Mars Express Power Challenge

CC71Q - Introducción a la Minería de Datos

Departamento de Ciencias de la Computación
Facultad de Ciencias, Física y Matemáticas
Universidad de Chile

Gabriel De La Parra

30 de mayo de 2016

1. Introducción

El Mars Express Power Challenge tiene como objetivo predecir el consumo de los circuitos eléctricos de calefacción y refrigeración de un satélite de exploración en la órbita de Marte, necesarios para el correcto funcionamiento de los equipos de exploración científica de la nave.

Para realizar dicha predicción se tienen un set de entrenamiento correspondiente a los datos de tres años marcianos (q año marciano son 687 días terrestres) y un set de pruebas correspondiente al cuarto año. Cada set está compuesto por cinco subsets de datos de entrenamiento y un subset de los valores que se desean predecir. Los datos para entrenamiento tienen que ver con comandos que se envían a la nave, ángulos de incidencia solar, posición del satélite con respecto a marte y la tierra, eventos tales como eclipses. Los datos a predecir indican el valor de corriente de cada uno de los 33 circuitos eléctricos.

La forma de evaluar la predicción consiste en enviar un archivo de predicción, con promedios por cada hora, para un año y los 33 circuitos. La precisión de la medición se calculará mediante el *Root Mean Square Error (RMSE)* que se calcula de la siguiente forma:

$$\epsilon = \sqrt{\frac{1}{NM} \sum (c_{ij} - r_{ij})^2}$$

ϵ: root mean square error
c_{ij}: valor i-ésimo de predicción en el cuarto año
r_{ij}: valor i-ésimo de referencia en el cuarto año
N: número total de muestras para un año $i \in [1, N]$ with $N <= 16488$
M: número total de parámetros $j \in [1, M]$ with $M = 33$

Para poder realizar una predicción, se utilizó un método de predicción de valores continuos. A esta categoría de predicción se le conoce como regresión. Existen varias técnicas de regresión, como primer acercamiento se aplicó RandomForest, visto en clase para clasificación, el cual tiene también aplicaciones sobre regresión.

Para la visualización y la implementación del algoritmo se utilizó R en RStudio. Lo anterior no descarta que puedan aplicarse otras herramientas como Python o MatLab para realizar el (pre) procesamiento de los datos.

2. Visualización

Con el objetivo de obtener una primera impresión de los datos, se procedió a graficar y tabular estos en bruto y buscar correlaciones superficiales entre los datos. De los sets de datos entregados, algunos se pueden visualizar:

