

# SDS 385: Stat Models for Big Data Lecture 11: Bootstrap and subsampling

Purnamrita Sarkar Department of Statistics and Data Science The University of Texas at Austin

https://psarkar.github.io/teaching

- So far we have talked about estimation, and ways to estimate statistical quantities quickly
- But often, you are interested in quantifying the variability of your estimate
- You can do this using the variance of your estimate or by producing a confidence interval
- What is a confidence interval?

- Data  $X_1, \ldots, X_n \stackrel{\text{iid}}{\sim} P$
- Some estimator  $\hat{\theta}$  of parameter of interest  $\theta$ .
- For some coverage  $\alpha,$  want to produce a lower and upper bound such that:

$$P\left(\hat{a} \le \theta \le \hat{b}\right) \ge 1 - 2\alpha$$

- Data  $X_1, \ldots, X_n \stackrel{\mathsf{iid}}{\sim} P$
- Some estimator  $\hat{\theta}$  of parameter of interest  $\theta$ .
- For some coverage  $\alpha$ , want to produce a lower and upper bound such that:

$$P\left(\hat{a} \leq \theta \leq \hat{b}\right) \geq 1 - 2\alpha,$$

• Say you know the distribution of  $(\hat{ heta}- heta)/\hat{\sigma}$ 

- Data  $X_1, \ldots, X_n \stackrel{\mathsf{iid}}{\sim} P$
- Some estimator  $\hat{\theta}$  of parameter of interest  $\theta$ .
- For some coverage  $\alpha$ , want to produce a lower and upper bound such that:

$$P\left(\hat{a} \leq \theta \leq \hat{b}\right) \geq 1 - 2\alpha,$$

- Say you know the distribution of  $(\hat{ heta} heta)/\hat{\sigma}$
- Then you will just return:

$$P\left(\hat{\theta} - \kappa_{1-\alpha}\hat{\sigma} \le \theta \le \hat{\theta} - \kappa_{\alpha}\hat{\sigma}\right) \ge 1 - 2\alpha,$$

where  $\kappa_{\alpha},\kappa_{1-\alpha}$  are the quantiles of  $(\hat{\theta}-\theta)/\hat{\sigma}$ 

• The distribution of  $(\hat{\theta} - \theta)/\hat{\sigma}$  depends on *P*.

- Data  $X_1, \ldots, X_n \stackrel{\mathsf{iid}}{\sim} P$
- Some estimator  $\hat{\theta}$  of parameter of interest  $\theta$ .
- For some coverage  $\alpha$ , want to produce a lower and upper bound such that:

$$P\left(\hat{a} \leq \theta \leq \hat{b}\right) \geq 1 - 2\alpha,$$

- Say you know the distribution of  $(\hat{ heta} heta)/\hat{\sigma}$
- Then you will just return:

$$P\left(\hat{\theta} - \kappa_{1-\alpha}\hat{\sigma} \le \theta \le \hat{\theta} - \kappa_{\alpha}\hat{\sigma}\right) \ge 1 - 2\alpha,$$

where  $\kappa_{\alpha},\kappa_{1-\alpha}$  are the quantiles of  $(\hat{\theta}-\theta)/\hat{\sigma}$ 

- The distribution of  $(\hat{\theta} \theta)/\hat{\sigma}$  depends on *P*.
- Often this distribution is normal, but with unknown parameters.

- The trouble is we don't know P.
- What will we do if we did know P?

- The trouble is we don't know P.
- What will we do if we did know P?
- Draw B datasets of size n from P

- The trouble is we don't know P.
- What will we do if we did know P?
- Draw B datasets of size n from P
- For the  $i^{th}$  dataset, calculate  $\hat{\theta}^{(i)}$

- The trouble is we don't know P.
- What will we do if we did know P?
- Draw B datasets of size n from P
- For the  $i^{th}$  dataset, calculate  $\hat{\theta}^{(i)}$
- Now get the distribution of  $\hat{\theta}^{(1)}, \dots, \hat{\theta}^{(B)}$  and get the C.I.

