Andy Dingler
Predict 413
Mid-term Project

Forecasting Rainfall Precipitation

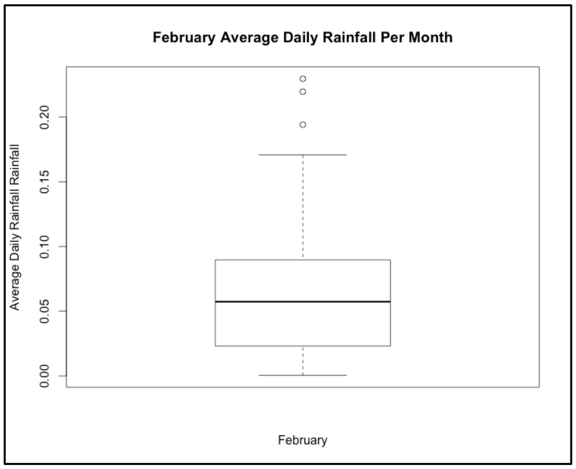
Problem under consideration

Weather is a difficult thing to forecast, and even the pros on news channels constantly get it wrong. In particular, rainfall precipitation can come in small or large quantities. I live in the Pacific Northwest, where we receive on average 42 inches of rain throughout the year, so my ability to accurately forecast rainfall precipitation is important to me. I love running outdoors but a rainy day can put a damper on my running experience. Just last month, I had to run a half marathon during a downpour. Being able to predict the amount of rain per month is a great start to understanding when it is going to rain and potentially figuring out what hour of the day it is going to rain. Being able to accurately predict this will allow me to figure out the best time in the day for me to get a run in.

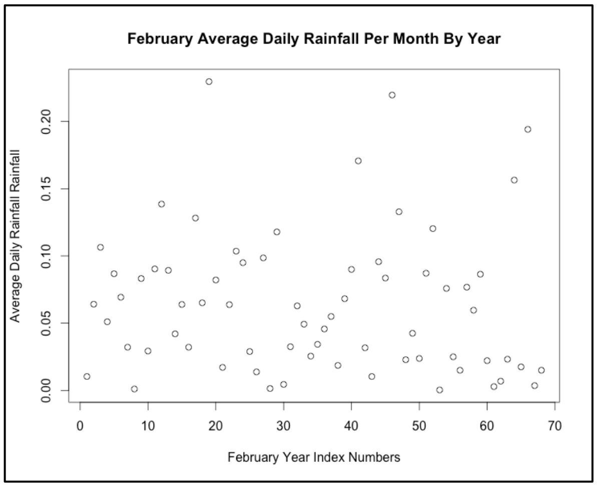
Data

The data used for this analysis is from the Global Historical Climatology Network (GHCN). The GHCN captures climate data for over 40 different meteorological elements ranging from daily temperatures, rainfall precipitation, and to the degree of cloudiness in the sky. The analysis in this report uses GHCN's daily rainfall precipitation in inches from an unknown station dating back to September 1946 and as present as July 2014. The data will be used to forecast monthly rainfall for August 2014 to July 2016.

Before using the data, it needed to be aggregated from the daily rainfall amount to the monthly rainfall. A problem with predicting monthly rainfall is that it is not standardized. This is because not every month has the same number of days. If we are not counting leap year, February usually has 28 days, and the other months are split between 30 and 31 days. To standardize the data, I calculated the average rainfall per day for each month by dividing the monthly rainfall from the number of days in their respective months. This did not dramatically improve the forecast results but it did provide a small improvement.



Since the data consists of more than 60 years of daily rainfall observations, there are outlier months that received more rain than their usual amount. To identify outliers, I decided to investigate each month individually. I did this to try and limit rainfall's seasonal impact on outliers. To simplify things in this report, I did not account for the extra day in February for leap year. In the example below, I will show you how I adjusted outlier rainfall values for the month of February. In the February boxplot to the right, you can see that there are three potential outlier data points that are significantly higher than the other 62 years of average rainfall for February. To adjust the outliers for any given month, I calculated the cap threshold for the month by using the month’s mean plus 2.5 * the month’s standard deviation which is 0.195 inches of rainfall for the month of February. The two data points in the boxplot above that are greater than 0.20 inches were adjusted downward to 0.195 inches. Overall, this approach did improve the model’s accuracy by 0.57% but it does have its limitations. By adjusting outliers by only looking at each month individually and not including the time component, I am not accounting for the impact of climate change over time. The plot below consists of all rainfall observations for February by year, and since I do not see a linear trend in the plot, I decided to omit the impact of climate change on rainfall when adjusting for outliers.



