Time series modeling of Google Trend Data

Zhongling Jiang

October 28, 2016

Abstract

This report is about fitting time series model to the Google Trend Data and forecasting the future trend. The dataset used is 1DS.csv, which contains weekly onservations of a certain subject over 4 years. The goal is to predict the next 52 datapoints, equivalently a year of observations.

Introduction

In this report, in order to yield better prediction accuracy and model interpretability, I will use Validation set method, together with two different models to approach datasets.

- main model: differencing + ARIMA
- complementary model: Exponential Smoothing

Methods

Validation Set Approach

The dataset is splited into a training set and a validation set (or test set). In this mode, the validation set takes the last 52 values of both datasets as validation set, which is roughly 1 year of data. The model derives from the training set and is tested in validation set. The reason I prefer validation set approach over a multi-fold cross-validation is due to limited numbers of data points, i.e., 209 observations splitted among multi 52-obs folders may decrease model accuracy.

Differencing model & ARIMA

The first model involves differencing the data until the trend and seasonality components are removed. Ideally, we could get a white noise series 'd2' on which we apply ARIMA models. By calling predict() we obtain the predicted value for the next 52 weeks.

Exponential Smoothing

Exponential Smoothing can be used to make short-term forcast for time series data. It produces prediction values by assigning weight to previous and current values (α values). In this model, I will apply simple exponential smoothing to 'd2' series as an alternative to ARMA model. The idea behind this is to remove the effect of the siginificant spikes detected in d2 and its acf values, since exponential smoothing could 'smoothen out' the effect.

Analysis

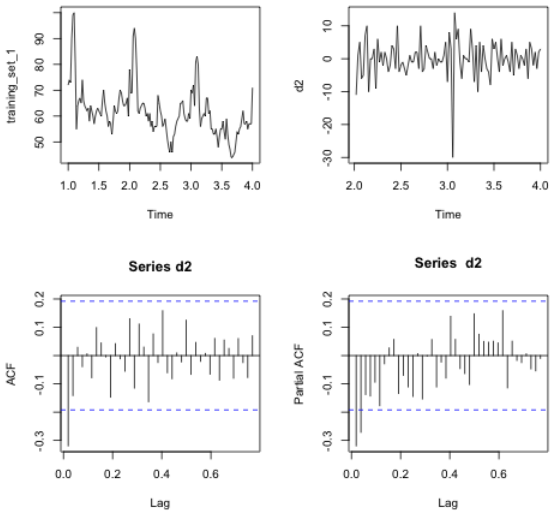
Pre-modeling Data Processing

There are several steps before we fit any model

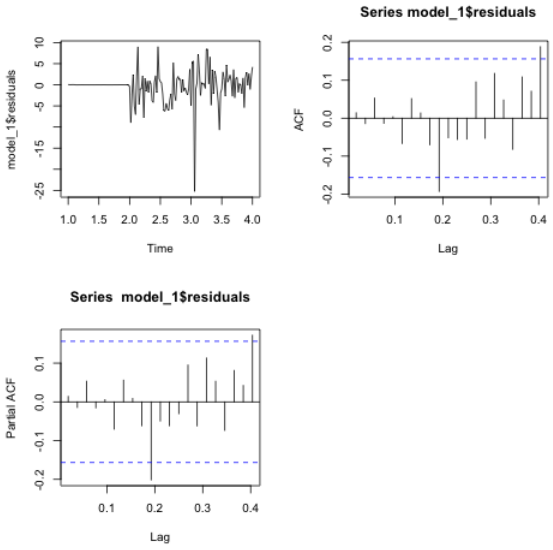
- turn data into time series object
- create training set and testing set

Differencing model and ARIMA

Figure 1.1 shows the time series d2 obtained by differencing once on seasonality component and once on trend component.



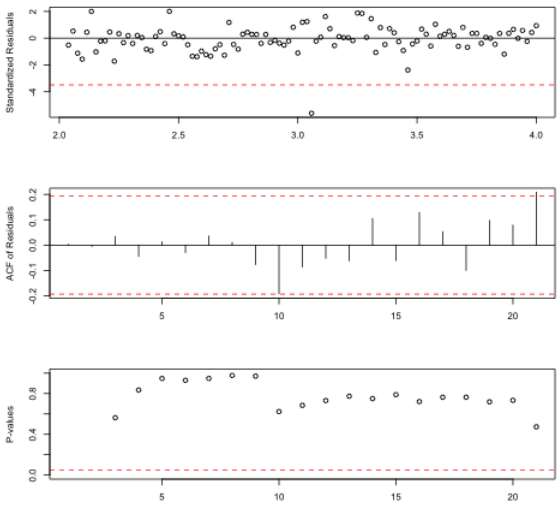
PACF tails off and acf also shows a tail off sign, so ARMA(1,1) could be a potentially fit model. Therefore, I fit ARIMA c(1,1,1) x c(0,1,0). Figure 1.2 shows the distribution of model residuals.



