

Dependent Random Weighting

Katherine Goode and Shan Yu

05/04/2018

Introduction

We were interested in learning about resampling methods for irregularly spaced time series data. This led us to read the paper

"The Dependent Random Weighting" (2015) by Srijan Sengupta, Xiaofeng Shao, and Yingchuan Wang.

The paper:

- Introduces a method that assigns random weights to the irregular time series data
- Weights are created using a dependence structure that mimics that of the observed data

Irregular Time Series Data

Irregular time series data can occur in two ways.

1. **Missing Values:** Time series occurs at equally space intervals but not all data points are observed



2. **Unequal Intervals:** Times when the data are observed are generated from a 1-D point process



Dependent Random Weighting

Assign a random weight to each observation

- A stationary time series $\{X_t\}_{t \in \mathbb{Z}}$. And the parameter of interest is $\theta = T(F)$, where T is a given function and F is the marginal distribution of $\{X_t\}$.
- The estimator of θ is $\widehat{\theta}_n = T(F_n)$, where F_n is the empirical distribution function based on observations $\{X_{t_j}\}_{j=1}^n$ and t_j are the time points at which the data are observed.
- The random weighted empirical distribution F_n^* is defined as

$$F_n^*(x) = \sum_{i=1}^n w(t_i) I(X_{t_i} \leq x),$$

where $\{w(t_i)\}_{i=1}^n$ are the random weights.

Dependent Random Weighting

- The bootstrap sample is

$$\hat{\theta}_{n,DRW}^* = T(F_n^*).$$

- **Example:** If we are interest in the marginal expectation of X_t , then we have

$$\bar{X}_{n,DRW}^* = \sum_{j=1}^n w(t_j) X_{t_j}.$$

Generate Weights

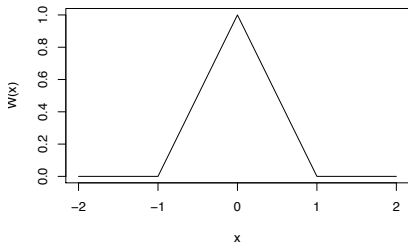
- Assume the random weights $\{w(t_i)\}_{i=1}^n$ take the form

$$w(t_i) = \frac{Z(t_i)}{\sum_{i=1}^n Z(t_i)}.$$

- $Z(t_i)$ are a realization from a **non-negative** and **l -dependent** process $Z(t)$, $t \in R$.
- l plays a similar role as the block size in the moving block bootstrap.

Generate Weights

Example: $Z(t_i) = (Y(t_i) + c)^2$,
where $\{Y(t_i)\}_{i=1}^n \sim N(0, \Sigma)$. Σ
is a $n \times n$ matrix with
 $\Sigma(i, j) = W\left(\frac{t_i - t_j}{l}\right)$, where $W(\cdot)$
is a symmetric kernel function.



$$l = 2$$

$$\begin{pmatrix} 1 & 0.5 & 0 & 0 \\ 0.5 & 1 & 0.5 & 0 \\ 0 & 0.5 & 1 & 0.5 \\ 0 & 0 & 0.5 & 1 \end{pmatrix}$$

$$l = 3$$

$$\begin{pmatrix} 1 & 2/3 & 1/3 & 0 \\ 2/3 & 1 & 2/3 & 1/3 \\ 1/3 & 2/3 & 1 & 2/3 \\ 0 & 1/3 & 2/3 & 1 \end{pmatrix}$$

Equally Spaced Time Series

Theorem 1. Assume that the function H is differentiable in a neighborhood of μ , under the assumptions in paper, then

$$\sup_{x \in \mathbb{R}} |P[\sqrt{n}\{H(\bar{X}_n) - H(\mu)\} \leq x] - P^*[\sqrt{n}\{H(\bar{X}_{n,DRW}) - H(\bar{X}_n)\} S_Z \leq x]| \rightarrow o_p(1).$$

Irregularly Spaced Time Series

Based on a stochastic sampling design, $t_j = \lambda_n v_j$, where v_j takes values in a Borel subset of $(-1/2, 1/2]$. Let $k = \lim_{n \rightarrow \infty} n/\lambda_n$.

