Example of a simulation project

Bootstrapping in a Time Series

Introduction

Bootstrapping is a statistical tool used to estimate a parameter of interest. Commonly it is used when said parameter of interest does not possess an asymptotic distribution which is well known or easy to work with. For an example of this consider an independent and identically distrusted (iid) sample, $x_1, x_2, \dots, x_n$ from some population and we want to estimate the population median using the sample median, $M(x_1, x_2, \dots, x_n)$. Since we do not know the population median it is unlikely we know anything about the population distribution, therefore figuring out the sampling distribution of the sample median is no easy task. Instead statisticians commonly use bootstrap methods, where one resamples n observations with replacement, $x_{i_1}, x_{i_2}, \dots, x_{i_n}$ from the original sample $x_1, x_2, \dots, x_n$ and then calculate the median of this new sample $M(x_{i_1}, x_{i_2}, \dots, x_{i_n})$. This process is then repeated several times and we obtain an empirical distribution of $M(x_1, x_2, \dots, x_n)$. This empirical distribution can then be used for construction of confidence intervals and hypothesis testing.

This technique has proved itself in the statistical community and works well for the iid case, but what happens when we violate the assumption of independence? In an autoregressive time series of order three, AR(3), for example we cannot employ the techniques applied above. This is because, in an AR(3) model and observation, x_t, depends on the three observations that preceded it, x_{t-1}, x_{t-2} and x_{t-3}, therefore if we use the technique above we would lose the dependence structure of this data. Therefore when boot strapping a time series we have to use

directly but also indirectly on all past values

techniques which induce this dependence relationship on our resamples. Several bootstrapping techniques have been proposed such as Block, Sieve, Markov, wild and local bootstraps (Wolfgang et. al. 2003). In this paper we will primarily focus on Block bootstrapping.

Block Bootstrap

Overview

This technique relies on the assumption that the time series is stationary. It is closely related to the nonparametric iid case described above, in the sense that we draw observations with replacement. It differs in that we draw the observations in blocks of size l. So if we have a sample of size n then we draw $^{n}/_{l}$ blocks of size l. After this we calculate the estimate of our parameter of interest and repeat. Then we use all these resample to estimate the distribution of the sample statistic, which allows us to construct confidence intervals and conduct hypothesis test (Bhattacharya 2005).

Determining the Blocks

Determining the block length, l, has no firm rule. When deciding on l the researcher must take into consideration the length of the time series, n, and the correlation structure. Remember the point of Block Bootstrapping is to preserve the correlation amongst the observations, so if the ACF or PACF shows significant correlation at lag, k, block length needs to be at least k and preferable much bigger than k (Buhlmann 1999). How much bigger will depend on n and number of blocks the research wants in the bootstrap, again no clear cut rule for this.

There are several variations on how to define the blocks that can be sampled in the bootstrap process. The most simple of which just partitions the data into $^{n}/_{l}$ length l blocks. In

the case where there are100 observations and block length is 10, our blocks would consist of $x_1 : x_{10}$, $x_{11} : x_{20}$, ... , $x_{91} : x_{100}$. This is referred to as non-overlapping blocks technique as no two unique blocks share an element in common. Another cleverly named technique is overlapping blocks. In this method all n/l length subset of the original time series form the blocks that we sample for the bootstrap. So in the above example our blocks would be $x_1 : x_{10}$, $x_2 : x_{11}$, $x_3 : x_{12}$, ... , $x_{91} : x_{100}$. A critique of the over lapping blocks technique is that now the first and last nine observations have a lower probability of being selected. This is because observations x_{10} to x_{90} are all in 10 of the blocks, but this is not the case for the other observations mentioned above (Mammen & Nandi 2006).

