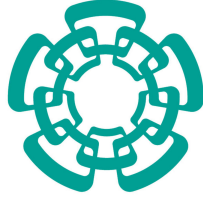


Ejercicio 5: Aplicación de técnicas de agrupación a dataset del mundo real



Análisis de datos

Profesor: Iván López Arévalo,

Rafael Pérez Torres

LTI CINVESTAV Tamaulipas

1 Introducción

Las técnicas de aprendizaje no supervisado permiten descubrir información en los datos sin tener información *a priori* de los mismos. Dichas técnicas han permitido la obtención de conocimiento a simple vista oculto en enormes datasets de información proveniente de la industria y de los ámbitos empresarial y científico.

Dos de las grandes familias de algoritmos para generar grupos (clústers) a partir de datos son las técnicas jerárquicas y particionales. Las técnicas jerárquicas buscan hacer divisiones incrementales en los datos, creando nuevos niveles con grupos específicos en cada iteración. Por otro lado, las técnicas particionales buscan dividir el espacio de objetos en áreas que se encuentran representadas por un punto central (prototipo) denominado centroide.

En este documento se muestran los resultados del análisis de un dataset de vuelos comprendidos durante el año 2002 en Estados Unidos, así como las observaciones encontradas al utilizar técnicas de clústering jerárquico (*híbrido*) y particional (*kmeans*) utilizando la suite *R*.

2 Metodología implementada

Esta sección describe los pasos más importantes que fueron realizados para analizar los datos, particularmente las etapas de preprocesamiento y el análisis de datos.

2.1 Preprocesamiento

2.1.1 Reducción de dimensionalidad

A partir de una primera inspección de los datos se eliminaron algunas de las columnas cuyo valor se mantenía constante, se encontraba ausente en todas las instancias, o que no aportaba valor alguno para el descubrimiento de información. A través de instrucciones como las que se muestran en PseudocódigoX, fue posible realizar la remoción de dichas columnas.

```
1 dsCrudo$CarrierDelay <- NULL
2 dsCrudo$WeatherDelay <- NULL
3 dsCrudo$NASDelay <- NULL
4 dsCrudo$SecurityDelay <- NULL
5 dsCrudo$LateAircraftDelay <- NULL
6 dsCrudo$Year <- NULL
```

Listing 1: Eliminación de columnas con valores constantes

Después de realizar una segunda vista rápida a los datos y a las descripciones de cada uno de los atributos, se determinó realizar la separación de los datos en tres familias: vuelos cancelados, vuelos desviados y vuelos con la información completa.

La determinación de los vuelos cancelados se realizó a través de:

```
1 dsVuelosCancelados <- ( subset(dsCrudo, dsCrudo$Cancelled == 1) )
```

Listing 2: Detección de vuelos cancelados

La determinación de los vuelos desviados se realizó a través de:

```
1 dsVuelosDesviados <- ( subset(dsCrudo, dsCrudo$Diverted == 1) )
```

Listing 3: Detección de vuelos desviados

Finalmente, la determinación de los vuelos con información de horarios completa fue determinada de forma manual, omitiendo aquellos registros que caracieran de algún valor en las columnas:

```
1 dsTiemposCompleto <- ( subset(dsCrudo, !is.na(dsCrudo$DepTime)) )
2 dsTiemposCompleto <- ( subset(dsTiemposCompleto, !is.na(dsTiemposCompleto$
  ArrTime)) )
```

```

3 dsTiemposCompleto <- (subset(dsTiemposCompleto, !is.na(dsTiemposCompleto$
  CRSArrTime)))
4 dsTiemposCompleto <- (subset(dsTiemposCompleto, !is.na(dsTiemposCompleto$
  CRSDepTime)))
5 dsTiemposCompleto <- (subset(dsTiemposCompleto, !is.na(dsTiemposCompleto$
  ActualElapsedTime)))
6 dsTiemposCompleto <- (subset(dsTiemposCompleto, !is.na(dsTiemposCompleto$
  AirTime)))
7 dsTiemposCompleto <- (subset(dsTiemposCompleto, !is.na(dsTiemposCompleto$
  DepDelay)))
8 dsTiemposCompleto <- (subset(dsTiemposCompleto, !is.na(dsTiemposCompleto$
  ArrDelay)))

```

Listing 4: Detección de vuelos con información de horarios completa

2.2 Tratamiento de los registros con información de horarios completa

Los registros con la información de horarios completa fueron los únicos considerados para ingresar en la etapa de descubrimiento de información (clústering). Su preparación fue realizada a través de las actividades descritas a continuación.

