Matrix Completion of Weather Data

Zhi Su (301146205), Pascal Schmidt (301267941), Shenjunwei Lu (301300461)

November 11, 2019

Introduction

This module was about imputing missing values in a times series weather data set, acquired from 76 different weather stations in the United Kingdom. The data set consisted of 1,880,892 rows and 11 columns. In total, there were 2,074,873 missing values where we only had to make predictions for 29,000 of them. The final score was evaluated by calculating the absolute mean error. Our approaches for missing data imputations included linear interpolation, random forests, and linear regressions. In the end however, the crucial part for this module was to figure out for which NA values we needed to use linear interpolation, and which ones we needed to predict with modeling.

Data Description

The data set consisted of 1,880,892 rows and 11 columns. `USAF` (weather station), `YEAR`, `MONTH`, `DAY`, `HOUR`, `MINUTE`, `WINDDIR`, `WINDSPEED`, `TEMPERATURE`, `DEWPOINT`, and `PRESSURE`. Only missing values occurred in the `WINDDIR`, `WINDSPEED`, `TEMPERATURE`, `DEWPOINT`, and `PRESSURE` columns. Hence, we only needed to impute data for these columns. It is also noteworthy that `DAY` and `HOUR` values were only available in the year of 2012. For all other years we have zeros for these two columns. The `MINUTE` column has all zeros throughout the entire data set. Lastly, for all columns where we had to make predictions for were being standardized with mean zero and variance one.

`TEMPERATURE` and `DEWPOINT` were highly correlated (r = 0.9). There is also a moderate correlation between `MONTH` and `DEWPOINT`, `TEMPERATURE`, and `PRESSURE`. There is also a moderate correlation between `PRESSURE` and `WINDSPEED`.

Table 1: Correlation Matrix

	YEAR	MONTH	WINDDIR	WINDSPEED	TEMPERATURE	DEWPOINT	PRESSURE
YEAR	1.000	-0.008	0.041	-0.003	-0.051	-0.046	0.033
MONTH	-0.008	1.000	-0.007	0.027	0.284	0.203	-0.149
WINDDIR	0.041	-0.007	1.000	0.063	-0.002	-0.031	-0.022
WINDSPEED	-0.003	0.027	0.063	1.000	-0.141	-0.099	-0.334
TEMPERATURE	-0.051	0.284	-0.002	-0.141	1.000	0.897	0.097
DEWPOINT	-0.046	0.203	-0.031	-0.099	0.897	1.000	0.045
PRESSURE	0.033	-0.149	-0.022	-0.334	0.097	0.045	1.000

Data Processing

Because the data was standardized with mean zero and variance one, we considered data values above 4 and below -4 for `WINDDIR`, `WINDSPEED`, `TEMPERATURE`, `DEWPOINT`, and `PRESSURE` to be non-accurate data. Hence, we decided to remove these values and impute NA for them to not interfere with any modeling later.

Analysis

Linear Interpolation

At first, we used the `imputeTS` package and the `na_interpolation()` function to impute missing values by linear interpolation. This gave us a 0.21275 mean absolute error score on the public leaderboard. Linear interpolation seemed to be a very reliable imputation technique for a time series data set. However, we knew that there are missing values that we had to predict that had more than 100 NA values above them and more

than 100 NA values below them. For these kinds of NA values, it did not make much sense to use linear interpolation. Hence, we tried out a mix of interpolation and modeling.

Linear Interpolation + Modeling First Attempt

Instead of imputing all NA values by linear interpolation, we used the `maxgap` argument in the `na_interpolation()` package to only impute NA values for a maximum number of 5 successive NAs. This number was chosen by domain knowledge about weather. Within 5 hours, there are not a lot of weather changes. Afterwards, we fit a linear regression and a random forest model by station, with `TEMPERATURE` as response variable and `DEWPOINT`, `MONTH`, `WINDDIR`, `WINDSPEED`, and `PRESSURE` as predictors. In addition to that we also fit a linear regression with `DEWPOINT` as response variable and `TEMPERATURE`, `MONTH`, `WINDDIR`, `WINDSPEED`, and `PRESSURE` as predictors. We used these two models first because `DEWPOINT` and `TEMPERATURE` had the highest correlation among each other and so we thought that a linear regression or random forest would give good results.