Notice the above two techniques produce bootstrapped time series that are not stationary. To see this we again consider the example. Say we are using either of the two techniques described above and we have an AR(1) process. Then if the first two blocks selected are $x_1 : x_{10}$ and $x_{81} : x_{90}$ Then in this bootstrapped series the covariance between the 9th and 10th observations, $\gamma(x_9, x_{10}) = \alpha_1^{10-9} = \alpha_1$, but the covariance between the 10th and 11th element, $\gamma(x_{10}, x_{81}) = \alpha_1^{81-10} = \alpha_1^{71}$ therefore the autocovariance function does not only depend on the lag. We can get around this non-stationary issue by determining block length using a geometric random variable. Again how define the probability of success parameter has no clear rule but the expected value of the length should equal the desired block length l.

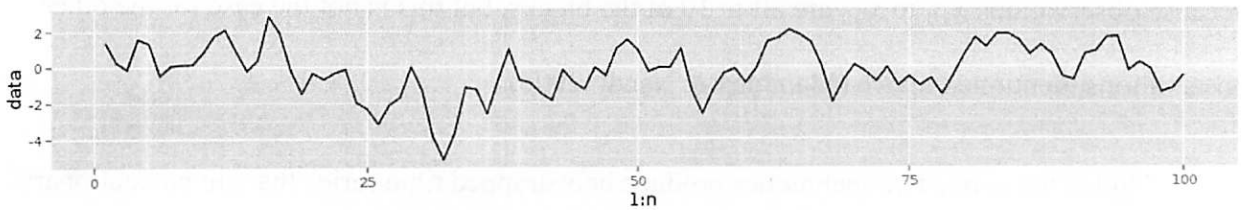
Confidence Interval

Constructing the confidence interval in a block bootstrapped time series is the same as the usual bootstrap case. Simulate a generous number of bootstraps and for each one calculate the parameter of interest. In doing so, we create the empirical distribution for our estimate of the

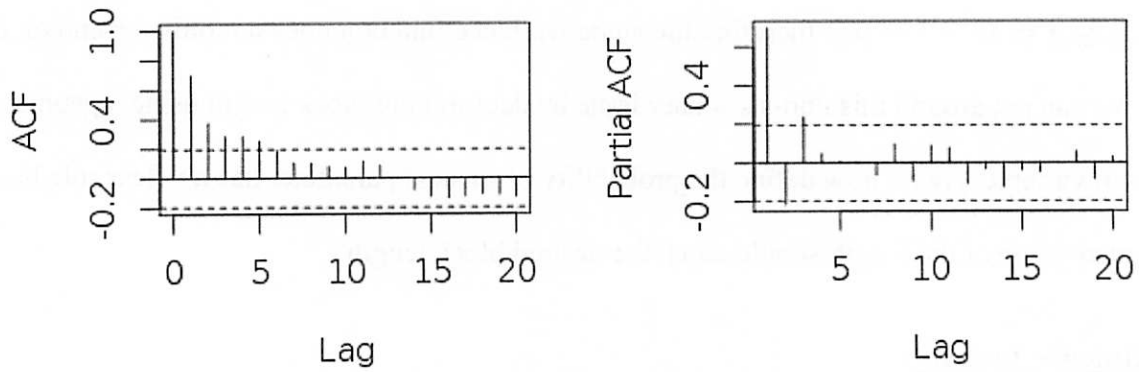
parameter. Then for a $(1-\alpha)*100\%$ report the $\frac{\alpha}{2}$ and $1-\frac{\alpha}{2}$ quantiles from the empirical distribution.

Example

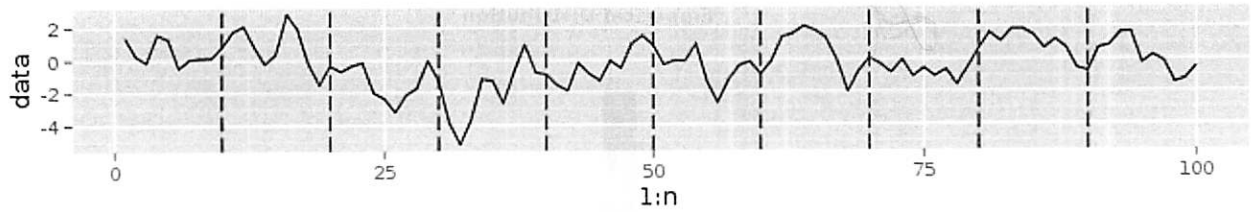
Consider the $AR(3)$ process $x_t = 0.9x_{t-1} - 0.5x_{t-2} + 0.3x_{t-3} + w_t$ where w_t is white noise with variance one. And say the parameter of interest is the median. Using `arima.sim` in R a simulated time series from this model could be



