A comparison of ARIMA and GP model in time series prediction

Baosen Luo

December 2021

1 Introduction

The aim of the project is to compare two machine learning model's performance in predicting the evolution of a time series in the medium to long term. The models to be analyzed are Auto Regressive Integrated Moving Average(ARIMA) and Gaussian Process(GP). Both models are frequently used in the time series context, because they could model the dependencies across time like trend, periodicity, and seasonality reasonably well.

The dataset, NBA.CSV, used here is a time series of popularity of a term, NBA(National Basketball Association), searched in Google. [1] The dataset is obtained from Google Trends, which records Chrome users' browsing activity since 2004. The NBA.CSV is consisted by monthly popularity of the term from 2006 to 2019(168 samples, which correspond to 14 years). Github Link: A comparison of ARIMA and GP

2 Preliminary Data Analysis

The times series of NBA popularity from 2006 to 2019 is displayed in figure 1. Before building the model, the original dataset should be analyzed to uncover the underlying dependencies across time, which could help in model selection and parameter tuning.

One could easily identify three main features from the plot. Firstly, there is a gradual but consistent increasing trend. Secondly, there is a periodic repeating pattern of popularity within a year. Thirdly, there is a short term irregularity. Those features are consistent with reality and common sense. In the past two decades, the NBA's influence has continued to expand and become increasingly popular around the world. The periodic repeating pattern also synchronizes with the yearly calendar of the NBA. The play-offs and conference finals are held in summer, indicating the peak of popularity in the mid-year; the all-star games are usually held in winter, indicating the sub peak in the year end; off-seasons and regular-seasons in the rest of the year draw lower attention and thus correspond to the valley in the plot.

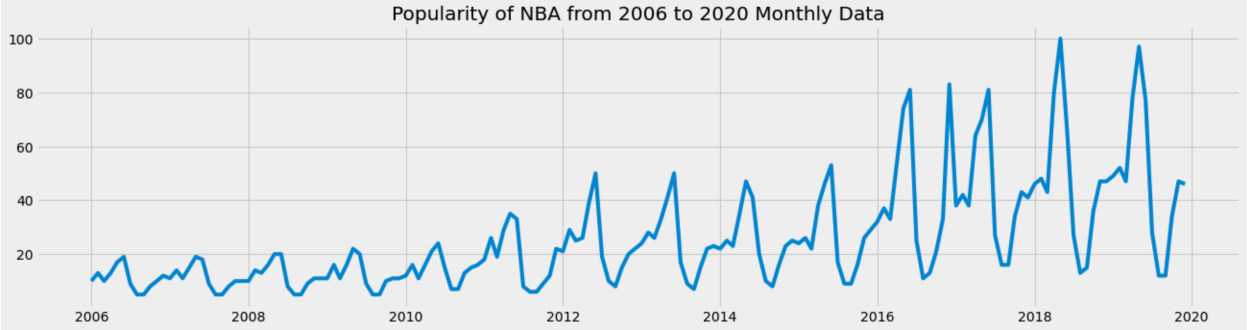

Figure 1: NBA Popularity from 2006 to 2019

3 ARIMA Model

3.1 An introduction to ARIMA Model

ARIMA stands for Auto Regressive Integrated Moving Average, which is a combination of the "autoregressive"(AR) terms and "moving average"(MA) terms, with a complementary "Integrated" component as a series might need to be diffrenced to be made stationary. A specific ARIMA model is often denoted by ARIMA(P,D,Q), where P, D and Q stand for the order of autoregressive, integration and moving average terms respectively. A series displays autoregressive behavior if it feels a "restoring force" that tends to pull it towards the mean. For example, in an AR(1) model, the magnitude of the coefficient determines how fast the series is converging towards the mean. If the coefficient is near zero, the series returns to its mean quickly; if the coefficient is near 1, the series returns to its mean slowly. A series displays moving average behavior if the error term or the "random shocks" that took place in the past continue to influence the consecutive periods. In an MA(1) model, the magnitude of the coefficient determines the fraction of the last period's shocks that is still felt in the current period. [2]

