RAA 2021 Vol. **21** No. **4**, 94(10pp) doi: 10.1088/1674-4527/21/4/94

http://www.raa-journal.org *http://iopscience.iop.org/raa*

***R**esearch in*
***A**stronomy and*
***A**strophysics*

BoxCox multi-output linear regression for 10.7 cm solar radio flux prediction

Rui-Fei Cui[1], Ya-Guang Zhu[1], Huan Zhang[1], Ri-Wei Zhang[1], Hong-Yu Zhao[1] and Zheng-Lian Li[2]

[1] State Key Laboratory of Astronautic Dynamics, Xi'an 710043, China; *crf_dqq@126.com*
[2] Beijing Information and Communication Research Center, Beijing 100840, China

Received 2020 June 2; accepted 2020 October 21

Abstract We consider the problem of predicting the mid-term daily 10.7cm solar radio flux (F10.7), a widely-used solar activity index. A novel approach is proposed for this task, in which BoxCox transformation with a proper parameter is first applied to make the data satisfy the property of homoscedasticity that is a basic assumption of regression models, and then a multi-output linear regression model is used to predict future F10.7 values. The experiment shows that the BoxCox transformation significantly improves the predictive performance and our new approach works substantially better than the prediction from the US Airforce and other alternative methods like Auto-regressive Model, Multi-layer Perceptron, and Support Vector Regression.

Key words: Sun: radio radiation — methods: data analysis — methods: statistical

1 INTRODUCTION

The 10.7 cm solar radio flux (F10.7) refers to the amount of solar radio emission that has a wavelength of 10.7 cm (Lee 2007), which is a widely-used indicator of solar activity intensity with a variety of applications ranging from satellite systems, astronomy to communications (Tobiska 2003). As a popular index representing the solar chromosphere, transition region, and corona extreme ultraviolet radiation, F10.7 is an important input parameter to the atmospheric density model (Hedin 1991; Jacchia 1977), which in turn plays a key role in determining satellite orbit. In the common ionosphere models, F10.7 is of great importance as well and a good prediction can improve the accuracy of subsequent tasks like amending radar error caused by ionosphere disturbance. In addition, F10.7 is also a crucial influential factor of various space environment event prediction models such as the solar wind model, the geomagnetic storm early warning model, etc. Therefore, prediction on F10.7 has been a hot topic in the field of space environment over the past few decades.

There are various organizations that provide F10.7 prediction service for public, including the US National Oceanic and Atmospheric Administration (NOAA), the US Space Weather Prediction Center (SWPC), the US Airforce, the British Geological Survey (BGS), the Space Environment Prediction Center (SEPC), the National Space Science Center (NSSC), the Chinese Academy of Sciences (CAS), the Solar Activity Prediction Center (SAPC), and the National Astronomical Observatories, CAS (NAOC). Their products include short-term prediction (usually 1–3 days), mid-term prediction (usually 27, 45, or 90 days), and long-term prediction (a couple of solar cycles). In this paper, we mainly talk about the short and middle term prediction, focusing on the 1–27 days ahead daily forecast of the F10.7 index.

In the previous study, time series models and machine learning methods are the mainstream, where the main idea is to take the lagged observed values of a time series as input of a model to predict the forthcoming values. The BGS employs Auto-regressive Integrated Moving Average Model (Box & Tiao 1975) to make their prediction. Chatterjee (2001) uses an Artificial Neural Network with two hidden layers to predict 1-day ahead values of F10.7. Huang et al. (2009) applies Support Vector Regression (SVR) to make a short-term prediction (3 days ahead). Liu et al. (2010) uses a 54-order Auto-regressive Model (AR) to do a 27 days ahead prediction, where the mean absolute percentage error is around 15% for highly-active years and 5% for low-active years. The prediction of SEPC is based on the results of the AR model. Du (2020) systematically analyzes the predictive power of AR models with different orders for different prediction lengths. Wang et al. (2014) proposes a model that takes the second-order Fourier features to fit the solar self-rotation periodic law, in which they use 135-day

lagged values to train the model and make 54 days ahead prediction.

Among the models mentioned above, a common issue exists that may have a serious impact on the performance, that is, the violation of homoscedasticity. Specifically speaking, these models normally assume the data follow a property of homoscedasticity, whereas the F10.7 values show a much bigger variance during highly-active years than those during law-active years. To solve this issue, we propose to apply BoxCox transformation to the original F10.7 data such that our data fed to the subsequent regression model have similar variance regardless of the activity level of the Sun. A nonparametric method follows in order to choose a proper parameter of the BoxCox transformation. A second problem we need to solve is to transfer a univariate linear regression model to the multi-output case. For this goal, we introduce three commonly-used methods and choose the one that fits our problem best through experiments.

The reminder of this paper is organized as follows. Section 2 presents our method where we analyze the problem of homoscedasticity violation in detail, propose our solution to this issue, and introduce the multi-output linear regression model. Section 3 shows numerical experiments, in which we compare the three methods to enable multi-output regression, evaluate the effect of BoxCox transformation, and compare our approach with alternative methods. Section 4 concludes this paper and gives some discussion.

2 METHOD

In this section, we first analyze a common issue that current prediction models are faced with, that is, the violation of homoscedasticity. Then we present how to apply BoxCox transformation to solve this issue. Lastly we introduce the multi-output linear regression model by which we make the F10.7 mid-term daily forecast.

2.1 Problem Analysis

Models for F10.7 prediction can be generally cast as a regression problem, where past lagged observations of the time series are the inputs while future values are the outputs. This can be formulated as

$$y_t = f(y_{t-1}, y_{t-2}, \ldots, y_{t-p}) + \epsilon_t, \text{ with } \epsilon_t \sim \mathcal{N}(0, \sigma^2)\,, \tag{1}$$

where y_t is the observation at time t, p is the number of past observations used as features for prediction, and σ^2 is the variance of residuals (usually assumed to be normally distributed).

