Prediction Function (R) Behaviour Analysis Report

Prepared by Xinyu Huang - 20615673, Bozhao Shi - 20653471, Lingli Xu - 20600103, Zihan Xu - 20610709

INTRODUCTION

Basic Concept of Time Series Analysis

A time series is a series of data points indexed in time order. It is considered stationary if its mean, variance and autocorrelation structure do not change over time, and non-stationary otherwise. Time series forecasting is the use of a model to predict future values based on previous observations (Zigu). In this report, the focus will be on the autoregressive model *AR(p)*. The model specifies that the output variable linearly depends on a certain number (*p*) of its own previous values and on a stochastic term (2018). *AR(p)* is one of the two components of the more general ARMA model, generalizing which we further get the ARIMA model *ARIMA(p,d,q)*. It implies that a differencing step can be applied d times on *ARIMA(p,d,q)* to obtain *ARMA(p,q)*. Thus, *ARIMA(p,0,0)* is the same as *AR(p)*, based on which we can use the `arima` function to generate the AR model in our analysis.

R's Methods of Estimating Parameters of ARIMA model

To fit an AR model on the data points, we will need to estimate its parameters. In R, the three methods of estimation in the `arima` function include CSS (Conditional Sum of Squares), ML (Maximum likelihood function) and CSS-ML. The ML method has a longer run time but its estimate is more accurate. Also, ML can only be used in a stationary process while CSS does not have the restriction. CSS-ML is a combination of the two and is default in the `arima` function. It first generates a rough estimate by CSS and uses that to initiate a more accurate estimate by ML. Thus, the results from CSS-ML and ML are close but CSS-ML will have a shorter run time. Since it involves ML, CSS-ML still requires the sample to be stationary. That limitation has affected how we chose the method for the two cases when fitting the AR model, which will be discussed in detail in the analysis section.

ANALYSIS

Main Steps

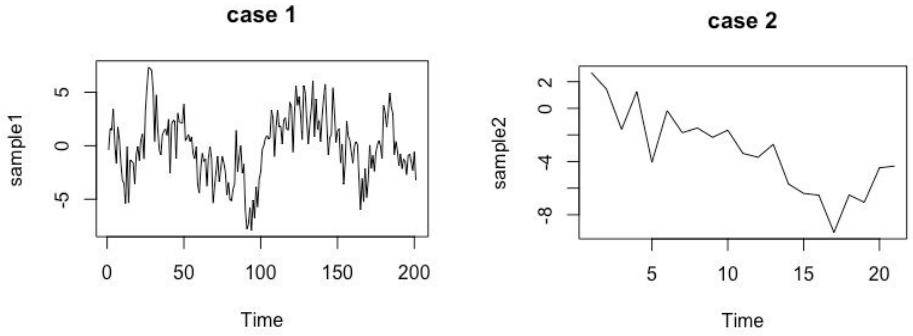
1. Choose an appropriate `N.sim`, the number of simulations, and repeat steps 2-4 for `N.sim` times
2. Use the `arima.sim` function to generate samples for case1 and case2, and choose the first 200 and 20 observations respectively as the training set. These values are considered to be observed linearly in time. Use the 201 and 21 values as test values of the two cases.
   The AR(2) model we used in this step is the one we would like to test:
   $X_t = 0.5X_{t-1} + 0.3X_{t-2} + Z_t$, where Z_t are independent and identically distributed as N(0,4).
3. Use the `arima` function to fit a new *AR(2)* model to the samples respectively, use the model to predict the value in $time_{201}$ for case1 and $time_{21}$ for case2, and then construct a 95% prediction interval for each case
4. Check if the test values fall in the corresponding prediction intervals
5. Get the two $\hat{\alpha}$, the estimated proportion of times that 95% prediction interval covers a test value
6. Evaluate how well the `arima` function for time-series prediction works based on $\hat{\alpha}$

Choose an Appropriate N.sim

Having $\hat{\alpha}$ calculated as $\hat{\alpha} = (\sum_{i=1}^{N.sim} I\{95\%\ prediction\ interval\ covers\ test\ value\})/N.sim$, where $I \sim$ Bernoulli(α), the standard error of $\hat{\alpha}$ is $se(\hat{\alpha}) = \sqrt{var(\hat{\alpha})} = \sqrt{\alpha * (1-\alpha)/N.sim}$. A smaller standard error will give us a more accurate estimate. The above formula indicates that $se(\hat{\alpha})$ decreases as `N.sim` increases, which means that we should choose `N.sim` as large as possible for accuracy. However, an overflow in `R` may occur if running too many simulations. Thus, we believe 10000 is a reasonable `N.sim` for our analysis since it is large enough for accuracy while it will not result in an overflow in `R`.

