# Big Data + Bayesian Nonparametrics

**Matt Taddy**

Microsoft Research and University of Chicago

http://taddylab.com

## Big Data

The sample sizes are enormous.

- We'll see 21 and 200 million today.
- Data can't fit in memory, or even storage, on a single machine.

The data are super weird.

- Internet transaction data distributions have a big spike at zero and spikes at other discrete values (e.g., 1 or $99).
- Big tails (e.g., $12 mil/month eBay user spend) that matter.
- The potential feature space is unmanageably large.
- We cannot write down or measure believable models.

Both 'Big' and 'Strange' beg for nonparametrics.

## Bayesian nonparametrics

In usual BNP you *model* a complex generative process with flexible priors, then apply that model directly in prediction and inference.

$$\text{e.g.,} \qquad y = f(\mathbf{x}) + \varepsilon \qquad \text{or} \qquad f(y|\mathbf{x}) = f(\mathbf{x}, y)/f(\mathbf{x})$$

'Flexibility' comes from a huge number of variables,

$$\text{e.g.,} \qquad f(\mathbf{z}) = \sum_{k=1}^{\infty} \omega_k \mathrm{N}(\mathbf{z};\ \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

and learning requires integration over uncertainty on nuisance parameters, e.g., $k_i$ component membership for observation $\mathbf{z}_i$.

For years this integration was done via MCMC. But these algorithms did not scale for big data. Can we do scalable BNP?

## Classical distribution-free nonparametrics

Frequentists are great at finding simple procedures (e.g. $[\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'y$) and describing how they work under a variety of possible true DGP.

( DGP = Data Generating Process )

This is 'distribution free' nonparametrics.

1: Find some statistic that is useful regardless of DGP.

2: Derive the distribution for this stat under minimal assumptions.

Practitioners apply the simple stat and feel happy that it will work.

## Distribution-free Bayesian nonparametrics

Find some *statistic of the DGP* that you care about:

- derive from first principles, e.g. moment conditions
- *an algorithm* that we know works, e.g. CART
- think about geometric projections, e.g. OLS

Call this statistic $\boldsymbol{\theta}(g)$ where $g(\mathbf{z})$ is the DGP (e.g., for $\mathbf{z} = [\mathbf{x}, y]$).

Then you write down a flexible model for the DGP $g$, and study properties of the posterior on $\boldsymbol{\theta}(g)$ induced by the posterior over $g$.

## A flexible model for the DGP

Say $\mathbf{z} = [\mathbf{x}, y]$ is a single *independent* data point.

Each data point assumes one of a *finite* number of possible values, $[\zeta_1 \dots \zeta_L]$, with probabilities proportional to $[\theta_1 \dots \theta_L]$.

$$g(\mathbf{z}) = \frac{1}{|\boldsymbol{\theta}|} \sum_{l=1}^{L} \theta_l \mathbb{1}_{[\mathbf{z}=\zeta_l]}$$

We complete specification with a conjugate prior on the weights:

$$\frac{\boldsymbol{\theta}}{|\boldsymbol{\theta}|} \sim \mathrm{Dir}(a) \propto \frac{1}{|\boldsymbol{\theta}|^{L(a-1)}} \prod_l \theta_l^{a-1} \quad \text{where} \quad a, \theta_l > 0.$$

This is the Dirichlet-multinomial sampling model (Ferguson 1973).

Now you've observed some data, say $\mathbf{Z} = \{\mathbf{z}_1 \dots \mathbf{z}_n\}$.
(say every $\mathbf{z}_i = [\mathbf{x}_i, y_i]$ is unique).

The posterior over weights has $\theta_l \overset{ind}{\sim} \mathrm{Exp}\left(a + \mathbb{1}_{[\boldsymbol{\zeta}_l \in \mathbf{Z}]}\right)$.

**A convenient limiting case**

$a \to 0$ leads to $\mathrm{p}(\theta_l = 0) = 1$ for $\boldsymbol{\zeta}_l \notin \mathbf{Z}$.

In this case, we can focus on only the *observed support* and write the posterior for our DGP

$$g(\mathbf{z}) = \tfrac{1}{|\boldsymbol{\theta}|} \sum_{i=1}^{n} \theta_i \mathbb{1}[\mathbf{z} = \mathbf{z}_i], \quad \theta_i \overset{iid}{\sim} \mathrm{Exp}(1).$$

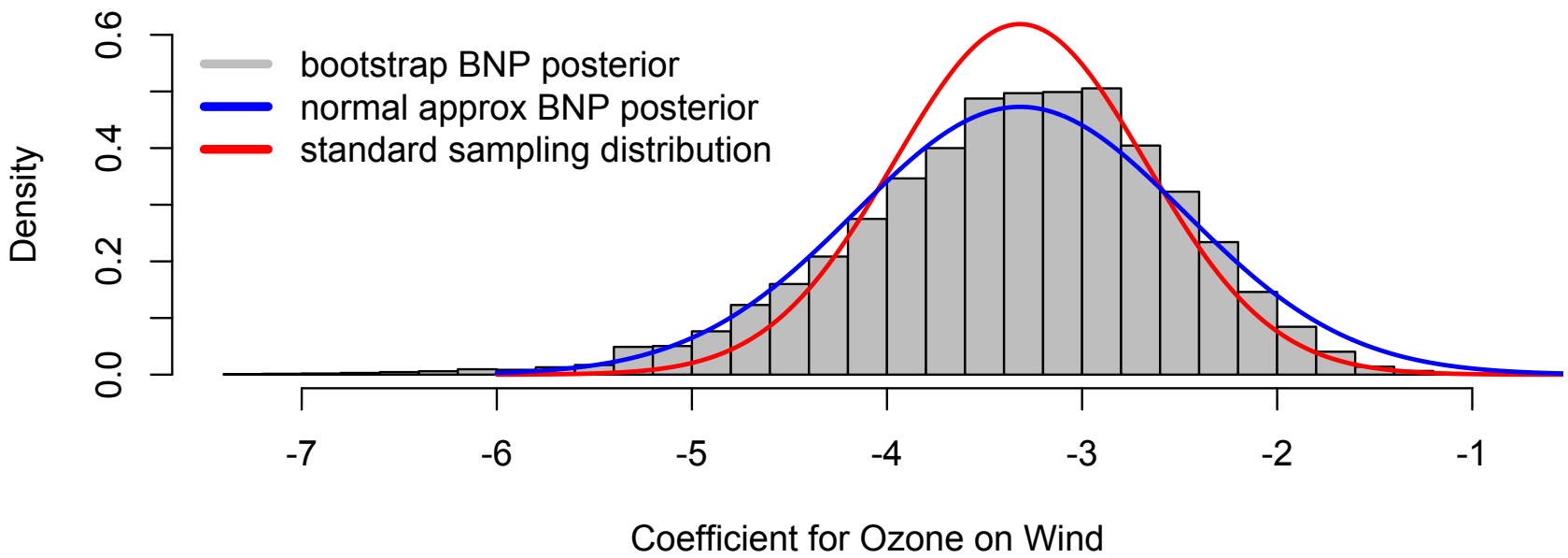
This is just the Bayesian bootstrap. (Rubin 1981)

## Example: **Ordinary Least Squares**

*Population* OLS is a posterior functional

$$\beta = (\mathbf{X}'\mathbf{\Theta}\mathbf{X})^{-1}\mathbf{X}'\mathbf{\Theta}\mathbf{y}$$

where $\mathbf{\Theta} = \mathrm{diag}(\boldsymbol{\theta})$. This is a random variable. (sample via BB)



## What is the blue line?

BB sampling is great, but analytic approximations are also useful.

Consider a first-order Taylor series approximation,

$$\tilde{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}} + \nabla \boldsymbol{\beta}\big|_{\boldsymbol{\theta}=\mathbf{1}}(\boldsymbol{\theta} - \mathbf{1})$$

where $\hat{\boldsymbol{\beta}} = [\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'y = \boldsymbol{\beta}\big|_{\boldsymbol{\theta}=\mathbf{1}}$ is the sample OLS line.

We can derive *exact* posterior moments for $\tilde{\boldsymbol{\beta}}$. e.g.,

$$\mathrm{var}(\boldsymbol{\beta}) \approx \mathrm{var}(\tilde{\boldsymbol{\beta}}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathrm{diag}(\mathbf{e}^2)\mathbf{X}'(\mathbf{X}'\mathbf{X})^{-1}$$

where $\mathbf{e} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$ are the sample OLS residuals.

This is the economist's Huber-White 'Sandwich' variance formula.

(Lancaster 2003 or Poirier 2011.)

## Example: **Heterogeneous Treatment Effects**

eBay runs lots of experiments: they make changes to the marketplace (website) for random samples of users.

Every experiment has response $y$ and treatment $d$ [0/1].
In our illustrative example, $d_i = 1$ for bigger pictures in my eBay.

$i \in$ control : $d_i = 0$


$i \in$ treatment : $d_i = 1$


This is a typical 'A/B experiment'.

What is 'heterogeneity in treatment effects'? (HTE)

Different units [people, devices] respond differently to some treatment you apply [change to website, marketing, policy].

I imagine it exists.

We know $\mathbf{x}_i$ about user $i$. About 400 features in our example.

- Their previous spend, items bought, items sold...
- Page view counts, items watched, searches, ...
- All of the above, broken out by product, fixed v. auction, ...

Can we accurately measure heterogeneity: index it on $\mathbf{x}$?

