Machine Learning Theory

Lecture 3: Bounded Variation Regression

David A. Hirshberg

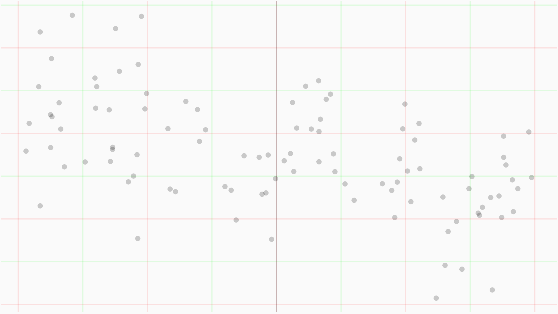
May 24, 2024

Emory University

Review

A Fictionalized RDD Example

- Question. Are smaller classes better for 5th graders?
- Data. A state caps class sizes at 40.
  - When there are $x \leq 40$ 5th graders enrolled in a school, they run one class of size x.
  - When there are $x > 40$, they run two classes of average size $x/2$.

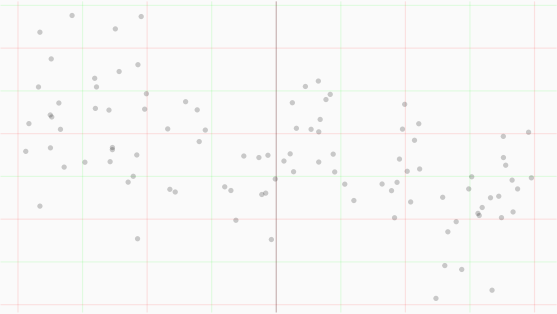


This is a fake-data version of a study of Angrist and Lavy [1999]. The state is Israel.

- It has been simplified to make our discussion easier.
- Real schools sometimes had more than 80 5th-graders enrolled.
- And they didn't follow this cap perfectly.

A Fictionalized RDD Example

- Question. Are smaller classes better for 5th graders?
- Data. A state caps class sizes at 40.
  - When there are $x \le 40$ 5th graders enrolled in a school, they run one class of size x.
  - When there are $x > 40$, they run two classes of average size $x/2$.



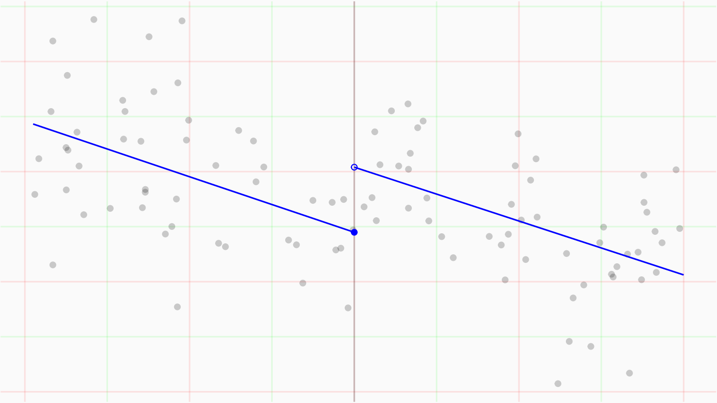
We can use this to estimate the average effect of having 20 vs. 40 students/class.

$$\text{effect} = \mu(40+) - \mu(40-) \qquad \text{where}$$

$$\mu(x) = \mathrm{E}[\text{avg. test score}_i \mid \text{enrolled 5th graders}_i = x].$$

All we have to do is estimate $\mu(x)$ just to the left and to the right of 40.

The Simple Approach

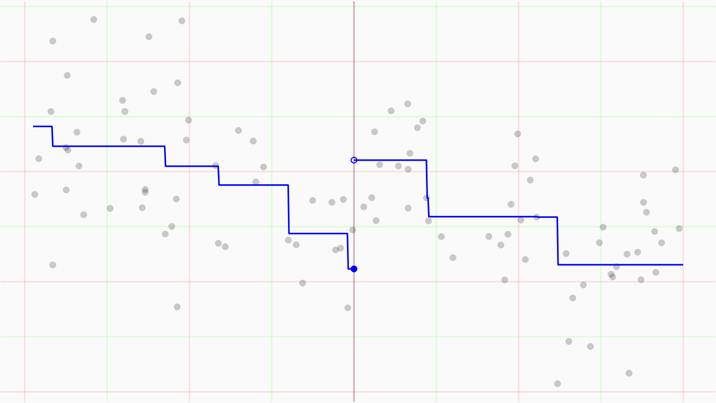


$$\widehat{\text{effect}} = \hat{\mu}_{\text{right}}(40) - \hat{\mu}_{\text{left}}(40) = 0.59$$

where

$$\hat{\mu}_{\text{left}} = \operatorname*{argmin}_{m \in \mathcal{M}} \sum_{i: X_i \leq 40} \{ Y_i - m(X_i) \}^2 \quad \text{and} \quad \hat{\mu}_{\text{right}} = \operatorname*{argmin}_{m \in \mathcal{M}} \sum_{i: X_i > 40} \{ Y_i - m(X_i) \}^2.$$

The Simple Approach

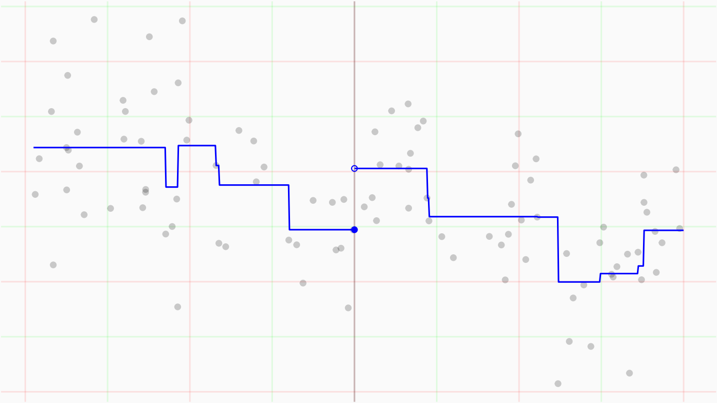


$$\widehat{\text{effect}} = \hat{\mu}_{\text{right}}(40) - \hat{\mu}_{\text{left}}(40) = 0.99$$

where

$$\hat{\mu}_{\text{left}} = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \sum_{i: X_i \leq 40} \{Y_i - m(X_i)\}^2 \quad \text{and} \quad \hat{\mu}_{\text{right}} = \underset{m \in \mathcal{M}}{\operatorname{argmin}} \sum_{i: X_i > 40} \{Y_i - m(X_i)\}^2.$$

The Simple Approach

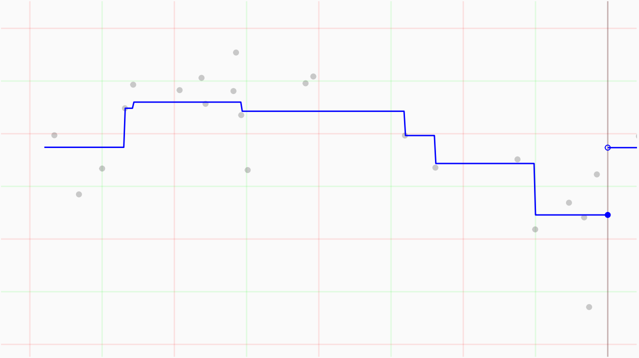


$$\widehat{\text{effect}} = \hat{\mu}_{\text{right}}(40) - \hat{\mu}_{\text{left}}(40) = 0.56$$

where

$$\hat{\mu}_{\text{left}} = \operatorname*{argmin}_{m \in \mathcal{M}} \sum_{i: X_i \leq 40} \{Y_i - m(X_i)\}^2 \quad \text{and} \quad \hat{\mu}_{\text{right}} = \operatorname*{argmin}_{m \in \mathcal{M}} \sum_{i: X_i > 40} \{Y_i - m(X_i)\}^2.$$

The Bounded Variation Model



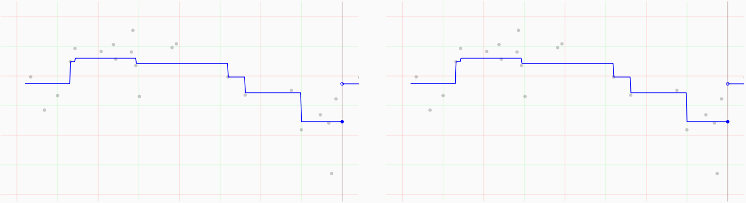
$$\hat{\mu} = \operatorname*{argmin}_{m \in \mathcal{M}} \frac{1}{n} \sum_{i=1}^{n} \{ Y_i - m(X_i) \}^2 \quad \text{where} \quad \mathcal{M} = \{ m : \int |m'(x)| \leq B \}.$$

This is too vague to actually implement.

1. We haven't specified the domain we're integrating over.
2. Nor have we said what this means for non-differentiable functions.

Let's fix that.

The integration domain is easy. It's a matter of convention.
We shift and scale our X_i into the unit interval $[0, 1]$. That's our integration domain.



$$X_i \in (0, 40] \qquad \xrightarrow{x \to x/40} \qquad X_i' \in (0, 1]$$

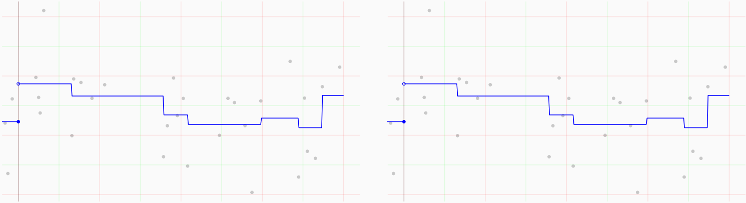
What's nice about this is that it's an average.

$$\int_0^1 |m'(x)| dx = \mathrm{E}|m'(\tilde{X})| \text{ for } \tilde{X} \text{ uniformly distributed in } [0, 1]$$

Averages are easy to think about and compare to other things.

$$\mathrm{E}|m'(\tilde{X})| \le \sqrt{\mathrm{E}\, m'(\tilde{X})^2} \le \max_x |m'(x)|$$

The integration domain is easy. It’s a matter of convention.
We shift and scale our X_i into the unit interval $[0,1]$. That’s our integration domain.



$$X_i \in (40,80] \qquad \xrightarrow{x \to (x-40)/40} \qquad \tilde{X}_i \in (0,1]$$

What’s nice about this is that it’s an average.