ARIMA is the most classical model for predicting time series that could be transformed into stationary series. A stationary time series has constant statistical properties. First of all, a stationary series has constant mean and no trend, so that the variations around the mean have constant amplitude. Secondly, the auto-correlation function(correlations of a random variable with its lagged counterpart) is only dependent on time lag and not on time. Generally speaking, a stationary series could be viewed as a combination of signal and noise. ARIMA model is trying to separate the signal from noise by fitting a pre-selected model that maximizes the likelihood of observations. And then, the signal is extrapolated to obtain forecasts in the future. The forecasting equation of a stationary time series is a linear equation in which the predictors consist of lagged dependent variables and lagged forecasting errors. [3] That is:

$$\hat{y}t = \mu + \phi_1 \cdot y_{t-1} + \cdots + \phi_p \cdot y_{t-p} + \theta_1 \cdot w_t + \cdots + \theta_q \cdot w_{t-q} \tag{1}$$

In addition, if the time series has a strong and consistent seasonal pattern, one has to add to the ARIMA model a seasonal difference term. For monthly data, in which there are

12 periods in a season, the seasonal difference of Y at time t is $Y_t - Y_{t-12}$. Based on the StatsModel's implementation, an ARIMA model with seasonal component is denoted as SARIMA(P,D,Q,s), where the first three terms mean exactly the same thing as above, and the last term corresponds to the order of seasonal difference. In practice, it is often set to be 12 for monthly data.

3.2 Construction of SARIMA Forecasting Model

The classical way to fit an ARIMA model is to follow the Box-Jenkins methodology, which uses ACF(auto-correlation function) and PACF(partial auto-correlation function) plots, and summary statistics to identify the trend, seasonality, and autoregression elements to get an idea of the amounts of difference and number of lags that will be required. However, the dataset used here presents complex underlying structure, and plots and statistics are not enough to identify the orders. Therefore, a random but reasonable guess of the parameters would be initiated and then cross validation technique would be performed to fine tune the parameters.

Since the seasonal difference term is set to be 12 by experience, the tasks now involve finding the best coefficients for the other three terms. By studying the original time series, one could observe a linear increasing trend. Therefore, it is a good idea to start with the first order integration difference. After taking the first order integration difference, the series $Y_t - Y_{t-1}$ is approximately stationary, but both the ACF and PACF plots tail off and could not provide robust suggestion for the number of lags that should be included in the model. Thus, a SARIMA(1,1,1,12) is set to be the initial baseline model. Then, the cross validation technique is performed to search for the best fitting parameters and it concludes that the SARIMA(4,1,1,12) yields the best forecasting performence in terms of minimizing the Mean Squared Error(MSE).

When the training set includes data from January, 2006 to July, 2018, and the testing set includes data from August, 2018 to December, 2019, the SARIMA model yields a MSE of 39.96 on the testing set. Finally, a rolling forecast would be performed on the same testing set. Unlike the train-test-split forecast, rolling forecast would train an initial SARIMA, but predict only one step into the future. After that, a new SARIMA model, incorporating that step's observation, would be re-created. The iteration process continues until all the testing data has been given a prediction. The rolling forecast yields a MSE of 30.53, which outperforms the train-test-split forecast. This result is consistent with the fact that rolling forecast has more information available at each prediction and thus has more predictive power.

4 GP Model

4.1 An introduction to GP Model

GP stands for Gaussian Process, which is a combination of the Gaussian Distribution and Stochastic Process. For the Gaussian Distribution, when the random variable is one dimensional, it is called univariate Gaussian with a probability density function(PDF) of $N(\mu, \sigma^2)$; when the dimensionality of the random variable rises to a finite p-dimensions, it is called a multivariate Gaussian with a PDF of $N(\mu_p, \sum_{p\times p})$. Gaussian Process is a stochastic process such that every set of those random variables has a multivariate normal distribution, in other words, Gaussian Process is an infinite-dimensional Gaussian Distribution. For a p-dimensional Gaussian, its statistical property is determined by two parameters: a p dimensional mean vector μ_p that indicates each random variable's expectation, and a $p \times p$ dimensional co-variance matrix $\sum_{p\times p}$ that reflects each random variable's variance and different random variables' co-variance. Similarly, for a GP that is defined on the continuous domain T, it needs to describe each time point's expectation and any two time points' co-variance. Since GP is equivalent to an infinite-dimensional Gaussian Distribution, mean vector and co-variance matrix could not work. Instead, a function about time $m(t)$ would be used to describe each time point's expectation and a kernel function $k(s,t)$ would be defined to describe any two time points' co-variance. [4]