An important underlying assumption behind the model shown in Equation (1) is homoscedasticity, in the sense that the residual variance does not change over time, i.e., $\epsilon_t \sim \mathcal{N}(0, \sigma^2),\ \forall t$. However, this property is obviously violated for the F10.7 daily observations (solar flux units, sfu = $10^{-22}\mathrm{Wm}^{-2}\mathrm{Hz}^{-1}$). Figure 1 shows the yearly mean and variance of the F10.7 daily values[1], in which each data point is a summary statistic (mean or variance) over all the days within a year. From Figure 1, we can see that the variance of F10.7 observations fluctuate dramatically, ranging from a few to more than 1500 (sfu^2). Therefore, it is not a good idea to directly apply a regression model to the F10.7 daily observations to make prediction, and from our study we find that this significant violation of homoscedasticity is indeed one of the bottlenecks for the current models to achieve better performance.

2.2 BoxCox Transformation

From Figure 1, we also find that there is a strong association between the mean and variance; they have similar behavior over time. In order to get deeper insight of this relation, we perform a more detailed analysis shown in Figure 2, from which we see a significant positive linear correlation (the coefficient reaches 0.92) between the mean and variance. This property enables us to apply data transformation techniques to solve the problem of heteroscedasticity. The main idea is to impose a non-linear monotonic transformation to the original data such that the variance is approximately the same regardless of how big the mean is in the transformed space. A way to achieve this goal is the logarithm transformation. As illustrated in Figure 3, uniformly-distributed values within the interval [x2, x3] have bigger variance than values within [x1, x2] in the original space (since [x2, x3] is longer than [x1, x2]), whereas the values within [y2, y3] have the same variance as those within [y1, y2] after the logarithm transformation (since [y2, y3] has the same length with [y1, y2]).

However, the logarithm transformation is not always sufficient especially when the variance increases very quickly with a growing mean. A more general technique is the BoxCox transformation (Dey 2006), which is parameterized by λ, shown as follows

$$\tilde{y}(\lambda) = \begin{cases} \dfrac{y^\lambda - 1}{\lambda}, & \lambda \neq 0, \\ \ln(y), & \lambda = 0, \end{cases} \tag{2}$$

where y and $\tilde{y}$ are original and transformed data respectively. The parameter λ controls the shape of transformation function, and which value to use depends on the data.

[1] The F10.7 data used in this paper is downloaded from http://www.swpc.noaa.gov/Data, where they provide purely observed values and adjusted values subject to local effect. Here we take the adjusted values.

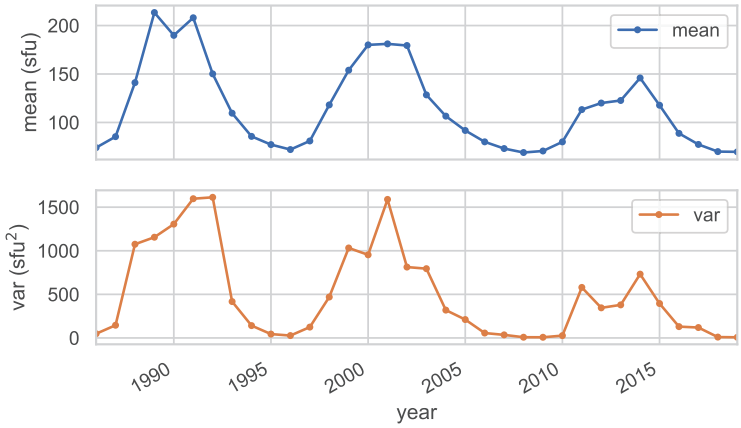

Fig. 1 Yearly mean (*top panel*) and variance (*bottom panel*) of the F10.7 observations from 1986 to 2019.

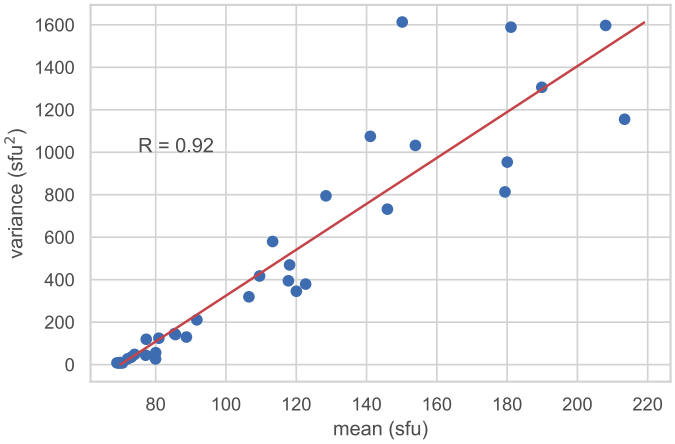

Fig. 2 Scatter plot of yearly mean and variance of the F10.7 observations from 1986 to 2019, in which the red straight line is the result of linear regression analysis and the correlation coefficient (R) between the two variables is 0.92.

