EnorNOC-GAM

Yu-Chen Xue

2018 年 6 月 13 日

GAM methods are implemented in R in the package mgcv.

For visualizations packages ggplot2, grid and animation will be used. One useful function from package car will be used too.

Let's scan all of the needed packages.

1. Dataset 介紹：

名稱：EnerNOC GreenButton Data
來源：open enernoc data
簡述：該資料集由 EnerNOC 電力公司提供，它依循時間序列記錄了 2012 年 100 棟不記名的商業大樓每 5 分鐘的用電情況。
解釋變數：
- timestamp: 以秒為單位的絕對時間數值 (Type: int)
- dttm_utc: 即 datetime，日期與時間 (Type: chr)
- value: 特定時間點下的電耗值 (Type: num)
- estimated: 若為 1 表示當下的電耗值為估計值，否則為 0 (Type: chr) - anomaly: 若有值表示當下的電耗值有誤，否則沒有值 (Type: chr)

2. 要用迴歸分析回答哪些問題

1. 電耗的日週期變化
2. 電耗的週週期變化
3. 預測兩週以內的電耗值
4. 預測三個月內的電耗
5. 預測半年的電耗

3. 使用的方法：

Generalized additive model (GAM)

4. 仔細的 model 解讀

```
library(feather)
library(data.table)
library(mgcv)
library(car)
library(ggplot2)
library(grid)
library(animation)
```

讀取資料

```
library(data.table)
library(feather)
DT <- as.data.table(read_feather("D:/WORKSPACE/RProjects/EnorNOC-GAM/DT_4_ind"))
```

使用 package `car` 中的 function `record`，將週次改為 interger

```
DT[, week_num := as.integer(car::recode(week,
    "'Monday'='1';'Tuesday'='2';'Wednesday'='3';'Thursday'='4';
    'Friday'='5';'Saturday'='6';'Sunday'='7'"))]
```

從讀取的資料中獲取 industry, date, weekday and period 等信息，並使用變量來儲存。

```
n_type <- unique(DT[, type])
n_date <- unique(DT[, date])
n_weekdays <- unique(DT[, week])
period <- 48
```

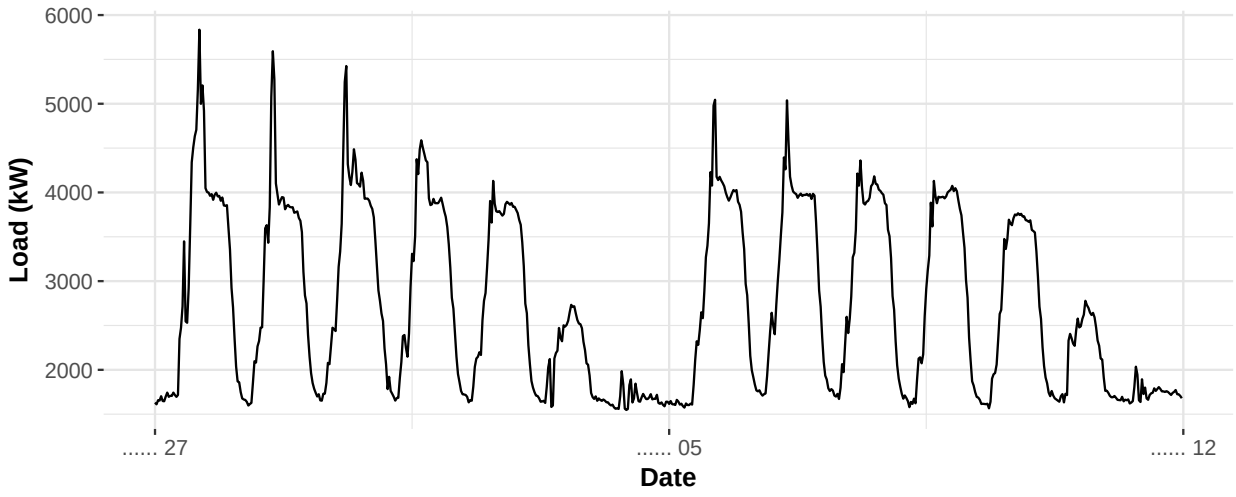
截取兩個禮拜內的商業用樓房的電耗記錄，並儲存在 data_r 變量中。之後畫圖展示之。

```
data_r <- DT[(type == n_type[1] & date %in% n_date[57:70])]

ggplot(data_r, aes(date_time, value)) +
  geom_line() +
  theme(panel.border = element_blank(),
        panel.background = element_blank(),
        panel.grid.minor = element_line(colour = "grey90"),
        panel.grid.major = element_line(colour = "grey90"),
        panel.grid.major.x = element_line(colour = "grey90"),
```

```
        axis.text = element_text(size = 10),
        axis.title = element_text(size = 12, face = "bold")) +
  labs(x = "Date", y = "Load (kW)")
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鉴  27' in 'mbcsToSbcs': dot substituted for <e4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鉴  27' in 'mbcsToSbcs': dot substituted for <ba>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鉴  27' in 'mbcsToSbcs': dot substituted for <8c>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鉴  27' in 'mbcsToSbcs': dot substituted for <e6>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鉴  27' in 'mbcsToSbcs': dot substituted for <9c>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鉴  27' in 'mbcsToSbcs': dot substituted for <88>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鈥  05' in 'mbcsToSbcs': dot substituted for <e4>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鈥  05' in 'mbcsToSbcs': dot substituted for <b8>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鈥  05' in 'mbcsToSbcs': dot substituted for <89>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鈥  05' in 'mbcsToSbcs': dot substituted for <e6>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鈥  05' in 'mbcsToSbcs': dot substituted for <9c>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on '鈥  05' in 'mbcsToSbcs': dot substituted for <88>
```

```
## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '鉢  05' in 'mbcsToSbcs': dot substituted for
## <88>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '鉢  12' in 'mbcsToSbcs': dot substituted for
## <e4>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '鉢  12' in 'mbcsToSbcs': dot substituted for
## <b8>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '鉢  12' in 'mbcsToSbcs': dot substituted for
## <89>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '鉢  12' in 'mbcsToSbcs': dot substituted for
## <e6>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '鉢  12' in 'mbcsToSbcs': dot substituted for
## <9c>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on '鉢  12' in 'mbcsToSbcs': dot substituted for
## <88>
```



