UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

ANÁLISIS DE DATOS

HIPERTIROIDISMO

ADOLFO GUZMÁN

IGNACIO IBAÑEZ

Profesor: Felipe Bello

Ayudantes: Fernanda Lobos

Alfonso Guzmán

Santiago - Chile

9 de abril de 2017

TABLA DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

CAPÍTULO 1. INTRODUCCIÓN

El cuerpo necesita sistemas de regulación para mantener su homeostasis, la que consiste en el equilibrio de un medio interno. Básicamente existen dos mecanismos homeostáticos: el sistema nervioso y el sistema endocrino. Este último sistema, regula el cuerpo a través de vías endocrinas manejando niveles de glucosa, relaciones de macronutrientes y control de procesos anabólicos.(“Sistema endócrino”, 2016).

Por lo tanto, es de suma importancia estudiar las enfermedades del sistema endocrino, ya que pueden tener efectos adversos de largo plazo e incluso irreparables. Una de estas enfermedades corresponde al hipertiroidismo, es un cuadro provocado por el exceso de secreciones de hormonas tiroideas tales como la tiroxina(T4) o la triyodotironina(T3) que puede generar síntomas como dificultad para concentrarse, fatiga e incluso problemas cardíacos y osteoporosis (Wisse, 2016).

En la presente experiencia se utilizan árboles de decisión que permiten la clasificación de los pacientes dentro de los cuatro posibles diagnósticos contemplados en la base de datos. Además, hay que tener en consideración que estos datos son orientados a la aplicación del algoritmo C50 de Quinlan, el que corresponde al utilizado, pero bajo su implementación en R.

El informe está estructurado en seis capítulos. En la introducción se describe brevemente el contexto de la experiencia, en el marco teórico se exponen los rudimentos sobre los árboles de decisión, la obtención de reglas en donde se aplica el algoritmo de Quinlan a la base de datos. Más tarde, en el análisis de los resultados se realiza la revisión de las reglas obtenidas y en la Comparación estos resultados son contrapuestos con las reglas generadas en la experiencia anterior. En el capítulo final, la Conclusión, se rescata el aprendizaje y las ideas principales obtenidas de la experiencia, destacando lo que se hizo bien y lo que se debe mejorar.

CAPÍTULO 2. MARCO TEÓRICO

2.1 ÁRBOLES DE DECISIÓN

El procedimiento árbol de decisión consiste en la creación de un modelo de clasificación basado en árboles que clasifica casos en grupos intentando distinguir entre un conjunto de características y poder seleccionar aquellas que nos permitan realizar una mejor clasificación de los individuos aportando una mayor ganancia de información ("Creación de árboles de decisión", 2016).

Por lo antes mencionado se pueden encontrar dentro de la clasificación dos tipos de atributos:

- Atributos estudiantes (V^j) : corresponden a la totalidad de atributos con los cuales se intentara realizar una caracterización de los objetos.
- Atributos experto (C_i) : será seleccionado dentro del universo de atributos y tienen la cualidad de poder separar los n objetos en diferentes clases.

Como se puede ver en el árbol de decisión presentado a continuación, se encuentra un atributo experto separado en dos clases N (No presente) o P (Presente), además se utilizan diferentes atributos estudiantes en este caso son ambiente, viento y humedad.