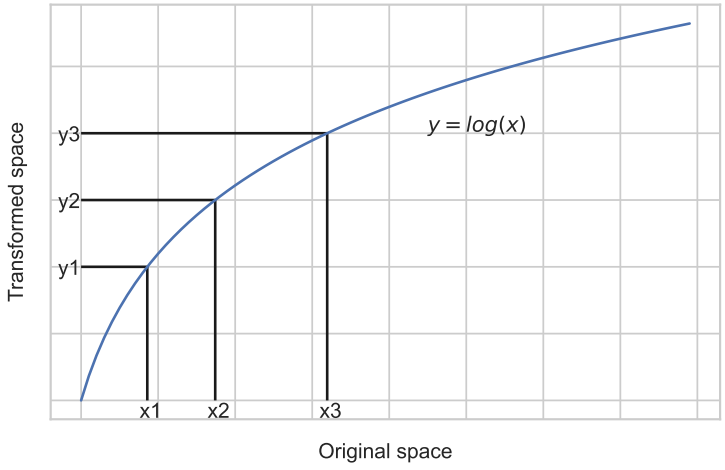

Fig. 3 Illustration of the logarithm transformation: in the original space the interval [x1, x2] is shorter than [x2, x3] while in the transformed space [y1, y2] has the same length with [y2, y3], acting as if variation of values within [x2, x3] is larger than those within [x1, x2] in the original space whereas values within [y1, y2] and values within[y2, y3] achieve the same variation in the transformed space.

What follows is how to choose a proper parameter λ[2]. To this end, we define a loss function of λ as

$$L\big[\tilde{y}(\lambda)\big] = \max\left(\frac{\mathrm{Var}[\tilde{y}_h(\lambda)]}{\mathrm{Var}[\tilde{y}_l(\lambda)]}, \frac{\mathrm{Var}[\tilde{y}_l(\lambda)]}{\mathrm{Var}[\tilde{y}_h(\lambda)]}\right) - 1, \quad (3)$$

where $\mathrm{Var}[\tilde{y}_h(\lambda)]$ and $\mathrm{Var}[\tilde{y}_l(\lambda)]$ denote the variance of transformed data over highly-active years and low-active years respectively. This loss function is a symmetric metric that will go to zero if the homoscedasticity property holds and will be a positive number otherwise. It can be interpreted as the extent to violation of homoscedasticity. Then, the optimal λ is learn by minimizing the loss function, i.e.,

$$\hat{\lambda} = \arg\min_{\lambda} L\big[\tilde{y}(\lambda)\big]. \quad (4)$$

One question that remained here is how to estimate $\mathrm{Var}[\tilde{y}_h(\lambda)]$ and $\mathrm{Var}[\tilde{y}_l(\lambda)]$. We take the data from 1986 to 2019 as an example to explain our procedure[3]: (1) compute and sort (by descending order) the yearly mean of the original data; (2) choose the first six as highly-active years (group 1) and the last six as low-active years (group 2); (3) compute the yearly variance of the transformed data with a specific λ for the chosen twelve years; (4) take the average variance over the first group as the estimate of $\mathrm{Var}[\tilde{y}_h(\lambda)]$ and that over the second group as the estimate of $\mathrm{Var}[\tilde{y}_l(\lambda)]$.

To make the learning processing of λ clearer, we summarize these steps in Algorithm 1. When taking the F10.7 observations from 1986 to 2019 as input data, we get an optimal λ of -1.338 with Algorithm 1.

Algorithm 1 Steps of the learning process of λ

Input: F10.7 daily observations.
Output: The learned optimal λ.
Step 1: Estimate $\mathrm{Var}[\tilde{y}_h(\lambda)]$ and $\mathrm{Var}[\tilde{y}_l(\lambda)]$ with the procedure introduced in the last paragraph;
Step 2: Put the learned $\mathrm{Var}[\tilde{y}_h(\lambda)]$ and $\mathrm{Var}[\tilde{y}_l(\lambda)]$ into Eq. (3) to get the loss function;
Step 3: Solve the optimization problem in Eq. (4) to get the optimal λ (the solver used here is the function 'fmin' implemented in the 'optimize' module of Python Scipy library.).

Figure 4 displays the original F10.7 daily observations from 1986–01–01 to 2019–12–31, the logarithm transformed data (BoxCox with $\lambda = 0$), and the BoxCox transformed data with the learned optimal λ. We see that for the original data, fluctuation over highly-active years is much larger than that over low-active years, as described in Section 2.1. With the logarithm transformation, the problem of heteroscedasticity is mitigated substantially, but the fluctuation over highly-active years is still higher. For the BoxCox transformation with optimal λ, we notice that there is no obvious difference in terms of the variation over years, showing a good property of homoscedasticity. If we conduct similar analysis for the transformed data with optimal λ as in Figure 2, we will approximately get a horizontal line, meaning that the variance is almost independent of the mean. The losses (see Eq. (3)) for the original data and two kinds of transformations are $\{52.5, 6.7, 4.0 \times 10^{-7}\}$ respectively.

2.3 Multi-output Linear Regression

Usually, the target variable of a regression model is univariate as shown in Equation (1), in the sense that we can only predict a 1-day ahead forecast with such a model. To enable multiple days ahead forecast, further steps are needed. There are three commonly-used methods for this purpose: Recursive Strategy, Multi-target Regression, and Chain Regression. Next, we introduce these methods respectively.

Recursive Strategy means that one time step ahead prediction is conducted first and then the prior time step is used as an input for making a prediction on the following time step, shown as follows

$$\begin{cases} y_t = w_0 + w_1 y_{t-1} + w_2 y_{t-2} + \ldots + w_p y_{t-p} + \epsilon\,, \\ y_{t+1} = w_0 + w_1 y_t + w_2 y_{t-1} + \ldots + w_p y_{t-p+1} + \epsilon\,, \\ \quad \ldots \\ y_{t+h} = w_0 + w_1 y_{t+h-1} + w_2 y_{t+h-2} + \ldots + w_p y_{t+h-p} + \epsilon\,. \end{cases} \quad (5)$$

Note that all equalities in Equation (5) share the same set of parameters, so the complexity of this method is quite simple, totally $p + 1$ parameters.