There is possible to see two main seasonalities in plotted time series: daily and weekly. We have 48 measurements during the day and 7 days during the week so that will be our independent variables to model response variable - electricity load. Let's construct it: 根據每天的週期性變化和每週的週期性變化，重新構建資料

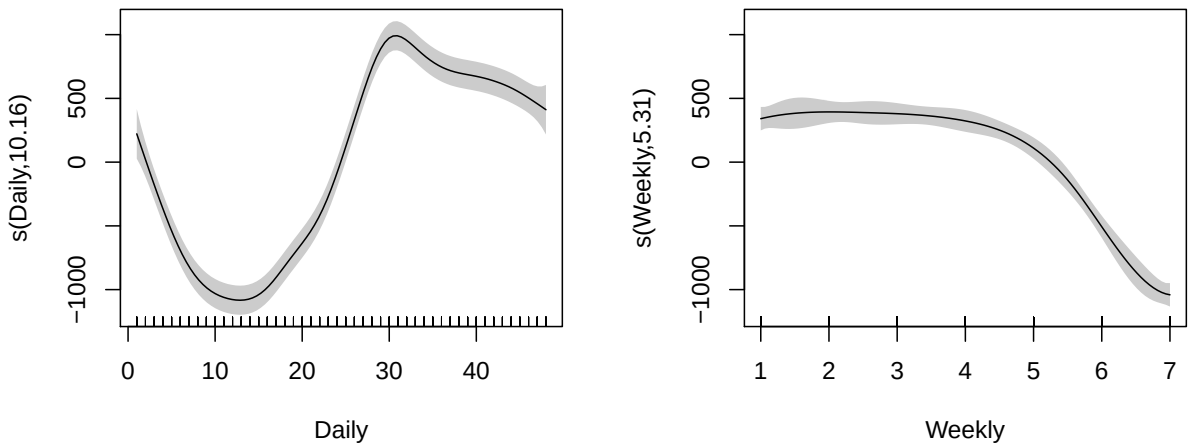
```
N <- nrow(data_r) # train set 中的資料筆數
window <- N / period # train set 所囊括的天數
matrix_gam <- data.table(Load = data_r[, value],
                         Daily = rep(1:period, window),
                         Weekly = data_r[, week_num])
```

Here we are! Train our first **GAM** with function `gam`. Independent variables are modeled by smoothing function `s`, for daily seasonality cubic regression spline is used, for weekly seasonality, P-splines is used, a number of knots are logically set to the number of unique values. Let's do it. 建立 GAM 模型，其中每天的週期性變化採用 cubic regression spline 模式來描述，每週的週期性變化採用 P-splines 來描述。

```
gam_1 <- gam(Load ~ s(Daily, bs = "cr", k = period) +
               s(Weekly, bs = "ps", k = 7),
             data = matrix_gam,
             family = gaussian)
```