The Ljung-Box test shows the following result:

```
##
##  Box-Ljung test
##
## data:  model_1$residuals
## X-squared = 0.032419, df = 1, p-value = 0.8571
```

From the large p-values, we know that there is little evidence to show residuals are auto-correlated. Therefore, the model is valid. Similarly, the diagnostic plots (Figure 1.3) also shows reasonable residual plots, acf values and high p-values.



Cons: However, we observe that there is a spike around the year 3. There may also be connection between the spike and the abnormalities in acf/pacf values of model residuals, as shown in Figure 1.2. Since the existence of the spike brings difficulties to the modeling, it is better to remove it through smoothing techiniques, e.g. Exponential Smoothing.

Exponential Smoothing

By calling Holtwinters() on d2 series and setting $/beta$ = FALSE, $/gamma$ = FALSE, we apply the simple exponential smoothing to d2 series, which contains no trend and seasonality component. The resulting model is:

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = d2, beta = FALSE, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.1361129
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##        [,1]
## a 0.5891962
```

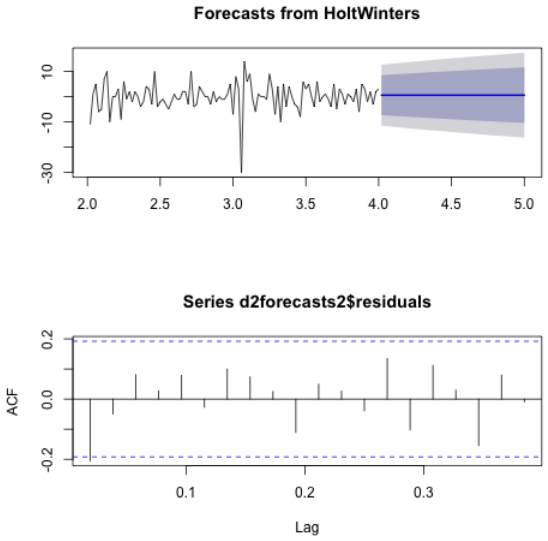
```
##
##  And the SSE is
```

```
## [1] 3920.655
```

```
##
##  Forecasted Values
```

```
## Time Series:
## Start = c(4, 2)
## End = c(5, 1)
## Frequency = 52
##  [1] 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962
##  [8] 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962
## [15] 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962
## [22] 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962
## [29] 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962
## [36] 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962
## [43] 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962 0.5891962
## [50] 0.5891962 0.5891962 0.5891962
```

Figure 2.1 shows the fitted values with forecasted 52 values with 95% confidence interval and acf plot of the new model.



I notice that the residuals acf plots behave more like white noise, and the Ljung-Box test also shows:

```
##
##  Box-Ljung test
##
## data:  d2forecasts2$residuals
## X-squared = 19.134, df = 20, p-value = 0.5131
```

Both indicate that Exponential Smoothing could be an ideal alternative in modeling d2 series. However, it is difficult to recover to the original series; only if we assume that the trend and seasonality components behave simiarly over time.

Model Selection

After we have confirmed the model, we want to select the best fitted model by testing the model on the testing set. One metric we use to measure the accuracy is Mean Square Error (MSE). We calculate the MSE using self-defined function computeMSE (see R Code) and eliminate models with overly large MSE. Another criteria we look at is Akaike information criterion (AIC), which penalizes if a model uses too many predictors. The ARIMA c(1,1,1) x (0,1,0) model has following MSE and AIC:

Property	Value
MSE	15.05677
AIC	617.9561

Model Fitting

At last I fit the model on the entire dataset. The predicted plot is shown in Figure 3.1

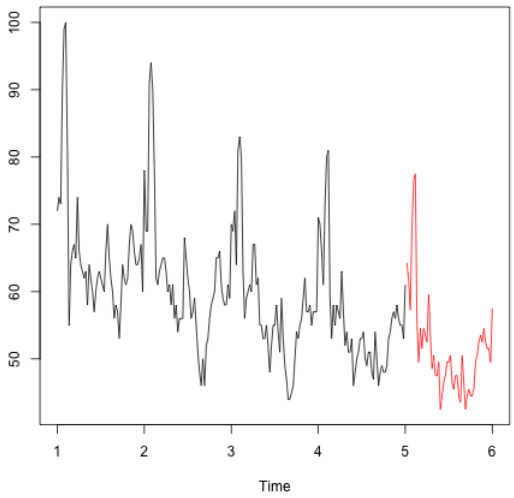


Figure 3.1

Conclusion

After comparing MSE and AICs from different ARIMA models, I finally choose ARIMA c(1,1,1)x c(0,1,0) as predicting model. However, the model fails to capture the spike that occurs in the residual plot, neither do slight abnomalities in acf and pacf plots. To address this problem, I try to fit Exponential Smoothing model to the 'de-trend' and 'de-seasonality' series. It turns out the residuals of forcasted values follows a white noise process, which shows Exponential Smoothing in fact performs better than ARIMA in forcasting residuals.