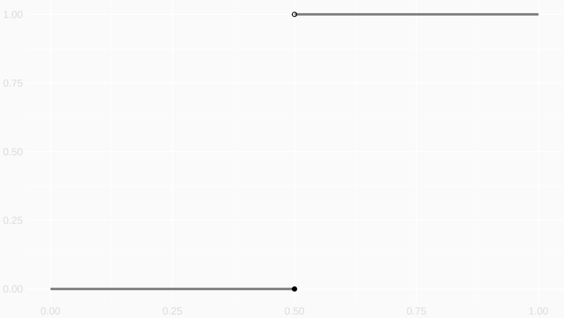
$$\int_0^1 |m'(x)|dx = \mathrm{E}|m'(\tilde{X})| \;\text{ for }\; \tilde{X} \text{ uniformly distributed in } [0,1]$$

Averages are easy to think about and compare to other things.

$$\mathrm{E}|m'(\tilde{X})| \le \sqrt{\mathrm{E}\, m'(\tilde{X})^2} \le \max_x |m'(x)|$$

The nondifferentiable thing is subtler.
Think about this problem—an infinite-data noiseless regression.

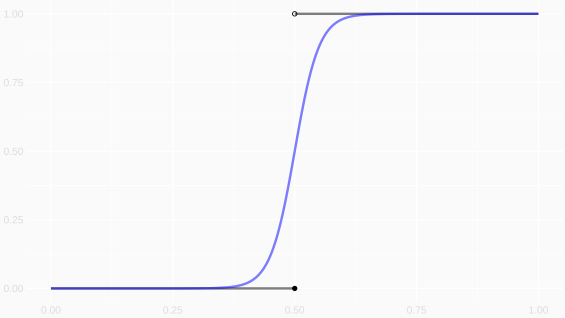
$$\hat\mu = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ \text{differentiable } m : \int_0^1 |m'(x)| \le B \right\}.$$



Here's μ.
What is the solution $\hat\mu$?

The nondifferentiable thing is subtler.
Think about this problem—an infinite-data noiseless regression.

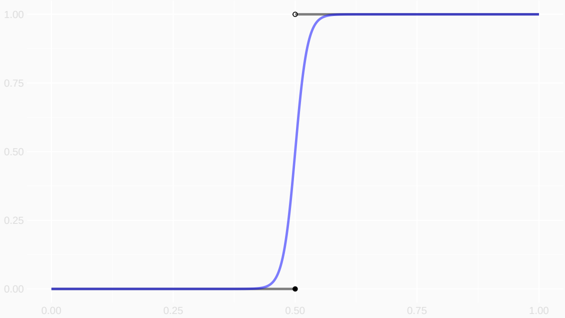
$$\hat{\mu} = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ \text{differentiable } m : \int_0^1 |m'(x)| \leq B \right\}.$$



It's not this function m.
We can find a better fit.

The nondifferentiable thing is subtler.
Think about this problem—an infinite-data noiseless regression.

$$\hat{\mu} = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ \text{differentiable } m : \int_0^1 |m'(x)| \leq B \right\}.$$

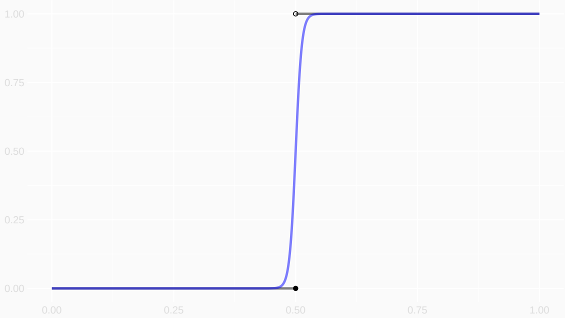


This is better, but we can keep going.
We can get arbitrarily close.

$$\mu = \lim_k m_k \quad \text{for} \quad m_k \in \mathcal{M}.$$

The nondifferentiable thing is subtler.
Think about this problem—an infinite-data noiseless regression.

$$\hat{\mu} = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ \text{differentiable } m : \int_0^1 |m'(x)| \leq B \right\}.$$



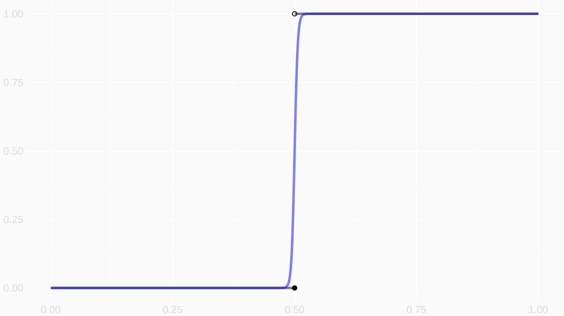
This is better, but we can keep going.
We can get arbitrarily close.

$$\mu = \lim_k m_k \quad \text{for} \quad m_k \in \mathcal{M}.$$

$$m_k(x) = \frac{1}{1 + e^{-k(x - .5)}}$$

The nondifferentiable thing is subtler.
Think about this problem—an infinite-data noiseless regression.

$$\hat{\mu} = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ \text{differentiable } m : \int_0^1 |m'(x)| \leq B \right\}.$$



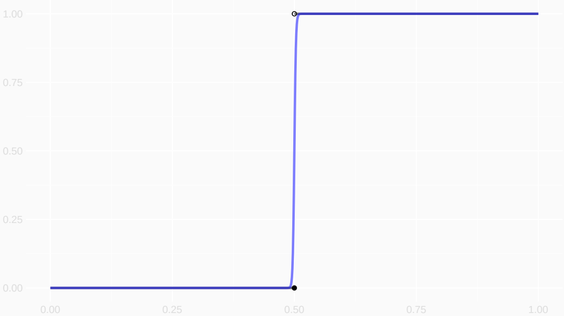
This is better, but we can keep going.
We can get arbitrarily close.

$$\mu = \lim_k m_k \quad \text{for} \quad m_k \in \mathcal{M}.$$

$$m_k(x) = \frac{1}{1 + e^{-k(x - .5)}}$$

The nondifferentiable thing is subtler.
Think about this problem—an infinite-data noiseless regression.

$$\hat{\mu} = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ \text{differentiable } m : \int_0^1 |m'(x)| \le B \right\}.$$



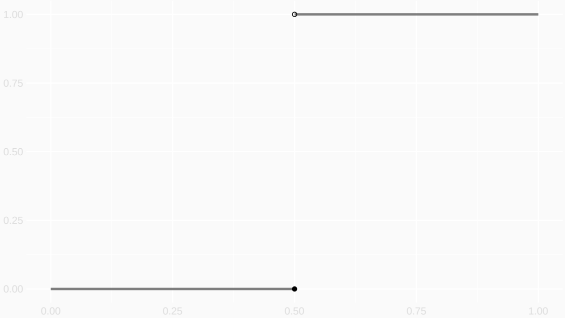
This is better, but we can keep going.
We can get arbitrarily close.

$$\mu = \lim_k m_k \quad \text{for} \quad m_k \in \mathcal{M}.$$

$$m_k(x) = \frac{1}{1 + e^{-k(x - .5)}}$$

The nondifferentiable thing is subtler.
Think about this problem—an infinite-data noiseless regression.

$$\hat\mu = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ \text{differentiable } m : \int_0^1 |m'(x)| \le B \right\}.$$



The **only reasonable answer** is $\hat\mu = \mu$
But μ isn't in our model $\mathcal{M}$.

The $\operatorname{argmin} \hat{\mu}$ **doesn't exist**.

$$\hat{\mu} = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ \text{differentiable } m : \int_0^1 |m'(x)| \le B \right\}.$$

Mean squared error does not have a minimum **in this model.**

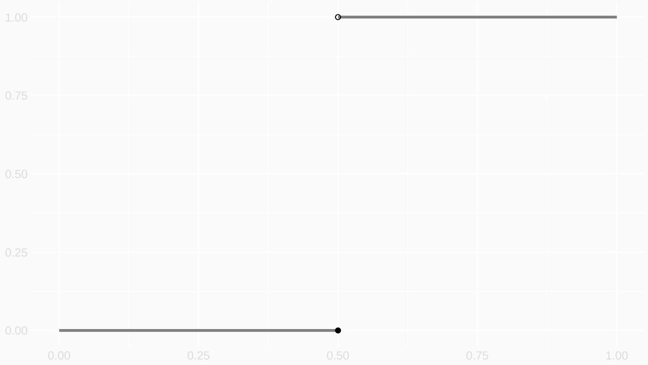
But being pedantic doesn't get us far. We know μ **should** be the solution.
We need a way to define our model so that it can be.
In a sense, we don't need to change much.

$$\hat{\mu} = \operatorname*{argmin}_{m \in \mathcal{M}} \int_0^1 \{\mu(x) - m(x)\}^2 \quad \text{where} \quad \mathcal{M} = \left\{ m : \int_0^1 |m'(x)| \le B \right\}.$$

All we've got to do is define the integral $\int_0^1 |m'(x)|$ for non-differentiable functions.