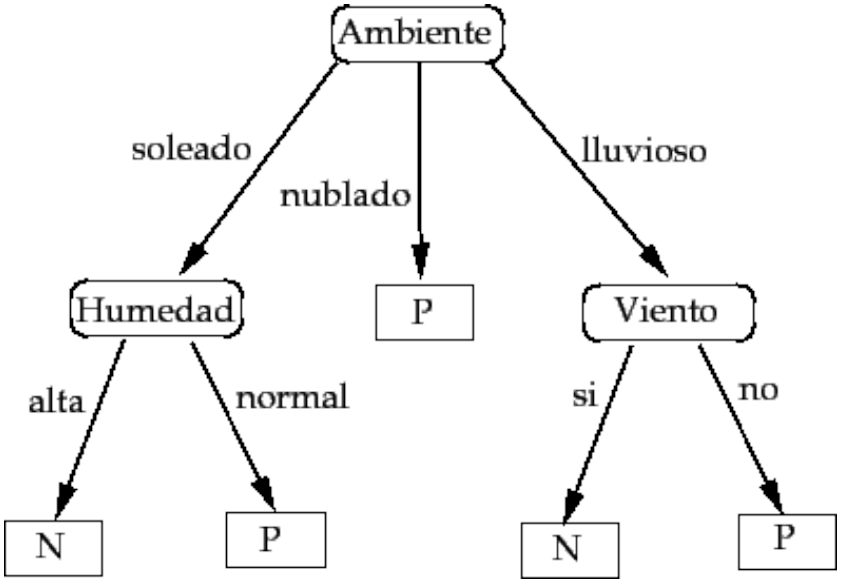

Figura 2-1: Ejemplo árbol de decisión para saber si se puede jugar Golf.

2.1.1 Algoritmo árbol de decisión

El procedimiento que siguen los diferentes algoritmos para la generación del árbol de decisión se inicia con el primer atributo que particiona los objetos (en el caso anterior fue ambiente), cada subconjunto restante es un nuevo problema de aprendizaje a su vez, con menos objetos y un atributo menos, luego de trabajar con cada atributo se pueden producir 4 casos diferentes ("Inducción de Árboles de Decisión", 2012).

1. Si existen objetos que pueden pertenecer a las distintas clases, escoge el mejor atributo para particionar.

2. Si todos los atributos restantes pertenecen a una sola clase, termina.

3. No quedan objetos (no ha sido observado un objeto con esa combinación de atributos). Regresa un default en base a la clasificación mayoritaria de su nodo padre).

4. No hay más atributos, pero seguimos con objetos perteneciente a diferentes clases.

El procedimiento de encontrar un árbol puede ser trivial, pero no son buenos para predecir casos no vistos. El problema que se produce es que solo se memorizan los casos vistos, por lo que no podemos esperar que extrapole.

2.1.2 Ganancia de información

Una vez finalizado con la generación del árbol se debe encontrar aquellos atributos estudiante que caracterizan de mejor forma las clases, es decir, realizan una mejor distinción entre ellas y para esto debe ser calculada la ganancia de información.

$$Ganancia(v^j, c) = \sum_i \sum_k p(v_i; c_k) ld(\frac{p(v_i; c_k)}{p(v_i)p(c_k)}) \tag{2.1}$$

CAPÍTULO 3. OBTENCIÓN DEL ÁRBOL

3.1 PREPROCESAMIENTO

El preprocesamiento en esta experiencia es realizado casi directamente, ya que según se muestra en la siguiente sección del capítulo la función utilizada en la experiencia, llamada C5.0, tiene la capacidad de trabajar automáticamente con atributos de tipo binario y atributos reales. Lo que permite trabajar con los atributos presentes en la base de datos tales como Tumor, Sick y pregnant junto con TSH o TT4.

Además, hay que tomar en consideración que el conjunto de datos fue diseñado para la utilización del algoritmo C50 sugerido por Quinlan y por tanto la aplicación de la función solicitada es sencilla comparando con experiencias anteriores.

3.2 FUNCIÓN UTILIZADA

La función utilizada en esta oportunidad pertenece a la biblioteca C50 y es conocida como C5.0. Retorna un objeto de la clase C5.0 y según lo especificado en los parámetros puede contener las reglas, el árbol generado, la matriz de confusión, etc.

Esta implementa el algoritmo creado por Quinlan para la generación de árboles de decisión. Por otro lado, ya que en ocasiones entender el significado de un árbol de decisión puede llegar a ser complejo, permite la generación de las reglas asociadas, mostrando su lift, el antecedente y el consecuente, para realizar una comparación directa con las reglas obtenidas en la experiencia anterior.

Por lo tanto, es necesario identificar el atributo experto el cual en este caso corresponde a la clase que muestra el diagnóstico Goitre, intoxicamiento por T3, hipertiroidismo y negativo.

Dentro de los parámetros utilizados se encuentran.

- X: Contiene la matriz de objetos con los respectivos atributos estudiantes. En este caso son utilizados todos los que aparecen en la base de datos menos Class.

- Y: Un vector con los valores del atributo experto, en este caso, negativo, goitre, hipertiroidismo e intoxicamiento por T3.