## Treatment effect regression

Potential outcomes: $v_i(d)$ is the response for user $i$ if $d_i = d$.

We'd love to model $\mathbb{E}[v_i(1) - v_i(0)|\mathbf{x}_i]$, but we can't directly.

Since $d \perp\!\!\!\perp [\mathbf{x}, \boldsymbol{v}]$, a good alternative is to consider

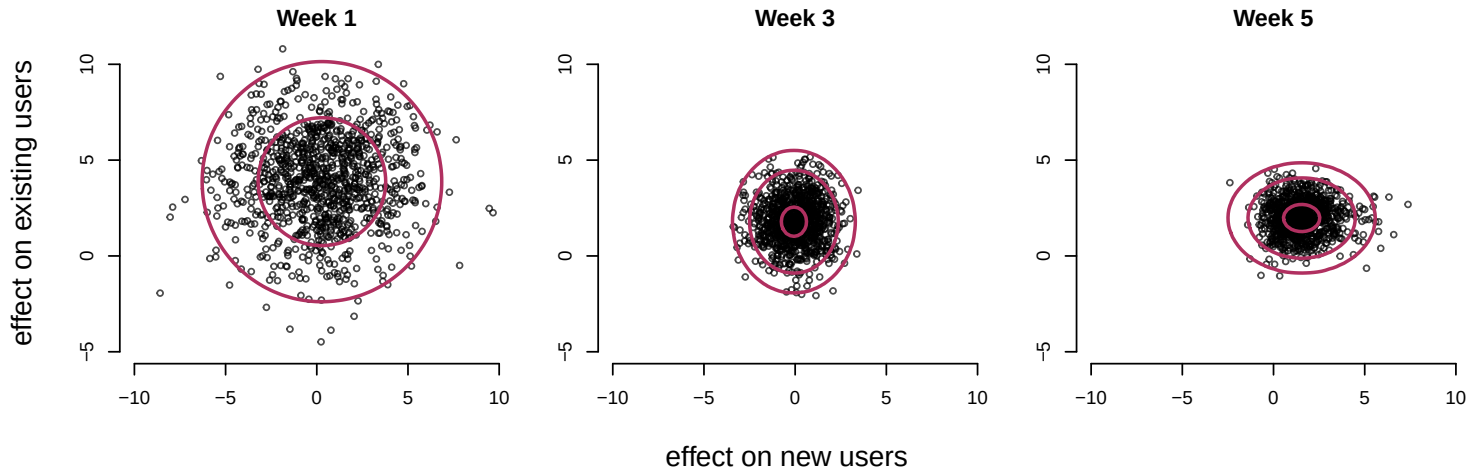
$$\mathbb{E}[v_i(1)|d_i = 1, \mathbf{x}_i] - \mathbb{E}[v_i(0)|d_i = 0, \mathbf{x}_i].$$

Even simpler, just diff the OLS projections for each group

$$\boldsymbol{\beta}_\mathsf{t} - \boldsymbol{\beta}_\mathsf{c} = (\mathbf{X}_\mathsf{t}'\boldsymbol{\Theta}_\mathsf{t}\mathbf{X}_\mathsf{t})^{-1}\mathbf{X}_\mathsf{t}'\boldsymbol{\Theta}_\mathsf{t}\mathbf{y}_\mathsf{t} - (\mathbf{X}_\mathsf{c}'\boldsymbol{\Theta}_\mathsf{c}\mathbf{X}_\mathsf{c})^{-1}\mathbf{X}_\mathsf{c}'\boldsymbol{\Theta}_\mathsf{c}\mathbf{y}_\mathsf{c}$$

e.g., coefficient on *[made purchase this year]* vs an intercept:

*million users:* *7.45* *10.97* *13.22*




Sample is from posterior, contours are normal approximation.

The most basic application of an A/B experiment is estimating the average treatment effect,

$$\text{ATE} = \mathbb{E}[\upsilon(\text{t})] - \mathbb{E}[\upsilon(\text{c})]$$

The usual estimator is $\bar{y}_\text{t} - \bar{y}_\text{c}$. But many authors advocate instead the 'regression adjusted' $\bar{\mathbf{x}}'(\hat{\boldsymbol{\beta}}_\text{t} - \hat{\boldsymbol{\beta}}_\text{c})$ (Lin, Berk+, Deng+, 2013).

Both are [frequentist] unbiased.

But quantifying uncertainty about the regression adjustment, and any variance reduction it provides, is tough without assumptions.

From our BNP persepective, both the usual and adjusted ATE estimators are random functions of the DGP.

$$\begin{aligned}
\text{ATE}_g^{\text{obs}} &= \frac{1}{|\boldsymbol{\theta}_\text{t}|}\boldsymbol{\theta}_\text{t}'\mathbf{y}_\text{t} - \frac{1}{|\boldsymbol{\theta}_\text{c}|}\boldsymbol{\theta}_\text{c}'\mathbf{y}_\text{c} \\
\text{ATE}_g^{\text{lin}} &= \frac{1}{|\boldsymbol{\theta}|}\boldsymbol{\theta}'\mathbf{X}\left((\mathbf{X}_\text{t}'\boldsymbol{\Theta}_\text{t}\mathbf{X}_\text{t})^{-1}\mathbf{X}_\text{t}'\boldsymbol{\Theta}_\text{t}\mathbf{y}_\text{t} - (\mathbf{X}_\text{c}'\boldsymbol{\Theta}_\text{c}\mathbf{X}_\text{c})^{-1}\mathbf{X}_\text{c}'\boldsymbol{\Theta}_\text{c}\mathbf{y}_\text{c}\right).
\end{aligned}$$

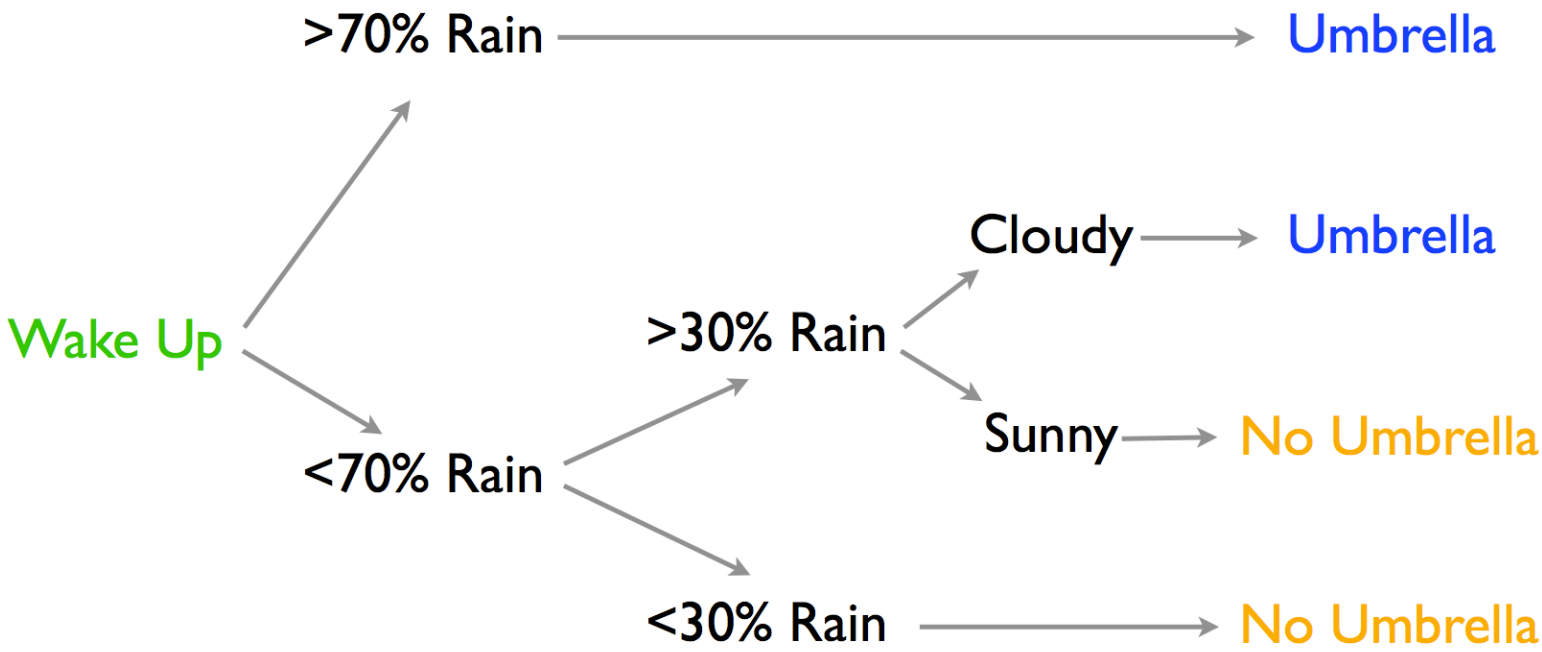
Using our usual tricks, we can resolve any *controversy*:

$$\text{var}\left(\text{ATE}_g^{\text{lin}}\right) \approx \text{var}(\text{ATE}_g^{\text{obs}}) - \left(\frac{R_\text{t}^2 s_{\mathbf{y}_\text{t}}^2}{n_\text{t}^2} + \frac{R_\text{c}^2 s_{\mathbf{y}_\text{c}}^2}{n_\text{c}^2}\right),$$

so that big variance reduction comes only for small sample $n_\text{d}$ or large $R_\text{d}^2 = \text{cor}(y_\text{d}, \hat{y}_\text{d})$. In digital experiments we have big samples and low signal, so we shouldn't expect much variance reduction.

