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

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

After observing $\mathbf{Z} = \{\mathbf{z}_1 \dots \mathbf{z}_n\}$, 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.

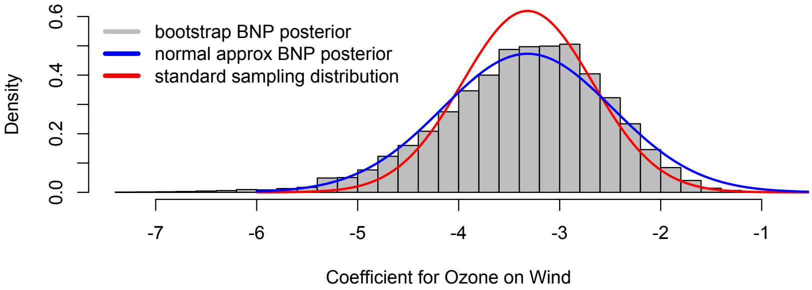
Ferguson 1973, Rubin 1981

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)



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 $\theta_i \stackrel{iid}{\sim} \mathrm{Exp}(1)$.

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 $\mathbf{x}$?

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

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

- Control Variant: Production (96 x 96px)

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 $\upsilon(\mathsf{t}) - \upsilon(\mathsf{c})$.

Since randomization implies that $\mathbb{E}[\mathbf{x}\upsilon(\mathsf{d})] = \mathbb{E}[\mathbf{x}\upsilon(\mathsf{d})|d]$, we get

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

As we did with OLS, consider a first-order approximation

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

where

$$\hat{\gamma} = \gamma\big|_{\boldsymbol{\theta}=\mathbf{1}} = n(\mathbf{X}'\mathbf{X})^{-1}\left(\frac{\mathbf{X}_{\mathrm{t}}'\mathbf{y}_{\mathrm{t}}}{n_{\mathrm{t}}} - \frac{\mathbf{X}_{\mathrm{c}}'\mathbf{y}_{\mathrm{c}}}{n_{\mathrm{c}}}\right).$$

This yields an approximate variance

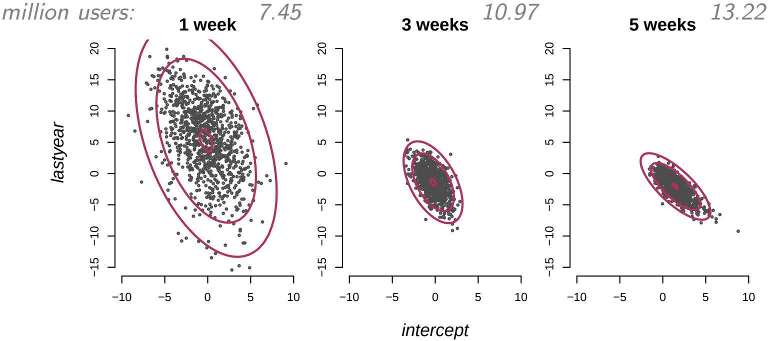
$$\mathrm{var}(\tilde{\gamma}) \approx (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathrm{diag}(\boldsymbol{e}^{\star})\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

with 'treatment effect residuals'

$$e_i^{\star} = \left(\frac{\mathbb{1}_{[i\in\mathrm{t}]}}{n_{\mathrm{t}}} - \frac{\mathbb{1}_{[i\in\mathrm{c}]}}{n_{\mathrm{c}}}\right) n y_i - \mathbf{x}_i'\hat{\gamma}.$$

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

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

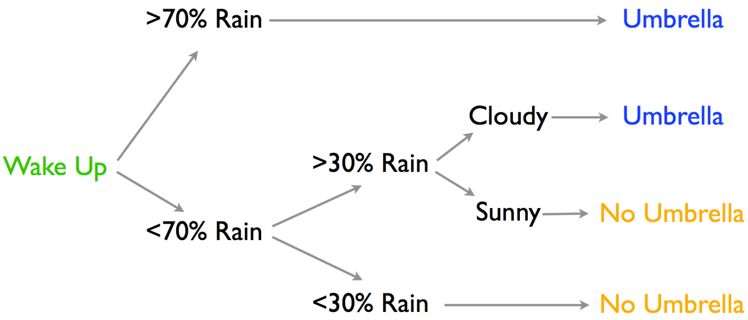


Sample is from posterior, contours are normal approximation.