Theorem 2. Under the assumptions in the paper, we have

1. if $k \in (0, \infty)$, then

$$\sup_{x \in \mathbb{R}} |P[\sqrt{n}\{\bar{X}_n - \mu\} \leq x] - P^*[\sqrt{n}\{\bar{X}_{n,DRW} - \bar{X}_n\}S_Z \leq x]| \rightarrow o_p(1).$$

2. if $k = \infty$, then

$$\sup_{x \in \mathbb{R}} |P[\sqrt{\lambda_n}\{\bar{X}_n - \mu\} \leq x] - P^*[\sqrt{\lambda_n}\{\bar{X}_{n,DRW} - \bar{X}_n\}S_Z \leq x]| \rightarrow o_p(1).$$

Our Simulations: Overview

We wanted to apply and compare DRW to methods learned in STAT 651. We decided to compare the following situations.

- **Methods:** DRW versus MBB
- **Data:** MA versus AR time series
- **Estimators:** mean versus median
- **Bandwidth:** block size versus l -dependence

Note on irregular data type:

- Paper used unequal time intervals (type 2)
- We used equal time intervals with missing values (type 1)

Our Simulations: The Procedure

We used the following procedure for our simulations.

1. **Generate irregular time series of size $n = 400$.**

- (i) Simulate Y_t for $t = 1, \dots, n$ from
 - an MA(2) process with $\mu = 0$, $\theta_1 = -1$, and $\theta_2 = 0.7$ or
 - an AR(2) process with $\mu = 0$, $\phi_1 = -0.1$, or $\phi_2 = 0.6$.
- (ii) Assign a weight ω_t to Y_t where

$$\omega_t = \sin \left(\frac{\pi \cdot t}{n} \right).$$

- (iii) Generate $Z_t \sim \text{binomial}(\omega_t)$ for $t = 1, \dots, n$.
- (iv) Let

$$X_t = \begin{cases} Y_t & \text{if } Z_t = 1 \\ \text{missing} & \text{if } Z_t = 0 \end{cases}$$

for $t = 1, \dots, n$.

- (v) Re-index the non-missing X_t as X_i for i from 1 to n_j and use as the observed sample.

Our Simulations: The Procedure

2. **Let $\ell = 1$, and apply the resampling method to $K = 1000$ samples.**
 - MBB: Draw block bootstrap samples from X_1, \dots, X_{n_j} with blocks of size $b = \ell$. (ignores missing values)
 - DRW: Randomly assign weights to X_1, \dots, X_{n_j} using the method from the paper assuming m -dependence with $m = \ell$.
3. **Compute the mean and median from the K samples.**
4. **Use the distributions of means and medians to compute evaluative measures.**
 - Determine if the 95% confidence interval contains the true value. (True process medians were approximated using 100,000 Monte Carlo simulations.)
 - Compute the standard deviation of the distribution. (Denote this as $\sigma_{n_j}^{(j)} / \sqrt{n_j}$.)