2.2.1 Separación por mes

Con la intención de realizar un análisis y muestreo estratificado que permitiera reducir la cantidad de instancias a analizar, los datos se separaron en base al mes de la fecha de partida del viaje. Instrucciones similares a la mostrada en el Listing X permitieron realizar la separación.

```

1 dsEnero <- subset(dsTiemposCompleto, dsTiemposCompleto$Month == 1)

```

Listing 5: Separación de registros de vuelos por mes

Un caso especial fue el de los registros pertenecientes al mes de abril, ya que algunos de los valores de las fechas no podían ser transformados a su representación en milisegundos (más información de esta transformación es encontrada en las secciones posteriores del documento). Por ello, se omitieron algunas de estas instancias (44 en total) si fueron registradas con una fecha del domingo 7 de abril y en el rango de las 02:00 horas. Dicha discriminación fue realizada a través del código mostrado en el Listing X.

```

1 dsAbril <- subset(dsTiemposCompleto, dsTiemposCompleto$Month == 4)
2 dsAbrilProbablesConflictivos <- subset(dsAbril, dsAbril$DayofMonth == 7)
3 dsAbrilConflictivosHraSalida <-

```

```

4  subset( dsAbrilProbablesConflictivos , dsAbrilProbablesConflictivos$DepTime
    >= 200 & dsAbrilProbablesConflictivos$DepTime < 300)
5  dsAbrilConflictivosHraEstimadaSalida <-
6  subset( dsAbrilProbablesConflictivos , dsAbrilProbablesConflictivos$
    CRSDepTime >= 200 & dsAbrilProbablesConflictivos$CRSDepTime < 300)
7  dsAbrilConflictivosHraLlegada <-
8  subset( dsAbrilProbablesConflictivos , dsAbrilProbablesConflictivos$ArrTime
    >= 200 & dsAbrilProbablesConflictivos$ArrTime < 300)
9  dsAbrilConflictivosHraEstimadaLlegada <-
10 subset( dsAbrilProbablesConflictivos , dsAbrilProbablesConflictivos$
    CRSArrTime >= 200 & dsAbrilProbablesConflictivos$CRSArrTime < 300)
11 dsAbrilConflictivos <- rbind(dsAbrilConflictivosHraEstimadaLlegada ,
    dsAbrilConflictivosHraEstimadaSalida , dsAbrilConflictivosHraSalida ,
    dsAbrilConflictivosHraLlegada)
12 diferencia <- dsAbril[!duplicated(rbind(dsAbrilConflictivos , dsAbril))[-seq-
    len(nrow(dsAbrilConflictivos))], ]
13 dsAbril <- diferencia

```

Listing 6: Separación de registros del mes de abril

2.2.2 Muestreo aleatorio

Se realizó un submuestro aleatorio de las instancias de cada uno de los meses. Para el caso del clústering particional (skmeans) se optó por un submuestro del 10% del total de los registros. Sin embargo, esta cantidad resultaba demasiado alta para el algoritmo jerárquico (el equipo de cómputo utilizado agotaba su memoria) por lo que para este caso se utilizó únicamente el 1%. A través de instrucciones similares a las mostradas en el Listing X se realizó este muestreo.

```

1 dsEnero <- dsEnero[sample(1:nrow(dsEnero) , as.integer(length(dsEnero$Month) *
    tamanyoMuestra) , replace=FALSE) ,]

```

Listing 7: Muestreo aleatorio de registros por mes

Cálculo de las fechas en formato de milisegundos Debido a que la información de fechas estimadas y reales tanto de llegada como de salida de los vuelos se encontraba dividida, se procedió a realizar su unificación en un solo atributo *sintético* que representa la cantidad de milisegundos transcurridos entre cada una de las fechas en cuestión y el 1 de enero de 1970 (conocido como *the epoch*).