- rules: Es un valor booleano para indicar si se deben crear o no las reglas. Un valor verdadero indica que se deben crear, un valor falso indica que no se deben crear. Como uno de los objetivos de esta experiencia es comparar las reglas generadas con el árbol con las reglas generadas en la experiencia anterior este atributo es ingresado con el valor verdadero.

No obstante, para complementar el modelo generado por el árbol se reingresan los datos para verificar la predicción. Para lograrlo se utiliza la biblioteca gmodels y la función CrossTable, que permite ver los niveles de error y la tabla de contingencia entre las clases a las que pertenecen y las clases que fueron predichas.

3.3 OBTENCIÓN DE REGLAS

Las reglas obtenidas a partir de la generación del árbol son seis, que corresponden a:

1.- $OnThyroxine = f, Psych = f, TSH <= 0{,}31, TT4 > 141, FTI > 170 \rightarrow classhyperthyroid.[0{,}849]$

2.- $TSH > 0{,}31 \rightarrow classnegative.[0{,}997]$

3.- $OnThyroxine = f, TT4 <= 141, \rightarrow classnegative.[0{,}993]$

4.- $Psych = t \rightarrow classnegative.[0{,}992]$

5.- $FTI <= 170 \rightarrow classnegative.[0{,}991]$

6.- $OnThyroxine = t \rightarrow classnegative.[0{,}989]$

A continuación se presentan los datos asociados a cada regla.

Regla	Bien clasificados	Mal clasificados	lift
1	51	7	35.2
2	1507	4	1.0
3	1570	10	1.0
4	129	0	1.0
5	1867	15	1.0
6	183	1	1.0

Cuadro 3.1: Datos reglas generadas

Finalmente, se muestran los resultados de la aplicación del modelo creado a los datos, a través de una matriz de confusión generada con la función CrossTable de la biblioteca gmodels. Considerar que estos valores e incluso la configuración de la matriz dependen de las muestras que utiliza C50, ya que internamente realiza bootstraping.

Las celdas tienen el siguiente formato: Cantidad de elementos clasificados / proporción de la muestra.

Predicción / Clase real	Goitre	Hipertiroidismo	Negativo	Total fila
Goitre	7/0.004	0/0.00	0/0.00	7/0.04
Hipertiroidismo	0/0.00	44/0.023	3/0.002	47/0.024
Negativo	1/0.001	7/0.004	1879/0.966	1887/0.97
Intoxicamiento T3	0/0.00	0/0.00	5/0.003	5/0.003
Total columna	8/0.004	51/0.026	1887/0.97	1946/1

Cuadro 3.2: Tabla de confusión tras ejecución de la predicción de datos.

CAPÍTULO 4. ANÁLISIS DE LOS RESULTADOS

4.1 REGLAS OBTENIDAS

Las reglas obtenidas en un árbol de decisión nacen del recorrido desde la raíz del árbol. Entonces, dado el método utilizado es posible obtenerlas de manera automática, ya fueron expuestas anteriormente.

La desventaja de este método es que el Lift de las reglas generadas no es prometedor en la mayoría de ellas y se vuelve necesario estudiar cada una por separado para entender su significado conceptual.

1.- $OnThyroxine = f, Psych = f, TSH <= 0{,}31, TT4 > 141, FTI > 170 \rightarrow classhyperthyroid.[0{,}849]$

Como se observa, el paciente no debe tener cuadros psiquiátricos y tampoco estar en tratamiento de tiroxina. Lo interesante de la regla viene dado por los niveles hormonales que están involucrados, el nivel de TSH debe ser bajo, mientras que las hormonas tiroideas deben ser muy altas. Esta relación tiene lógica dentro del contexto, ya que la TSH y las demás hormonas están relacionadas a través de un circuito cerrado de retroalimentación negativa, en otras palabras, el aumento de hormonas tiroideas en el organismo inhibe la secreción de hormonas tiroideas. Sin embargo, los niveles de TT4 y FTI son tan altos que llevan a la TSH a niveles bajos, haciendo coincidir el perfil con el del paciente hipertiroide.

Otra característica importante de esta regla es que tiene un Lift de 35.2, lo que sugiere una relación de dependencia entre el antecedente y el consecuente.