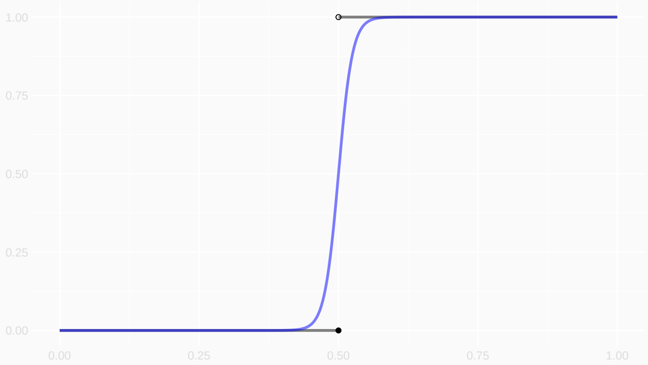
We know what we want from our definition.

$$\int_0^1 |m'(x)|dx = \ ?$$



We know what we want from our definition.

$$\int_0^1 |m'(x)|dx \approx \int_0^1 |m_k'(x)|dx$$



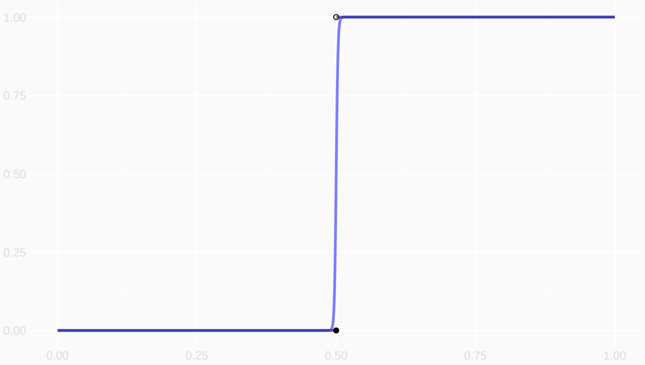
where

$$m_k(x) \approx m(x).$$

The rest is calculus.

We know what we want from our definition.

$$\int_0^1 |m'(x)|dx = \lim_k \int_0^1 |m_k'(x)|dx$$



where

$$m_k(x) \to m(x).$$

The rest is calculus.

If we have a dense partition $0 = x_1 < \ldots < x_k = 1$ in $[0,1]$, then:

$$\begin{aligned}
\int_0^1 |m'(x)| &\approx \sum_{j=1}^{k} \big|m'(x_j)\big|(x_{j+1} - x_j) && \text{a sum approximates our integral} \\
&\approx \sum_{j=1}^{k} \left|\frac{m(x_{j+1}) - m(x_j)}{x_{j+1} - x_j}\right|(x_{j+1} - x_j) && \text{a slope approximates our derivative} \\
&= \sum_{j=1}^{k} |m(x_{j+1}) - m(x_j)| && \text{and we've got no calculus stuff left.}
\end{aligned}$$

If we take denser and denser partitions ($\max_j x_{j+1} - x_j \to 0$), then this is exact.

So we've got a definition.

$$\int_0^1 |m'(x)| dx = \lim \sum_{j=1}^{k} |m(x_{j+1}) - m(x_j)| \quad \text{as} \quad \max_j x_{j+1} - x_j \to 0.$$

$$\int_0^1 |m'(x)|dx = \lim \sum_{j=1}^{k} |m(x_{j+1}) - m(x_j)| \quad \text{as} \quad \max_j x_{j+1} - x_j \to 0.$$

This definition is a bit complicated. It involves as weird kind of limit. You'd have to think about whether that limit exists. So we use another.

$$\int_0^1 |m'(x)|dx = \sup_{\substack{\text{finite partitions} \\ 0=x_1<\ldots<x_k=1}} \sum_{j=1}^{k} |m(x_{j+1}) - m(x_j)|.$$

We've replaced the *limit over denser partitions* with a *supremum over all partitions*.

Is that ok? Are these the same?

$$\sup_{\substack{\text{finite partitions}\\ 0=x_0<\ldots<x_k=1}} \sum_{j=1}^{k} |m(x_{j+1})-m(x_j)| \overset{?}{=} \lim \sum_{j=1}^{k} |m(x_{j+1})-m(x_j)| \text{ as } \max_j x_{j+1}-x_j \to 0.$$

I'm not being totally rigorous here. But yes, they're the same.
The sum can only get bigger if we *refine* our partition by adding intermediate points.

$$\begin{aligned} |m(x_{j+1}) - m(x_j)| &= |m(x_{j+1}) - m(\tilde{x}) + m(\tilde{x}) - m(x_j)| \\ &\leq |m(x_{j+1}) - m(\tilde{x})| + |m(\tilde{x}) - m(x_j)| \quad \text{for} \quad \tilde{x} \in (x_j, x_{j+1}). \end{aligned}$$

That's the triangle inequality in action.
And it means that our supremum is *always* a limit for increasingly dense partitions.

It doesn't however, mean we can't stop at some point.
There may be a finite partition that gives us the maximal sum.
Thinking about these partitions can help us understand our 'integral'.

The Bounded Variation Model

$$\mathcal{M} = \{m : \rho_{TV}(m) \le B\}$$

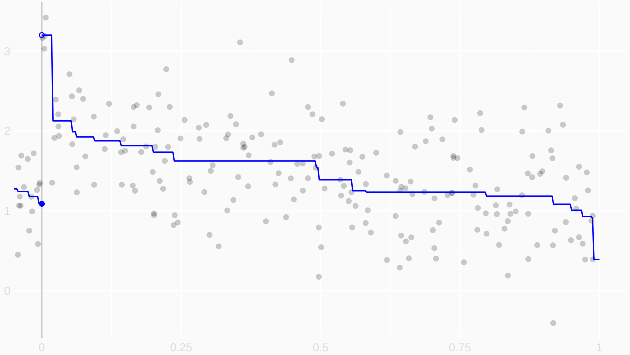
where

$$\begin{aligned} \rho_{TV}(m) &= \int_0^1 |m'(x)|dx && \text{for differentiable } m \\ &= \sup_{\substack{\text{finite partitions} \\ 0=x_0<\ldots<x_k=1}} \sum_{j=1}^{k} |m(x_{j+1}) - m(x_j)| && \text{in complete generality .} \end{aligned}$$

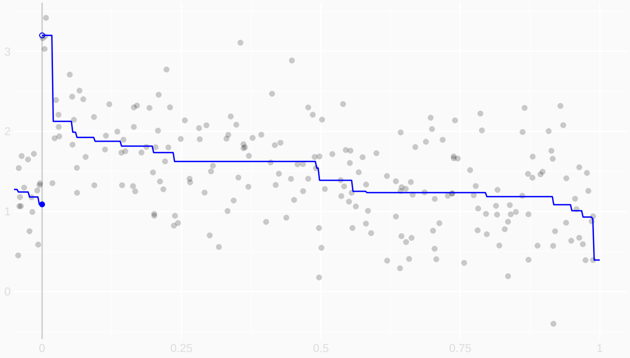
It's a **ball** in the **total variation seminorm** ρ_{TV}.

How do we know that it's a seminorm?

Understanding Total Variation



Here’s a monotone regression estimator $\hat{\mu}$.
What is its total variation $\rho_{TV}(\hat{\mu})$?

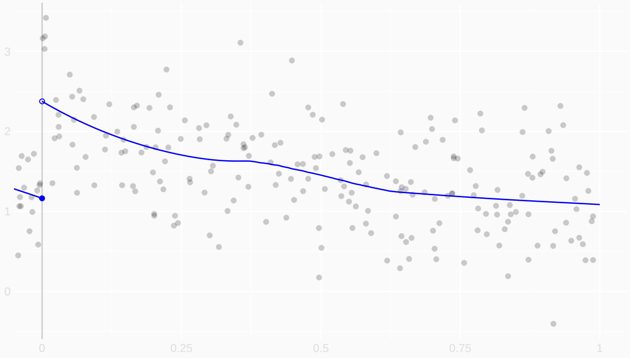


Here's a monotone regression estimator $\hat{\mu}$.
What is its total variation $\rho_{TV}(\hat{\mu})$?

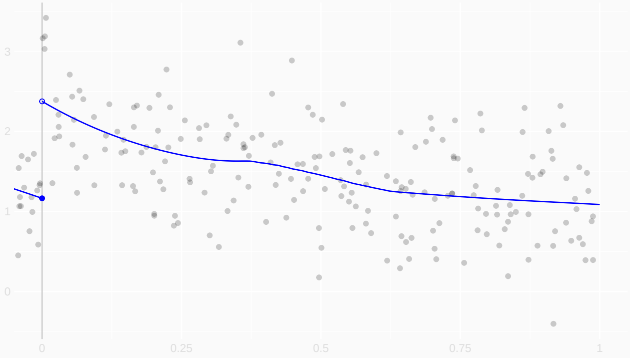
It's $\hat{\mu}(0) - \hat{\mu}(1) \approx 1$.

$$\sum_{j=1}^{k} |\hat{\mu}(x_{j+1}) - \hat{\mu}(x_j)| = \sum_{j=1}^{k} \hat{\mu}(x_j) - \hat{\mu}(x_{j+1}) = \hat{\mu}(x_0) - \hat{\mu}(x_k).$$

For a monotone function, it doesn't matter what partition we sum over.
The sum is always the same.



Here's another estimator $\hat{\mu}$. It's a *local polynomial regression estimator.* What is its total variation $\rho_{TV}(\hat{\mu})$?



Here's another estimator $\hat{\mu}$. It's a *local polynomial regression estimator.*
What is its total variation $\rho_{TV}(\hat{\mu})$?