POWER: Consumo energético, variable a predecir

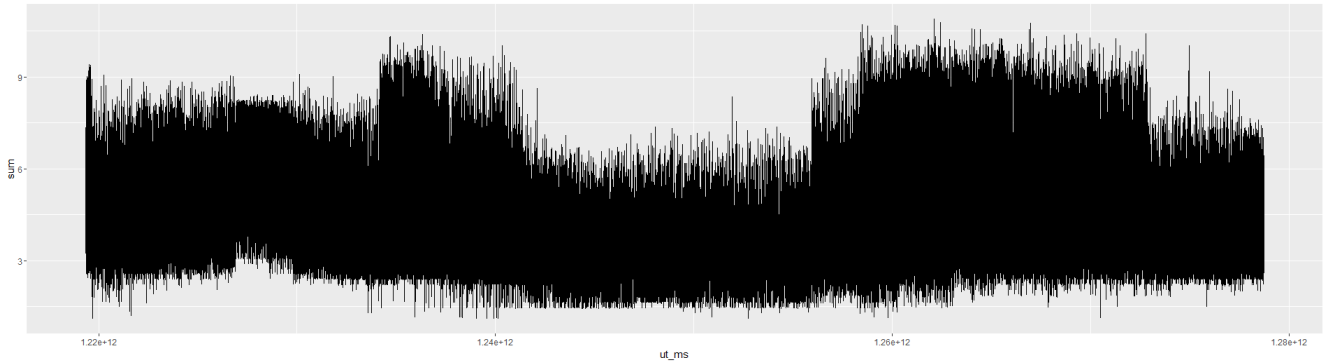


Figura 1. Suma de potencias para el primer año *power*

SAAF: Aspectos solares

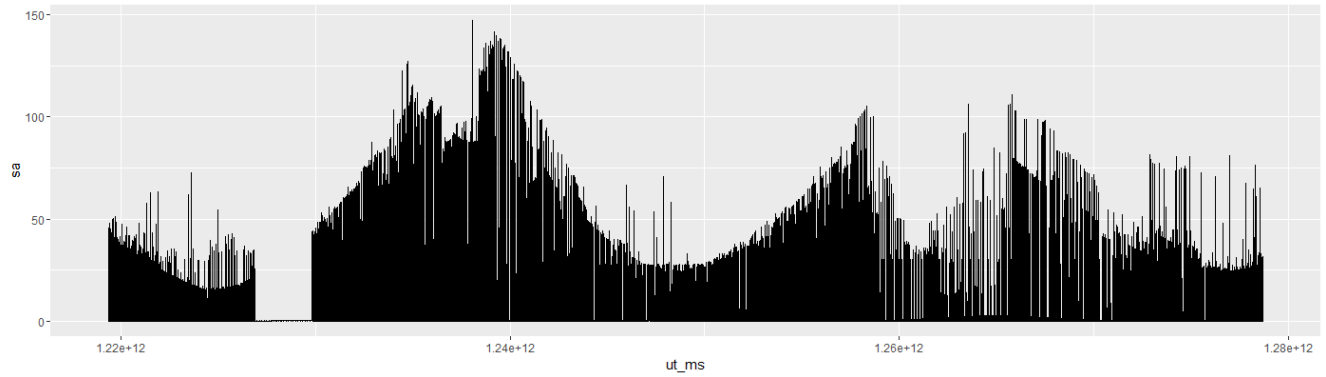


Figura 2. Incidencia Solar para el primer año $saaf\$sa$

LTDATA: Información de periodos extendidos

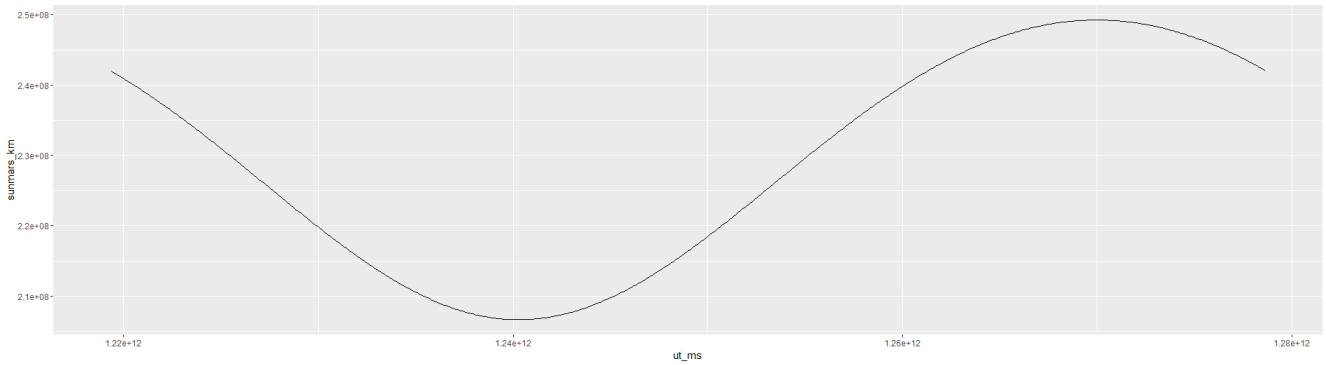


Figura 3. Distancia Sol-Marte para el primer año

Los otros sets del problema tienen datos que no se pueden visualizar de forma directa. Estos están estructurados en el siguiente formato:

DMOP: Detalles de la planificación de operación

Tabla 1. Muestra registros de dmop.csv

ut_ms	subsystem
1219363211000	AXXX301A
1219364909000	AAAAF20C1
1219364924000	AAAAF60A1
1219366035000	AXXX380A
1219366635000	ASEQ4200
1219367381000	ATTTF301E

FTL: Eventos de la trayectoria de la nave

Tabla 2. Muestra registros de ftl.csv

utb_ms	ute_ms	type	flagcomms
1219363213000	1219365494000	EARTH	FALSE
1219369619000	1219370253000	SLEW	FALSE
1219370253000	1219373093000	NADIR	FALSE
1219373093000	1219374563000	SLEW	FALSE
1219376633000	1219381144000	EARTH	TRUE

EVTF: Otros eventos

Tabla 3. Muestra registros de evtf.csv

ut_ms	description
1219365755000	"NNO_AOS_05_/_RTLT_02373"
1219368640000	"4000_KM_DESCEND"
1219369280000	"MRB_/_RANGE_06000KM_START"
1219369855000	.ºCC_MARS_200KM_START_/_RA_181.68_/_OMP_(296.35 -46.48)_/_SZA_077"
1219369949000	.ºCC_MARS_START_/_RA_181.69_/_DE_-00.08_/_OMP_(299.32 -43.44)_/_SZA_076"

Al graficar los valores de potencia, los datos de incidencia solar y los datos de misión, se puede apreciar que existe una ligera correlación entre los anteriores, por lo que se procedió a trabajar sobre estos datos en primera instancia.

3. Hipótesis

La hipótesis inicial es que existe una predicción *gruesa* y una predicción *fina*. La predicción gruesa tiene que ver con los datos de incidencia solar y eventos de la nave. La predicción fina tiene que ver con los datos de eventos y de comandos de la nave.

Los datos de comandos y de eventos se dejarán por fuera en esta primera instancia. Se considera que el procesamiento sobre estos datos debe ser mayor, ya que requiere generar periodos de ventana, lo cual se considera más complejo, ya que requiere identificar correlaciones entre los distintos circuitos y entre los distintos comandos, los cuales no están especificados como 'ON' y 'OFF' o de forma similar, como se puede apreciar en las Tablas 1, 2 y 3.

En este trabajo se avanzará sobre la hipótesis de la predicción gruesa. Para poder realizar predicción sobre los datos es necesario pre-procesar los datos para colocarlos en escalas de tiempos similares. Para esto se requerirá realizar un match en la escala temporal ($'ut\_ms'$) de los valores de incidencia solar con los de de potencia de entrada. Esta escala se encuentra en tiempo UNIX de milisegundos(POSIX), por lo que será necesario convertirlos a DateTime para facilitar su análisis.

Debido a que los valores para predicción se deben entregar como promedio por hora, será necesario calcular el promedio de los datos de entrenamiento, agrupados por hora y crear un nuevo dataframe con estos.

Posteriormente se realizará un merge sobre las tablas de valores antes calculadas para tener una sola tabla sobre la cual se entrenará al sistema.

Con el fin de poder realizar una correcta predicción, es necesario interpolar los valores faltantes para los valores de la misión de la nave, ya que estos últimos se reciben una vez al día.

Con la intención de probar que el modelo ha sido correctamente entrenado, se entrenará al sistema con un conjunto de los datos y se evaluará con otro conjunto. Para este fin se ha dividido el set de un año en dos.

Finalmente se realizarán mediciones para comprobar si este esquema de predicción es satisfactorio.