## Example: **Decision Trees**

Trees are great: nonlinearity, deep interactions, heteroskedasticity.




The 'optimal' decision tree is a statistic we care about (s.w.c.a).

## CART: greedy growing with optimal splits

Given node $\{\mathbf{x}_i, y_i\}_{i=1}^n$ and DGP weights $\boldsymbol{\theta}$, find $x$ to minimize

$$|\boldsymbol{\theta}|\sigma^2(x, \boldsymbol{\theta}) = \sum_{k \in \text{left}(x)} \theta_k (y_k - \mu_{\text{left}(x)})^2 + \sum_{k \in \text{right}(x)} \theta_k (y_k - \mu_{\text{right}(x)})^2$$

for a regression tree. Classification impurity can be Gini, etc.

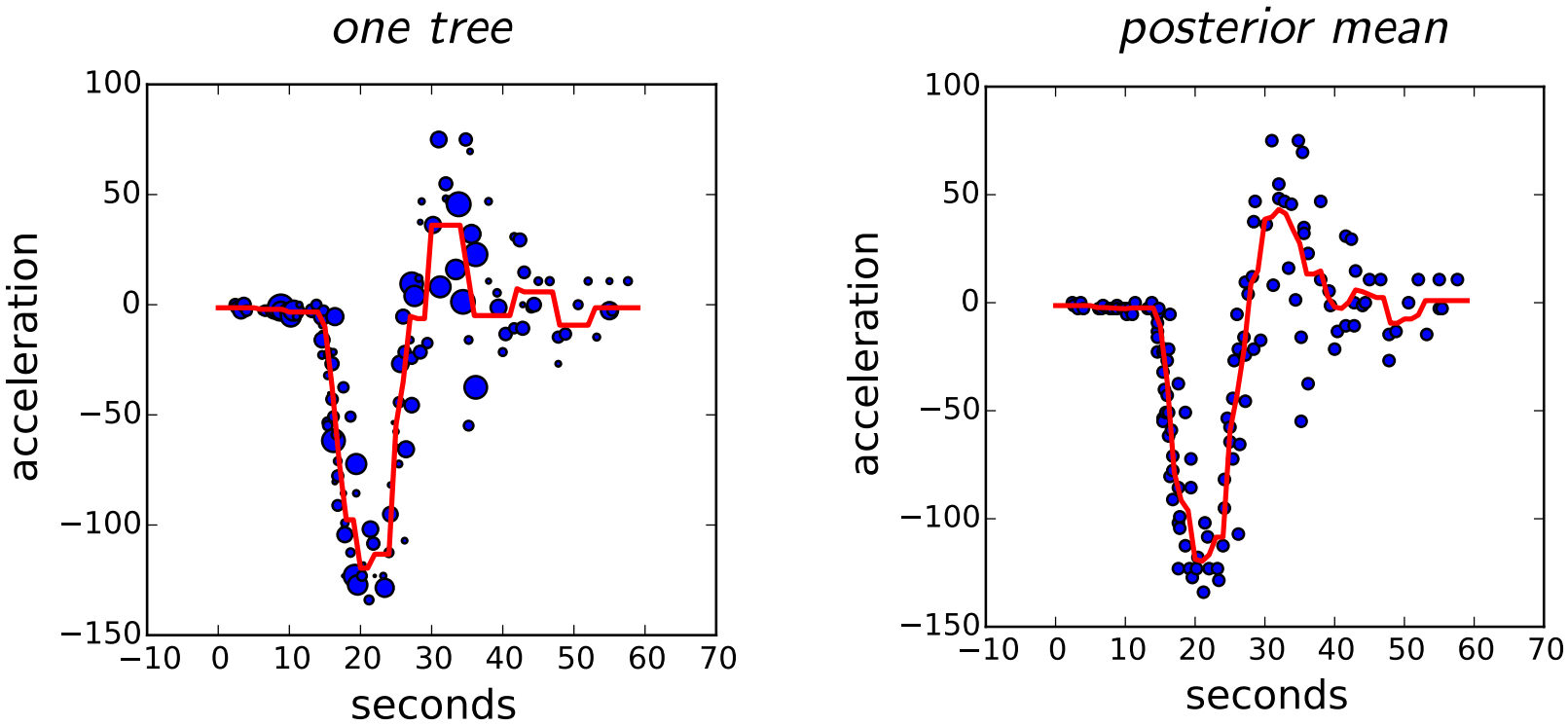
Population-CART might be a statistic we care about.

Or, in settings where greedy CART would do poorly (big $p$), a randomized splitting algorithm might be a better s.w.c.a.

## Bayesian Forests: a posterior for CART trees

For $b = 1 \ldots B$:

- draw $\boldsymbol{\theta}^b \overset{iid}{\sim} \mathrm{Exp}(\mathbf{1})$
- run weighted-sample CART to get $\mathcal{T}_b = \mathcal{T}(\boldsymbol{\theta}^b)$




Random Forest $\approx$ Bayesian forest $\approx$ posterior over CART fits.

## Theoretical trunk stability

Given forests as a posterior, we can start talking about *variance*.

Consider the first-order approximation

$$\begin{aligned}\sigma^2(x, \boldsymbol{\theta}) \approx \quad & \sigma^2(x, \mathbf{1}) + \nabla\sigma^2\big|_{\boldsymbol{\theta}=\mathbf{1}}(\boldsymbol{\theta} - \mathbf{1}) \\ & = \tfrac{1}{n}\textstyle\sum_i \theta_i \left[y_i - \bar{y}_i(x)\right]^2\end{aligned}$$

with $\bar{y}_i(x)$ the sample mean in $i$'s node when splitting on $x$.

Based on this approx, we can say that for data at a given node,

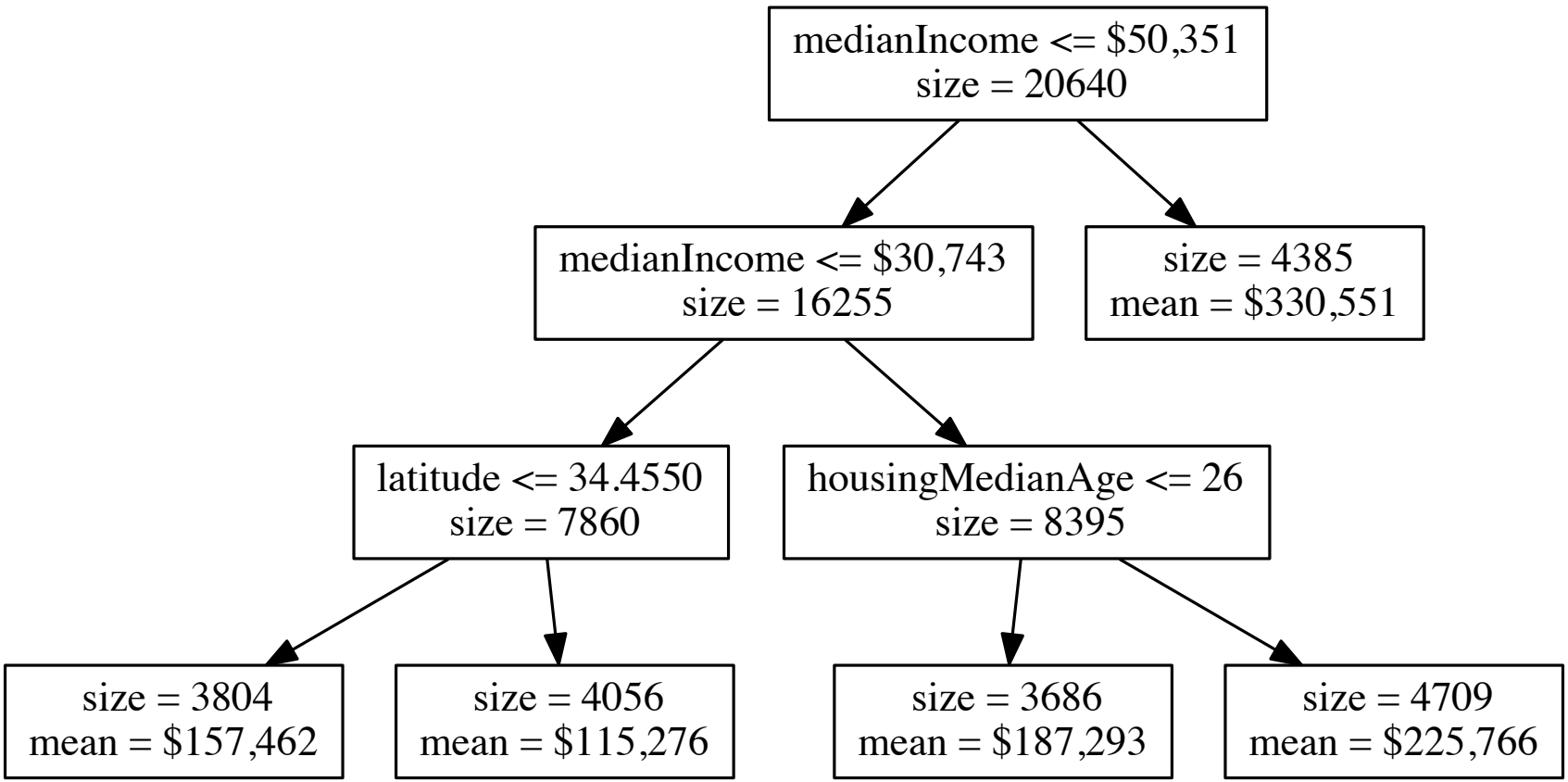
$$\mathrm{p}\left(\text{optimal split matches sample CART}\right) \gtrsim 1 - \frac{p}{\sqrt{n}} e^{-n},$$

with $p$ split locations and $n$ observations.

