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 (the process)

- 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

$$\widehat{\theta}^*_{n,DRW} = T(F^*_n).$$

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

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

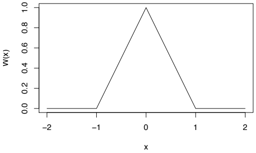
Dependent Random Weighting (theorems)

- 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 $Z(t)$, $t \in R$.

- **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.



Dependent Random Weighting (their simulations)

...

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**: blocksize 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 $n = 200$.**

   (i) Simulate y_t for $t = 1, ..., n$ from
   - an MA process with $\mu = 0$, $\theta_1 = -1$, and $\theta_2 = 0.7$ or
   - an AR 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 $z_t \sim \mathit{binomial}(\omega_t)$ for $t = 1, ..., n$.

   (iv) Let

   $$x_t = \begin{cases} y_t & \text{if } z_t = 1 \\ \text{missing} & \text{if } z_t = 0 \end{cases}$$

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

   (v) Reindex 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 $x_1, ..., x_{n_j}$ with blocks of size $b = \ell$. (ignores missing values)
   - DRW: Randomly assign weights to $x_1, ..., x_{n_j}$ 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. (Note: 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
   - MSE:

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

   - Normalized MSE:

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

   - Note: $\sigma_n = \sqrt{n}Var(\hat{\theta}_n)$ where $\hat{\theta}_n$ is the estimator of interest. σ_n was approximated using 100,000 Monte Carlo simulations for both the mean and median.

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

Our Simulations: Results for Means

. . .

Our Simulations: Results for Medians

. . .

Our Simulations: Results for Computing Time

...

Conclusion and Ideas for Future Work

...