4. Pre-procesamiento de los datos

4.1. Escala temporal

Para convertir la escala temporal de los datos se utiliza el comando as.POSIXct, considerando el cambio entre milisegundos a segundos y con inicio de valores en 1970-01-01.[1]

```
1 power1DT <- power1
2 power1DT$ut_ms <- as.POSIXct((((power1['ut_ms'])/1000)[,]), origin="1970-01-01")
```

Tabla 4. Conversión de la escala temporal

[1] UNIX TIME: `https://en.wikipedia.org/wiki/Unix_time`

old(ut_ms)	new(DateTime)
1.2193632130E+12	21/08/2008 20:00:13
1.2193632350E+12	21/08/2008 20:00:35
1.2193632950E+12	21/08/2008 20:01:35
1.2193633550E+12	21/08/2008 20:02:35

4.2. Agrupación por hora

Se agruparon los valores por horas con dos intenciones. En primera instancia, el resultado de las predicciones debe entregarse en promedios por hora. Si bien, se considera que las predicciones pueden ser mucho más adecuadas si se utilizan todos los valores existentes, por motivos de capacidad computacional se decidió promediar para disminuir la cantidad de valores en los dataFrames, con lo que el procesamiento se puede acelerar.

Para agrupar los valores por hora, se ocupa el comando *cut*, seguido de *group_by*[2]:

```
power1DT$ut_ms <- cut(power1DT$ut_ms, breaks="hour")
power1DTHourMean <- power1DT %>% group_by(ut_ms) %>% summarise_each(funs(mean))
```

El resultado de esta operación disminuyó, para el frame *power*, el número de filas de *1830121* a *16454*.

4.3. Match de escalas temporales

Para poder entrenar el modelo, es necesario que los valores estén en el mismo instante temporal, de lo contrario, un valor de potencia en un instante t podría tener un valor de incidencia solar *NA*. El match se hace considerando como origen el tiempo del vector de potencias.

```
power1DTHourMeanMS <- power1DTHourMean$ut_ms

for (i in 1:nrow(ltdata1DTHM)) {
  nearest <- findInterval(ltdata1DTHM$ut_ms[i],power1DTHMms)
  ltdata1DTHM$ut_ms[i] <- power1DTHMms[nearest]
}
```

El resultado de esta operación busca en *ltdata* y *saaf* el valor de *ut_ms* más cercando en *power* y lo reemplaza.

4.4. Interpolación de valores faltantes

Los valores de *ltdata* se entregan originalmente uno por día. Para una correcta predicción es necesario interpolar estos valores para cada hora. Como se trata de distancias y angulos entre planetas y el sol, se puede interpolar linearmente todos los puntos faltantes. Lo anterior se realiza mediante *na.spline* y *na.approx*[3]

```
ltdata1DTHM$sunmars_km <- na.spline(ltdata1DTHM[,2],na.rm = FALSE)
ltdata1DTHM$earthmars_km <- na.spline(ltdata1DTHM[,3],na.rm = FALSE)
ltdata1DTHM$sunmarsearthangle_deg <- na.spline(ltdata1DTHM[,4],na.rm = FALSE)
ltdata1DTHM$solarconstantmars <- na.spline(ltdata1DTHM[,5],na.rm = FALSE)
ltdata1DTHM$occultationduration_min <- na.spline(ltdata1DTHM[,6],na.rm = FALSE)
ltdata1DTHM$eclipseduration_min <- na.approx(ltdata1DTHM[,7],na.rm = FALSE, rule=2)
```

4.5. Unión de valores

Posterior a realizar todos los cambios en los frames *power*, *ltdata* y *saaf*, se deben unir estos valores para tener un solo dataframe para entrenamiento. Esto se puede realizar mediante *merge*:

[2] group_by: library(dplyr)
[3] na.spline, na.approx: library(zoo)

```
power1DTHM<-merge(x=power1DTHM, y=saaf1DTHM, by="ut_ms", all.x=TRUE)
power1DTHM<-merge(x=power1DTHM, y=ltdata1DTHM, by="ut_ms", all.x=TRUE)
```

El resultado es un frame que tiene la escala de tiempo, todas las columnas de *power*, *ltdata* y *saaf*.