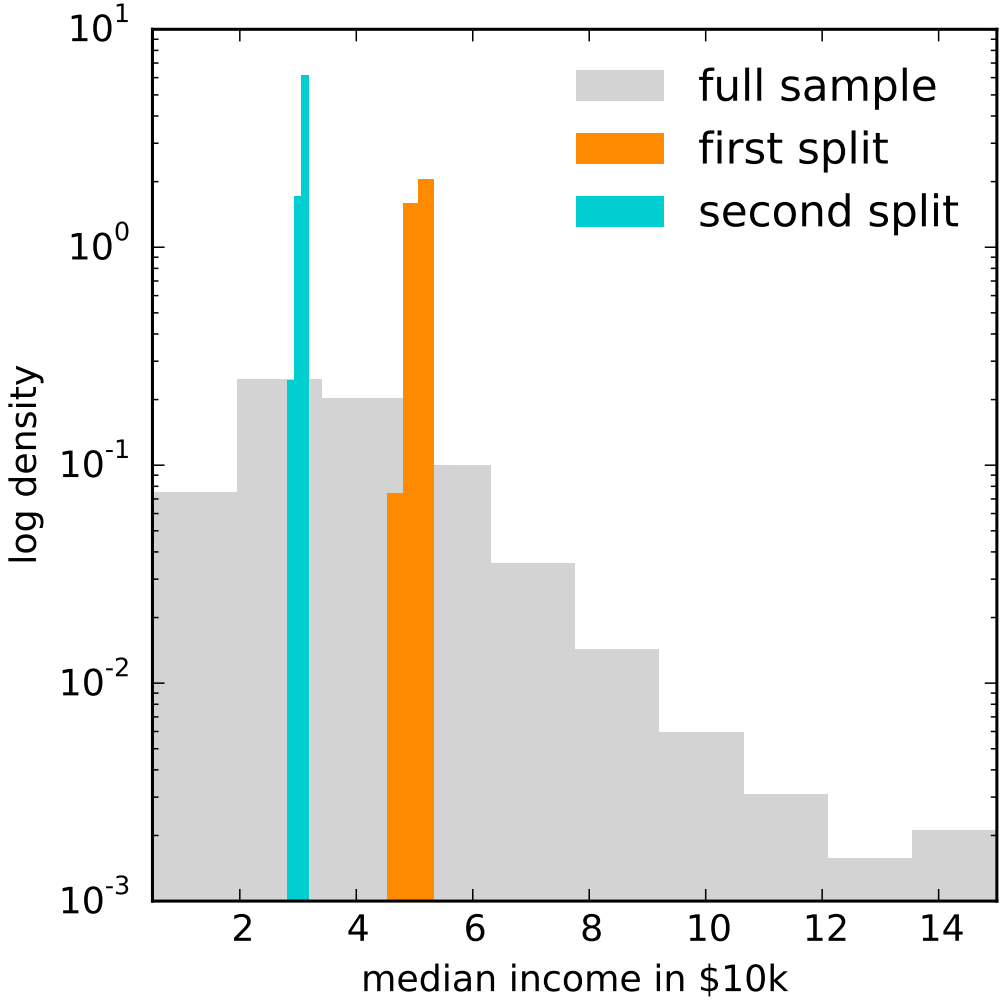
(Taddy, Chen, Yu, Wyle 2015)

## California Housing Data

20k observations on median home prices in zip codes.




Above is the trunk you get setting min-leaf-size of 3500.




- sample tree occurs 62% of the time.
- 90% of trees split on income twice, and then latitude.
- 100% of trees have 1st 2 splits on median income.

Empirically and theoretically: trees are stable, at the trunk.

## Random Forest bottlenecks

RFs (or BFs) are awesome, but when the data are too big to fit in memory or on a single machine they get extremely expensive.
(e.g., Google PLANET, Panda 2009).

A common solution is a 'sub-sampling forest': instead of drawing with-replacement, or re-weighting, draw $m \ll n$ sub-samples.

This defeats the whole purpose of trees: these are rules that are designed to grow in complexity with the amount of data available.
(that's why we bother storing so much data!)

If you starve the individual trees of data you lose.

## Empirical Bayesian Forests (EBF)

RFs are expensive. Sub-sampling hurts bad.

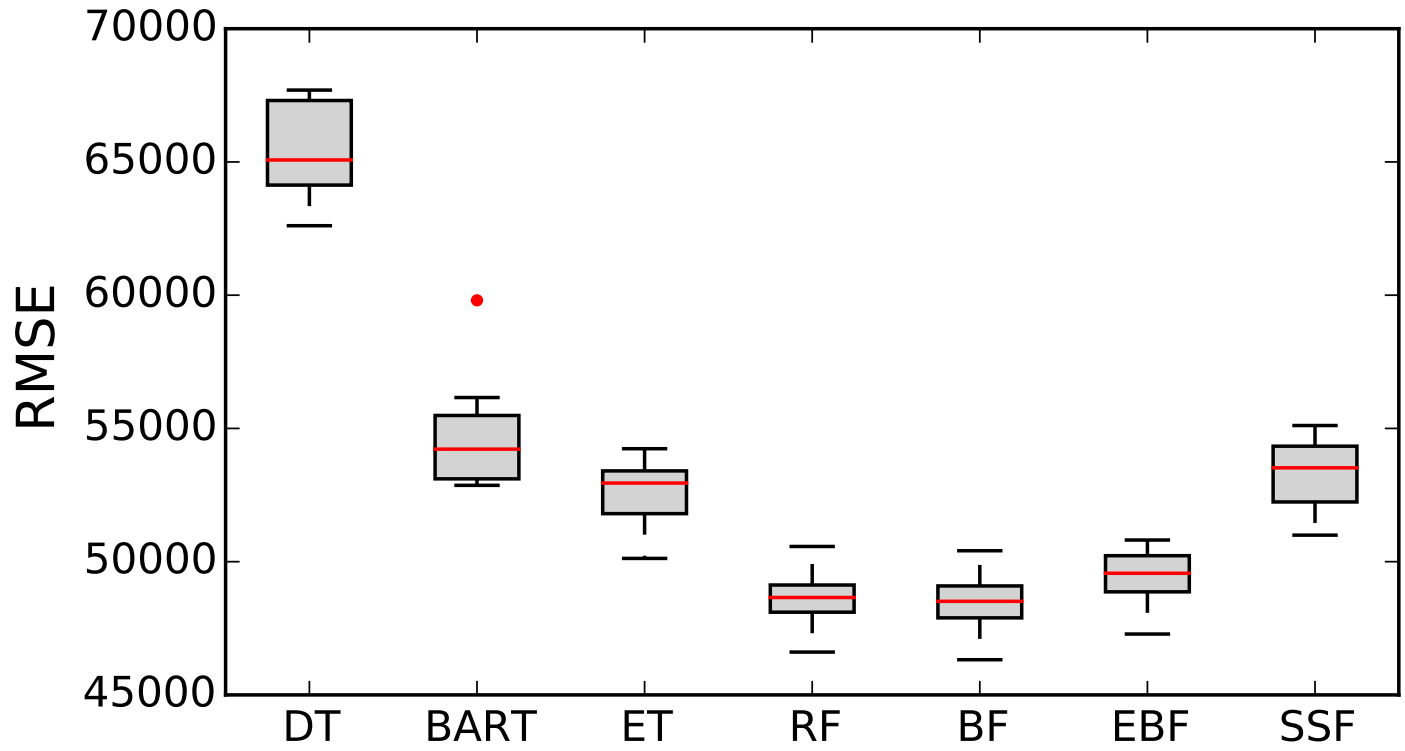
Instead:

- fit a single tree to a shallow trunk.
- Map data to each branch.
- Fit a full forest on the smaller branch datasets.

Since the trunks are all similar for each tree in a full forest, our EBF looks nearly the same at a fraction of computational cost.

It's an updated version of classical Empirical Bayes:
use plug-in estimates at high levels in a hierarchical model,
focus effort at full Bayesian learning for the the hard bits.

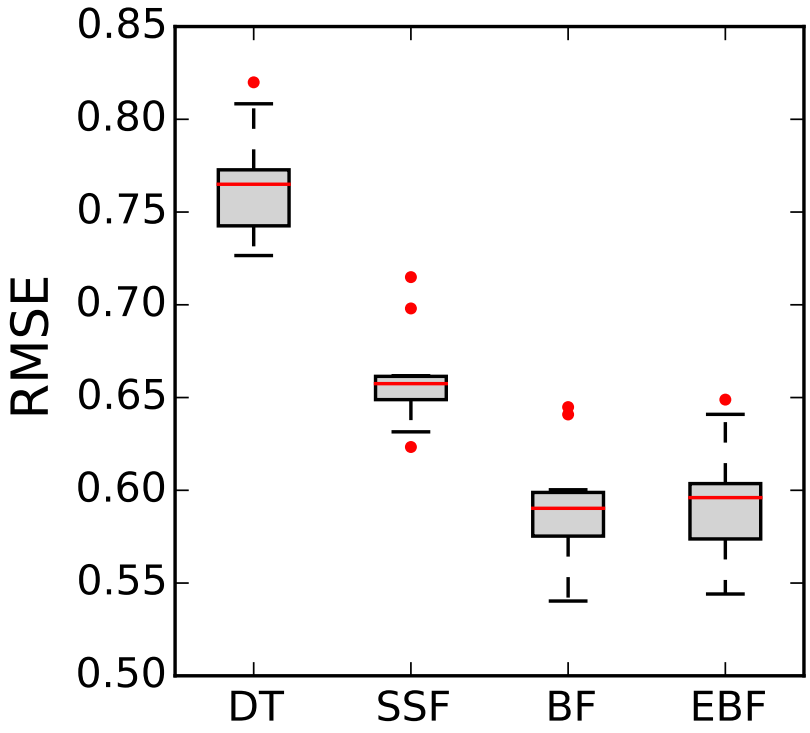
## OOS predictive performance on California Housing




Here EBF and BF give nearly the same results. *SSF does not.*

EBFs crunch more data faster without hurting performance.

