stat 628 Module 1 Report

Bodi Yang, Mengqi Li, Shikun Liu

September 2020

1 Overview

We try to build up a model to predict the storage of the server, changing from time to time. According to the behavior of the server, this should be a time dependent linear model. The key point for our algorithm is to choose appropriate data points to build such linear model and make prediction of what time the server will reach its maximum storage capacity.

In order to send an early warning to the engineers, we choose to use the model to predict the moment when the server reaches 90% of the maximum capacity. In this way, they can have a certain amount of time to prepare accordingly. Our main idea is: 1) Get the valid data; 2) Choose the data we need for the model; 3) Build up the model and make some predictions.

2 Algorithm Description

2.1 Data cleaning and preparation

We started with turning the data into numeric. For the strings, it will be dealt into missing values. Then we delete all the missing data and use the remaining data to build up the model.

2.2 Select the suitable data

In our opinion, since the engineers may deal with the former groups, the former group has little effect on the follow-up. We assume that predicted model and the model for the former group are uncorrelated. Thus we use the most recent data points to construct our model.

The key point is how to find most recent data points. According to the storage behavior of the server, the point when engineer cleaned the server is the point when there is great gap of the storage before and after the server being cleaned. Considering that there are two kinds of breakpoints in the data: sudden increasing and sudden dropping. We decided to calculate the difference of the response variable between all two adjacent moments. Then we take the absolute value of the difference and find out the point with the largest absolute difference. We set this point as break-point and we treat the points after the break-point as most recent points.

As the storage capacity problem described, the storage data can be divided into two groups: the first group is the data before t=1100 which records the storage before the engineers cleaned the server and the second group is the data after t=1100 which records the storage after the engineers cleaned the server.

Thus, we will try to build our model based on the data after engineers cleaned the server.

2.3 Build up the linear model

After we get the great gap point (as mentioned in 2.2), we are able to build our linear model based on the data points after the break-point. We don't use the lm() function in the R, instead we calculate the intercept and slope directly since this is just simple linear regression in order to accelerate the code running time and save the storage.

2.4 Predict the time reaching 90% of maximum capacity

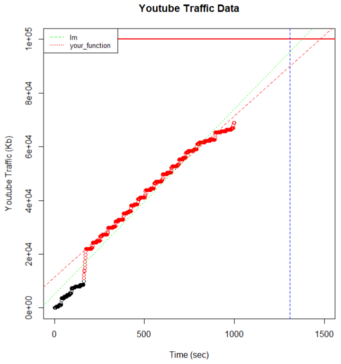
Finally we calculate the predicted time when server reaches 90% storage capacity since we want to warn the engineers ahead of the max out time. And we get relatively smaller running time compared to using lm() function directly.

3 Data Testing

We use three datasets to test the robustness of our algorithm and use linear regression algorithm using lm() function in R upon all data points as benchmark to make comparison. For the following figures, red points represent data that we used to fit the linear regression model. Red line is the regression line of our algorithm while the green line is the linear regression line using lm() function upon all data points. The vertical blue line denotes the time we use to warn that the storage is going to max out.

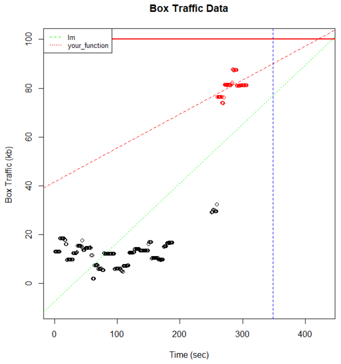
3.1 The first test dataset: Youtube Traffic Data

For the first dataset, the trend of the data before and after the break-point is basically the same and we use data points after the break-point to build the linear regression model. As the figure shows, we make an accurate prediction ahead of the time that storage will max out.



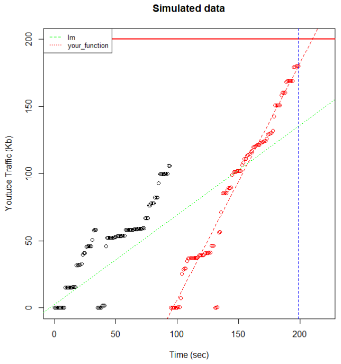
3.2 The second test dataset: Box Traffic Data

For the second dataset, there is a sudden increasing break-point. Before the break-point, the data is almost stable and time lag will exist if we use all data points to construct the linear model. However, we use the latter group of data points to construct the linear model and it predicts more accurate caompared to lm() function. As the figure shows we also warn the engineer timely.



3.3 The third test dataset: Simulation Data

For the third dataset, there is a similar pattern compared to the hard-drive problem. Around t=100, the traffic was cleaned up and after that the trend is upward again. The data points ahead of the cleaning time is meaningless because the trend is another stochastic behavior after cleaning up. As the figure shows, the time predicted by linear regression using all data point is much larger than our prediction. Our algorithm have better performance with regard to this dataset.



4 Summary

Our linear model has advantages in CPU time and storage compared with running lm() in R. And because of the strategy we adopting, by only using the last group of data after engineers cleaning the server, the accuracy also be improved with regard to this kind storage problem. In practical industry, our "90%" maximum capacity standard will change according to the specific situation to offer engineers enough time to do the adjustment to the server.