- The trouble is we don't know *P*.
- All we have at hand is the *n* datapoints  $x_1, \ldots, x_n$

- The trouble is we don't know *P*.
- All we have at hand is the *n* datapoints  $x_1, \ldots, x_n$
- So we put a 1/n mass on each datapoint to get a empirical distribution  $\hat{P}$

- The trouble is we don't know P.
- All we have at hand is the *n* datapoints  $x_1, \ldots, x_n$
- Drawing *n* points from this distribution boils down to?

- The trouble is we don't know P.
- All we have at hand is the *n* datapoints  $x_1, \ldots, x_n$
- So we put a 1/n mass on each datapoint to get a empirical distribution  $\hat{P}$
- Drawing *n* points from this distribution boils down to?
- Sampling with replacement!

#### True model Bootstrapped model



How do you estimate P?

Empirical Bootstrap

$$\hat{P} = \frac{1}{n} \sum_{i} \delta(x_i)$$

Generate *m* samples  $(X_1^*, \ldots, X_n^*)^{(j)}$ , j = 1 : m. Each giving a  $(\hat{\theta}^*, \hat{\sigma}^*)$  pair. Compute the  $\kappa_{\alpha}$  quantile of the distribution of  $\frac{\hat{\theta}^* - \hat{\theta}}{\hat{\sigma}^*}$ 

Parametric bootstrap  $\hat{P} = P_{\hat{\theta}}$ 

Lets try the simplest setting with  $\theta = \mu := E[X_1]$ 

• What is the expectation of the bootstrapped mean  $\bar{X}^{\ast}$  given the data?

Lets try the simplest setting with  $\theta = \mu := E[X_1]$ 

• What is the expectation of the bootstrapped mean  $\bar{X}^*$  given the data?

$$E[\bar{X}^*|X_1,\ldots,X_n] = E\left[\frac{1}{n}\sum_i X_i^*|X_1,\ldots,X_n\right]$$
$$= E[X_1^*|X_1,\ldots,X_n]$$
$$= \sum_{i=1}^n X_i \times n = \bar{X}$$

# **Empirical bootstrap**

Lets try the simplest setting with  $\theta = \mu := E[X_1]$ 

• What is the variance of the bootstrapped mean  $\bar{X}^*$  given the data?

## **Empirical bootstrap**

Lets try the simplest setting with  $\theta = \mu := E[X_1]$ 

• What is the variance of the bootstrapped mean  $\bar{X}^*$  given the data?

$$\operatorname{var}[\bar{X}^*|X_1,\ldots,X_n] = \operatorname{var}\left[\frac{1}{n}\sum_{i=1}^n X_i^*|X_1,\ldots,X_n\right]$$
$$= \frac{1}{n}\operatorname{var}\left[X_1^*|X_1,\ldots,X_n\right]$$
$$= \frac{1}{n}\left(E[(X_1^*)^2|X_1,\ldots,X_n] - \bar{X}^2\right)$$
$$= \frac{1}{n}\underbrace{\left(\frac{1}{n}\sum_i X_i^2 - \bar{X}^2\right)}_{\text{Sample Variance}}$$

### **Empirical bootstrap**

Lets try the simplest setting with  $\theta = \mu := E[X_1]$ 

• What is the variance of the bootstrapped mean  $\bar{X}^*$  given the data?

$$\operatorname{var}[\bar{X}^*|X_1,\ldots,X_n] = \operatorname{var}\left[\frac{1}{n}\sum_{i=1}^n X_i^*|X_1,\ldots,X_n\right]$$
$$= \frac{1}{n}\operatorname{var}\left[X_1^*|X_1,\ldots,X_n\right]$$
$$= \frac{1}{n}\left(E[(X_1^*)^2|X_1,\ldots,X_n] - \bar{X}^2\right)$$
$$= \frac{1}{n}\underbrace{\left(\frac{1}{n}\sum_i X_i^2 - \bar{X}^2\right)}_{\text{Sample Variance}}$$

• This makes sense, since the sample variance converges to the true variance, and we all know that the variance of  $\bar{X}$  is exactly  $\sigma^2/n$ 

But whats the point then? Can't you just do some Taylor approximation or something, and get the final distribution?

