Report of MRR project

Assessing Beijing PM2.5 pollution

Hao LIU Sokchivin SANN

21st January 2018

Dataset description:

Number of Instances:42824
Number of Attributes: 13
Missing Values: Yes
Associated Tasks: Regression

Variables : From 01/01/2010 - 31/12/2014

By learning the PM 2.5 readings and meteorological records from 2010–2015, the severity of PM 2.5 pollution in Beijing can be quantified with a set of statistical measures.

Data source : This hourly data set contains the PM2.5 data of US Embassy in Beijing. Meanwhile, meteorological data from Beijing Capital International Airport are also included.

No: row number
Year: year of data in this row
Month: month of data in this row
Day: day of data in this row
Hour: hour of data in this row
PM2.5: PM2.5 concentration (ug/m^3)
DEWP: Dew Point (â„ƒ)
TEMP: Temperature (â„ƒ)
PRES: Pressure (hPa)
cbwd: Combined wind direction
Iws: Cumulated wind speed (m/s)
Is: Cumulated hours of snow
Ir: Cumulated hours of rain

Object:

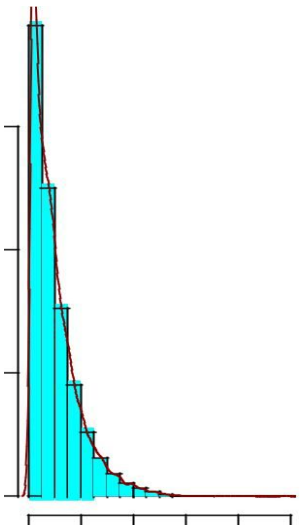
By building a linear regression model, we want to predict the PM2.5, at a given situation
Problems:

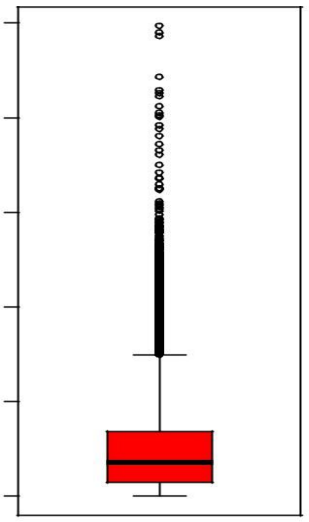
1: Certain PM2.5 values (Y) are missing NA
2: Quite a lot instances, low performance with a normal computer
3: Four variables they are in some way related with each other. (Year, Month, Day, Hour)
4: Categorical variable instead of numerical variable (Combined wind direction)
Introduction :

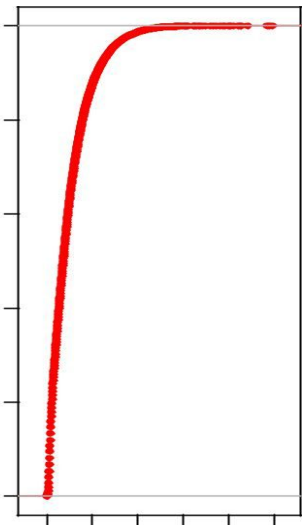
PM2.5 refers to atmospheric particulate matter (PM) that have a diameter of less than 2.5 micrometers, which is about 3% the diameter of a human hair.
Descriptive analysis
Response variable : pm2.5
In this data set, the response variable **pm2.5** takes value between 0 and 994 with average value of 98.6, variance of 8470.619 and other statistical descriptions as shown in the below table.

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max
##	0.0	29.0	72.0	98.6	137.0	994.0

This variable appears not to follow the Guassian distribution by immediately look at its distribution in the below graphic.
Moreover, depending on the Boxplot, we can see that there are some weired data points which will be considered as another problem for our regression. However, we will try to see the effect of those outlier in our model later.