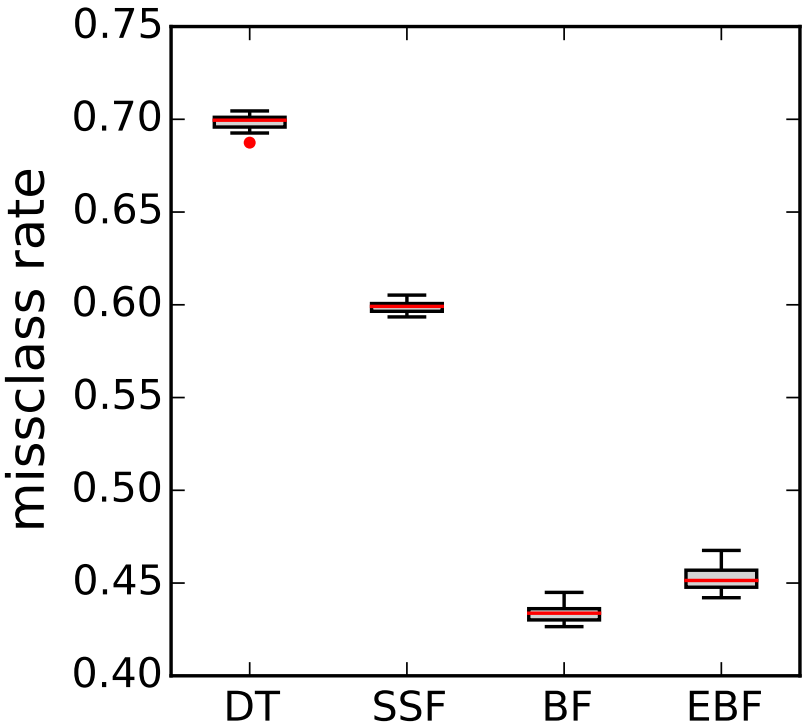
## EBFs work all over the place




| $\overline{\text{RMSE}}$ | % WTB | |
|---|---|---|
| 0.5905 | 0.0 | BF |
| 0.5953 | 0.8 | EBF |
| 0.6607 | 11.9 | SSF |
| 0.7648 | 29.5 | DT |

Predicting wine rating from chemical profile

## EBFs work all over the place




| $\overline{\text{MCR}}$ | % WTB | |
|---|---|---|
| 0.4341 | 0.0 | BF |
| 0.4531 | 4.4 | EBF |
| 0.5989 | 38.0 | SSF |
| 0.6979 | 60.8 | DT |

or beer choice from demographics

## Choosing the trunk depth

Distributed computing perspective: fix only as deep as you must!

How big is each machine? Make that your branch size.

| | CA housing | | | Wine | | | Beer | | |
|---|---|---|---|---|---|---|---|---|---|
| *Min Leaf Size in* $10^3$ | 6 | 3 | 1.5 | 2 | 1 | 0.5 | 20 | 10 | 5 |
| *% Worse Than Best* | 1.6 | 2.4 | 4.3 | 0.3 | 0.8 | 2.2 | 1.0 | 4.4 | 7.6 |

Still, open questions: e.g., more trees vs shallower trunk?

## Catching Bad Buyer Experiences at eBay

BBE: 'not as described', delays, etc.

$\mathrm{p}$(BBE) is a key input to search rankings, and elsewhere.

Big Data axiom: best way to improve prediction is more data.

(a.k.a. 'data beats model' )

On 200 million transactions, EBF with 32 branches yields a 1.5% drop in misclassification over the SSF alternatives.

EBFs via Spark $\Rightarrow$ more data in less time.

Putting it into production requires some careful engineering, but this really is a very simple algorithm. Big gain, little pain.

## Back to HTE: Treatment Effect Trees

Recall our earlier example: A/B experiment with user covariates $\mathbf{x}_i$.

Athey+Imbens propose indexing user HTE by fitting CART to

$$y_i^\star = y_i \frac{d_i - q}{q(1-q)} = \begin{cases} y_i/(1-q) & \text{if } d_i = 0 \\ y_i/q & \text{if } d_i = 1 \end{cases}$$

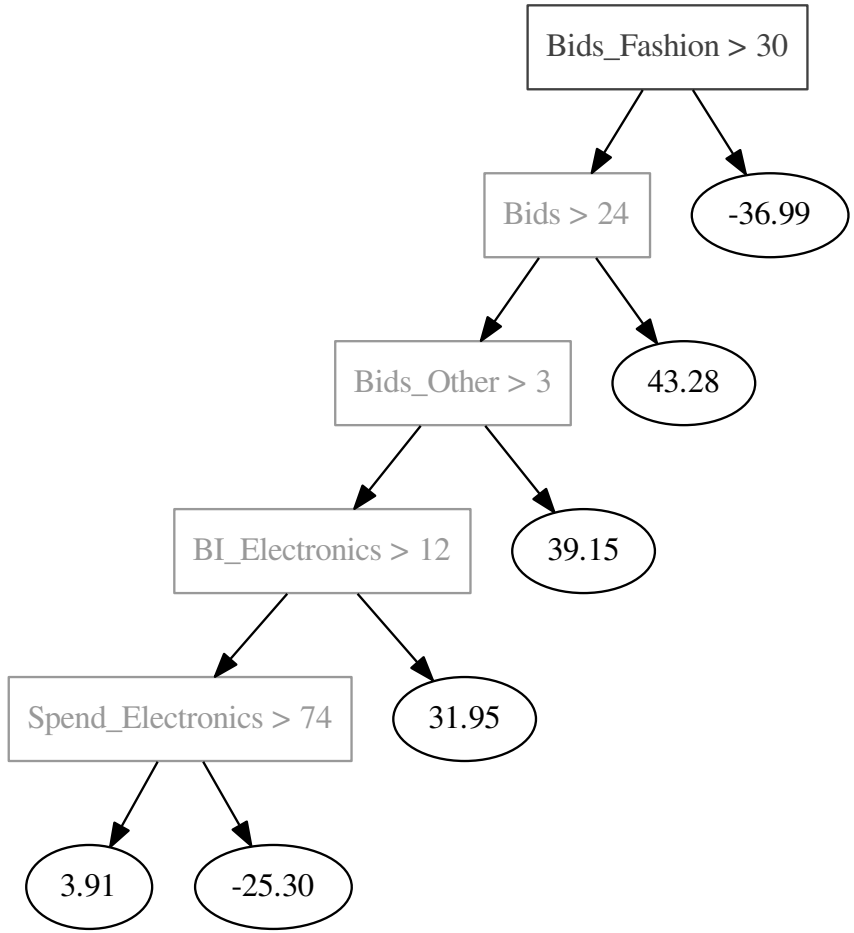
where $q$ is the probability of treatment (2/3 in our example).

This works because

$$\mathbb{E}[y_i^\star | \boldsymbol{\upsilon}_i] = \upsilon_i(1) - \upsilon_i(0).$$