It's $|\hat{\mu}(0) - \hat{\mu}(x_\star)| + |\hat{\mu}(x_\star) - \hat{\mu}(1)| \approx 1.5$ for $x_\star = \operatorname{argmax}_x \hat{\mu}(x)$.

$$\sum_{j=1}^{k} |\hat{\mu}(x_{j+1}) - \hat{\mu}(x_j)| = \sum_{j=1}^{j_\star - 1} \hat{\mu}(x_{j+1}) - \hat{\mu}(x_j) + \sum_{j=j_\star}^{k} \hat{\mu}(x_j) - \hat{\mu}(x_{j+1})$$

for $x_{j_\star} = x_\star$ because m is increasing on $[0, x_\star]$ and decreasing on $[x_\star, 1]$.

As long as our partition contains all local extrema, the sum is always the same.

Properties

$$\begin{aligned}\rho_{TV}(m) &= |m(1) - m(0)| && \text{if } m \text{ is monotone (increasing or decreasing) on } [0,1] \\ &= \sum_j |m(x_{j+1}) - m(x_j)| && \text{if } m \text{ is monotone on each interval } [x_j, x_{j+1}].\end{aligned}$$

Try these!

What is $\rho_{TV}(m)$ for …

1. $m(x) = x$
2. $m(x) = x^2$
3. $m(x) = e^x$
4. $m(x) = \sin(\pi x)$
5. $m(x) = \begin{cases} 0 & \text{if} \quad x < 1 \\ 1 & \text{if} \quad x = 1 \end{cases}$
6. $m(x) = \sin(1/x)$

We'll look at some more interesting cases in this week's homework.

Why the Bounded Variation Model is Useful

How does a constraint on total variation keep us from overfitting?

Hint.

- A monotonicity constraint helps because noise jumps up and down.
- To fit noise would violate monotonicity.

How does a constraint on total variation keep us from overfitting?

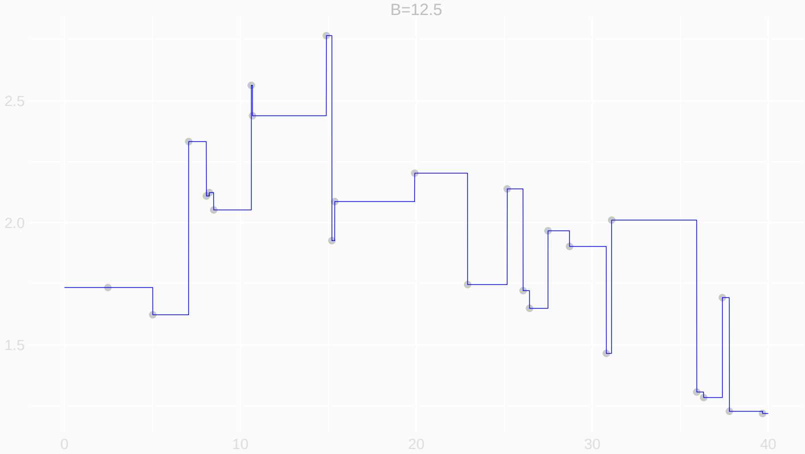
Hint.

- A monotonicity constraint helps because noise jumps up and down.
- To fit noise would violate monotonicity.

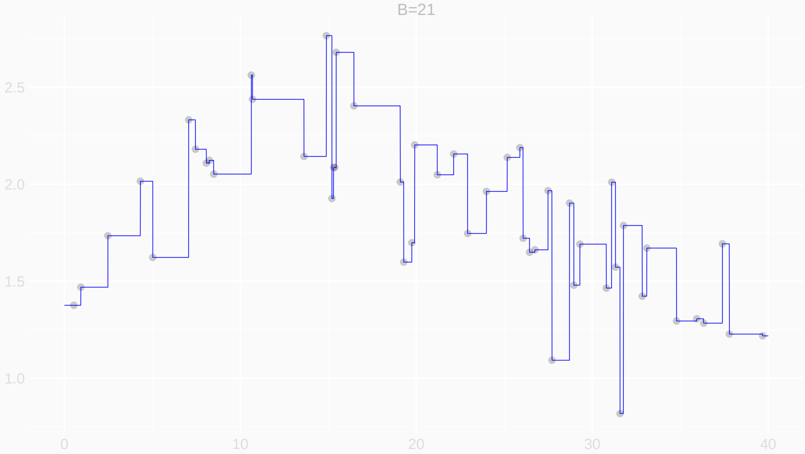
Answer.

- A TV constraint helps because noise jumps up and down *a lot*.
- To fit it, we'd need a curve m with huge total variation $\rho_{TV}(m)$.

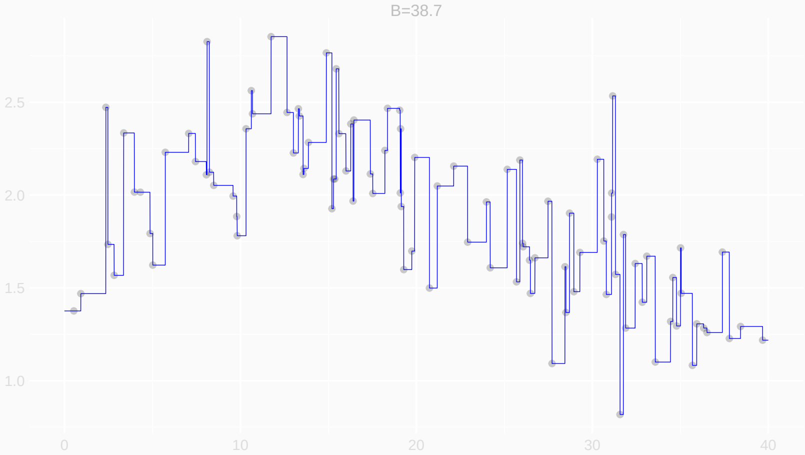
To fit perfectly, we need to increase our variation budget with sample size. Fast.



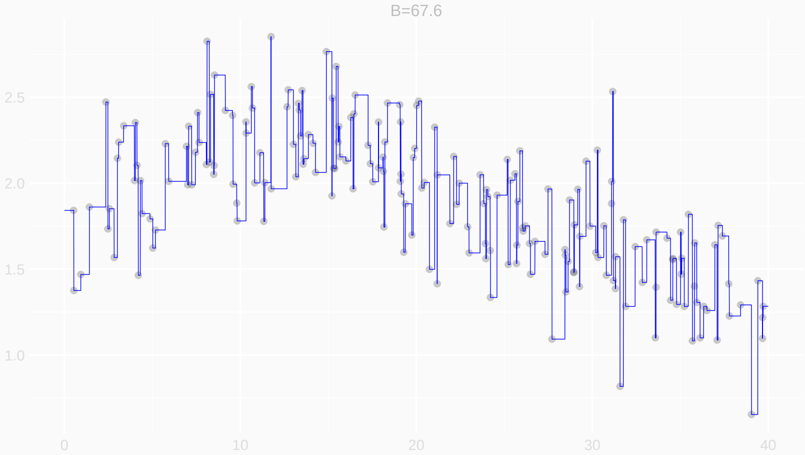
To fit perfectly, we need to increase our variation budget with sample size. Fast.



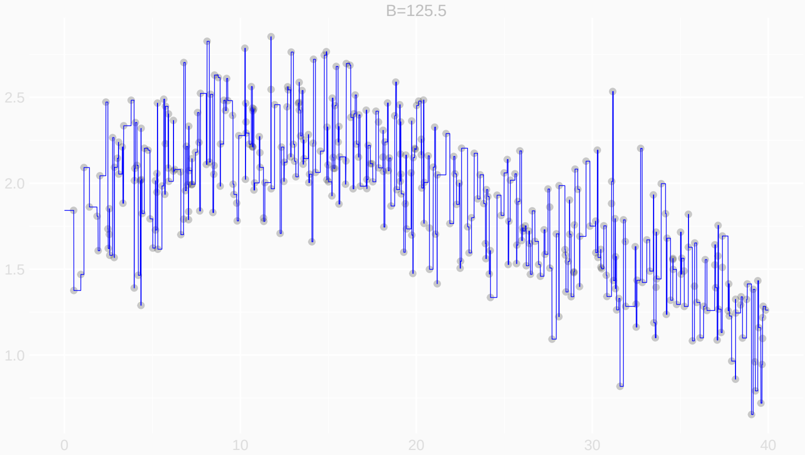
To fit perfectly, we need to increase our variation budget with sample size. Fast.



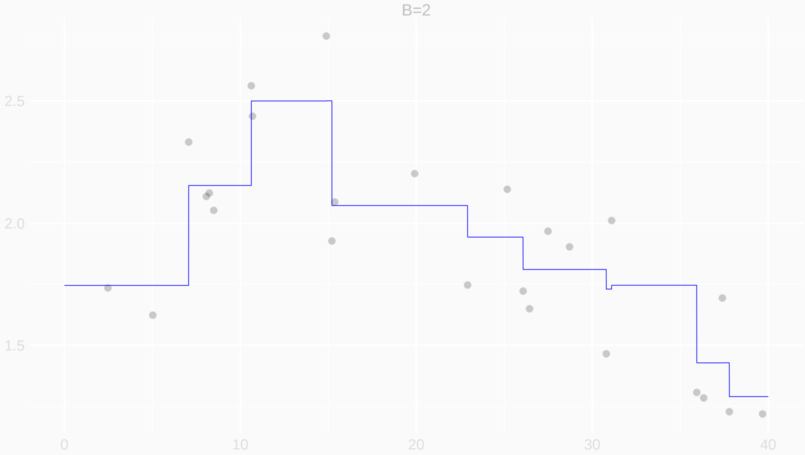
To fit perfectly, we need to increase our variation budget with sample size. Fast.