Al final, la intención es reducir la cantidad de columnas y mantener una variable continua que permitiera ser utilizada por los algoritmos. La función mostrada en el Listing

X. Permite realizar el cálculo de las representaciones de las fechas en milisegundos.

```
1 calcularMsHoras <- function (ds){
2   parteFecha <- sprintf("2002-%02d-%02d", as.numeric(ds$Month), as.numeric(ds$
   DayofMonth))
3
4   horaSalida <- sprintf("%04d", ds$DepTime)
5   horaLlegada <- sprintf("%04d", ds$ArrTime)
6   horaEstimadaSalida <- sprintf("%04d", ds$CRSDepTime)
7   horaEstimadaLlegada <- sprintf("%04d", ds$CRSArrTime)
8
9   # Hora real salida
10  fechaSalida <- sprintf("%s %s:%s:00", parteFecha, substr(horaSalida, 1, 2),
   substr(horaSalida, 3, 4))
11  msHoraSalida <- as.integer( ymd_hms(fechaSalida, tz = "America/Mexico_City")
   )
12
13  # Hora real llegada
14  fechaLlegada <- sprintf("%s %s:%s:00", parteFecha, substr(horaLlegada, 1, 2),
   substr(horaLlegada, 3, 4))
15  msHoraLlegada <- as.integer( ymd_hms(fechaLlegada, tz = "America/Mexico_City
   "))
16
17  # Hora estimada salida
18  fechaEstimadaSalida <- sprintf("%s %s:%s:00", parteFecha, substr(
   horaEstimadaSalida, 1, 2), substr(horaEstimadaSalida, 3, 4))
19  msHoraEstimadaSalida <- as.integer( ymd_hms(fechaEstimadaSalida, tz = "
   America/Mexico_City"))
20
21  # Hora estimada llegada
22  fechaEstimadaLlegada <- sprintf("%s %s:%s:00", parteFecha, substr(
   horaEstimadaLlegada, 1, 2), substr(horaEstimadaLlegada, 3, 4))
23  msHoraEstimadaLlegada <- as.integer( ymd_hms(fechaEstimadaLlegada, tz = "
   America/Mexico_City"))
24
25  mss <- list("msHoraEstimadaSalida" = msHoraEstimadaSalida,
26             "msHoraSalida" = msHoraSalida,
27             "msHoraEstimadaLlegada" = msHoraEstimadaLlegada,
28             "msHoraLlegada" = msHoraLlegada)
29  mss
30 }
```

Listing 8: Transformación de las horas (timestamps) a milisegundos

2.2.3 Eliminación de información superflua y/o redundante

Debido a los cambios introducidos por el cálculo de milisegundos, algunas de las columnas pierden su razón de ser y se procedió a eliminarlas de los datasets. Por ejemplo, los datos señalando al mes, día del mes y a las fechas en particular son sustituidos por las columnas creadas.

Adicionalmente, datos como los que indican si el vuelo ha sido cancelado o desviado fueron también eliminados, ya que los registros de estos datasets cuentan con la información de horarios completa (recordando que los registros de vuelos desviados o cancelados ya fueron separados al inicio del pre-procesamiento).

Es importante destacar que otras columnas como el número de vuelo, el identificador del carrier y del avión también fueron eliminadas dado que representan identificadores que, según la perspectiva seguida, no aportan mayor información al proceso.

Finalmente, se realizó la detección de información redundante a través del cálculo de correlaciones. La Figura X muestra las correlaciones de uno de los datasets. Puede observarse que existe gran correlación entre los datos que indican información temporal, lo cual es una situación esperada ya que, por ejemplo, el crecimiento en la hora de partida de un avión incrementa también la hora de llegada. Otro ejemplo evidente es lo que sucede entre las variables de tiempo total de vuelo y distancia recorrida, que tienen una correlación amplia debido a que mientras más tiempo tenga el avión en vuelo, mayor distancia habrá recorrido.

FIGURA DE CORRELACIONES!