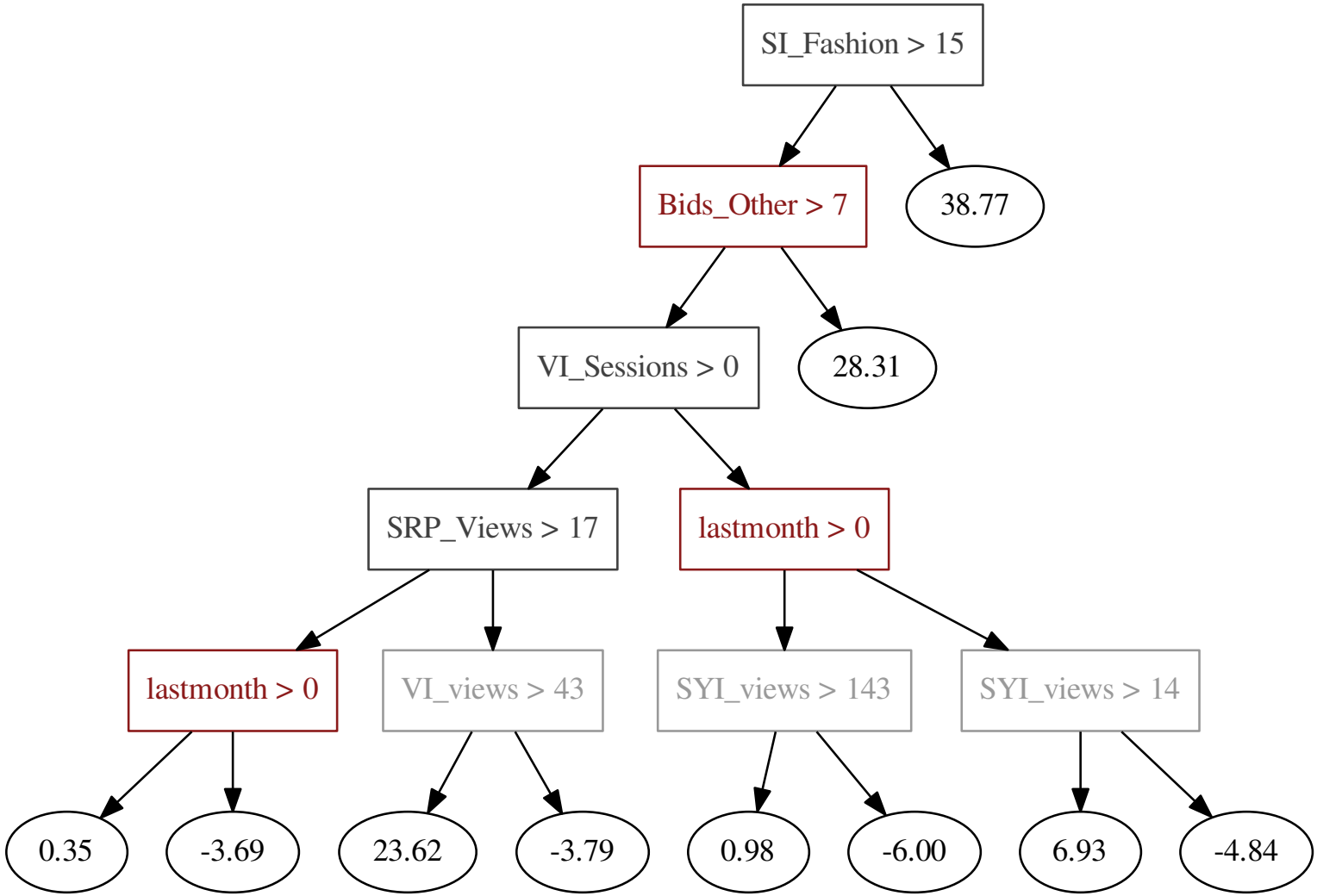
How sure are we of the fitted tree? Bayes forest is a posterior.

e.g., Pruned CART with after one week:



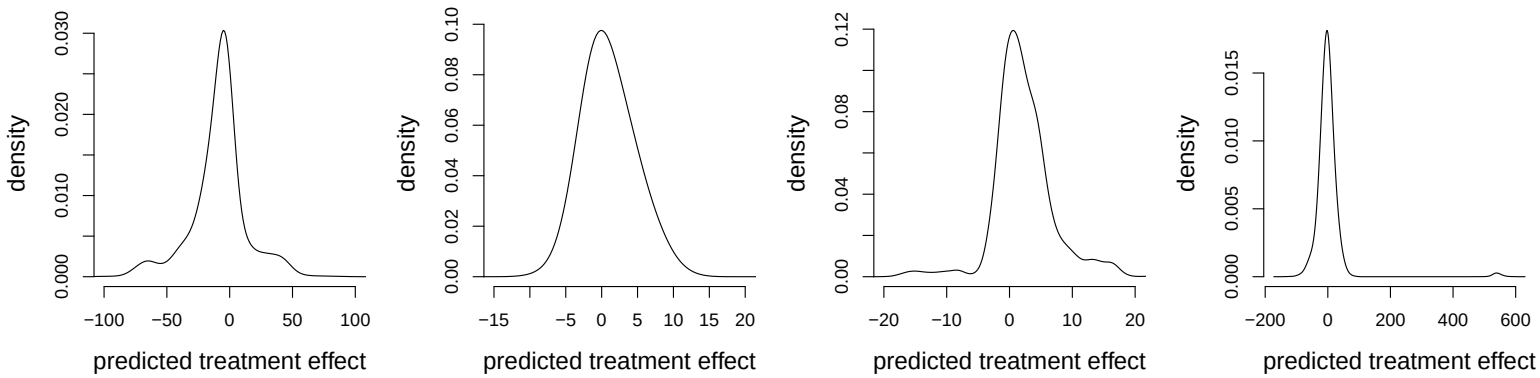

prob variable is node in tree: $\mathrm{p} < \frac{1}{3}$, $\mathrm{p} \in \left[\frac{1}{3}, \frac{1}{2}\right)$, and $\mathrm{p} \geq \frac{1}{2}$.

After 5 weeks the tree is much more stable.

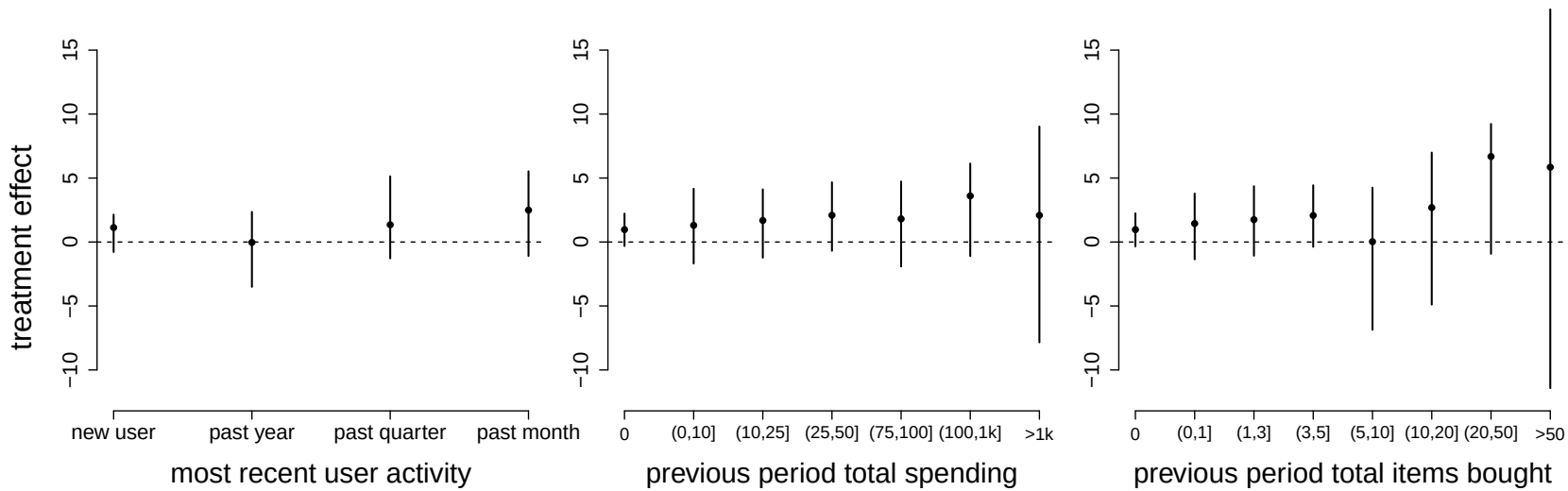



But you should still use a forest below a certain point.

*It is wise to look to the full forest instead:*



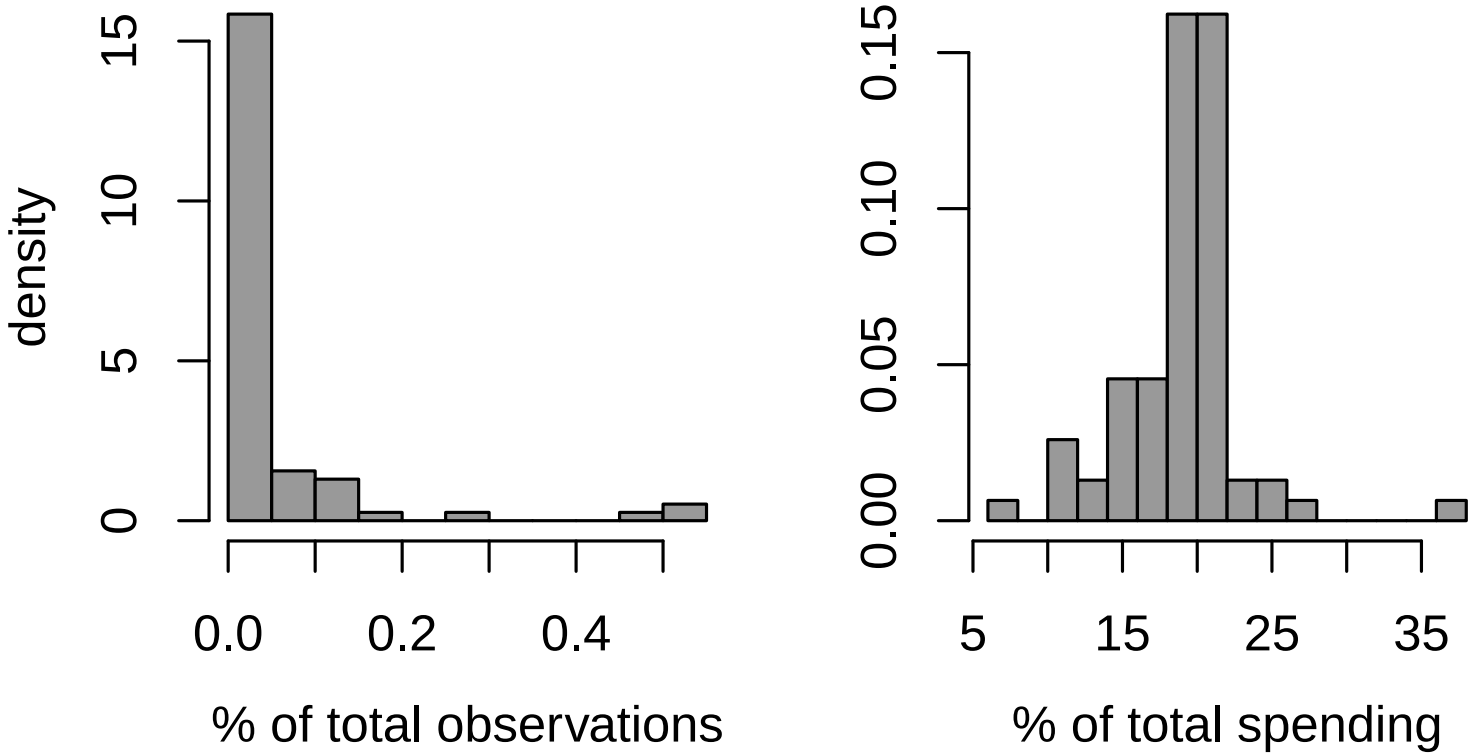

Treatment effect posteriors for sampled users (5-week data).