Rainfall Models

The models I selected to forecasting monthly rainfall are Autoregressive Integrated Moving Average (ARIMA), Neural Networks (NN) and an ensemble model that weighs the both models. I chose these two models because they are able to capture different pattern components within a time series. To test each model’s performance, I split the data set into train and test samples. Since the final model is forecasting the model 24 months into the future, the test set only consists of 24 months. Instead of only using the last 24 months in the data set to test the accuracy of each model, I used cross validation with rolling time periods of 36 months and then moved up 6 months. This allowed me to test each model’s accuracy of using 36 months as an input into each model and validate the model on the next 24 months 60 different times. I then averaged the RMSE for each 24 month forecasted period to calculate the RMSE of each model.

Autoregressive Integrated Moving Average Model

ARIMA models have been one the most popular linear models for time series forecasting over the past three decades (V. Ediger, S. Akar, 2007). ARIMA time series forecasting depends on past-observed values in previous time periods. They create dependent variables using moving averages (MA) on pervious values of the errors rather than the variable itself (M. Hisyamm, 2012).

To figure out the optimal moving average windows, I examined the Autocorrelation (ACF) and Partial Autocorrelation (PACF) to find the lag variables that were most significant. The Autocorrelation plots showed the best moving average windows were likely to be between 1 to 5 months and possibly 12 months. A moving average of 12 makes sense since it is the moving average over the previous year, but in order to have a moving average of 12. When using a

moving average of 12, you are adding all the moving averages between months 5 and 12 that are not significant. This may lead to overfitting the forecast and is why I decided not to use 12.
In order to pick the optimal moving average window, I ran an exhaustive search using the 1 to 5 month averages, and the 5 month moving average performed the best with an in-sample RMSE of 0.023.

The ARIMA model did pretty well on the in-sample data set. The in sample root mean square error (RMSE) and the out-of-sample RMSE were 0.087 and 3.02 respectively. The large increase from the in-sample to out-of-sample showed that we might have over-fit the model or that there were some outliers skewing the ARIMA forecast.

One of the disadvantages of using an ARIMA model is that the time series has to be stationary and there cannot be any missing data points. In this situation, we did not have any missing values, but if there were in the future we would have to figure out an approach to handle missing values. Even though we can decompose the time series or use a BoxCox transformation to make it linear, if the time series is non-linear in nature, ARIMA models will not have stable forecast results.

Neural Network Model

ARIMA models have a hard time forecasting a time series when the underlying time series correlation structure is non-linear. An advantage of Neural Networks (NN) is that they are additive in nature. The network's generalization capabilities remain accurate and robust in a non-stationary environment. (P.Chakradhara, V. Narasimhan, 2007).

The Neural Networks model did better than the ARIMA model on both the in-sample and out-of-sample cross validation data sets. On the in-sample data set, Neural Networks had an RMSE of 0.03 compared to the ARIMA's RMSE of 0.087. The neural network even had a smaller drop from the in sample to the out-of-sample data set. On the test data set, Neural Networks had an RMSE of 2.87 compared to the 3.02 from the ARMA. This is telling me, rainfall precipitation has both linear and non-linear time components. Even though the Neural Networks did better on both data splits, it does not mean it will perform better when forecasting new data points.

Using the Neural Network purely by themselves to model linear problems has yielded mixed results. Neural Networks inherently assumes the time series is non-stationary and is based on non-linear patterns (P.Chakradhara, V. Narasimhan, 2007). If the rainfall time series environment becomes more stationary, the Neural Network's performance will decrease and the ARIMA model's will eventually outperform the Neural Network.