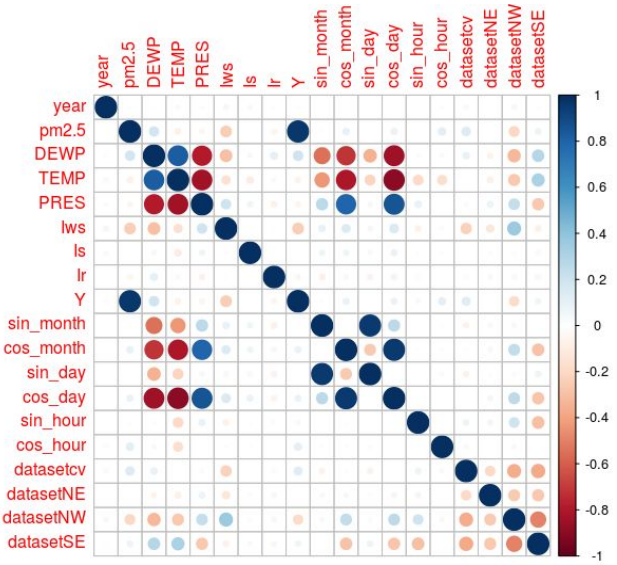




Study of dataset:

Variables correlation matrix

```
correlation = cor(dataset)
corrplot(correlation, method="circle")
```



Predictors and correlation

Firstly there are 12 predictors in our data set and one of them are categorical variable.After that, we transformed time variables into two dimensions and the categorical variable into binary. And we also add variable pm2.5 at time t(n − 1). So that we have 18 preditors. In this part, we will focus on two important kind of correlations. One is between independent variables, and another one is between independent and response variable . Since there are two kinds of predictors in our data set, we will look at those relationships separately. The figure below contains informations of correlations between numerical variables. We can see that there are some pairs of predictors appear to be highly correlated with each other which is what we don't want to see. These redundant variables will lead to

overfitting in our regression. On the other hand, we also can see that there is no predictor appear to be highly correlated with the response variable(except variable Y(pm2.5 at t(n − 1)) which is not so good for our regression as well. The high correlations between Y and response variable will lead to good prediction (large value of r2). That's why we needed to add the variable Y.

Data processing :

Change of variables : Month, year, days, hour

```
X_cos = cos(i*2π/X)

X_sin  = sin(i*2π/X)
```

Missing values: `dataset = na.omit(dataset)`

Dummy Coding : `cbwd = dummy(dataset[,'cbwd'])`

Time serie :
```
Y = dataset[,'pm2.5']
Y = Y[1:length(Y)-1]
dataset = cbind(dataset,Y)
```

Result : 19 variables

Separation dataset to training and test sets :

```
sample = sample(c(1:nrow(dataset)),n)

subset = dataset[sample,]

training_size = 0.8*nrow(subset)
test_size = 0.2*nrow(subset)
train_Y = Y[1:training_size]
train_X = X[1:training_size,]
test_Y = Y[(training_size+1):nrow(subset)]
test_X = X[(training_size+1):nrow(subset),]
```

Construction Multilinear Models

Model, $Y = X\beta + \varepsilon$ where X is $n \times (p + k)$ matrix where :

- Y is the response variable
- X is the predictor
- n is number of observation
- p is number of variables
- k dummy variable

The vector $\beta = \{\beta_0, \beta_1, \beta_2, ..., \beta_k\}$ and $E(\varepsilon) = 0$, $V(\varepsilon) = \sigma^2$ where :

- β is parameter
- ε is error term

Linear models :

```
reg =  lm('train_Y~.', data = as.data.frame(cbind(train_X,train_Y)))
```

Mean square error : sum((train_Y-reg$fitted.values)^2)/training_size

MSE(train) = 547.4764 MSE(test) = 573.7022

Model selection contructed by stepwise (Forward/Backward)

This algorithm considers all models starting from null model (using only intercept) to more complex model with more variables called Forward regression. Another regression starts from full model (using all variables) to more simple model with less variables called Backward regression. In this case, we will use stepwise regression with both directions and we measure each model using two criteria AIC and BIC. We name those two models with best value of AIC and BIC, step.AIC and step.BIC respectively.

```
regbackward = step(reg,direction = 'backward');
```

```
regforward = step(reg,list(upper=reg),direction = 'forward');
```

We check for the diagnostic of these two models. These models get a high R-square and their mean square errors are a bit larger than normal linear regression model. The residuals seem to be symmetric in the graphic above. It appears to be good models.