5. Procesamiento de los datos

5.1. Random Forest

Para la regresión se ocupó *Random Forest*[4]. La forma de ocupar la librería es intuitiva, se debe entrenar el modelo con un set y probarlo con otro. La forma de entrenar el set está dado por el siguiente código:

```
predictField <- 5 #Indice de la columna que se va a predecir
predictCols <- colnames(power1DT[,-1])
train <- power1DTHourMean[1:12000,-1]
test <- power1DTHourMean[12001:16000,-1]
colName <- predictCols[predictField]
rf <- randomForest(as.formula(paste(colName," ~ .")) ,data=train, ntree=10)
```

Para utilizar el modelo generado se utiliza el comando *predict*.

```
predicted <- predict(rf, test)
```

Existen varias limitantes encontradas durante el procesamiento de los datos, estas se discutirán en las conclusiones, sin embargo, se consiguió lograr un error bajo para varios campos, por lo que se consideró aceptable el método empleado.

5.2. Calculo de error

Para el cálculo del error es necesario definir la función entregada por la evaluación del Mars Challenge.

```
RMSE = function(predicted, reference){
  sqrt(mean((predicted - reference)^2))
}
```

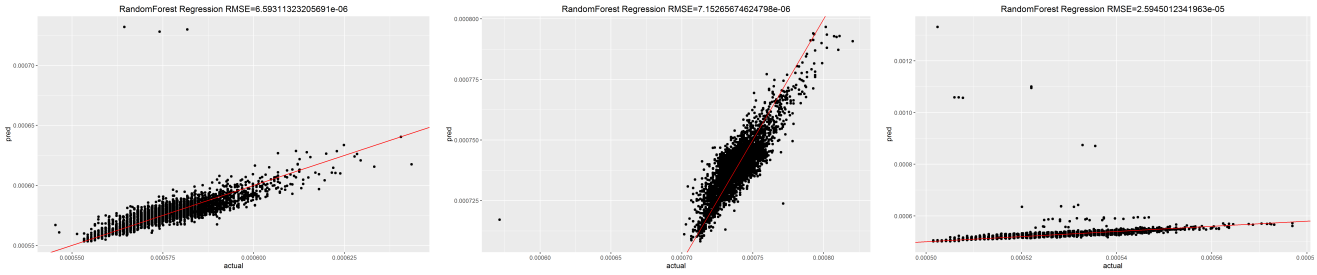
Para emplear dicha función, se requieren los valores predichos y la columna de los valores originales. De la misma manera se propone graficar los puntos de entrenamiento versus el cálculo obtenido para un análisis y comparación visual.

```
r2 <- RMSE(predCol, predicted)
p <- ggplot(aes(x=actual, y=pred),
            data=data.frame(actual=predCol, pred=predict(rf, test)))
p + geom_point() +
          geom_abline(color="red") +
          ggtitle(paste("RandomForest Regression RMSE=", r2, sep=""))
```

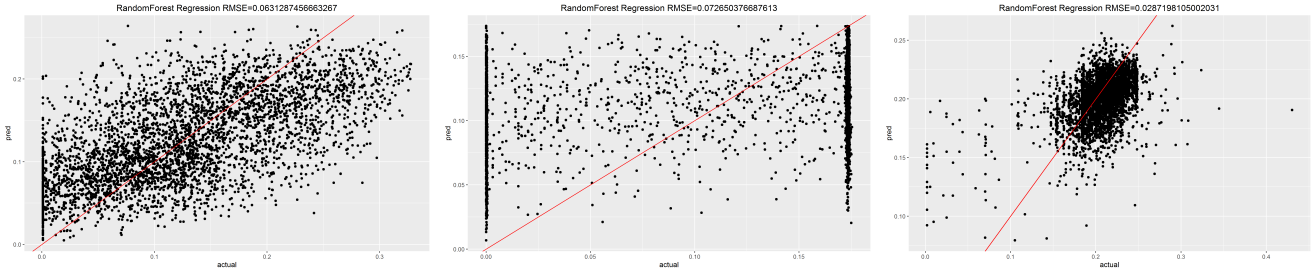
6. Resultados y Conclusiones

Al aplicar el método anterior se puede realizar una predicción bastante acertada para varios de los circuitos eléctricos. A continuación se presentan algunos resultados.

