Big Data and Bayesian Nonparametrics

Matt Taddy, Chicago Booth

`faculty.chicagobooth.edu/matt.taddy/research`

Challenges of being Big and Bayesian

The sample sizes are enourmous.

- we'll see 21 and 200 million today.
- Data can't fit in memory, or even storage, on a single machine.
- Our familiar MCMC algorithms take too long.

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.
- We cannot write down believable models.

Both ‘Big’ and ‘Strange’ beg for nonparametrics.

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

$$\text{e.g.,} \quad y = f(\mathbf{x}) + \epsilon, \quad \text{or even just} \quad f(y|\mathbf{x})$$

However averaging over all of the nuisance parameters we introduce to be ‘flexible’ is a hard computational problem.

Can we do scalable BNP?

Frequentists are great at finding simple procedures (e.g. $[\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'y$) and showing that they will 'work' regardless of the true DGP.

(DGP = Data Generating Process)

This is classical '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.

No need to re-model the underlying DGP each time, and you don't need to have a PhD in Bayesian Statistics to apply the ideas.

Can we Bayesians provide something like this?

distribution free Bayesian nonparametrics

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

- Derive it from first principles, e.g. moment conditions.
- Or a statistic could be an algorithm that we know works.

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

$$g(\mathbf{z}) = \frac{1}{|\boldsymbol{\theta}|}\sum_{l=1}^{L}\theta_l \mathbb{1}[\mathbf{z} = \boldsymbol{\zeta}_l], \quad \theta_l/|\boldsymbol{\theta}| \overset{iid}{\sim} \mathrm{Dir}(a)$$

After observing $\mathbf{Z} = \{\mathbf{z}_1 \ldots \mathbf{z}_n\}$, posterior has $\theta_l \sim \mathrm{Exp}(a + \mathbb{1}_{\boldsymbol{\zeta}_l \in \mathbf{Z}})$.
(say every $\mathbf{z}_i = [\mathbf{x}_i, y_i]$ is unique).

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

$$\Rightarrow g(\mathbf{z} \mid \mathbf{Z}) = \tfrac{1}{|\boldsymbol{\theta}|}\sum_{l=1}^{L}\theta_l \mathbb{1}[\mathbf{z} = \mathbf{z}_l], \quad \theta_i \sim \mathrm{Exp}(1)$$

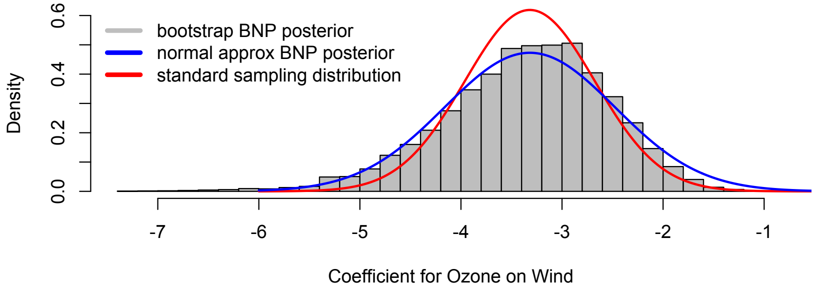
This is just the Bayesian bootstrap.
Ferguson 1973, Rubin 1981

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)



```
data(airquality); n <- nrow(airquality)
B <- 1000; beta <- vector(length=B)
for(b in 1:B){
  fit <- lm(Ozone ~., data=airquality, weights=rexp(n))
  beta[b] <- coef(fit)["Wind"] }
sampfit <- lm(Ozone ~ ., data=airquality)
coef <- summary(sampfit)$coef["Wind",1:2]
x <- as.matrix(cbind(1,na.omit(airquality)[,-1]))
xxi <- solve(crossprod(x))
sandwich <- xxi%*%t(x)%*%diag(sampfit$resid^2)%*%x%*%xxi
hist(beta, col=8, main="",
  xlab="Coefficient for Ozone on Wind",
  freq=FALSE,ylim=c(0,0.6),breaks=25)
grid <- seq(-6,5,length=500)
lines(grid, dnorm(grid,coef[1],coef[2]),col=2,lwd=2)
lines(grid, dnorm(grid,coef[1],sqrt(sandwich[3,3])),col=4,lwd=2)
legend("topleft",col=c(8,2),lwd=4,
  legend=c("bootstrap BNP posterior",
           "normal approx BNP posterior",
           "standard sampling distribution"),bty="n")
```

What is the blue line?

