vignettes that you could use for the problem set

Next lecture: Least Absolute Shrinkage and Selection Operator (LASSO)