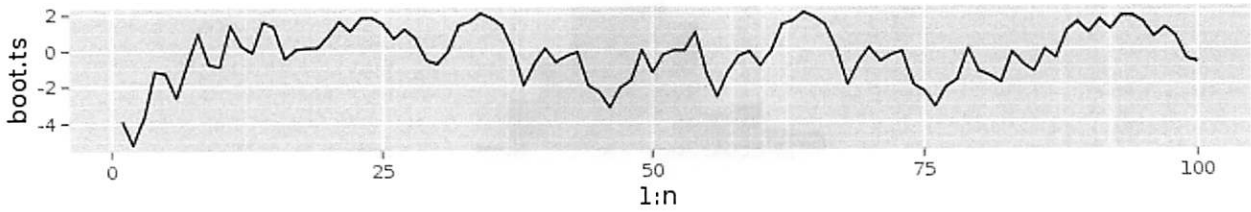
For illustration purposes I am going to use the non-overlapping block technique. To bootstrap this series we would first consider the acf and pacf, of course we know the underlying model to this time series but in practice we would not.



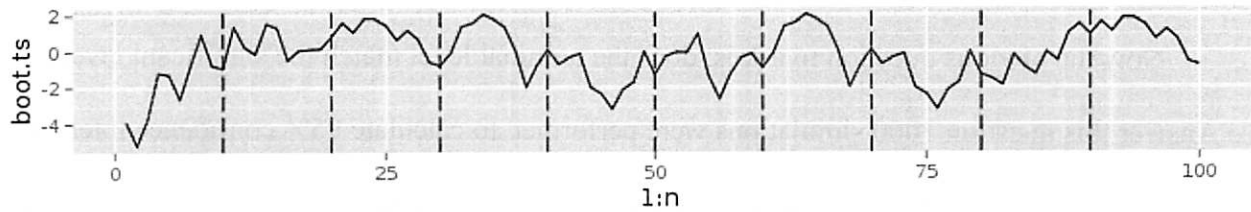
As we can see the highest significant lag comes from the ACF and it happens at lag 5. Since there is 100 observations, blocks of length 10 will be enough capture the dependence relationship. Now take the time series and divided it into 10 blocks of length 10.



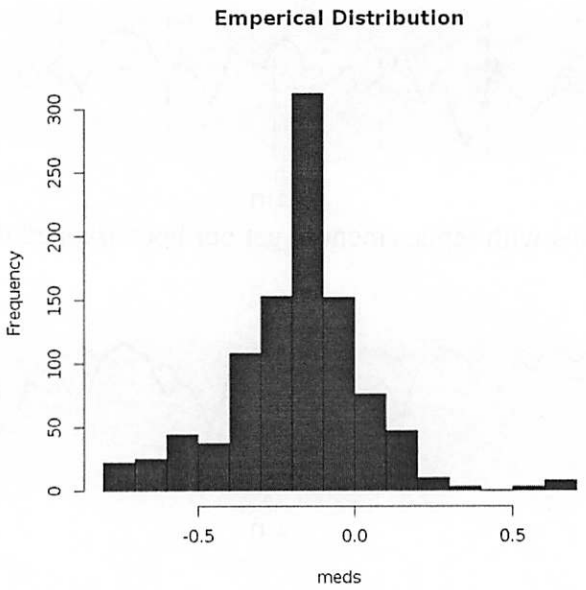
Now choose 10 of these bins with replacement to get our bootstrapped time series.