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

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

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

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

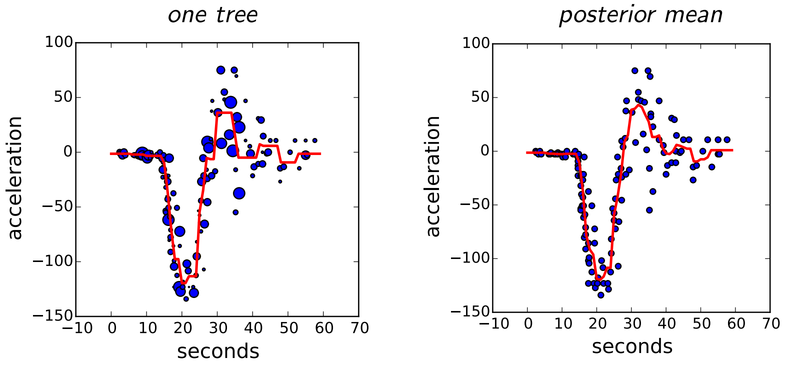
Population-CART might be a statistic we care about.

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 ≈ Bayesian forest ≈ posterior over CART fits.

Treatment Effect Trees

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).

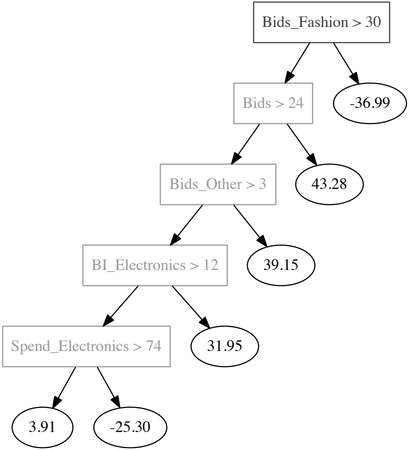
This works because

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

where $\upsilon_i(d)$ is the potential outcome for user i if $d_i = d$.
We only get to observe one of these: $y_i = \upsilon_i(d_i)$.

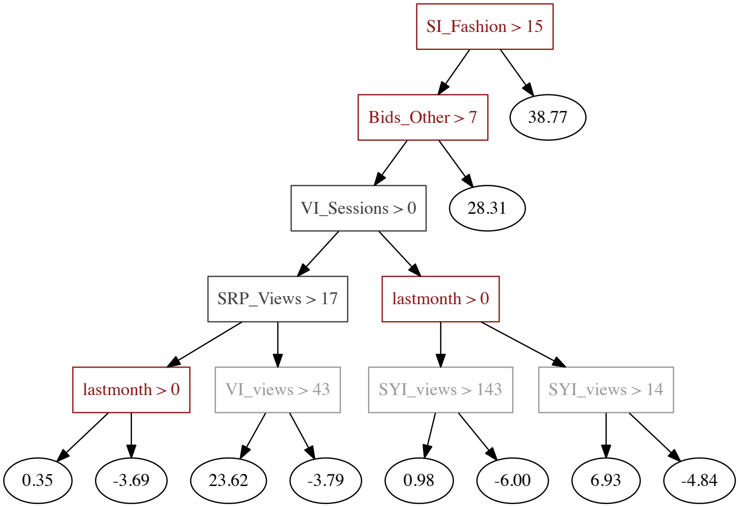
We can apply the Bayesian bootstrap (i.e., fit a BF) to assess *posterior uncertainty* about these treatment-effect trees.

e.g., sample depth-5 CART with min-leaf-size-10^5 after one week:



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

After 5 weeks the tree is much more stable.



We can quantify uncertainty about all sorts of structure.

Posterior probability of CART splitting on variable

Week 5	*depth in tree*				
	1	≤ 2	≤ 3	≤ 4	≤ 5
SI Fashion	.45	.50	.50	.50	.50
Bids Other	.30	.75	.75	.75	.75
VI sessions	.05	.05	.05	.10	.35
lastmonth	.05	.10	.15	.40	.65
SRP views	.00	.10	.15	.25	.40
VI views	.00	.00	.10	.20	.25
SYI views	.00	.00	.05	.15	.25

$\Rightarrow$ easy scalable uncertainty quantification for complex algorithms.

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!