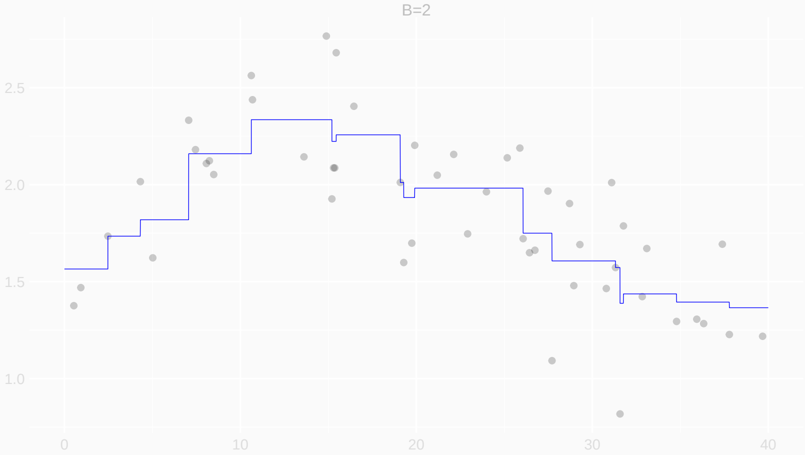
To fit perfectly, we need to increase our variation budget with sample size. Fast.



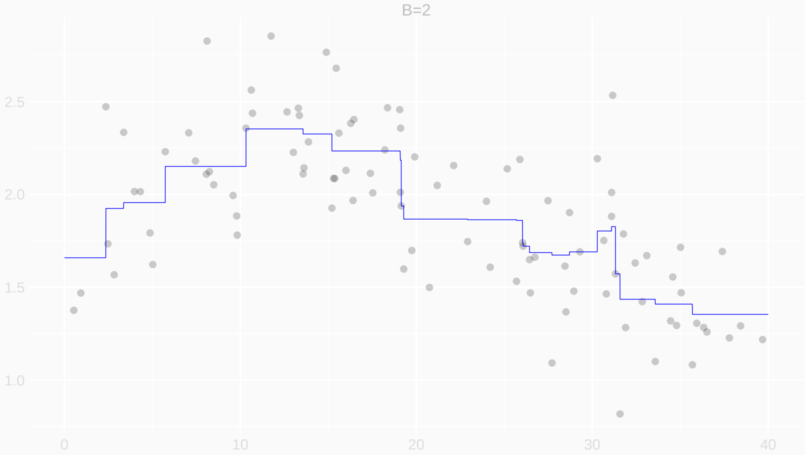
To fit the overall trend, we don't.



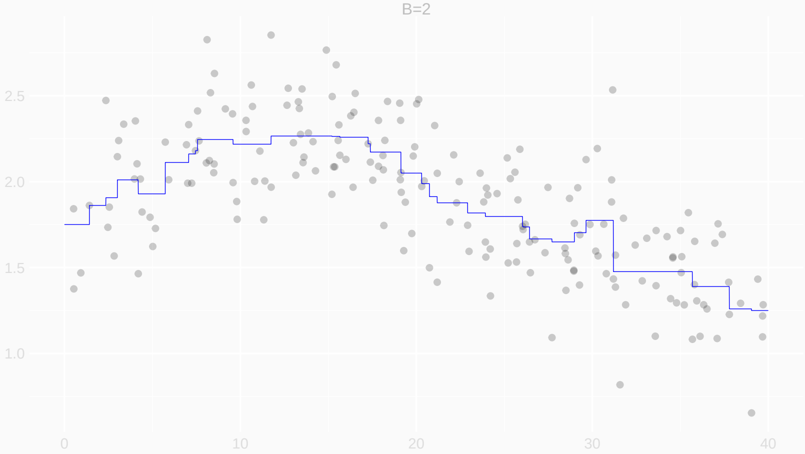
To fit the overall trend, we don’t.



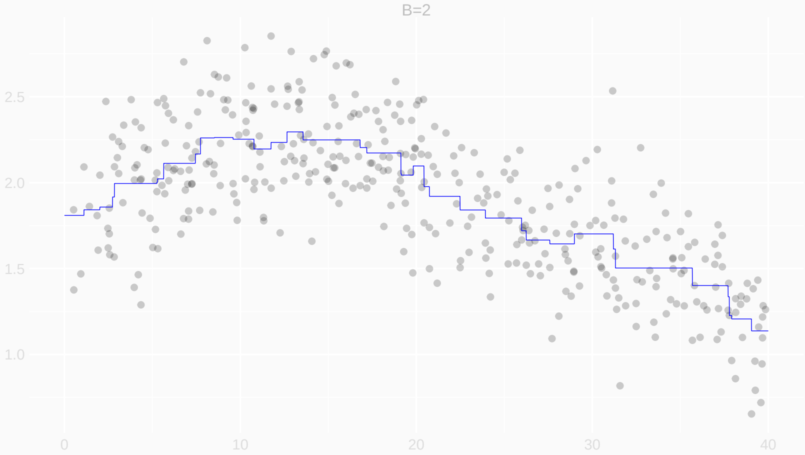
To fit the overall trend, we don’t.



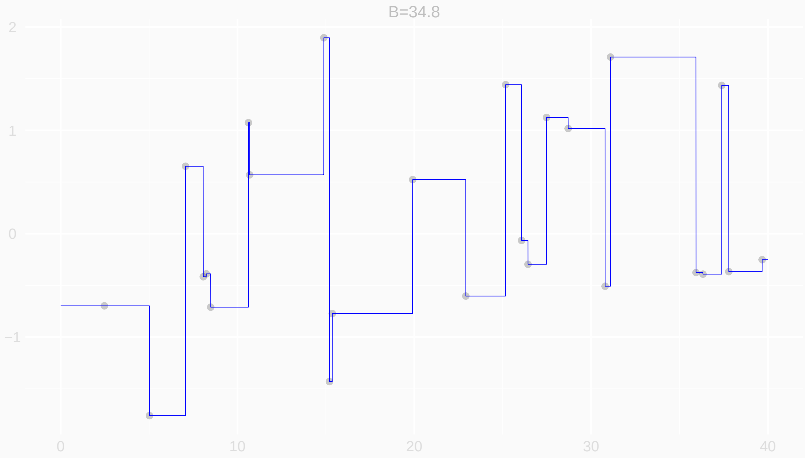
To fit the overall trend, we don’t.



To fit the overall trend, we don’t.

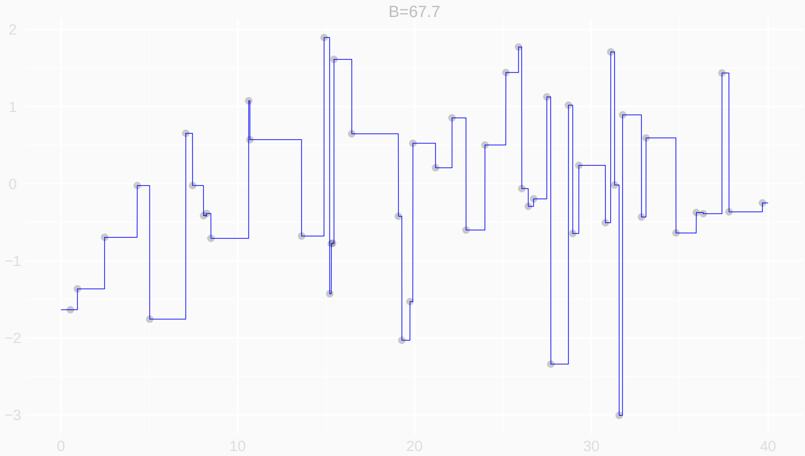


It takes a similar increase to fit pure gaussian noise.



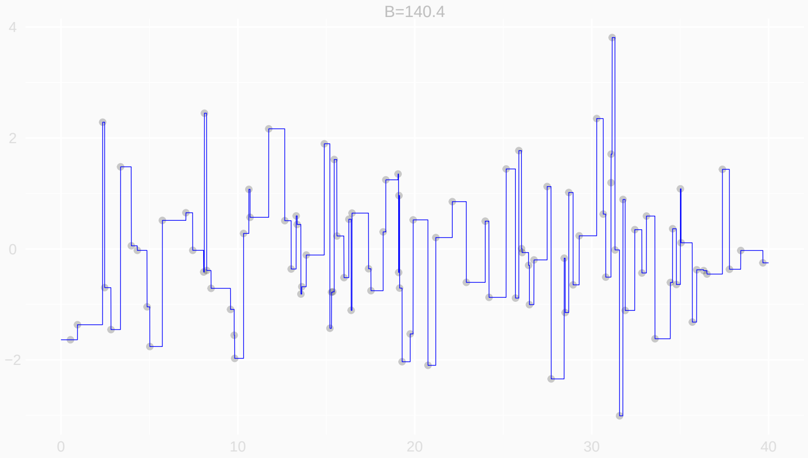
$$Y_i \sim N(0,1)$$

It takes a similar increase to fit pure gaussian noise.



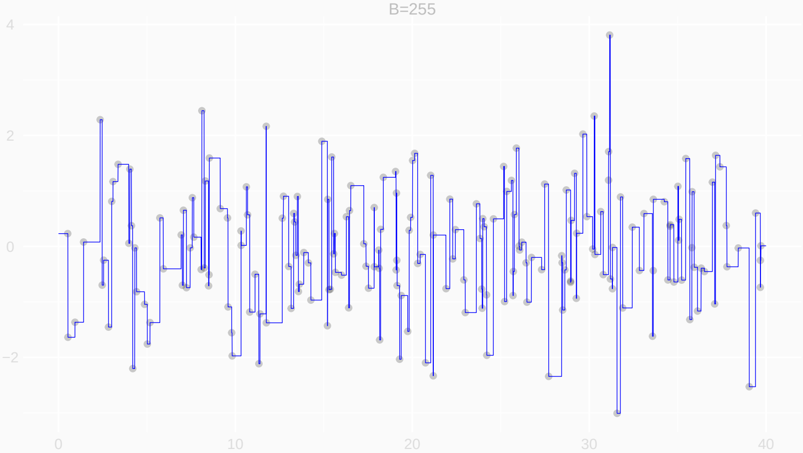
$$Y_i \sim N(0,1)$$

It takes a similar increase to fit pure gaussian noise.