Multi-target Regression means that each of the targets is modeled with input features respectively, shown as follows

$$\begin{cases} y_t = w_{00} + w_{01} y_{t-1} + w_{02} y_{t-2} + \ldots + w_{0p} y_{t-p} + \epsilon\,, \\ y_{t+1} = w_{10} + w_{11} y_{t-1} + w_{12} y_{t-2} + \ldots + w_{1p} y_{t-p} + \epsilon\,, \\ \quad \ldots \\ y_{t+h} = w_{h0} + w_{h1} y_{t-1} + w_{h2} y_{t-2} + \ldots + w_{hp} y_{t-p} + \epsilon\,. \end{cases} \quad (6)$$

Note that all equalities in Equation (6) employ different parameters while they share the same set of input features. This method is more complex than Recursive Strategy, having a total of $(h + 1) \times (p + 1)$ parameters.

[2] There are some off-the-shelf automatic procedures for learning the parameter of BoxCox transformation like maximum likelihood estimation (Sakia 1992), but over there the goal is to make the data follow a normal distribution as approximately as possible, which is different from our purpose. Therefore, those methods do not work here.

[3] One could try a different procedure to estimate $\mathrm{Var}[\tilde{y}_h(\lambda)]$ and $\mathrm{Var}[\tilde{y}_l(\lambda)]$, which may lead to a different λ but it is indistinguishable and does not change the general conclusion.

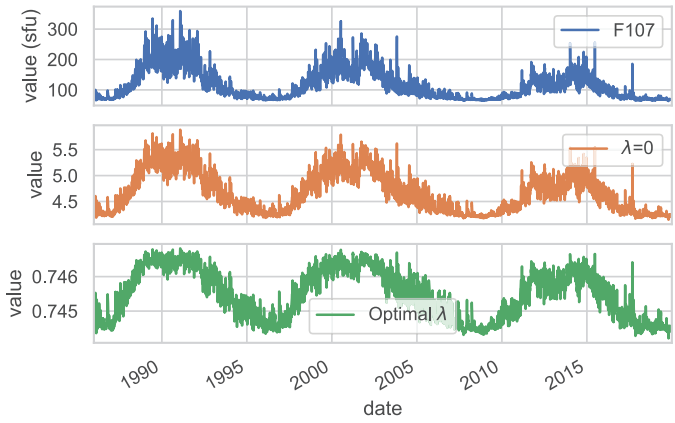

Fig. 4 *Top panel*: the F10.7 daily observations from 1986–01–01 to 2019–12–31; *Middle panel*: the BoxCox transformation with $\lambda = 0$ (reduces to the logarithm transformation) of the original F10.7 data; *Bottom panel*: the BoxCox transformation with optimal $\lambda(= -1.338)$ of the original F10.7 data.

Table 1 MAPE (%) of Chain Regression and Recursive Strategy with their Difference for Several Selected Days

Ahead of days	1	5	10	15	20
Chain	2.34	6.01	7.68	7.90	7.95
Recursive	2.33	5.97	7.57	7.71	7.68
Difference	0.01	0.04	0.11	0.19	0.27

Table 2 MAPE (%) of LReg and BoxCox-LReg with their difference for several selected days.

Ahead of days	1	5	10	15	20
LReg	2.4	6.4	8.3	8.6	8.6
BoxCox-LReg	2.3	6.0	7.5	7.7	7.7
Difference	0.1	0.4	0.8	0.9	0.9

Chain Regression means a multi-output model that arranges regressions into a chain, in which each model makes a prediction in the order specified by the chain using all of the input features plus the predictions of models that are earlier in the chain. This can be illustrated as

$$\begin{cases} y_t & = w_{00} + w_{01}y_{t-1} + w_{02}y_{t-2} + \ldots + w_{0p}y_{t-p} \\ & \quad +\epsilon\,, \\ y_{t+1} & = w_{10} + w_{11}y_{t-1} + w_{12}y_{t-2} + \ldots + w_{1p}y_{t-p} \\ & \quad +\alpha y_t + \epsilon\,, \\ & \quad \ldots \\ y_{t+h} & = w_{h0} + w_{h1}y_{t-1} + w_{h2}y_{t-2} + \ldots + w_{hp}y_{t-p} \\ & \quad +\beta_0 y_t + \ldots + \beta_{h-1}y_{t+h-1} + \epsilon\,. \end{cases} \tag{7}$$

This method is the most complex one, which has a total of $(h+1)\times(p+1) + h(h+1)/2$ parameters.

In Section 3.2, we will explore how the three methods perform through experiments and choose the one that fits our problem best.

To summarize Section 2 and turn the procedures aforementioned into a working algorithm, we profile our BoxCox multi-output linear regression approach in Algorithm 2.

Algorithm 2 Profile of the BoxCox multi-output linear regression approach

Step 1: BoxCox transformation of original data;
Step 2: Convert time series data into supervised learning form;

Step 3: Train a multi-output linear regression model;
Step 4: Make prediction with the trained model;
Step 5: Inverse BoxCox transformation of predicted values.

3 EXPERIMENTS

3.1 Dataset and Performance Metric

In this paper, we conduct our experiments with the F10.7 daily values from 1986–01–01 to 2019–12–31 covering around three solar cycles, in which data before 2009–01–01 are used as a training set (around two cycles) with the rest as a testing set (around one cycle). The number of input features and predicted time steps are chosen to be 54 and 27 respectively.

To evaluate the prediction performance, we use the mean absolute percentage error (MAPE)[4] between true values and predictions, defined as

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i} \times 100\%\,, \tag{8}$$

where $\hat{y}_i$ and y_i denote predicted and true values respectively.