Variable average effects for three important user features.

## Scalable semiparametrics for heavy-tailed distributions

In some settings we have really heavy tailed data.



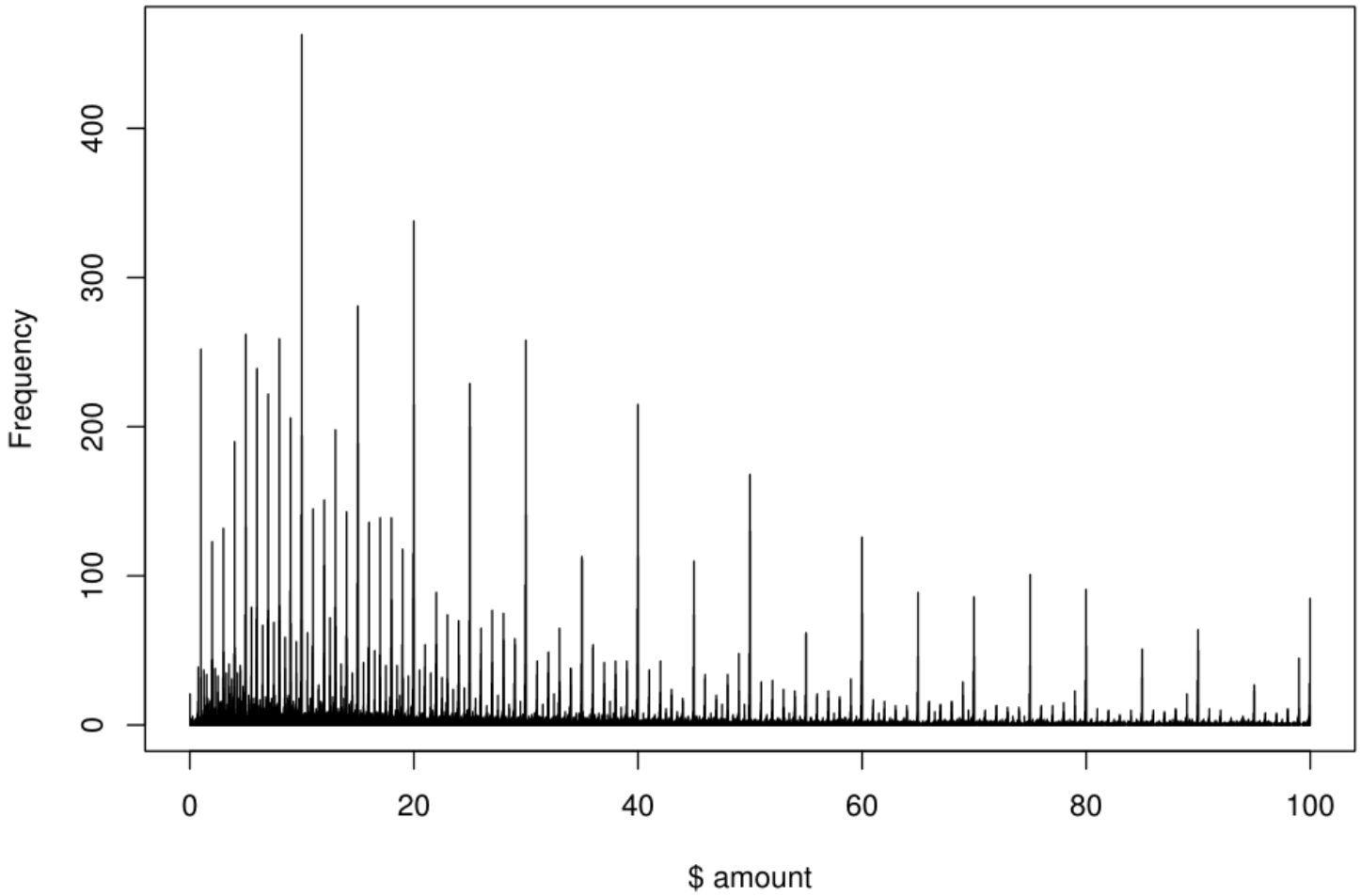

The standard bootstrap fails for infinite variance distributions.

We should suspect our Bayesian bootstrap will have similarly bad frequentist properties. e.g., the posterior will not be consistent.

**Lots of spikes!**

14,521 unique $ amounts from 50,000 users.

78.18% of the $ amounts are less than $100.




So the distribution below the tail is also screwy.

**Semi-parametrics: only model where you really need it.**

We can replace our DGP with a model that includes a parametric tail above some threshold $u$:

$$\mathrm{p}(z) = \frac{1}{|\boldsymbol{\theta}|}\sum_{l=1}^{L}\theta_l \mathbb{1}[z = \zeta_l] + \frac{\theta_{L+1}}{|\boldsymbol{\theta}|}\mathrm{GPD}(z-u;\ \xi,\ \sigma)\mathbb{1}[z \geq u]$$

where $\mathrm{GPD}(\cdot;\ \xi,\ \sigma)$ is the generalized Pareto density function

$$\mathrm{p}(y) = \frac{1}{\sigma}\left(1 + \xi\frac{y}{\sigma}\right)^{-(\frac{1}{\xi}+1)}$$

with $\xi > 0$ the tail index and $\sigma > 0$ the scale.

Say $\mu = \mathbb{E}[z]$ and $\lambda = \mathbb{E}[z - u | z \geq u]$.

$\mathbb{E}[\mu]$ and $\mathrm{var}(\mu)$ are available in closed form up-to $\mathbb{E}[\lambda]$ and $\mathrm{var}[\lambda]$.

## Inference for tail parameters

Everything for the weights is as before (exponential posterior).

For the GPD tail parameters, we use the prior

$$\pi(\sigma, \xi) = \frac{1}{\sigma} \xi^{a-1} (1 - \xi)^{b-1} \mathbb{1}_{\xi \in (0,1)},$$

For noninformative inference, use $a = b = 1$. But it is cooler to use a bank of related 'background' tail transactions to estimate $\hat{\alpha}, \hat{\beta}$.

We have a super simple independence MH posterior sampler that is based upon re-sampling from a parametric bootstrap.

Or, you can use a laplace approximation.