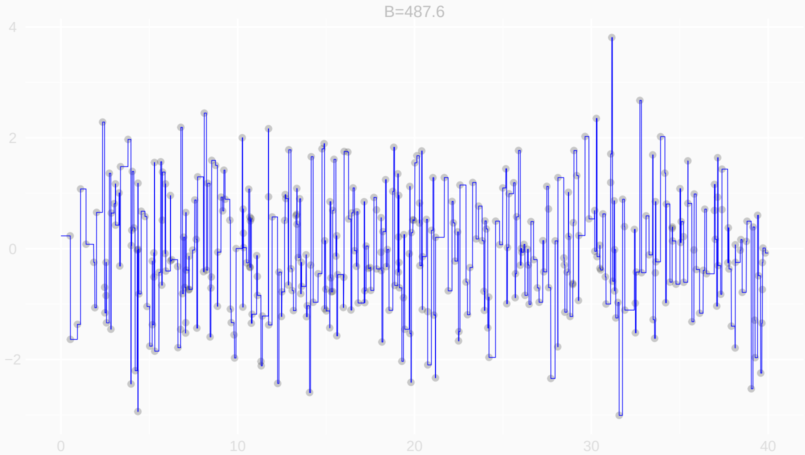
$$Y_i \sim N(0,1)$$

It takes a similar increase to fit pure gaussian noise.



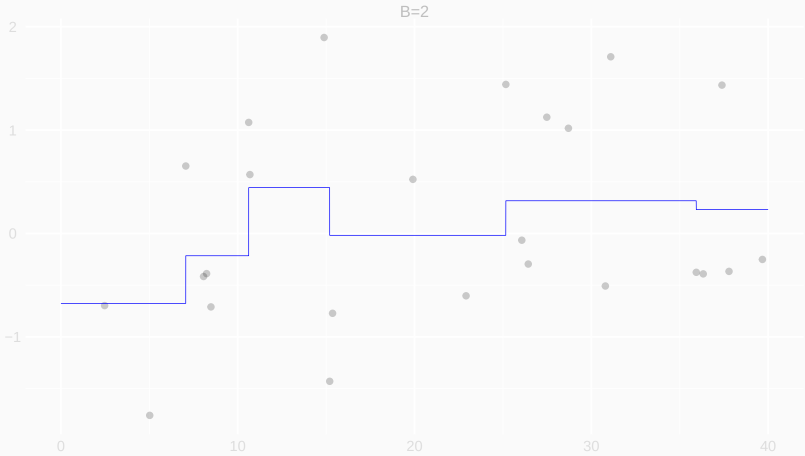
$$Y_i \sim N(0,1)$$

It takes a similar increase to fit pure gaussian noise.



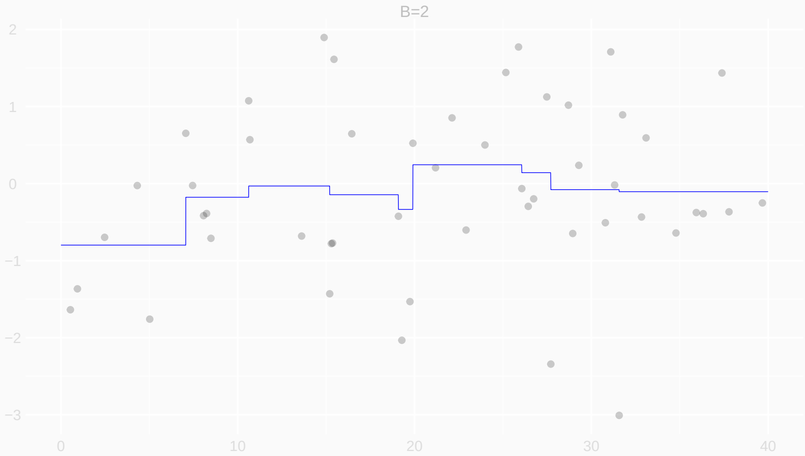
$$Y_i \sim N(0,1)$$

And we can find that there’s no trend if we keep our budget small.



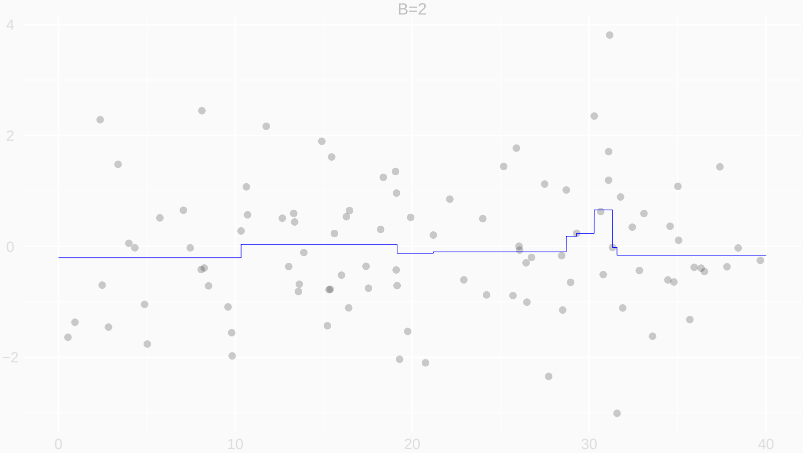
$$Y_i \sim N(0,1)$$

And we can find that there’s no trend if we keep our budget small.



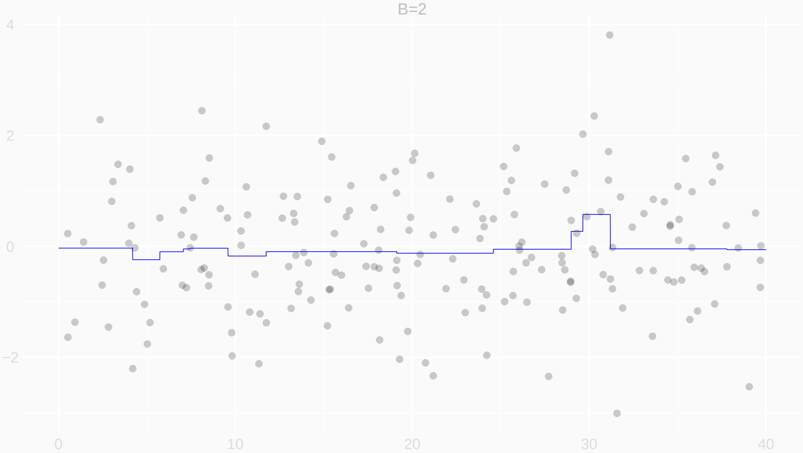
$$Y_i \sim N(0,1)$$

And we can find that there’s no trend if we keep our budget small.



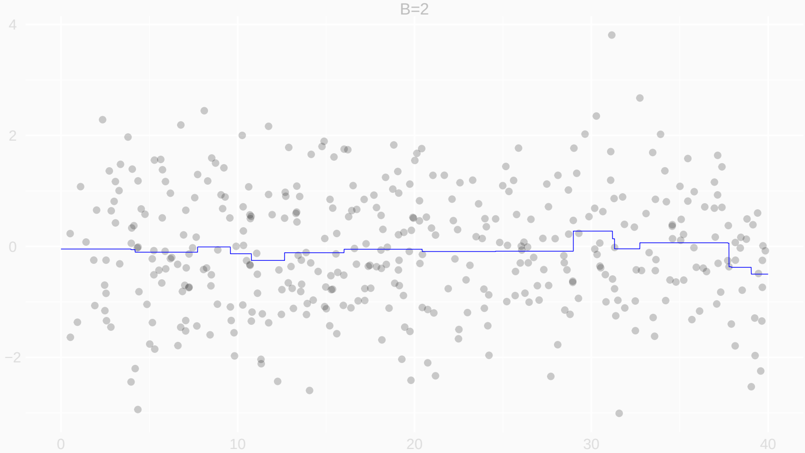
$$Y_i \sim N(0,1)$$

And we can find that there's no trend if we keep our budget small.



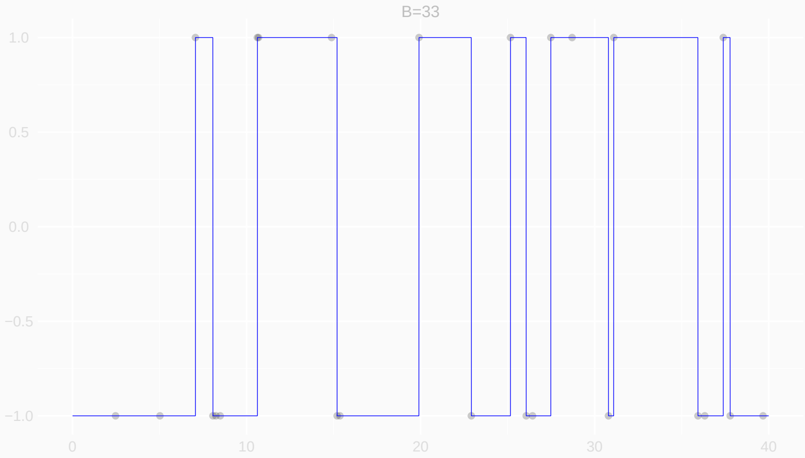
$$Y_i \sim N(0,1)$$

And we can find that there's no trend if we keep our budget small.