[4] library(randomforest)



Figuras 4, 5, 6. Predicción en buenos casos



Figuras 7, 8, 9. Predicción en malos casos

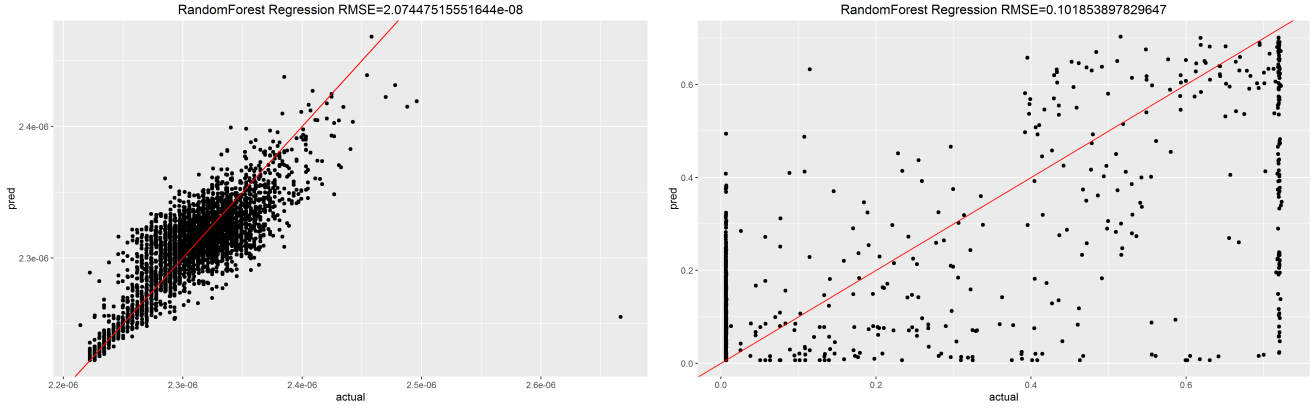


Figura 10. Mejor predicción: $\epsilon = 2{,}0744e^{-08}$
Figura 11. Peor predicción: $\epsilon = 0{,}10185$

Como se puede apreciar, la predicción realizada tiene valores de error muy bajos. En el mejor de los casos, se pudo predecir con un ϵ de $2{,}0744e^{-08}$. En el peor de los casos, el error fue de 0,10185. Al sumar el error para los 33 circuitos se pudo llegar a un error de 0,05079785, lo cual se considera suficiente para esta experiencia.

Como conclusión directa de lo anterior se desprende que el consumo de potencia tiene una estrecha relación con la cantidad de energía generada, lo que es intuitivo. Esta idea afirma la hipótesis de que existe una predicción gruesa y una fina.

Para poder considerar los otros sets de datos se considera realizar un pre-procesamiento adicional a los datos, como por ejemplo, encontrar la relación entre el consumo de los circuitos y la emisión de los comandos. De la misma manera, la incidencia solar tiene una dependencia sobre los eclipses solares. Este también sería una hipótesis sobre la cual avanzar.

Durante el (pre)procesamiento de los datos se encontraron varias dificultades, principalmente relacionadas con las capacidades computaciones. Procesar todos los datos sin agruparlos por hora conllevó tiempos extendidos, este tradeoff seguramente podría incrementar los valores de predicción, sin embargo se descarta para el desarrollo de este trabajo.