Due to the long run time of the random forest models, we stuck with linear regression models. We dropped some regression model that had an adjusted r-squared lower than 0.85. Our reason behind that was that linear interpolation does still better for models with low predictability (low adjusted r-squared). After we had fit our models, we ran the `na_interpolation()` function again without the `maxgap` argument and decreased out mean absolute error on the public leaderboard.

Next, we tried to use the same predictors as mentioned above with `PRESSURE`, `WINDDIR`, or `WINDSPEED` as response variables. However, our public leaderboard score increased and rarely any adjusted r-squared value was above 0.8.

We figured out that it did not make much sense to fit linear regression models for `PRESSURE`, `WINDDIR`, or `WINDSPEED` as response variable.

Linear Interpolation + Modeling Second Attempt

We noted some flaws in how we fit our linear regression models in our first approach. For example, a row could consist of some predictors having NA values. By including these predictor that are missing in a particular row, for which we want to predict an NA value that is in the test data set, we do not get a prediction, only NA. Example: We want to predict `DEWPOINT` for this particular row and include `WINDDIR`, `WINDSPEED`, `TEMPERATURE`, and `PRESSURE` in our model, then we will not get a prediction and `DEWPOINT` will stay missing.

Table 2: Example 1

WINDDIR	WINDSPEED	TEMPERATURE	DEWPOINT	PRESSURE
NA	0.4888963	-0.1042561	NA	NA

We also figured out that there is a flaw in how we approached the `maxgap` argument in the `na_interpolation()` function. For example, consider a column vector like this one where we only need to predict the third position (third position of vector appears in test set data) in the following vector.

Table 3: Example 2

1	2	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	6

Even though the gap of successive NA values is large (14), it makes sense to use the value in position two for position three or linear interpolation. In case of interpolation, the imputed value in position three will be very close to the value in position two.

Hence, we developed a `C++` function, which identifies the missing values we needed to predict, that appeared

in the test set, in the training data, and then counts how many missing values there are above and below that NA value.

After we have filtered out values for which there are six or more values missing above and below, we fit a linear regression on each missing data point that appeared in the test set separately. The reason for this was Example 1 above. We chose a number of six because we thought that a six hour gap to the next value is enough time for weather data to change drastically. Our code identified which predictors are not missing in a row (for NA values appearing in the test set) that we had to predict and included that predictor into our regression model. Then, we checked how many missing values there are in a column for each predictor. If there were more than 40% missing values in a column of a predictor, we excluded that predictor from our regression. Moreover, if there were less than 500 observations for a station, we used the entire training data set for our model and included station as a categorical variable in our model.

When the predictors in a row were all missing (except month), then we used a mean imputation of the column vector we wanted to predict for a station. As mentioned above, a predictor is not included in the model when it was missing in a row or when the column had more than 40% missing values.

This resulted in around 3000 individual linear regression models and resulted in a public leaderboard score of 0.16205.

Afterwards, we grabbed rows for `TEMPERATURE` and `DEWPOINT`, for which a regression model was failing. We only refitted models for these response variables because of the high adjusted r-squared these two response variables in a linear regression model yield. A regression model was failing because a month category in a row, for which we made a prediction, did not appear when we fitted a model. Therefore, we refitted these models again only with `TEMPERATURE` or `DEWPOINT` and `MONTH` as predictors, depending on the response variable.

In the end, there were around 300 NA values left to predict. These consisted of values where the regression model failed, or where there were not six or more NA values above and below a NA value appearing in the test set. We decided that we just use interpolation for these values without the `maxgap` argument.

Our second approach seemed to be very good approach because we decreased our public leaderboard score by 0.02.

Conclusion

In retrospect, our final model came down to indentifying for which NA values that appeared in the test set we had to use interpolation, and for which ones we had to use a linear regression. Furthermore, for this data set, it was crucial to fit a linear regression for each NA value we had to model seperately, due to the unpredictable missingness pattern in each row. We could have improved our analysis by developing a cross validation strategy that justified for how many successive NA values we use interpolation. However, due to the fact that we did not know in which way missing values were being deleted from the data set, we chose to not do that.