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

Consider a first-order Taylor series approximation,

$$\tilde{\boldsymbol{\beta}} = [\mathbf{X}'\mathbf{X}]^{-1}\mathbf{X}'y + \nabla\beta\big|_{\boldsymbol{\theta}=\mathbf{1}}(\boldsymbol{\theta} - \mathbf{1})$$

We can derive *exact* posterior moments for $\tilde{\boldsymbol{\beta}}$ under $\theta_i \overset{iid}{\sim} \mathrm{Exp}(1)$.

e.g., $\mathrm{var}(\tilde{\boldsymbol{\beta}}) \approx (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathrm{diag}(\mathbf{e})^2\mathbf{X}'(\mathbf{X}'\mathbf{X})^{-1}$, where $e_i = y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}$.

This is the familiar Huber-White ‘Sandwich’ variance formula.

See Lancaster 2003 or Poirier 2011.

Example: **User-Specific Behavior in Experiments**

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 $[0/1]$.

We know $\mathbf{x}_i$ about user i.

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

$\mathbf{x}_i$ are possible sources of heterogeneity. About 400 in our example.

What is ‘heterogeneity in treatment effects’? (HTE)

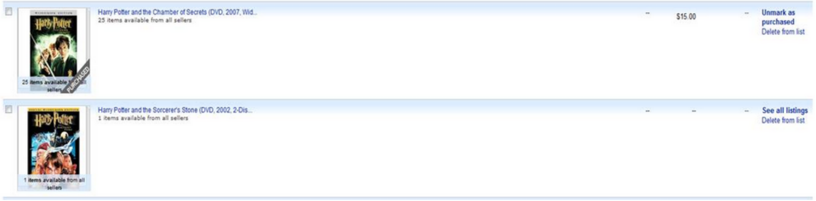
Different units [people, devices] respond differently to some treatment you apply [change to website, marketing, policy].

I imagine it exists.

Can we accurately measure heterogeneity (index it on $\mathbf{x}$),
and how is this useful for decision making?

In our illustrative example, d_i = bigger pictures in my eBay.

- Test Variant: Larger Image (140 x 140px)

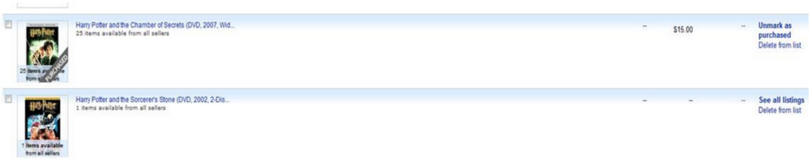


Harry Potter and the Chamber of Secrets (DVD, 2007, Wid...
25 items available from all sellers
-- $15.00 -- Unmark as purchased
Delete from list

Harry Potter and the Sorcerer's Stone (DVD, 2002, 2-Dis...
1 items available from all sellers
-- -- -- See all listings
Delete from list

25 items available from all sellers
PURCHASED
1 items available from all sellers

- Control Variant: Production (96 x 96px)



Harry Potter and the Chamber of Secrets (DVD, 2007, Wid...
25 items available from all sellers
-- $15.00 -- Unmark as purchased
Delete from list

Harry Potter and the Sorcerer's Stone (DVD, 2002, 2-Dis...
1 items available from all sellers
-- -- -- See all listings
Delete from list

25 items available from all sellers
PURCHASED
1 items available from all sellers

21 million tracked visitors over 5 weeks, 2/3 in treatment.

a statistic we care about ...

Potential outcome: $\upsilon_i(\mathsf{d})$ is $\approx$ \$ for user i under d.