[4] In the literature, it is also called mean relative error (MRE).

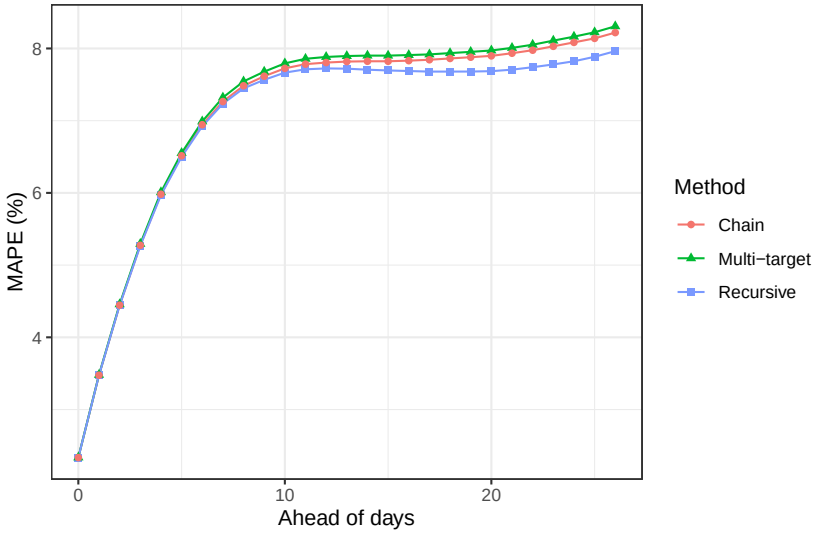

Fig. 5 MAPE of multi-output linear regression with Recursive Strategy, Multi-target Regression, and Chain Regression for 1–27 days prediction, showing the mean over all testing days.

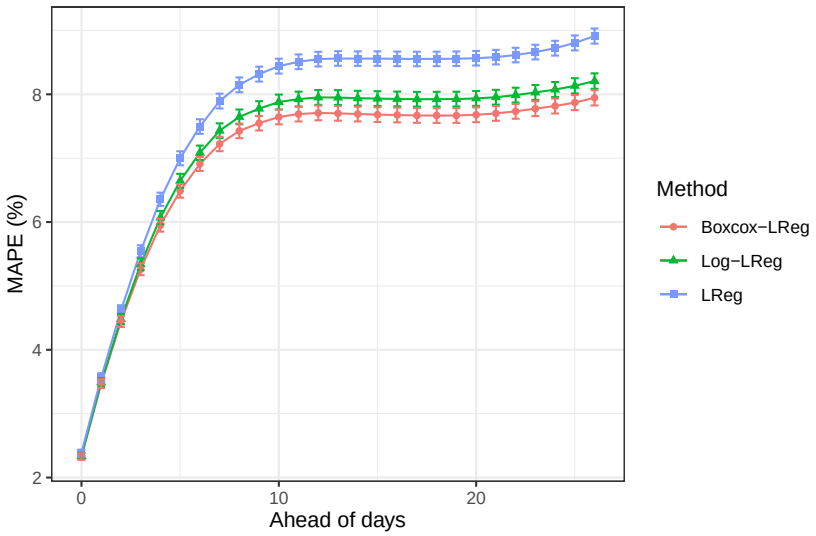

Fig. 6 MAPE of linear regression with original data (LReg), linear regression with logarithm transformed data (Log-LReg), and linear regression with optimal BoxCox transformed data (BoxCox-LReg) for 1–27 days prediction, showing the mean over all testing days with error bars representing one standard error.

3.2 Influence of Multi-output Strategies

In this section, we compare the three methods introduced in Section 2.3 that produce multi-output regression from a univariate model. Figure 5 shows the average MAPE over all testing days of Recursive Strategy, Multi-target Regression and Chain Regression for 1-27 days prediction. Table 1 shows the quantitative MAPE of Chain Regression and Recursive Strategy with their difference for several selected days. We see that for short-term prediction all methods are almost indistinguishable. This is because the shorter the prediction length is, the more similar they are, and they reduce to the same thing in the extreme case of 1-day ahead prediction. As the prediction length grows, the three methods begin to show different performance: Recursive Strategy performs substantially better than Chain, which in turn slightly outperforms Multi-target Regression. This tends to be more prominent for a longer time ahead prediction. This result arises possibly for two reasons: (1) Multi-target and Chain Regression are so complex that they overfit the data; and (2) Multi-target does not well employ the recent observations for longer-step predicting. Maybe there are some other reasons, but anyway we can conclude that Recursive Strategy is the method that fits our problem best. Therefore, in the subsequent part of this paper, when we mention multi-