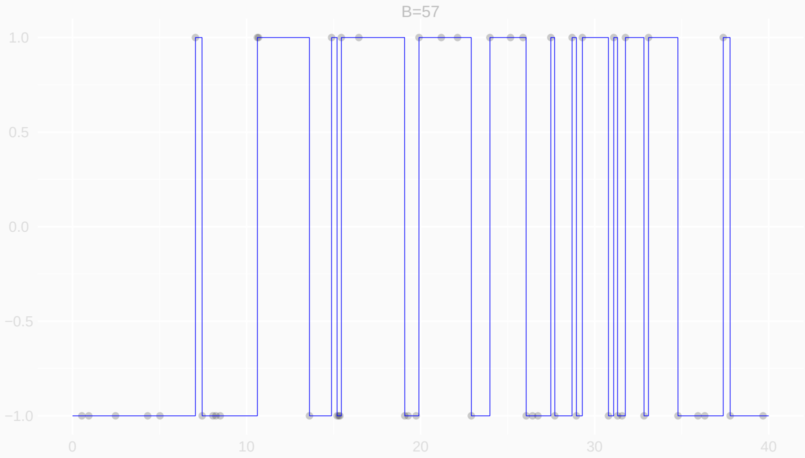
$$Y_i \sim N(0, 1)$$

Same deal for fitting random signs ±1. Coin flips.



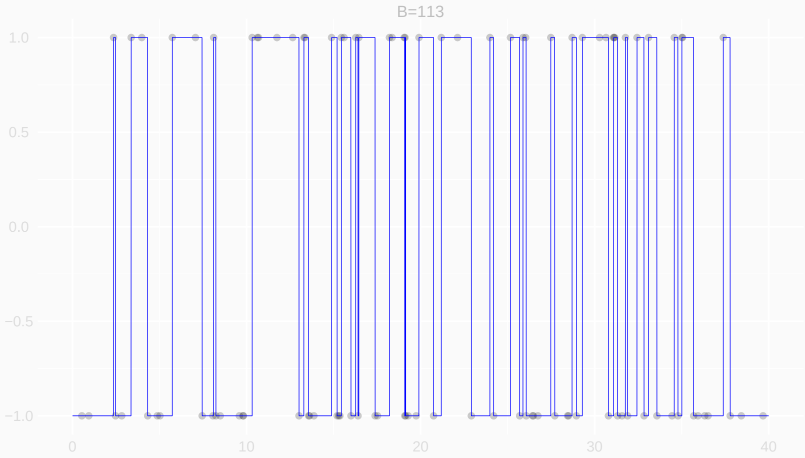
$$Y_i = \pm1 \quad \text{each with probability} \quad 1/2$$

Same deal for fitting random signs ±1. Coin flips.



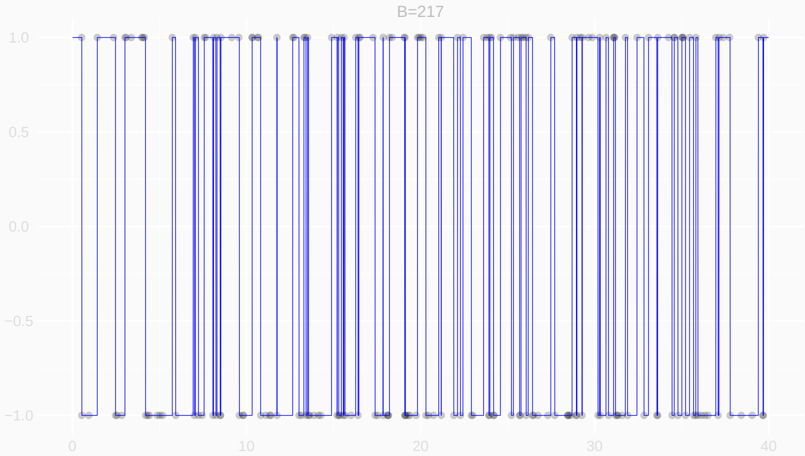
$Y_i = \pm 1$ each with probability $1/2$

Same deal for fitting random signs ± 1. Coin flips.



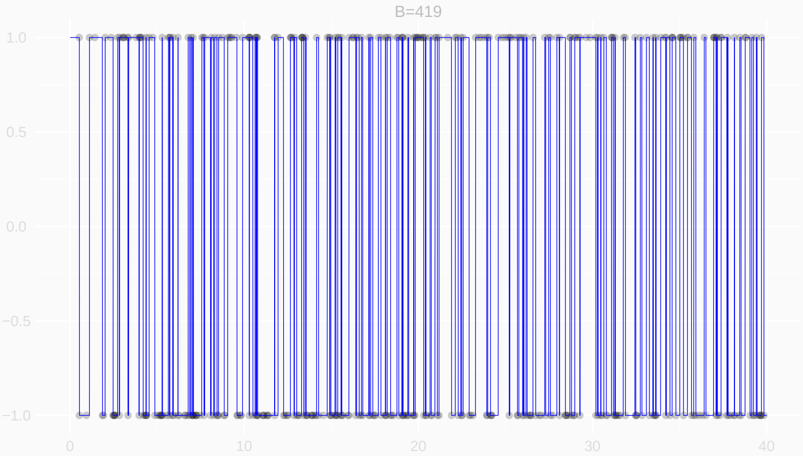
$Y_i = \pm 1$ each with probability $1/2$

Same deal for fitting random signs ±1. Coin flips.



$$Y_i = \pm 1 \quad \text{each with probability} \quad 1/2$$

Same deal for fitting random signs ± 1. Coin flips.



$$Y_i = \pm 1 \quad \text{each with probability} \quad 1/2$$

How big is way too big?

If a curve fits noise perfectly, i.e. if $m(X_i) = \varepsilon_i$, then what do we know?

How big is way too big?

If a curve fits noise perfectly, i.e. if $m(X_i) = \varepsilon_i$, then what do we know?

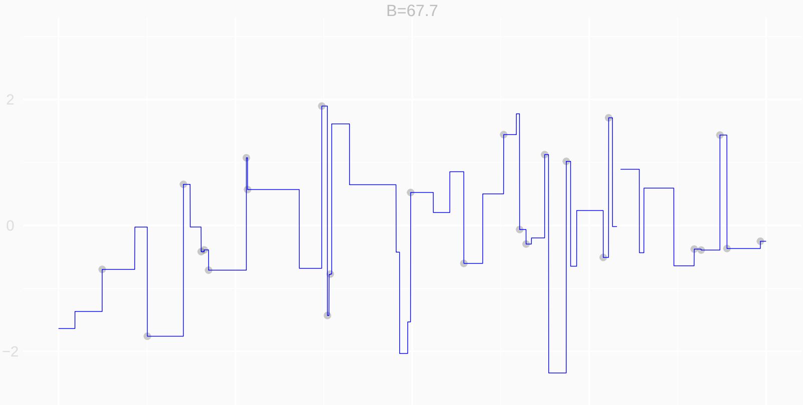
$$\rho_{TV}(m) \geq \sum_{i=1}^{n-1} |m(X_{i+1}) - m(X_i)| = \sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i|.$$

How big is way too big?

If a curve fits noise perfectly, i.e. if $m(X_i) = \varepsilon_i$, then what do we know?

$$\rho_{TV}(m) \geq \sum_{i=1}^{n-1} |m(X_{i+1}) - m(X_i)| = \sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i|.$$

The observations are just one partition, so all this tells us is a lower bound.
We didn't say what this curve does between the observations.



How big is way too big? Coin flip edition.

What if our noise ε_i is a coin flip.

$\varepsilon_i = \pm 1$ each with probability $1/2$.

How much total variation would we need for a perfect fit?
First, translate this into a question about the noise.

How big is way too big? Coin flip edition.

What if our noise ε_i is a coin flip.

$\varepsilon_i = \pm 1$ each with probability $1/2.$

How much total variation would we need for a perfect fit?
First, translate this into a question about the noise.

What is $\sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i|$?

How big is way too big? Coin flip edition.

What if our noise ε_i is a coin flip.

$$\varepsilon_i = \pm 1 \quad \text{each with probability} \quad 1/2.$$

How much total variation would we need for a perfect fit?
First, translate this into a question about the noise.

What is $\sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i|$?

In the worst case, it's $(n-1) \times 2$.

- We'd get that if we flipped sign every time.
- Heads, Tails, Heads, Tails, ...
- $|1 - \ -1| + |-1 - 1| + |1 - -1| + \ldots \quad (n-1$ times).
- Not very random-looking. Not very likely.

How big is way too big? Coin flip edition.

What if our noise ε_i is a coin flip.

$\varepsilon_i = \pm 1$ each with probability $1/2$.

How much total variation would we need for a perfect fit?
First, translate this into a question about the noise.

What is $\sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i|$?

In the average case, it's half that.

$$|\varepsilon_{i+1} - \varepsilon_i| = \begin{cases} 0 & \text{with probability } 1/2 \\ 2 & \text{with probability } 1/2 \end{cases}$$

and therefore

$$\mathrm{E}|\varepsilon_{i+1} - \varepsilon_i| = \frac{0+2}{2} = 1 \quad \text{and} \quad \mathrm{E}\sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i| = (n-1) \times 1.$$

How big is way too big? Gaussian edition.

What if our noise ε_i is standard normal?

$$\text{i.e. what is } \sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i| \quad \text{for} \quad \varepsilon_i \sim N(0,1) \quad ?$$

Is it bigger or smaller?

How big is way too big? Gaussian edition.