Our Simulations: The Procedure

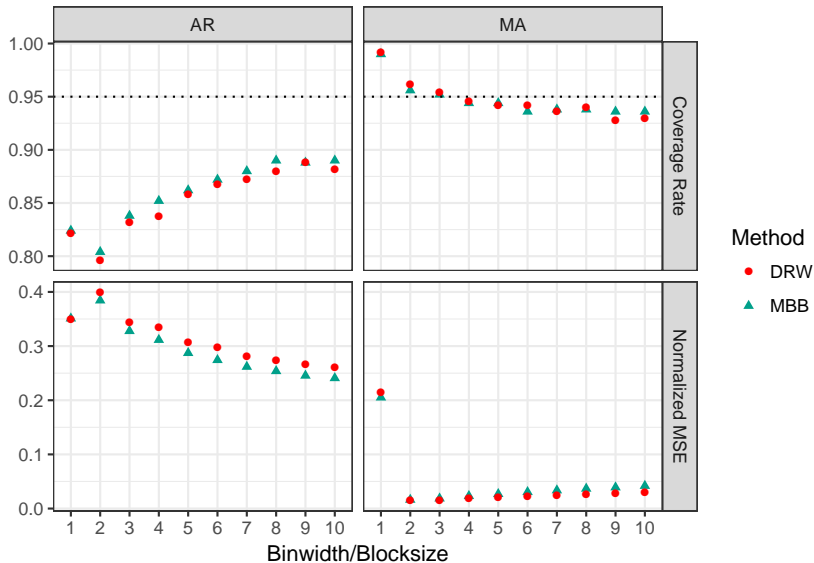
5. Repeat steps 1 to 4 for $M = 500$ times.
6. Compute final evaluative measures.
 - Coverage rate for both the mean and median
 - Normalized MSE:

$$\frac{1}{M} \sum_{j=1}^M \left(\frac{n_j \sigma_{n_j}^{(j)}}{n \sigma_n} - 1 \right)^2$$

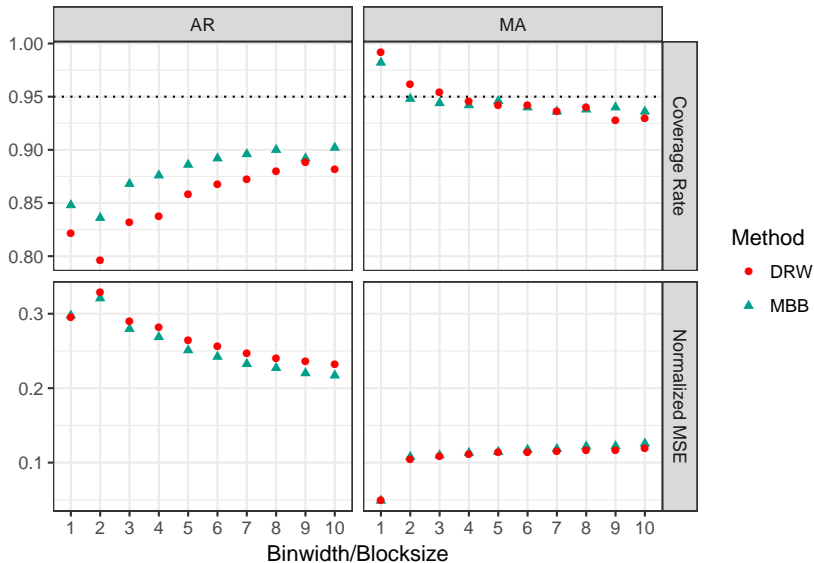
where $\sigma_n = \sqrt{n \text{Var}(\hat{\theta}_n)}$ with $\hat{\theta}_n$ denoting the estimator of interest, was approximated using 100,000 Monte Carlo simulations for both the mean and median

7. Repeat steps 1 to 6 for $\ell = 2, \dots, 10$.

Our Simulations: Results for Means



Our Simulations: Results for Medians



Our Simulations: Results for Computing Time

We wanted to compare computing times since the paper mentioned that DRW should be easier to implement.

- MBB simulations run on a personal computer (1 core)
- DRW simulations run on the ISU Condo Cluster (10 cores)

We found that the process took much longer for the DRW than the MBB even when run on a more powerful computer.

	MBB (personal computer)	DRW (ISU Condo)
AR	0.42	6.63
MA	0.41	6.48

Table 1: Computing times (in hours) for full simulation process within a category

Conclusions

Our simulations provided us with the following information.

- DRW is a new way to conduct bootstrap
- DRW results were usually similar or worse than MBB results
- DRW took more time than MBB

It would be interesting to run more simulations to consider:

- Would results change if different parameters were used to simulate AR and MA processes?
- How would different amounts or locations of missingness affect the results?
- How much would different sample sizes affect the results?