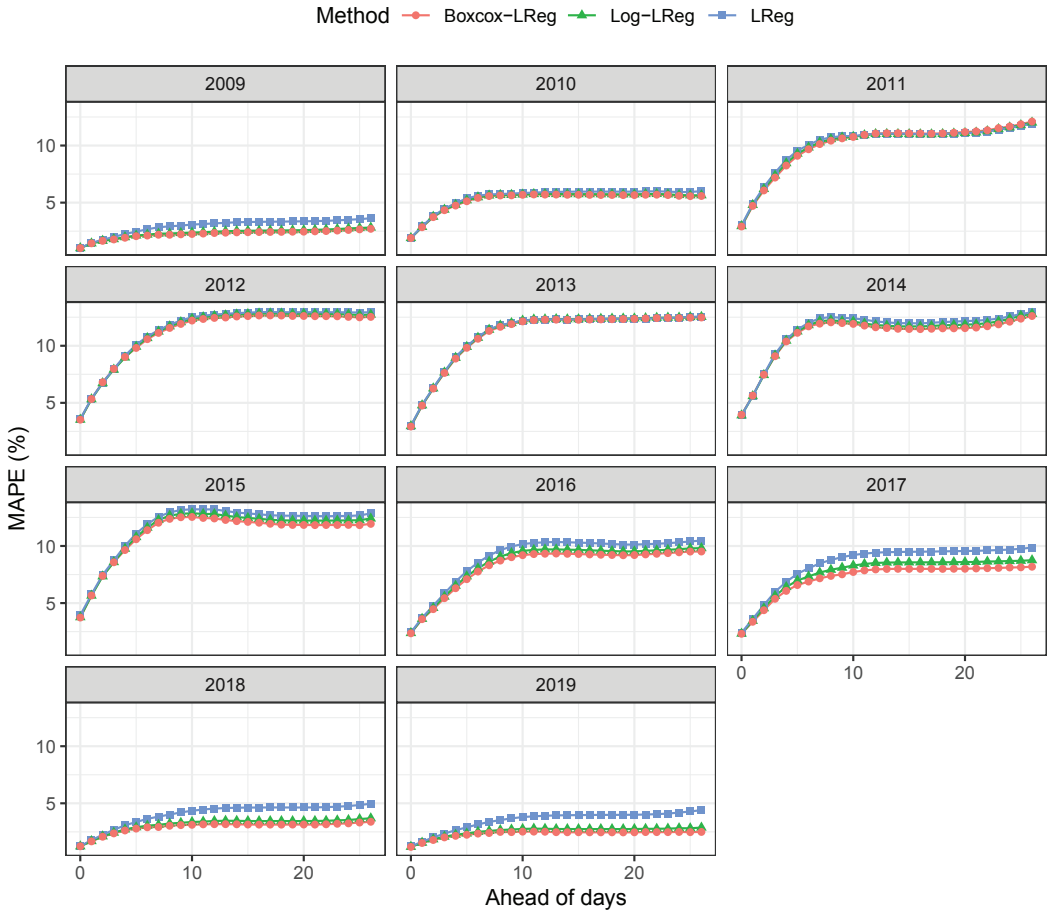

Fig. 7 MAPE of LReg , Log-LReg, and BoxCox-LReg for 1–27 days prediction grouped by years from 2009 to 2019, showing the mean over testing days within the corresponding year.

output regression it refers to the one with Recursive Strategy.

3.3 Influence of BoxCox Transformation

In this section, we evaluate the influence of BoxCox transformation on F10.7 prediction performance with the testing set. Three methods are considered: multi-output linear regression model with original data (LReg), multi-output linear regression with logarithm transformed data (Log-LReg), and multi-output linear regression with optimal BoxCox transformed data (BoxCox-LReg).

Figure 6 shows the MAPE of LReg, Log-LReg, and BoxCox-LReg for 1–27 days[5] prediction, providing the mean over all testing days with error bars representing one standard error. We see that BoxCox-LReg performs better than Log-LReg, which in turn outperforms LReg significantly. This is because the original data suffer from a serious issue of heteroscedasticity, the logarithm transformation mitigates this issue to some extent but it is insufficient, whereas the BoxCox transformation with optimal parameters makes the data approximately follow the property of homoscedasticity, resulting in the best performance. A detailed quantitative comparison between LReg and BoxCox-LReg is shown in Table 2, providing their respective MAPE and difference for several selected days. We see that BoxCox-LReg is substantially better than LReg, which becomes more prominent for longer prediction length.

In order to get deeper insights, we conduct our experiments year by year. The results are shown in Figure 7, providing MAPE of LReg, Log-LReg, and BoxCox-LReg for 1-27 days prediction from 2009 to 2019. From Figure 7, BoxCox-LReg shows a clear advantage over LReg for most years including 2009, 2015, 2016, 2017, 2018, and 2019; BoxCox-LReg performs slightly better than LReg for 2012 and 2015; the three methods give similar performance in 2010, 2011, and 2013. Overall, BoxCox-LReg is the best performer.

[5] In the figure, x-tick ranges from 0 to 26 since the day when we make forecast is taken as the first day to predict as shown in Equation (5).

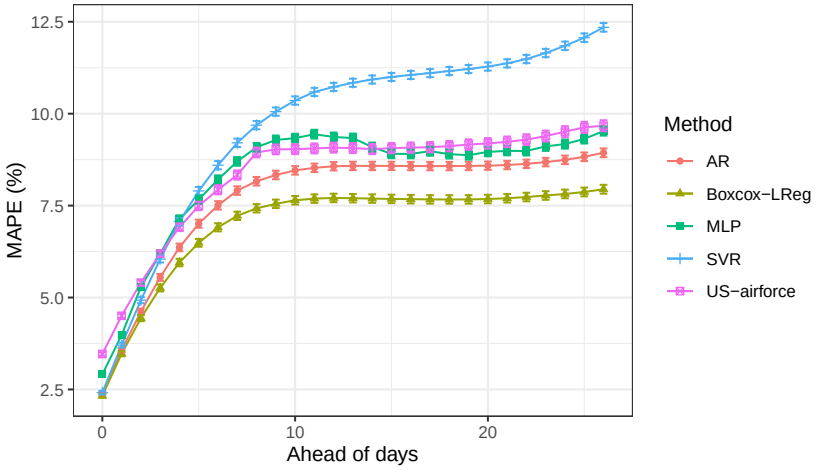

Fig. 8 MAPE of 54-order auto-regressive model (AR), BoxCox-LReg, multi-layer perceptron (MLP), support vector regression (SVR), and the US Airforce for 1-27 days prediction, showing the mean over all testing days with error bars representing one standard error.