Choose a Method of Estimating Parameters of ARIMA model

When using the `arima` function to fit an *AR(2)* model to our simulated data points, we found that the default CSS-ML method could only be applied in case1 but often gives an error message when was applied in case2. The following plots for the samples of the two cases are to explain the cause.



The graph of a random sample1 implies that this sample is stationary while the one of a random sample2 shows a decreasing trend, so it's non-stationary. That is possibly caused by the small number of observations for case2 and the training set is not representative. This issue will induce the error message if using the default CSS-ML method. Thus, we used CSS to generate the fitted *AR(2)* model for case2.

Results and Thoughts

The results we obtained are presented as follows.

```
> #Case 1
> #Use simulator to estimate proportion with 200 obervations
> alpha1.hat=ar_simulator(200,"CSS-ML")
> alpha1.hat
[1] 0.9497
> #Calculator M-C standard error
> mc_error(alpha1.hat)
[1] 0.002185633
```

```
> #Case 2
> #Use simulator to estimate proportion with 20 observations
> alpha2.hat=ar_simulator(20,"CSS")
> alpha2.hat
[1] 0.9044
> #Calculate M-C standard error
> mc_error(alpha2.hat)
[1] 0.002940419
```

The results show that $\hat{\alpha}$ for case1 is very close to 0.95 while the one for case2 is not. Since we have used a 95% prediction interval, we consider as $\hat{\alpha}$ approaching 0.95, the prediction function is working better. To see the reason why the function works better for case1, we will take a closer look at the fitted AR model for the two cases in one simulation.

```
Call:
arima(x = train1, order = c(2, 0, 0), include.mean = F)

Coefficients:
         ar1     ar2
      0.5156  0.2917
s.e.  0.0674  0.0686

sigma^2 estimated as 4.333:  log likelihood = -430.87,  aic = 867.74
```

```
Call:
arima(x = train2, order = c(2, 0, 0), include.mean = F, method = "CSS")

Coefficients:
         ar1     ar2
      0.4222  0.2099
s.e.  0.2274  0.2323

sigma^2 estimated as 3.031:  part log likelihood = -39.47
```

The figure on the left presents the fitted *AR(2)* process where the first 200 data points from a random sample1 were used as the training set (case1). More specifically, the parameters ar_1 was 0.5156 and ar_2 was 0.2917, which were really close to the ar_1 and ar_2 (0.5 and 0.3 respectively) used to generate the sample. That is likely because the parameters were estimated by the CSS-ML method. Therefore, that fitted *AR(2)* process will give us a prediction close to the sample value produced by the `arima.sim` function. In other words, the $time_{201}$ observation in sample1 should have a probability approaching 95% to fall in the range of the 95% confidence interval of the predicted $time_{201}$ value by the fitted model, which means that $\widehat{\alpha}_1$ should approach to 0.95.

The figure on the right shows the fitted *AR(2)* process for case2. In this case, the parameters ar_1 was 0.4222 and ar_2 was 0.2099, which were far off compared to the desired ar_1 and ar_2. Since the number of observations is too small to form a stationary data series, we have used the CSS method in case2. The result indicates that the CSS method does not always produce the accurate *AR(2)* model (ar_1 as 0.5, ar_2 as 0.3). Therefore, the $time_{21}$ observation in the sample2 generated by the accurate *AR(2)* model, is not likely to have a probability approaching 95% to fall in the 95% prediction interval produced by that fitted *AR(2)* model, which means that $\widehat{\alpha}_2$ will not be close to 0.95.

CONCLUSION

For case 1, there are enough observations to compose stationary data points, thus we can use the CSS-ML method to give more accurate estimates on parameters to obtain a more precise fitted model. However for case 2, we do not have enough observations. Therefore, with non-stationary data, we can only apply the CSS method, which is likely to result in a less accurate *AR(2)* model.

To summarize, as what we have observed in the two cases, the prediction works well if the fitted model is close to the original model used to generate the sample. Furthermore, we can conclude that the prediction function works better if we have more previous observations since the fitted AR model can better describe the time series.

REFERENCES

1. Zigu. “Time Series Definition: Statistics Dictionary.” *MBA Skool-Study.Learn.Share.*, www.mbaskool.com/business-concepts/statistics/12649-time-series.html.
2. “Autoregressive Model: Definition & The AR Process.” *Statistics How To*, 1 June 2018, www.statisticshowto.datasciencecentral.com/autoregressive-model/.