Gaussian Process Regression(GPR) adopts the Bayesian Inference framework. At first, a prior version of $m(t)$ and $k(s,t)$ is set to define the GP. As the observations come in, GP's mean function and kernel function are then revised to obtain the posterior GP. Let the values and time index of a set of observations be denoted by vectors Y and X respectively. Accordingly, let the values and time index of non-observations be denoted by vectors $f(X^*)$ and X^*. Gaussian Distribution has a nice property, which is that Gaussian Distribution's joint, conditional, and marginal distributions are all normally distributed. It follows that:

$$\begin{bmatrix} Y \\ f(X^*) \end{bmatrix} \sim N(\begin{bmatrix} \mu_x \\ \mu_{x^*} \end{bmatrix}, \begin{bmatrix} k(X,X) & k(X,X^*) \\ k(X^*,X) & k(X^*,X^*) \end{bmatrix}) \tag{2}$$

$$f(X^*)|Y \sim N(\mu^*, k^*) \tag{3}$$

$$\mu^* = \mu_{x^*} + k(X^*,X)k(X,X)^{-1}(Y - \mu_x) \tag{4}$$

$$k^* = k(X^*,X^*) - k(X^*,X)k(X,X)^{-1}k(X,X^*) \tag{5}$$

With those formulas, one could obtain the posterior Gaussian Distribution of the non-observations and takes the expectations of the posterior as predictions.

4.2 Construction of GP Forecasting Model

As illustrated above, a GP is determined by a mean function and a kernel function. In practice, the prior of mean function $m(t)$ is set to be a constant value $\hat{X}$, which is the mean of the observations. However, the prior of kernel components is harder to construct. By studying the structure of NBA popularity time series and experimenting with different kernels, a combination of RBF, ExpSineSquared, RationalQuadratic, and white kernels are used to model the dependencies across time:

$$\begin{aligned} kernel = & RBF(length_scale = 30) + \\ & 2.0 * RBF(lengthscale = 100) * \\ & ExpSineSquared(lengthscale = 20, periodicity = 12) + \\ & RationalQuadratic(alpha = 20, lengthscale = 1) + \\ & WhiteKernel(noiselevel = 1) \end{aligned} \tag{6}$$

The RBF component could explain the gradual increasing trend, and a relatively large length scale could enforce smoothness. The ExpSineSquared component could explain the periodic seasonal structure. In order to allow decaying away from exact periodicity, the product with an RBF kernel is taken here. The RationalQuadratic component could explain the smaller, medium term irregularities. Finally, the WhiteKernel "noise" term could explain the correlated noise displayed in the time series. The specific parameters in the kernel function like length scale are selected by performing cross validation technique. In order to compare the forecasting performance of GP to that of ARIMA, the same testing and training set is applied. The GP model yields a MSE of 25.53 on the testing set.

5 Results and Conclusions

Figure 2 depicts the results from SARIMA train-test-split forecast, including both the actual data and the predictions on the testing set. Figure 3 depicts the results from SARIMA rolling forecast. Figure 4 depicts the forecasting results from Gaussian Process.

The results shown below suggest that GP performs much better than the SARIMA. Moreover, GP even outperforms the rolling forecast of SARIMA, indicating that GP is a better method in capturing the underlying trending, seasonal, and periodic structure displayed in the NBA popularity time series. The SARIMA model's predictions have much more wiggles, suggesting that the SARIMA model is over-fitting to the errors and thus has a worse performance than the smoother GP model.