We only ever get to see one of $\upsilon_i(\mathsf{t})$ and $\upsilon_i(\mathsf{c})$: ‘y’.

We care about ‘$\boldsymbol{\gamma}$’ from the *moment condition*

$$\mathbb{E}\left[\mathbf{x}(\upsilon(\mathsf{t}) - \upsilon(\mathsf{c}) - \mathbf{x}'\boldsymbol{\gamma})\right] = \mathbf{0}$$

This says $\mathbf{x}'\boldsymbol{\gamma}$ is uncorrelated with the treatment effect $\upsilon(\mathsf{t}) - \upsilon(\mathsf{c})$.

An extra randomization condition impies

$$\boldsymbol{\gamma} = \mathbb{E}\left[\mathbf{x}\mathbf{x}'\right]^{-1}\Big(\mathbb{E}[\mathbf{x}y|\mathsf{d} = \mathsf{t}] - \mathbb{E}[\mathbf{x}y|\mathsf{d} = \mathsf{c}]\Big)$$

since $\mathbb{E}[\mathbf{x}\upsilon(\mathsf{d})] = \mathbb{E}[\mathbf{x}\upsilon(\mathsf{d})|d]$ for randomized control trial.

approximate mean and variance ...

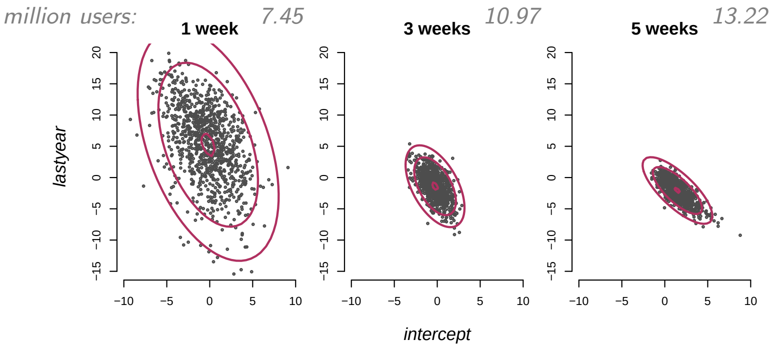
One can show that

$$\mathbb{E}\gamma \approx \hat{\gamma} = n(\mathbf{X}'\mathbf{X})^{-1}\left(\frac{\mathbf{X}_{\mathrm{t}}'\mathbf{y}_{\mathrm{t}}}{n_{\mathrm{t}}} - \frac{\mathbf{X}_{\mathrm{c}}'\mathbf{y}_{\mathrm{c}}}{n_{\mathrm{c}}}\right)$$

and we have an approximate variance (not too messy) $\mathbf{\Sigma}_{\tilde{\gamma}}$.

Or you can bootstrap, but it takes a long time.

e.g., coefficient on purchase within last-year vs an intercept:



Sample is from posterior, contours are normal approximation.

This is a statistic we care about, even if the truth is nonlinear.

Mining heterogeneity

We have 400 possible ‘directions’ of heterogeneity in γ.
Various levels of confidence in each, and they are highly correlated.

Can we pick out a few that look promising for exploration?

Be Bayesian: write down a *loss function* for what we decide to report, and choose the decision to minimize expected loss.

Say $\boldsymbol{\delta}$ is our 'decided' vector of heterogenaity.

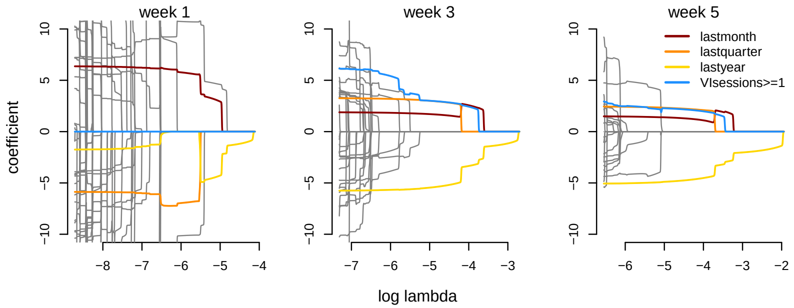
We want it to be mostly zero – keep count $|\delta_j \neq 0|$ small.