What if our noise ε_i is standard normal?

$$\text{i.e. what is } \sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i| \quad \text{for} \quad \varepsilon_i \sim N(0,1) \quad ?$$

Is it bigger or smaller?

In the average case, it's $(n-1) \times \sqrt{2}\,\mathrm{E}|\varepsilon_i| \approx (n-1) \times 1.1$.

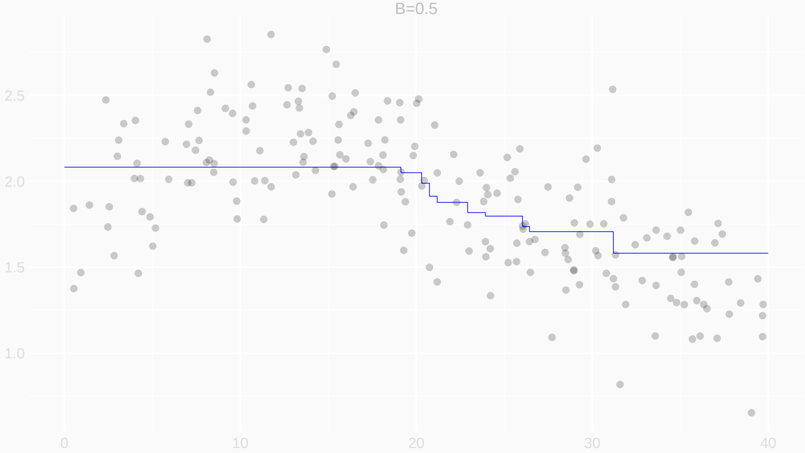
$$|\varepsilon_{i+1} - \varepsilon_i| \sim |N(0,2)| = \sqrt{2}|N(0,1)|$$

and therefore

$$\mathrm{E}|\varepsilon_{i+1} - \varepsilon_i| = \sqrt{2}\,\mathrm{E}|\varepsilon_i| \quad \text{and} \quad \mathrm{E}\sum_{i=1}^{n-1} |\varepsilon_{i+1} - \varepsilon_i| = (n-1) \times \sqrt{2}\,\mathrm{E}|\varepsilon_i|.$$

We don’t tend to use variation budgets anywhere near that big.
If μ were wildly discontinous, we might be a bit stuck even if we knew it exactly.

But we do have to choose a budget somehow.
And it we go too small, that’s not good either.

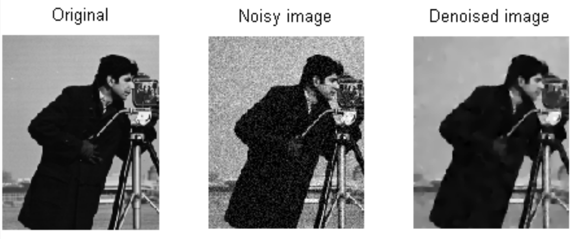


In lab, we’ll talk about how to choose it automatically.

BV Regression in Action.

Maybe you’ve heard of it. Or something like it.

Total Variation Denoising is BV Regression in 2D.



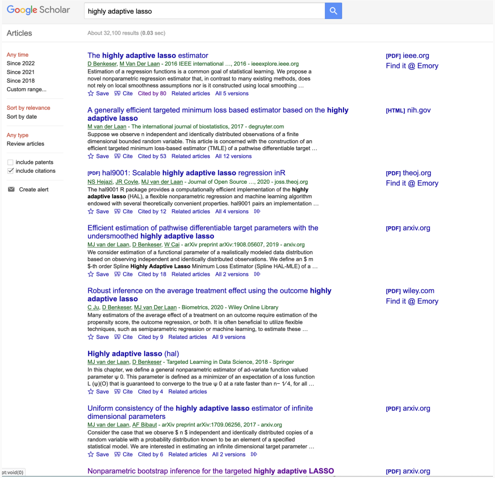
People use it all the time.

This image, by MAL, is licensed under CC BY-SA 3.0.

Total Variation Denoising is BV Regression in 2D.

They used it to capture the first-ever image of a black hole.
We’ll talk about and implement it in a few weeks.

The *Highly Adaptive Lasso* is a generalization of BV regression that works well even with high dimensional data.



About 32,100 results (**0.03** sec)

The **highly adaptive lasso** estimator — [PDF] ieee.org — Find it @ Emory
D Benkeser, M Van Der Laan - 2016 IEEE international …, 2016 - ieeexplore.ieee.org
Estimation of a regression functions is a common goal of statistical learning. We propose a novel nonparametric regression estimator that, in contrast to many existing methods, does not rely on local smoothness assumptions nor is it constructed using local smoothing …
☆ Save Cite Cited by 80 Related articles All 5 versions

A generally efficient targeted minimum loss based estimator based on the **highly adaptive lasso** — [HTML] nih.gov
M van der Laan - The international journal of biostatistics, 2017 - degruyter.com
Suppose we observe n independent and identically distributed observations of a finite dimensional bounded random variable. This article is concerned with the construction of an efficient targeted minimum loss-based estimator (TMLE) of a pathwise differentiable target …
☆ Save Cite Cited by 53 Related articles All 12 versions

[PDF] hal9001: Scalable **highly adaptive lasso** regression inR — [PDF] theoj.org — Find it @ Emory
NS Hejazi, JR Coyle, MJ van der Laan - Journal of Open Source …, 2020 - joss.theoj.org
The hal9001 R package provides a computationally efficient implementation of the **highly adaptive lasso** (HAL), a flexible nonparametric regression and machine learning algorithm endowed with several theoretically convenient properties. hal9001 pairs an implementation …
☆ Save Cite Cited by 12 Related articles All 4 versions

Efficient estimation of pathwise differentiable target parameters with the undersmoothed **highly adaptive lasso** — [PDF] arxiv.org
MJ van der Laan, D Benkeser, W Cai - arXiv preprint arXiv:1908.05607, 2019 - arxiv.org
We consider estimation of a functional parameter of a realistically modeled data distribution based on observing independent and identically distributed observations. We define an $ m $-th order Spline **Highly Adaptive Lasso** Minimum Loss Estimator (Spline HAL-MLE) of a …
☆ Save Cite Cited by 18 Related articles All 2 versions

Robust inference on the average treatment effect using the outcome **highly adaptive lasso** — [PDF] wiley.com — Find it @ Emory
C Ju, D Benkeser, MJ van Der Laan - Biometrics, 2020 - Wiley Online Library
Many estimators of the average effect of a treatment on an outcome require estimation of the propensity score, the outcome regression, or both. It is often beneficial to utilize flexible techniques, such as semiparametric regression or machine learning, to estimate these …
☆ Save Cite Cited by 9 Related articles All 9 versions

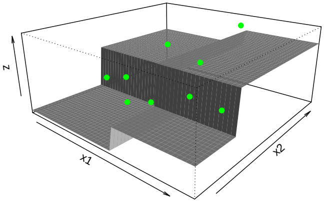
Highly adaptive lasso (hal)
MJ van der Laan, D Benkeser - Targeted Learning in Data Science, 2018 - Springer
In this chapter, we define a general nonparametric estimator of ad-variate function valued parameter ψ 0. This parameter is defined as a minimizer of an expectation of a loss function L (ψ)(O) that is guaranteed to converge to the true ψ 0 at a rate faster than n− 1⁄4, for all …
☆ Save Cite Cited by 4 Related articles

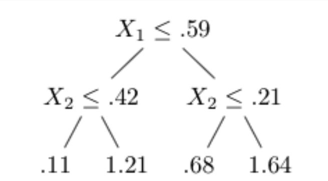
Uniform consistency of the **highly adaptive lasso** estimator of infinite dimensional parameters — [PDF] arxiv.org
MJ van der Laan, AF Bibaut - arXiv preprint arXiv:1709.06256, 2017 - arxiv.org
Consider the case that we observe $ n $ independent and identically distributed copies of a random variable with a probability distribution known to be an element of a specified statistical model. We are interested in estimating an infinite dimensional target parameter …
☆ Save Cite Cited by 6 Related articles All 2 versions

Nonparametric bootstrap inference for the targeted **highly adaptive LASSO** — [PDF] arxiv.org

It's popular with one of the biggest causal inference groups out there. We'll touch on this, too.

Random forests are a computationally-efficient approximation to BV regression.





- Forests, too, are constant except for jumps.
- TV regression uses the jumps that minimize mean squared error subject to a constraint on the curve's total variation $\rho_{TV}(\hat{\mu})$.
- This is fast enough in 1D or 2D, but can get slow high dimensions.
- Random forests use a greedy algorithm that tries to minimize MSE with constraints that are similar in intention and effect.

Variations

There’s more than one way to be a smooth curve.

1. Often, we think the curve we’re looking for shouldn’t jump around all the time.
2. Sometimes, we think it shouldn’t jump at all.
   And that it shouldn’t even get close to making big jumps.
3. Maybe we’ll even think it shouldn’t get steep at all.

These are all different ways to be smooth. Each is stronger than the last.

$$\underset{\text{Lipschitz smooth (3)}}{\max_{x\in[0,1]}|m'(x)| \le B} \implies \underset{\text{Sobolev smooth (2)}}{\sqrt{\int_0^1 |m'(x)|^2} \le B} \implies \underset{\text{Bounded Variation (1)}}{\int_0^1 |m'(x)| \le B}\ .$$

In many ways these kinds of models are similar, but they all have their quirks.
Which you use makes a difference, especially for behavior near the data’s edges.

- We’ll start talking about Lipschitz smoothness in this week’s homework.
- And we’ll add Sobolev smoothness to the mix later.

References

Joshua D Angrist and Victor Lavy. Using maimonides' rule to estimate the effect of class size on scholastic achievement. *The Quarterly journal of economics*, 114(2): 533–575, 1999.