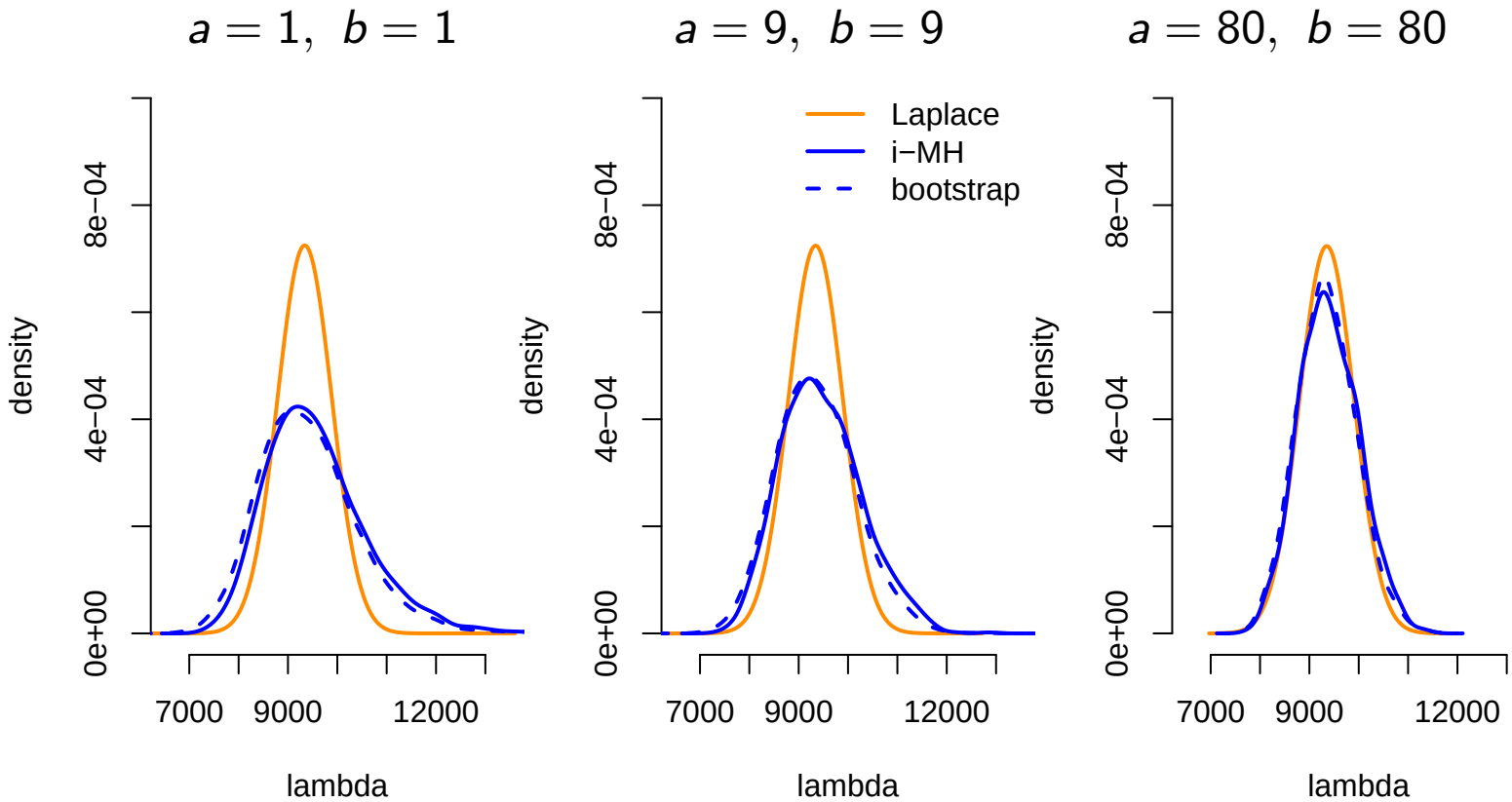
## Bootstrap-based posterior sampling

- Fit the MAP parameter estimates $[\hat{\xi}, \hat{\sigma}]$.
- Obtain $B$ draws $[\hat{\xi}_b, \hat{\sigma}_b]$ from the parametric bootstrap:
  - Generate a sample of size $n$ from the MAP GPD model.
  - Obtain new MAP estimates $[\hat{\xi}_b, \hat{\sigma}_b]$ for this simulated sample.
- Get a kernel estimate, say $r(\xi, \sigma)$, for the bootstrap density.
- Replace each $[\hat{\xi}_b, \hat{\sigma}_b]$ with $[\hat{\xi}_{b-1}, \hat{\sigma}_{b-1}]$ with probability

$$1 - \min\left\{ \frac{r(\hat{\xi}_{b-1}, \hat{\sigma}_{b-1}) \exp[l(\hat{\xi}_b, \hat{\sigma}_b)]}{r(\hat{\xi}_b, \hat{\sigma}_b) \exp[l(\hat{\xi}_{b-1}, \hat{\sigma}_{b-1})]}, \; 1 \right\}.$$

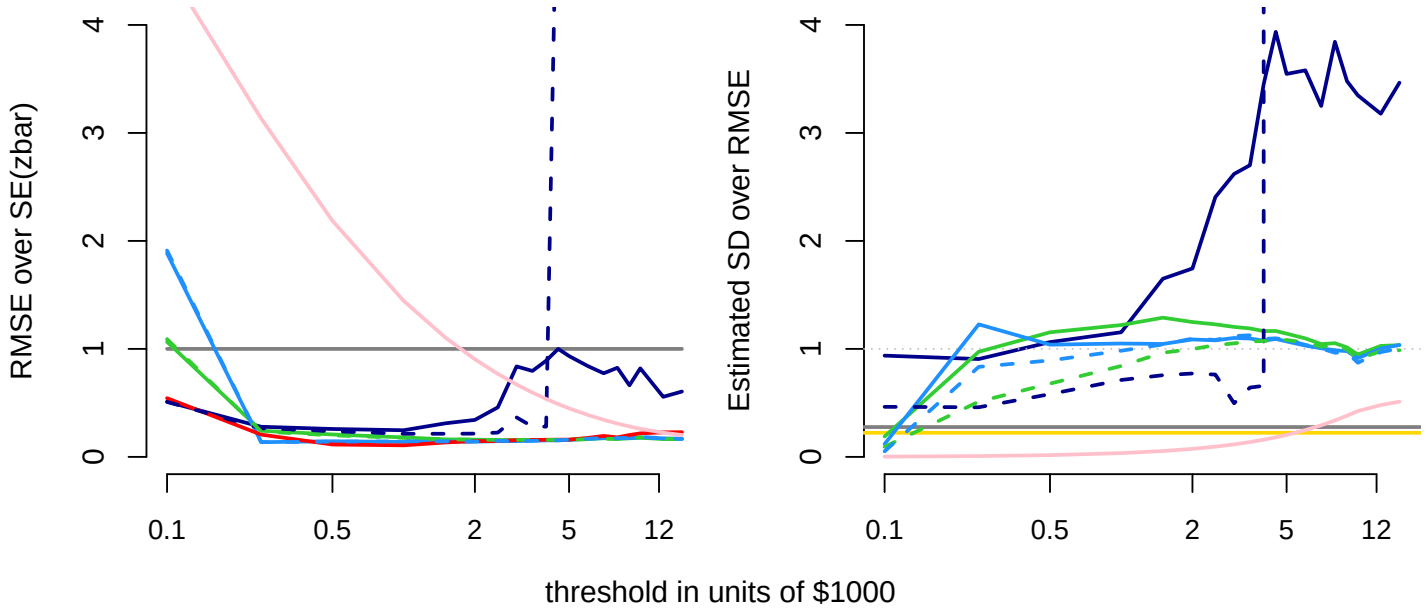
Super fast to run and code. And it relates Bayes to Frequentism!

Inference for the mean exeedance $\lambda = \sigma/(1-\xi)$,
beyond $u =$ \$9000, in one of our treatment groups.




These results give all we need for inference about the overall mean.

Performance over 100 resamples of $N = 50,000$ from $10^7$ total.



## Consistency and thresholds

The *semiparametric bootstrap* is consistent if $u_N = O(N^{\xi/(1+2\delta\xi)})$.
Which is cool, because the nonparametric bootstrap is not.

In this limit, $\sigma_N = \xi u_N$. So, one can increase $u$ until this holds.
Indeed, it holds roughly in our examples for good performing $u$.

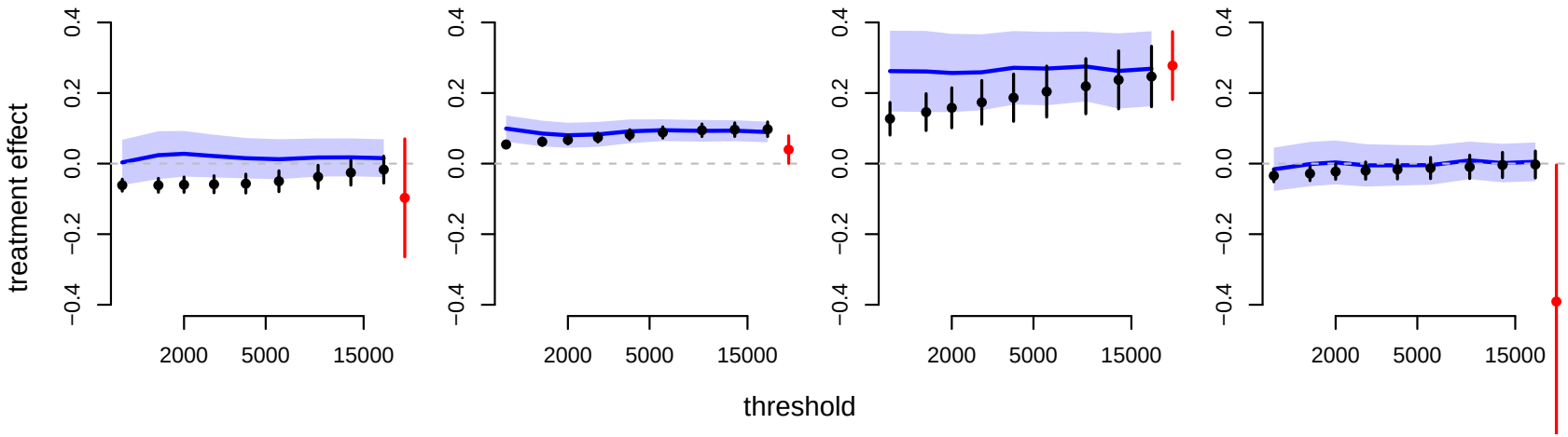
In practice, results are robust to a range of tail thresholds.

This also suggests why informative priors on $\sigma$ are not useful:
it is already informed by both the threshold and the tail index.

# Semi-parametrics in treatment effect estimation

Semiparametric inference makes a big difference in A/B analysis. Looking at ATEs, $\mu_1 - \mu_2$, in a few example experiments:

sp-bayes, capped, naive




Such care in uncertainty quantification is essential whenever you need to understand the full posterior (or sampling) distribution.

e.g., bandit learning with heavy-tailed rewards is another application.

## Big Data and distribution-free BNP

I think about BNP as a way to analyze (and improve) algorithms. Decouple action/prediction from the full generative process model.

This type of analysis is key for taking these algorithms from ML to fields like Economics that really care about uncertainty.

## Efficient Big Data analysis

To cut computation without hurting performance, we need to think about what portions of the 'model' are hard or easy to learn.

Once we figure this out, we can use a little bit of the data to learn the easy stuff and direct our full data at the hard stuff.

This is *the* future for Big Data.

**thanks!**