$$\mathbb{E}[\text{loss}] = \sum_i \frac{(\mathbf{x}_i\boldsymbol{\delta} - \mathbf{x}_i\hat{\boldsymbol{\gamma}})^2}{2n\mathrm{var}(\mathbf{x}_i\boldsymbol{\gamma})} + \lambda|\delta_j \neq 0|$$

Penalty $\lambda > 0$ is your 'cost of complexity': it's like a squelch.

Minimize to get answers 'close' to best, with closeness discounted by uncertainty about that best guess, using only a few covariates.

That actual minimization is tough, but we can get very close.



This shows $\boldsymbol{\delta}$ (vertical axis) as a function λ (cost of complexity).

Slices of this path provide representations of heterogenaity.

The best 3-factor model has treatment effects on capped GMB

```
newuser  lastmonth   lastyear       PC3
   4.66       6.60       0.10      2.23
```

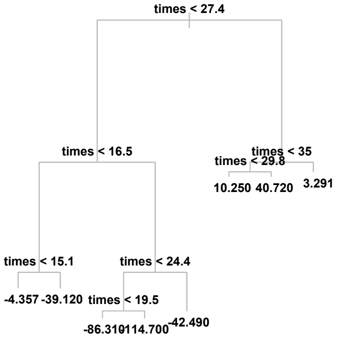
PC3 is big for sellers, viewers, and buyers of ‘unknown’ items.

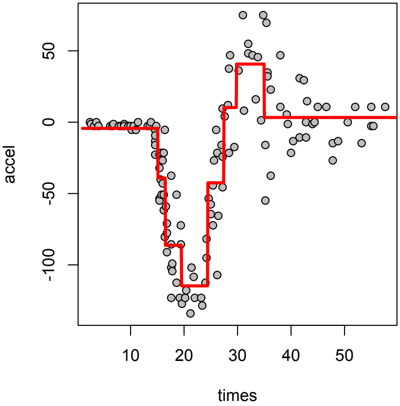
Our ‘treasure hunters’!

Heterogenaity! Get ’em with bigger pictures!

Example 2: Decision Trees are awesome

They learn non-linear response functions
discover interactions between variables,
and don't care about heteroskedasticity.



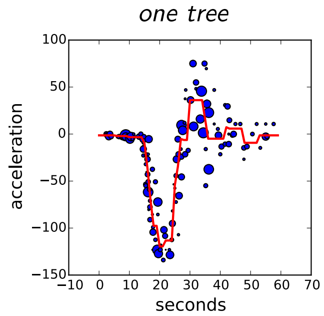


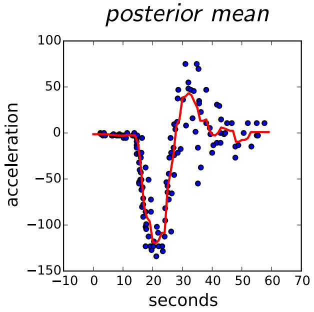
The CART tree is a statistic we care about.

Imagine CART on the population: it would predict well.

Bayesian Forest

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





Theoretical trunk stability

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

We are able to derive theoretically that the earliest structure in the tree – the trunk – should be very stable for large samples.