The first block of the bootstrapped time series came from the 4^{th} block of the original time series followed by the 1^{st}, 9^{th}, 7^{th} block and so on. This can be seen easier by dividing the bootstrapped time series into blocks



Once the bootstrapped time series is obtained simply calculate the parameter of interest. In this case it is the median, and for this bootstrapped series the median is -0.1067. Repeat this process 1000 times to obtain the empirical distribution.



Above is a histogram of the 1000 bootstrapped medians. Taking the 2.5 percentile and 97.5 percentile the 95% confidence interval from the median is -0.6831 to 0.1852. This does cover the population parameter which is 0.

Now the obvious question to ask is, does the nominal level match the significant level? To Answer this question 1000 simulations were performed to calculate 95% confidence intervals exactly the way describe above in the 1000 simulations the 95% confidence interval covered the true population median 872 times giving an average coverage probability 87.2%. Not what we were hoping for but not bad. Recall we only have 100 observations and there is so much variability associated with choosing the block length l.

Simulation

Lastly I would like to look at the issue of selecting block length while comparing non-overlapping block and overlapping block techniques. In order to do this I simulated 1000 time series using the same model as I did above $x_t = 0.9x_{t-1} - 0.5x_{t-2} + 0.3x_{t-3} + w_t$ where w_t is

white noise with variance one. Then for each of these time series I applied both of the techniques to obtain confidence intervals and then checked to see if the confidence interval contained 0. I repeated this for $l = 1, 2, 4, 5, 8, 10, 25, 50$.

Block Length	non-overlapping (coverage)	overlapping (coverage)	non-overlapping (length)	overlapping (length)
1	67.0%	67.4%	0.6946	0.9655
2	76.2%	75.6%	0.8171	0.8169
4	78.6%	77.8%	0.8717	0.8746
5	83.6%	82.6%	0.9897	0.9905
10	88.2%	85.6%	1.073	1.066
25	81.0%	79.4%	1.029	1.007
50	44.4%	54.6%	0.5146	0.6279

Notice first the case where block length is one, corresponds to the usual bootstrap where single elements are sampled with replacement, Therefore this case does not take into account the dependence relationship. As one can see it also preforms very poorly. In this simulation it appears the non-overlapping slightly outperformed the over-lapping block technique. This is not true for the simulations where block length is 1 or 50 but I am not concerned with these simulations as when $l = 1$ there is no difference in the techniques and when $l = 50$ the non-overlapping block technique only has two blocks per time series whereas the overlapping method has 51.It appears that coverage probability is maximized when block length is 10 for this case, but it still does not attain the desired coverage probability. This leads to the question of how sample size affects the coverage the probability. So let's dive into another simulation where we use the same underlying model, bootstrap with block size equals 10, and vary the sample size of the time series.

Sample Size	non-overlapping (coverage)	overlapping (coverage)	non-overlapping (length)	overlapping (length)
50	79.20%	76.00%	1.335	1.326
100	86.80%	85.20%	1.07	1.074
200	89.80%	88.60%	0.7843	0.7878
500	88.20%	87.80%	0.5141	0.5168
1000	89.60%	89.00%	0.3686	0.369

These results are surprising. I expected the coverage probability to attain 95% as sample size increased but it appears there is a maximum coverage probably. My predicted is that if you let block size increase with sample size the confidence intervals will attain the desired coverage probability. So if there really is 1000 observations we can allow our block size to be 50 or maybe even 100. These results again show the non-overlapping technique slightly outperforming the over-lapping block technique.

Conclusion

Bootstrapping can be used on time series data to construct confidence intervals and conduct hypothesis test. There are several techniques, but this paper focused on block bootstrap. Even within block bootstrapping there different options the researcher has for defining how they resample the blocks, this paper looked at overlapping and non-overlapping blocks and saw the non-overlapping technique slightly outperforming the over-lapping block technique. Defining block size is difficult and should be done while considering the sample size and the correlation structure.

References

Bhattacharya, A. (2005). Resampling in Time Series Models.1-34.

Buhlmann, P. (1999). Bootstraps for Time Series. Switzerland. 1-33.