• What is the asymptotic distribution of the median of *n* i.i.d r.v.s drawn from *P*?

- What is the asymptotic distribution of the median of *n* i.i.d r.v.s drawn from *P*?
- Its a normal, of course, like a lot of other estimators.

- What is the asymptotic distribution of the median of *n* i.i.d r.v.s drawn from *P*?
- Its a normal, of course, like a lot of other estimators.
- With variance  $\frac{1}{4nf(\tilde{\mu})^2}$ , where  $\tilde{\mu}$  is the population median and f is the density of P

- What is the asymptotic distribution of the median of *n* i.i.d r.v.s drawn from *P*?
- Its a normal, of course, like a lot of other estimators.
- With variance  $\frac{1}{4nf(\tilde{\mu})^2}$ , where  $\tilde{\mu}$  is the population median and f is the density of P
- If we don't know P, we can't evaluate the above.

Does it always work?

Does it always work? Lets try the maximum of  $X_1, \ldots, X_n \stackrel{\mathsf{iid}}{\sim} U([0, \theta])$ 

• What is the true limiting distribution?

Does it always work? Lets try the maximum of  $X_1, \ldots, X_n \stackrel{\text{iid}}{\sim} U([0, \theta])$ 

• What is the true limiting distribution?

$$P\left(\frac{n(\theta-X_{(n)})}{\theta} > x\right) = P\left(X_{(n)} \le \theta(1-x/n)\right) = (1-x/n)^n \to e^{-x}$$

• The bootstrapped limiting distribution

$$P\left(\frac{n(X_{(n)} - X_{(n)}^*)}{X_{(n)}} = 0\right) = P(X_{(n)}^* = X_{(n)}) = (1 - (1 - 1/n)^n) \to 1 - 1/e$$

Does it always work?

Does it always work?

- Rule of thumb: when the asymptotic distribution is normal.
- Another con is it will take forever if n is large, even if you parallelize
- What do you do when its not?

- Starts with the realization that instead of drawing with replacement, its better to draw without replacement smaller samples
- This is in some sense, a more honest representation or approximation of the unknow distribution *P*

- Starts with the realization that instead of drawing with replacement, its better to draw without replacement smaller samples
- This is in some sense, a more honest representation or approximation of the unknow distribution *P* 
  - Draw B size b subsamples without replacement
  - For each, compute your estimator  $\hat{\theta}$
  - Now get confidence intervals or variance of this distribution

- Starts with the realization that instead of drawing with replacement, its better to draw without replacement smaller samples
- This is in some sense, a more honest representation or approximation of the unknow distribution *P* 
  - Draw B size b subsamples without replacement
  - For each, compute your estimator  $\hat{\theta}$
  - Now get confidence intervals or variance of this distribution
- But now everything is on a different scale!
- For example, the standard dev. of the mean decays at a rate of  $1/\sqrt{n}$
- If you use subsampling, the numbers you will get will be  $1/\sqrt{b}$

- Starts with the realization that instead of drawing with replacement, its better to draw without replacement smaller samples
- This is in some sense, a more honest representation or approximation of the unknow distribution *P* 
  - Draw B size b subsamples without replacement
  - For each, compute your estimator  $\hat{\theta}$
  - Now get confidence intervals or variance of this distribution
- But now everything is on a different scale!
- For example, the standard dev. of the mean decays at a rate of  $1/\sqrt{n}$
- If you use subsampling, the numbers you will get will be  $1/\sqrt{b}$
- What to do? You will need to analytically correct the variability.

#### Pros

- Very fast, specially you have a super-linear estimation algorithm
- Works for statistics which bootstrap doesnt work for, i.e. requires far less conditions, as long as *b* grows to infinity with *n*, but at a slower rate.