2.2.4 Conversión de valores nominales a numéricos

En una de las experimentaciones se consideró que la información referente al origen y al destino podría tener algún grado de importancia, por lo que fueron transformadas de un tipo factor a uno numérico para su inclusión en el proceso de clasificación. Esto fue realizado a través de instrucciones similares a las mostradas en el Listing X:

```
1 dsEnero$Origin <- as.integer(factor(dsEnero$Origin))
2 dsEnero$Dest <- as.integer(factor(dsEnero$Dest))
```

Listing 9: Conversión de valores nominales (factor) a numérico

La conversión a valor numérico de los atributos nominales permitió realizar la ejecución del clústering particional. Sin embargo, los resultados arrojados por el clústering contenían alta dimensionalidad y estos atributos al ser identificadores fueron candidatos a ser eliminados par ala ejecución final.

Al final de estas tareas, se obtuvo la lista definitiva de atributos a considerar para el proceso de análisis de componentes principales (*PCA*), indicada en la Tabla 2.1.

Columna	Significado
DayOfWeek	Día de la semana
AirTime	Tiempo de vuelo
ArrDelay	Atraso en llegada
DepDelay	Atraso en salida
TaxiIn	Tiempo en hangar entrada
TaxiOut	Tiempo en hangar salida

Table 2.1: Conjunto final de columnas consideradas

2.2.5 Análisis de componentes principales (*PCA*)

Adicionalmente a los pasos anteriores, se implementó como etapa previa al clústering el análisis de componentes principales (*PCA*, por sus siglas en inglés). El proceso de *PCA* fue alimentado con los registros de los meses con sus datos escalados. De los componentes principales generados, originalmente se habían elegido cinco para realizar el proceso de clústering. Sin embargo esta dimensionalidad complicaba el análisis de los resultados por lo que se eligió un valor de 3 componentes para ejecutar el proceso de clústering.

Al final de estas tareas, se obtuvieron doce datasets con un promedio de X registros y Y columnas para el clústering particional y X registros y Y columnas para el clústering jerárquico. Originalmente, estos datasets tenían X registros y Y columnas en el dataset original.

2.3 Análisis de datos

2.3.1 En base a frecuencias

Será redactado

2.3.2 Con herramientas de clústering

Ejecución del proceso de clústering Las técnicas de clústering fueron implementadas sobre las versiones reducidas de los datasets. El equipo de cómputo donde se ejecutaron

los procesos es una iMac de 21.5 pulgadas, con 4GB de memoria RAM y un procesador Intel i5 con 4 núcleos con una velocidad de 2.5 GHz.

Clústering particional El algoritmo indicado para la realización del clustering particional fue el *K-Means*, en particular la función implementada dentro del paquete *skmeans* de *R*, llamada también *skmeans*. La ejecución de la función *skmeans* fue realizada con tamaños de clúster desde 2 hasta 8 por cada uno de los datasets de los meses. Algunos de los resultados técnicos de este experimento se encuentran descritos en la Tabla 2.2.

k=	Tiempo (seg)	Valor (criterion)
2	11.33	877.64
3	13.92	5591.99
4	16.97	4099.81
5	21.48	3147.88
6	22.94	2685.38
7	53.70	2350.11
8	42.13	2113.81

Table 2.2: Resultados técnicos de la ejecución de la técnica *skmeans* para el mes de enero

Clústering jerárquico La técnica indicada para la ejecución del clustering jerárquico fue el *hybridHclust* (Hybrid hierarchical clustering using mutual clusters), el cual sigue el enfoque jerárquico top-down. Este algoritmo no requiere parámetros adicionales al dataset a procesar. La ejecución de este algoritmo fue realizada en base a los datos escalados de cada uno de los meses. Algunos de los resultados técnicos de este experimento se encuentran descritos en la Tabla F2.

3 Interpretación de los resultados

La realización de los procesos de análisis en base a frecuencias y agrupamiento permitió obtener algunos detalles de la información contenida en los datos.

Es importante mencionar los siguientes puntos acerca de la ejecución de este algoritmo. Se ejecutaron varias rondas debido a modificaciones en los parámetros, sobre todo las columnas seleccionadas en cada uno de los datasets. En este sentido, inicialmente

se habían considerado datos que tenían relación con las componentes temporales (timestamp) de los registros, como por ejemplo las fechas reales y estimadas de llegada y partida; sin embargo, éstos dificultaban la generación y la representación de la información, ya que al representar eventos que suceden a lo largo del tiempo, su intersección con otros elementos era mínima.

INDICAR QUE HUBO UNA RETRAOLIMENTACIÓN EN LAS ETAPAS (no exitosa, pero existió).