Mammen, E & Nandi, S. 2006. Bootstrap and Resampling. *Computational statistics*. Section 2.4.1-2.4.7

Wolfgang Härdle, Joel Horowitz and Jens-Peter Kreiss *International Statistical Review / Revue Internationale de Statistique* , Vol. 71, No. 2 (Aug., 2003) , pp. 435-459

Appendix

R code

```
install.packages('ggplot2')
library(ggplot2)
n <- 100
l <- 10
nboot <- 1000
meds <- numeric(nboot)
true <- numeric(nsim)

##non-overlapping blocks
for(k in 1:nsim){
  data <- arima.sim(list(order=c(3,0,0),ar=c(0.9, -0.5, 0.3)),n=n)
  #qplot(1:n,data,geom='line')
  #qplot(1:n,data,geom='line')+geom_vline(xintercept =
seq(10,90,10),linetype = "longdash")
  for(j in 1:nboot){
    srts <- sample((0:(n/l-1)*l+1) , n/l, replace='T')
    boot.ts <- numeric(n)
    for(i in 1:(n/l)){
      boot.ts[(1+(i-1)*l):(l*i)] <- data[srts[i]:(srts[i]+l-1)]
    }
    #qplot(1:n, boot.ts, geom='line')
    #qplot(1:n, boot.ts, geom='line')+geom_vline(xintercept =
seq(10,90,10),linetype = "longdash")
    meds[j] <- median(boot.ts)
  }
  #true[k] <- ifelse(0 > quantile(meds,0.025) & 0 <
quantile(meds,0.975), 1,0)
  hist(meds,col='blue', main="Emperical Distribution")
}
mean(true)


##overlappin blocs
for(k in 1:nsim){
  data <- arima.sim(list(order=c(3,0,0),ar=c(0.9, -0.5, 0.3)),n=n)
  #qplot(1:n,data,geom='line')
  for(j in 1:nboot){
```

```
    srts <- sample(1:(n-l+1) , n/l, replace='T')
    boot.ts <- numeric(n)
    for(i in 1:(n/l)){
      boot.ts[(1+(i-1)*l):(l*i)] <- data[srts[i]:(srts[i]+l-1)]
    }
    #qplot(1:n, boot.ts, geom='line')
    meds[j] <- median(boot.ts)
  }
  true[k] <- ifelse(0 > quantile(meds,0.025) & 0 <
quantile(meds,0.975), 1,0)
  #hist(meds,col='blue')
}
mean(true)

##simulation code
n <- 1000
l <- 10
nboot <- 1000
meds <- matrix(NA, 2, nboot)
nsim <- 500
true <- leng <- matrix(NA,2,nsim)

for(k in 1:nsim){
  data <- arima.sim(list(order=c(3,0,0),ar=c(0.9, -0.5, 0.3)),n=n)
  for(j in 1:nboot){
    srts <- sample((0:(n/l-1)*l+1) , n/l, replace=T)
    boot.ts <- numeric(n)
    for(i in 1:(n/l)){
      boot.ts[(1+(i-1)*l):(l*i)] <- data[srts[i]:(srts[i]+l-1)]
    }
    meds[1,j] <- median(boot.ts)
  }

  for(j in 1:nboot){
    srts <- sample(1:(n-l+1) , n/l, replace=T)
    boot.ts <- numeric(n)
    for(i in 1:(n/l)){
      boot.ts[(1+(i-1)*l):(l*i)] <- data[srts[i]:(srts[i]+l-1)]
    }
    meds[2,j] <- median(boot.ts)
  }
  true[1,k] <- ifelse(0 > quantile(meds[1,],0.025) & 0 <
quantile(meds[1,],0.975), 1,0)
  true[2,k] <- ifelse(0 > quantile(meds[2,],0.025) & 0 <
quantile(meds[2,],0.975), 1,0)
  leng[1,k] <- quantile(meds[1,],0.975)-quantile(meds[1,],0.025)
  leng[2,k] <- quantile(meds[2,],0.975)-quantile(meds[2,],0.025)

}
apply(true,1,mean)
apply(leng,1,mean)
```