#### Cons

- Very sensitive to the choice of b (next two slides)
- You need to know the scaling factor to correct for using b < n

- Multivariate linear regression with d = 100 and n = 50,000 on synthetic data.
- *x* coordinates sampled independently from StudentT(3).
- y = w<sup>T</sup>x + ε, where w in R<sup>d</sup> is a fixed weight vector and ε is Gaussian noise.
- Estimate  $\theta_n = w_n$  in  $\mathbb{R}^d$  via least squares.
- Compute a marginal confidence interval for each component of w<sub>n</sub> and assess accuracy via relative mean (across components) absolute deviation from true confidence interval size.
- For subsampling, use  $b(n) = n^{\gamma}$  for various values of  $\gamma$ .
- Similar results obtained with Normal and Gamma data generating distributions, as well as if estimate a misspecified model.

# Subsampling - cons



- In between subsampling and bootstrap
- Draw size m w/o replacement samples from the data
- Draw size *n* with replacement samples from each subsample

- Three main parts+ $\epsilon$
- Large scale optimization:
  - Gradient descent, Newton Raphson
  - Stochastic gradient descent, proximal methods, subgradients, dual coordinate ascent, etc.

- Three main parts+ $\epsilon$
- Large scale optimization:
  - Momentum methods:
    - SGD has trouble navigating ravines, i.e. areas where the surface curves much more steeply in one dimension than in another , which are common around local optima.
    - Momentum helps accelerate SGD in the correct direction by damping oscillation
    - It does this by adding a fraction of the update vector of the past time step to the current update vector:

# Summary

- Three main parts
- Large scale optimization:
  - Adaptive methods:
    - John Duchi, Elad Hazan, Yoram Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization." Journal of Machine Learning Research 2011
    - Adaptively learn learning rates for different coordinates slow learning rates for frequent features, and large ones for infrequent features
    - Unfortunately the squared gradients keep accumulating and eventually learning rate goes to zero.
    - Diederik, Kingma; Ba, Jimmy (2014), "Adam: a Method for Stochastic Optimization"
    - ADAM uses exponentially decaying average of past squared gradients, and also does bias correction by estimating moments.

# Summary

- Large scale optimization:
  - Stochastic gradient descent
    - Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Advances in neural information processing systems, pages 315–323, 2013
    - Main point: Talks about dual coordinate ascent and shows how this leads to variance reduction

## Summary

- Large scale optimization:
  - Stochastic gradient descent
    - Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In Advances in neural information processing systems, pages 315–323, 2013
    - Main point: Talks about dual coordinate ascent and shows how this leads to variance reduction
    - Wilson et al., The Marginal Value of Adaptive Gradient Methods in Machine Learning (NeurIPS 2017)
    - Talks about pitfalls of Adaptive methods using a simple overparameterized problem
    - Feng Niu, Benjamin Recht, Christopher Re, Stephen J. Wright, Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent", NIPS 2011.
    - Asynchronous SGD without locks-use the sparsity in data

- Nearest neighbor methods: locality sensitive hashing, random projections and Johnson-Lindenstrauss, tree structures
  - Random Features for Large-Scale Kernel Machines, Ali Rahimi, Ben Recht, NIPS 2007
  - Random hash functions to project data to a low dimensional space so that the inner products of the transformed data are approximately equal to those in the feature space of a kernel.
  - Weinberger, Kilian, et al. "Feature hashing for large scale multitask learning." ICML, 2009.
  - Random projection type hash functions to bring high dimensional data down to lower dimensional space while not affecting the dot products (which are important for a various number of tasks).

- PCA, Spectral clustering
- Semisupervised learning, Pagerank, connection using random walks
- Power method for eigenvectors
- Networks: blockmodels, mixed membership models, connections to spectral clustering
- Topic models: connection to mixed membership models and corner finding algorithms
- Bootstrap and subsampling