Andy Dingler
Predict 413
Mid-term Project

Ensemble Model Using Both the Autoregressive Integrated Moving Average and Neural Network Models

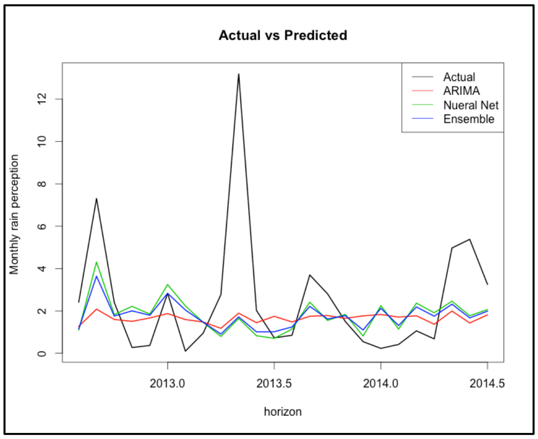
Ideally, all time series will be stationary, but in the real world they are not and it is difficult to determine whether a time series is generated from linear or non-linear correlation structures (M. Khashei, 2011). This is why ensemble models that decompose a time series into its linear and non-linear components have recently been shown to be successful for models (G.P. Zhang, 2003). The fact that both models above, the ARIMA (3.02 RMSE) and the Neural Network (2.87 RMSE) have similar RMSE on the cross validation data set shows rainfall has both a stationary and non-stationary time components.

An advantage of using an ARIMA and Neural Network ensemble model is that the model should be able to capture at least some portion of rainfall's stationary and non-stationary time components. Even though the models have similar RMSE values, they both are capturing different parts of the forecast. This is evident by ARIMA and Neural Network model's having a correlation coefficient of 0.11. The means that 89% of the time, both models are explaining a different portion of the rainfall time series.

To construct the ensemble model, I used the out-of-sample cross validation average RMSE for the 60 forecasted periods to calculate the weights for the ensemble model. The ensemble model returned an out-of-sample RMSE of 2.83, which is an improvement from using both models individually. Since the ensemble model uses both stationary and non-stationary time components, I am more confident in the reliability of this forecast model.

The plot to the right is both models' forecast 24-month horizon values. Ignoring the outlier monthly rainfall of greater than 12 inches in the plot to the right, the plot shows that the Neural Network in the green follows the general trend of the monthly rainfall better than the ARIMA model.

I am able to improve the performance of the model if I assign each model equal weights, which results in an RMSE of 2.89. I can probably continue to play with these weights but by doing so, I am overfitting the model. This is because I would be optimizing the weights on the test data set, which are a data set neither of the models have seen before and a data set that will not be available when forecasting future points.



The danger in using this approach is that there is an assumption that the relationship between the linear and non-linear components is additive and this may underestimate the relationship between the components and degrade performance; for example, a multiplicative time series (M.

Hisyamm, 2012). If rainfall precipitation starts to become purely stationary or non- stationary, one of the single model approaches will consistently out-perform the hybrid approach.

Future work

With more time to improve the model, I would incorporate different lag variables using the xreg parameter in the ARIMA. This could improve the ARIMA model's ability in capturing the trend in the time series that are not as easily captured with moving averages. In the hybrid model, I used the out-of-sample cross validation average RMSE for the 60 forecasted periods to calculate the weights for the ensemble model. The problem with this is every forecasted horizon RMSE mean has the same weight. With climate change, the weather is slowly changing throughout the years. To improve my ensemble weights, I would like to incorporate a dampening weighted average where we weigh the most recent RMSE periods more. This is because I believe a 24 month forecasted period in 2014 should not be the same as one in 1950.

Learning

Something I learned is that when you forecast with time series data that is seasonal, the seasonal window in ARIMA models needs to be at most half the size of the forecast horizon. For example, you cannot have a seasonal window of 12 and have a forecast horizon of 18 months. In addition, even though we can decompose a time series or normalize it using a transformation, we cannot always transform a non-stationary time series into a stationary time series. Sometimes, a non-traditional time series model should be used even though they yield mixed results from professionals.

References

M. Khashei, A novel hydrization of artificial of artificial Neural Networks and ARIMA models for time series forecasting, Applied Soft Computing 11 (2011) 2664-2675.

M. Hisyamm, Seasonal ARIMA for Forecasting Air Pollution Index: A Case Study, American Journal of Applied Sciences 9 (4): (2012) 570-578

P.Chakradhara, V. Narasimhan, Forecasting exchange rate better with artificial neural network, Journal of Policy Modeling 29 (2007) 227-236.

V. Ediger, S. Akar, ARIMA forecasting of primary enegery demand by fuel in Turkey, Energy Policy 35 (2007) 1701-1708.

G.P. Zhang, Time series forecasting using a hybrid ARIMA and neural network model, Neorcomputing 50 (2003) 159-175.