Forward selection :

train_Y ~ Y + NW + cos_month + sin_hour + NE + DEWP + TEMP + Ir + cv + sin_day + PRES

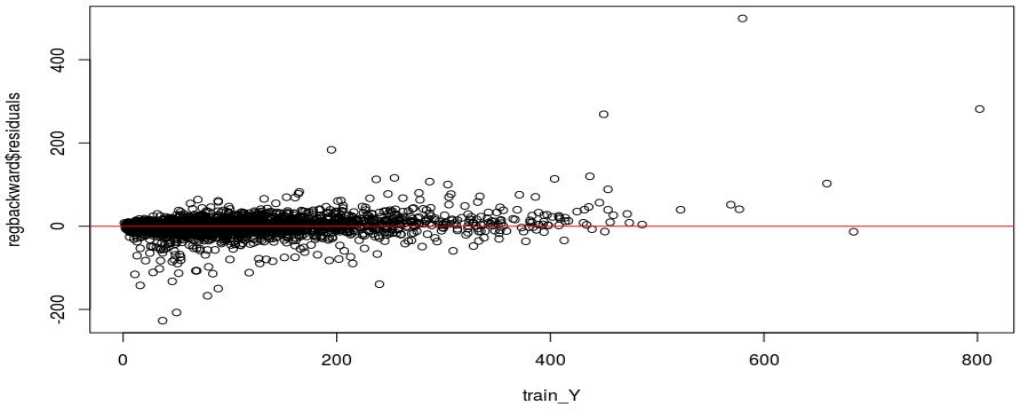
AIC : 12 202769.9 MSE(train) = 547.4968

Backward selection :

train_Y ~ DEWP + TEMP + PRES + Ir + Y + cos_month + sin_day + sin_hour + cv + NE + NW

AIC : 12 202769.9 MSE(train) = 547.4764

However, these two methods give us the same mount of variables and performance.



Residuals plot

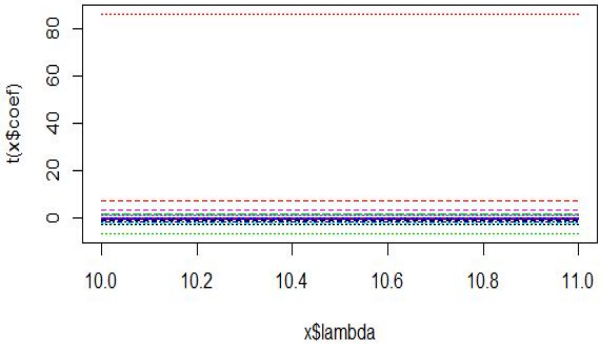
Penalisation Regression

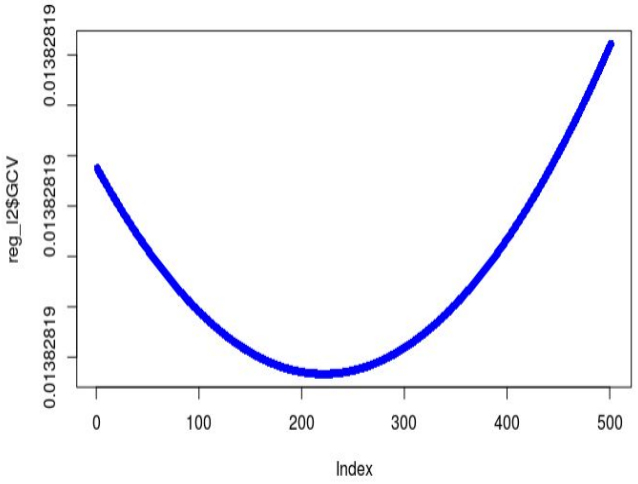
In this part we are interested in the regularised methods ridge and LASSO in order to constrain the variance of our estimator and eventually improve our prediction error. We trade between the Residual Sum of Square (RSS) and Variance of the estimator by constraining our estimator in a limited region.