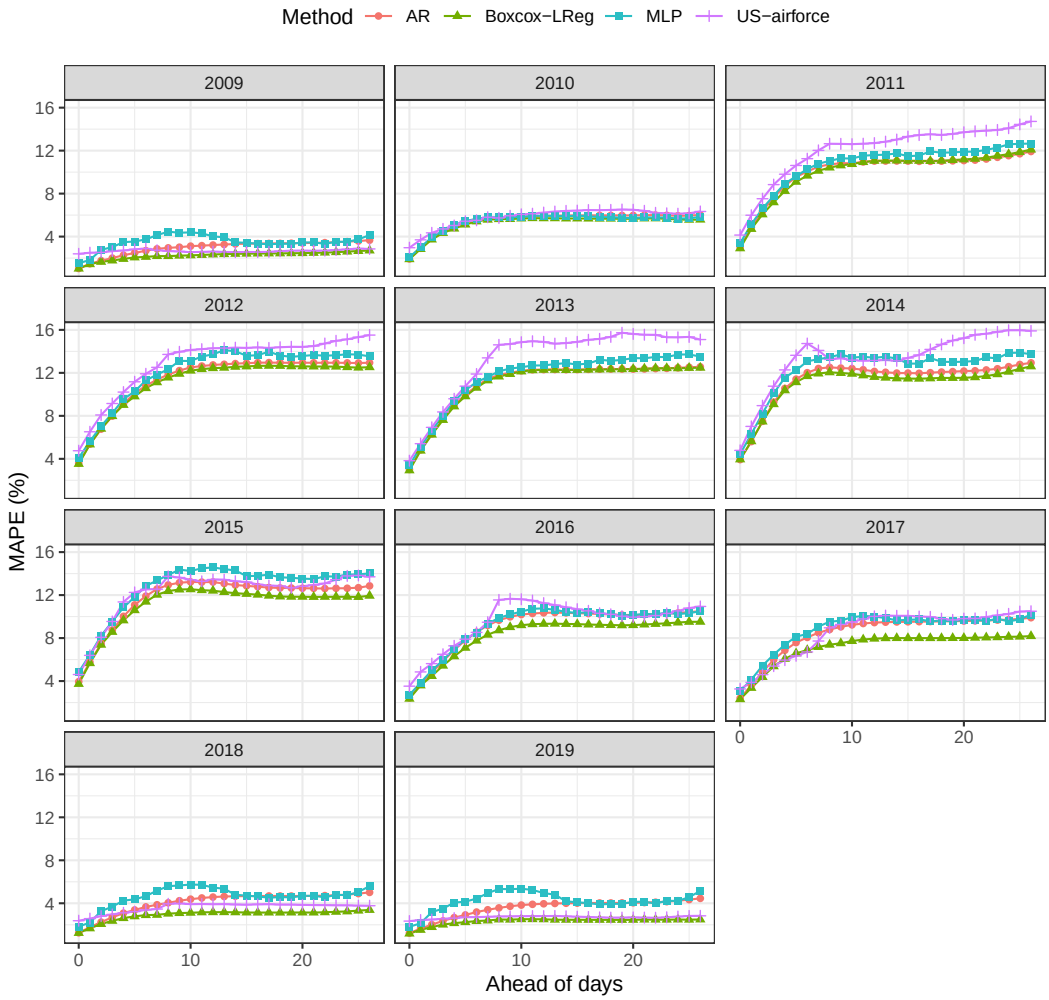

Fig. 9 MAPE of AR, BoxCox-LReg, MLP, and the US Airforce for 1–27 days prediction grouped by years from 2009 to 2019, showing the mean over testing days within the corresponding year.

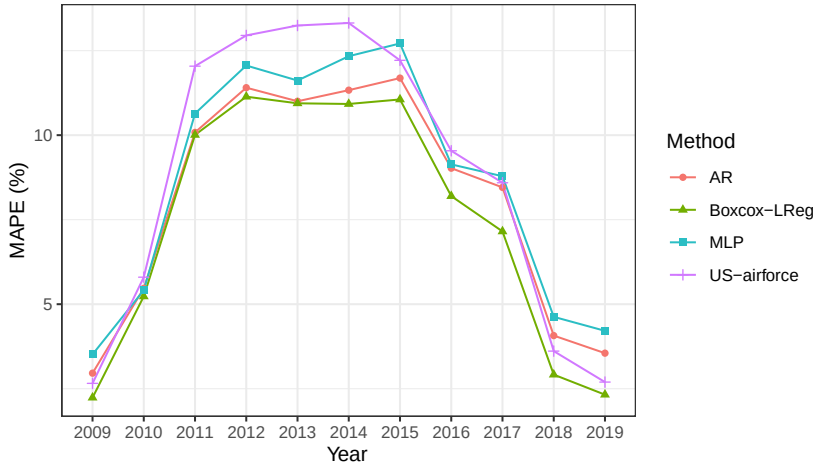

Fig. 10 Yearly MAPE of AR, BoxCox-LReg, MLP, and the US Airforce from 2009 to 2019, showing the mean over all testing days (365 or 366 days) and prediction lengths (27 days) within the corresponding year.

To conclude, the violation of homoscedasticity is indeed a serious issue that hampers a predictive model in the F10.7 prediction, and our proposal of using the BoxCox transformation can solve this issue very well to further improve the F10.7 forecast accuracy.