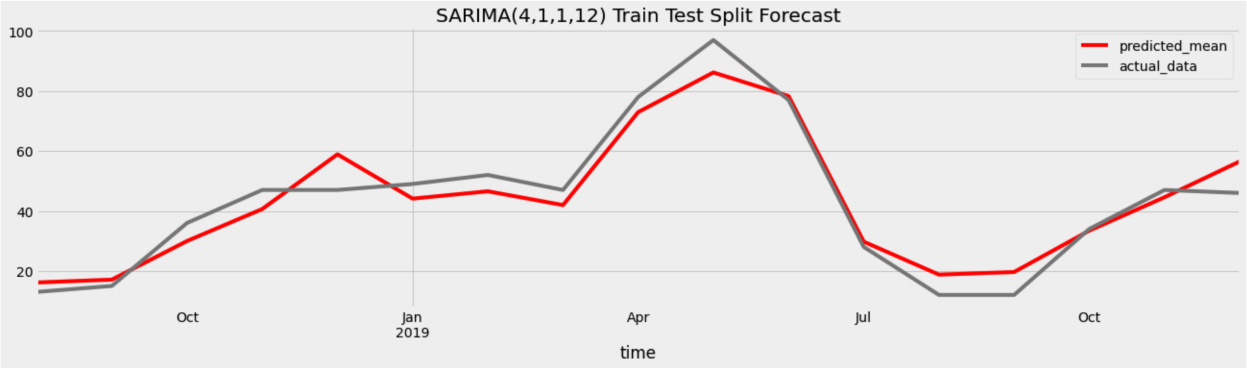

Figure 2: ARIMA Train Test Split Forecasting/ MSE: 39.96

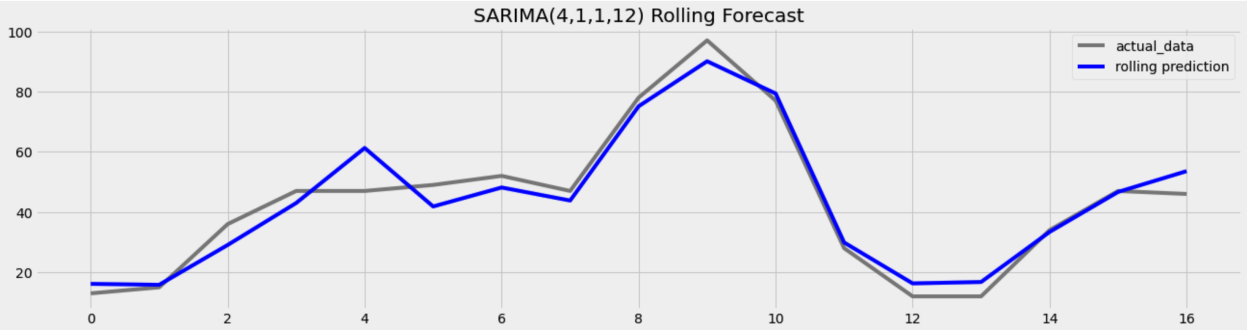

Figure 3: ARIMA Rolling Forecasting/ MSE: 30.53

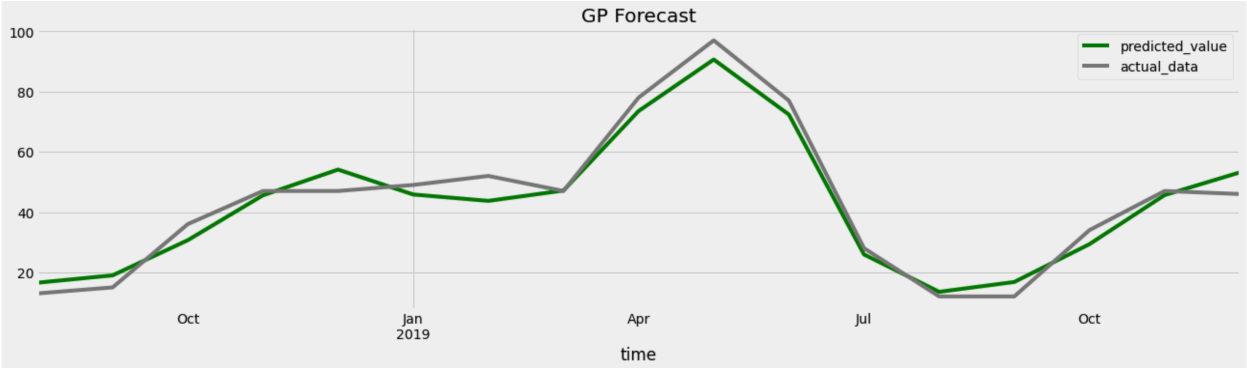

Figure 4: GP Forecasting/ MSE: 20.53

References

(1) Google Trends, https://trends.google.com/trends/?geo=US, Accessed: 2021-12-14.

(2) Hamilton, J. D., *Time series analysis*; Princeton university press: 2020.

(3) ARIMA models for time series forecasting, https://people.duke.edu/~rnau/411arim.htm, Accessed: 2021-12-14.

(4) Martino, L.; Laparra, V.; Camps-Valls, G. In *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp 1–6.