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|Y - X\beta\|^2 + \lambda\|\beta\|^2)$$

- Ridge Regression The Ridge regression constrains the variance of estimator using $k.k_2$. We measure each model through cross-validation with the **minimum** of GCV (the most penalized model with a "1se" distance from the model with the least error).

MSE(train) = 17784.68 MSE(test) = 17949.88

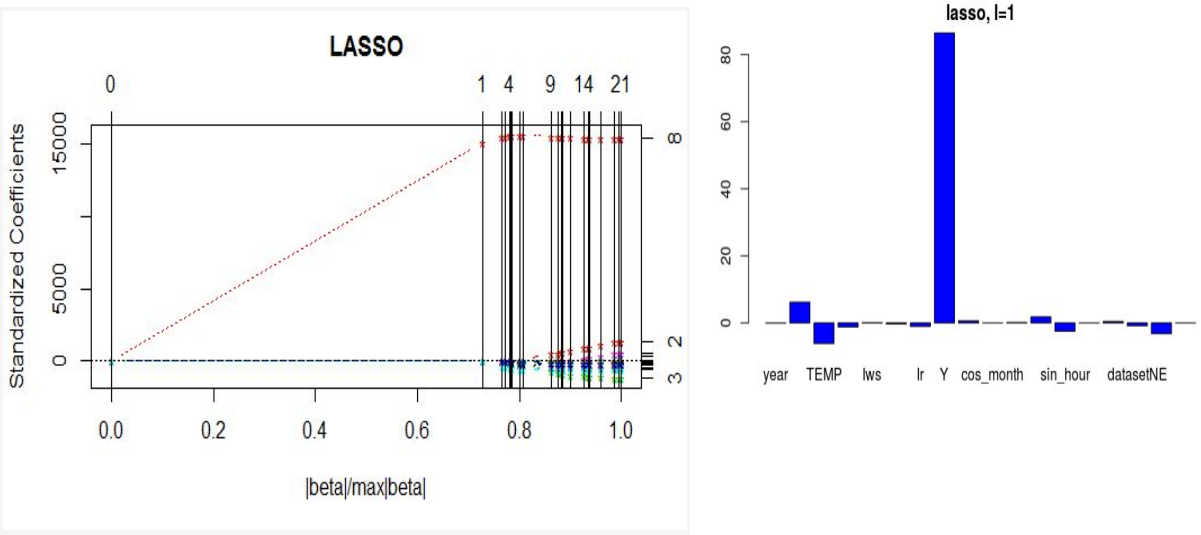




Lasso Regression

It is similar to the Ridge regression. The difference is in the LASSO regression we constrain the variance of estimator using k.k1. Because of the nature of this norm, it puts some variables to be 0 and the remaining variables are considered as significant for our model.

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|Y - X\beta\|^2 + \lambda\|\beta\|_1)$$



MSE(train) = 547.8338 MSE(test) = 574.3286

Elastic net regularization

Elastic net is a hybrid approach that blends both penalization of the L2 and L1 norms. Ridge and LASSO are particular cases of Elastic net.

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}}(\|Y - X\beta\|^2 + \lambda((1-\alpha)\|\beta\|^2 + \alpha\|\beta\|_1))$$

We consider the following cases with different values of α {0.05, 0.1, ..., 0.95}.

MSE(train) = 569.9181 MSE(test) = 592.3498

Polynomial regression

$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{33} x_3^2 + \ldots + \varepsilon$

```
poly_reg <- lm(train_Y ~ poly(train_X[,1],train_X[,2]train_X[,3],....,degree = 2))
```

MSE(train) = 516.3504

MSE(test) = 65424.36 (Overfitting)