For the data at a given node,

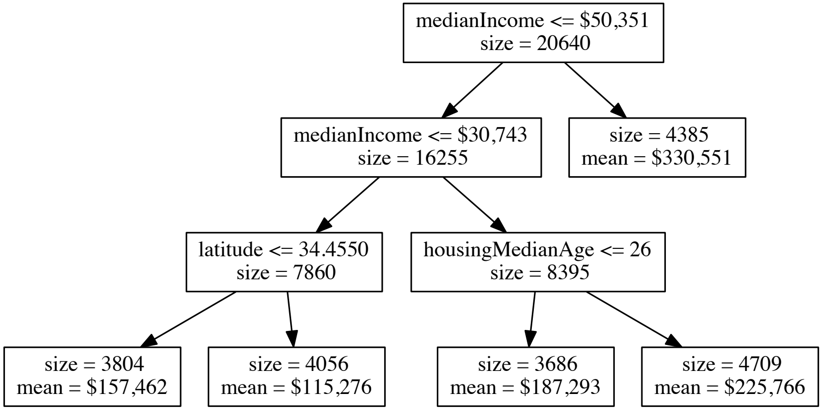
$$\mathrm{p}\left(\text{optimal split matches sample CART}\right) \gtrsim 1 - \frac{p}{\sqrt{n}} e^{-n},$$

with p split locations and n observations.

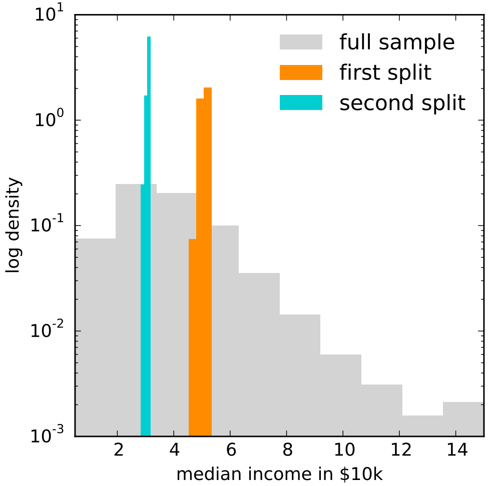
Things are pretty stable, until they aren't: as the tree grows, node sizes get smaller and chance of a non-optimal split multiplies.

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.

Forests are expensive when data is too big to fit in memory.

Subsampling forests lead to a big drop in performance.

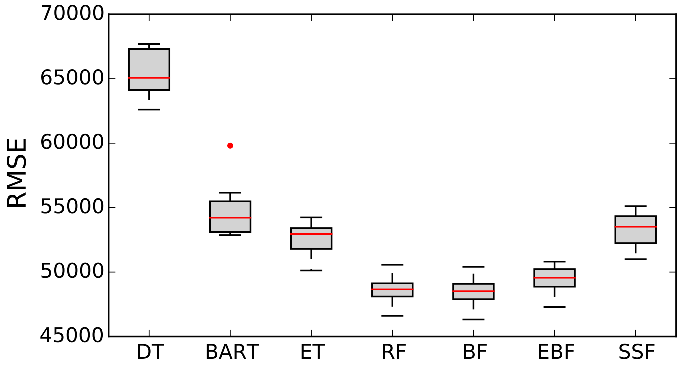
But wait: if the trunks are stable, can we just fit that once and then fit forests at each branch? Yes!

Empirical Bayesian Forests (EBF):

- fit a single tree to a shallow trunk.
- Use this as a mapper to direct full data to each branch.
- Fit a full forest on the smaller branch datasets.

This is classic Empirical Bayes: fix higher levels in a *hierarchical model*, and direct your machinery+data at learning the hard bits.

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



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

EBFs at EBay: predicting Bad Buyer Experiences

A BBE could be receiving something that is ‘significantly not as described’, or shipping delays, or any of many other problems.

The models are updated frequently, and information about p(BBE) is an input to search rankings and more.

The best thing to improve predictions is more data.
With millions of daily transactions, there’s little limit on data.

EBFs at EBay

Random forest runs take too long on full data.

Subsampling led to a noticeable and big drop in performance.

So: EBFs!

- trunk can be fit in distribution using Spark `MLLib`.
- this trunk acts as a sorting function to map observations to separate locations corresponding to each branch.
- Forests are then fit on a machine for each branch.

EBFs at EBay

On 12 million transactions, EBF with 32 branches yields a 1.3% drop in misclassification over the SSF alternatives.

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

A key point: EBFs are not inherently 'better' than forests fit to all of the data. But EBFs can be fit to more data in less time.

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.

I believe that this is the future for Big Data.

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.

thanks!