3.4 Comparison with Alternatives

In this section, we compare our method (BoxCox-LReg) with several alternative approaches, including the 54-order Auto-regressive Model (AR) (Liu et al. 2010), traditional feedforward neural network (a.k.a Multi-layer Perceptron, MLP), Support Vector Regression (SVR) (Huang et al. 2009), and the prediction from the US Airforce [6]. With MLP, the model structure consists of an input layer with 54 neurons, a hidden layer with 10 neurons, and an output layer with 27 neurons. The activation takes the hyperbolic tangent function, i.e., 'tanh', and the solver is set to the Adam (Kingma & Ba 2014) optimizer.[7] With SVR, since Huang et al. (2009) only makes prediction for 1–3 days, we extend it to work for 1-27 days through the recursive strategy, that is, one time step prediction is conducted first and then the prior time step is used as an input for making a prediction on the following time step. Following Huang et al. (2009), we use the RBF kernel as the kernel function in SVR.

Figure 8 shows comparative results of all the five methods for 1-27 days prediction, providing the mean over all testing days with error bars representing one standard error. It is encouraging that our BoxCox-LReg is dominating over its competitors especially when the predicted length becomes large. SVR displays a good performance for the first couple of days as reported by Huang et al. (2009), but it deteriorates quite fast and does not scale with a growing prediction length. Another interesting finding is that linear models (AR and BoxCox-LReg) perform better than complex models like MLP. Our conjecture is that the underlying relation between lagged and current values of the F10.7 data is approximately linear such that a linear model is a good choice for practical usage. By contrast, a complex model, although it could perform at least equally well to a linear model in theory, might slightly suffer from the problem of overfitting or could not reach its limit somehow, which makes it not work that well in practice.

To look into more details, further experiments were done. Figure 9 shows MAPE for 1–27 days prediction year by year from 2009 to 2019, where we leave out SVR since it performs obviously worse than other methods according to the results in Figure 8. Figure 10 shows yearly MAPE of AR, BoxCox-LReg, MLP, and the US Airforce, in which each data point is computed over all testing days (365 or 366 days) and prediction lengths (27 days) within the corresponding year (a data point is the mean over 365*27 or 366*27 original results). We see that the BoxCox-LReg method consistently outperforms the US Airforce across all years, which becomes more prominent for highly-active years. The reduced MAPE of BoxCox-LReg over the US Airforce ranges from 0.37% in 2019 to 2.39% in 2014. AR seems to show indistinguishable performance with our method in ascending years (from 2009 to 2013): the average reduced MAPE over these years is 0.27%, but BoxCox-LReg has a clear advantage over AR in descending years (from 2014 to 2019): the average reduced MAPE over these years increases to 0.9%. In a nutshell, BoxCox-LReg wins against its competitors in most cases.

[6] `swpc.noaa.gov/products`

[7] Here, the network structure and hyper-parameters are chosen empirically. One could explore different combinations.

4 CONCLUSIONS AND DISCUSSION

In this paper, we focused on the problem of forecasting 1-27 days F10.7 daily values. A new approach is proposed for this task, in which a BoxCox transformation is first applied to solve the issue of heteroscedasticity and then a multi-output linear regression model is used to make predictions. In the experiments, our BoxCox multi-output linear regression approach substantially outperformed the US-Airforce prediction and other alternative methods: the Auto-regressive Model, Multi-layer Perceptron, and Support Vector Regression. This is mainly because the BoxCox transformation with a proper parameter provides a good way to make the data satisfy homoscedasticity, which makes the subsequent regression model able to efficiently capture the relationship between features and targets. Another advantage of our approach is that it is very fast because of the linearity. Although our training set covers 22 years of data, the train time in our experience (python Scikit-learn implementation in a PC) is just 0.03s. The prediction time on a day is ignorable.

Our proposal of using the BoxCox transformation is a generic method to solve the issue of heteroscedasticity, which can be straightforwardly used in other problems than F10.7 daily prediction. The proposed learning procedure of optimal λ is not unique, e.g., one could define one's favorite loss function or try a different estimation method for $\mathrm{Var}[\tilde{y}_h(\lambda)]$ and $\mathrm{Var}[\tilde{y}_l(\lambda)]$ according to the specific problem.

The number of features used in this paper, 54, is empirically chosen as twice the days of solar self-rotation. Additional experiments show that these features are already sufficient to obtain a good forecast result, in the sense that inclusion of more features hardly improves the performance (we claim this is only correct for the current task under our experimental conditions). While we focused on 1–27 days prediction, it is quite straightforward to extend our approach to 90 days forecast (a scenario that is needed in a lot of applications) but perhaps requires more input features.

References

Box, G. E. P., & Tiao, G. C. 1975, Journal of the American Statistical Association, 70, 70

Chatterjee, T. N. 2001, MNRAS, 323, 101

Dey, D. K. 2006, Box-Cox Transformation (American Cancer Society)

Du, Z. 2020, Sol. Phys., 295, 134

Hedin, A. E. 1991, J. Geophys. Res., 96, 1159

Huang, C., Liu, D.-D., & Wang, J.-S. 2009, RAA (Research in Astronomy and Astrophysics), 9, 694

Jacchia, L. G. 1977, SAO Special Report, 375

Kingma, D. P., & Ba, J. 2014, in International Conference on Learning Representations, arXiv:1412.6980

Lee, J. 2007, Space Sci. Rev., 133, 73

Liu, S.-Q., Zhong, Q.-Z., Wen, J., & Dou, X.-K. 2010, Chinese Astronomy and Astrophysics, 34, 305

Sakia, R. M. 1992, Journal of the Royal Statistical Society: Series D, 41, 169

Tobiska, W. K. 2003, in Aerospace Sciences Meeting & Exhibit, doi: 10.2514/6.2003-1224

Wang, H. B., Xiong, J. N., & Zhao, C. Y. 2014, Acta Astronomica Sinica, 55, 302