Package `mgcv` have many advantages and nice features. First is its visualization capabilities. Let's try it: 作圖分析此模型

```
layout(matrix(1:2, nrow = 1))
plot(gam_1, shade = TRUE)
```



在左圖中可以看出，用電高峰出現在下午 3 點；在右邊的圖中可以看出，週末的用電量小於平日。

查看模型的 summary table

```
summary(gam_1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Load ~ s(Daily, bs = "cr", k = period) + s(Weekly, bs = "ps",
##     k = 7)
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2731.67      18.88   144.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df     F p-value
## s(Daily)  10.159 12.688 119.8  <2e-16 ***
## s(Weekly)  5.311  5.758 130.3  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.772   Deviance explained = 77.7%
## GCV = 2.4554e+05  Scale est. = 2.3953e+05  n = 672
```

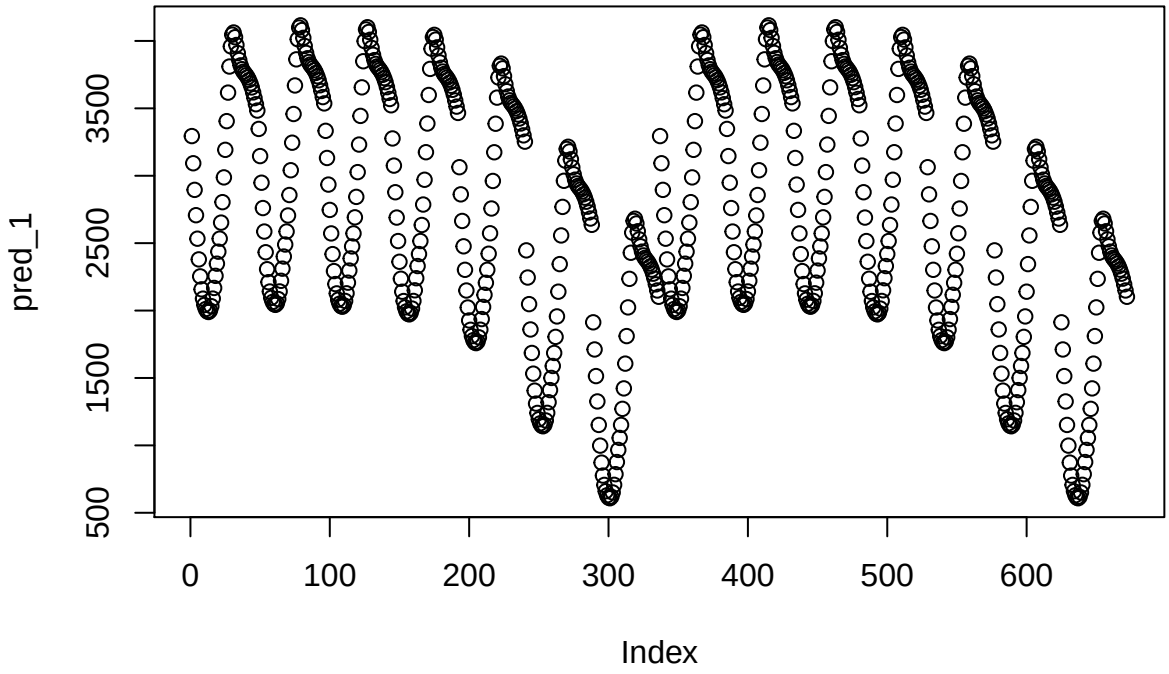
由 summary table 可以看出，根據 p-value，電耗波動有很強的每天週期性變化和每週週期性變化。，Deviance explained 數值不算很高，說明還有其他變量需要挖掘。

5. 預測

```
data_test <- DT[(type == n_type[1] & date %in% n_date[71:84])]
matrix_gam <- data.table(Load = data_r[, value],
                         Daily = rep(1:period, window),
                         Weekly = data_test[, week_num])
pred_1 <- predict(gam_1, matrix_gam)
summary(pred_1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   607.9  2087.5  2690.6  2731.7  3599.4  4115.6
```

```
plot(pred_1)
```



6. 課堂沒教的花俏的 r 指令

1. unique: 獲取某欄位下的所有出現過的值
2. feather::read_feather: 讀取 feather 格式的資料
3. car::recode: 將資料重新編碼