2.- $TSH > 0{,}31 \rightarrow classnegative.[0{,}997]$

Esta regla quiere decir que si los niveles de TSH son altos, el organismo busca aumentar las hormonas tiroideas en consecuencia de que sus niveles están muy bajos. Lo que lleva a que no hay características determinantes para pensar que el paciente está enfermo.

3.- $OnThyroxine = f, TT4 <= 141, \rightarrow classnegative.[0{,}993]$

La interpretación de esta regla es directa, debido a que los niveles de hormonas tiroideas son bajas no hay presencia de hipertiroidismo, el cuál se relaciona con excesos de secreciones endocrinas tiroideas.

4.- $Psych = t \rightarrow classnegative.$[0,992] Esta regla puede ser consecuencia de las incidencias de la base de datos, ya que en realidad los síntomas del hipertiroidismo pueden estar relacionados con trastornos como irritabilidad u otros.

5.- $FTI <= 170 \rightarrow classnegative.$[0,991] Esta regla, es interpretada igual que la regla 3, ya que el FTI corresponde a los niveles de T4 en estado activo.

6.- $OnThyroxine = t \rightarrow classnegative.$[0,989] Si está bajo un tratamiento de tiroxina, entonces el paciente tiene un déficit de esta hormona y no posee los niveles necesarios para llegar a tener hipertiroidismo.

Como se adelantó anteriormente, las reglas tienen un Lift que sugiere independencia en las variables, en las reglas desde la 2 a la 6 (como se observa en la tabla 3.1), lo que puede ser explicado por las incidencias de la base de datos o también se puede ver por la eliminación de los registros que contienen valores nulos, ya que esto puede traer como resultado un sesgo en los resultados. Entonces el valor de estas reglas necesita ser corroborado por la generación del árbol sobre una base de datos con los mismo atributos estudiantes y expertos de esta.

Desde otra perspectiva, si se toma en consideración la tabla 3.2 donde se muestra la predicción versus la clase real, se ve como los errores de clasificación tienden a ser muy bajos, concentrándose la mayoría de los errores en los pacientes negativos, sin embargo a nivel porcentual estos errores son casi despreciables. El otro error interesante, es que no aparecen casos predichos de intoxicamiento de T3, lo que puede ser consecuencia de que estos casos tienen un número muy pequeño en la base de datos, el poder generar una regla para clasificarlos está sobre un soporte muy pequeño y al generar el árbol se pierde la característica entre el grupo más significativo, es decir, el que tiene mayor cantidad de ocurrencias dentro de la base de datos, que en este caso son los pacientes negativos.

CAPÍTULO 5. COMPARACIÓN DE MÉTODOS

Antes de realizar la comparación es necesario realizar un recordatorio de la experiencia anterior, en donde se trabajo bajó el alero de la minería de reglas de asociación y se generaron reglas a partir de la función apriori implementada en R.

Las reglas obtenidas en esa oportunidad corresponden a:

1. Sex=F, OnThyroxine=f, QueryOnThyroxine=f, FTI (198,395]=1 ⇒ Class=hyperthyroid.
2. Sex=F, OnThyroxine=f, FTI (198,395]=1⇒Class=hyperthyroid.
3. OnThyroxine=f, QueryOnThyroxine=f, FTI (198,395]=1⇒Class=hyperthyroid.
4. OnThyroxine=f, FTI (198,395]=1⇒Class=hyperthyroid.
5. Sex=F, QueryOnThyroxine=f, FTI (198,395]=1⇒Class=hyperthyroid.
6. Sex=F, FTI (198,395]=1⇒Class=hyperthyroid.
7. QueryOnThyroxine=f, FTI (198,395]=1⇒ Class=hyperthyroid.
8. FTI (198,395]=1⇒Class=hyperthyroid.

También es importante recalcar los niveles de Lift de cada una de estas reglas (tabla 5.1).

Lo obvio es que la cantidad de reglas generadas en la experiencia anterior excede a la cantidad de reglas generadas en esta oportunidad. Una explicación puede corresponder a que los métodos se creación difieren, es decir, el árbol de decisión primero detecta qué variable es la que entrega mayor información con respecto al atributo experto, mientras que las reglas de asociación genera todas las reglas de manera independiente, recordar que el costo computacional de esta operación podía llegar a provocar que R fallara, entonces no tenía un criterio robusto para la elección de las reglas como lo es la información y solo escogió las reglas a partir de los niveles de confianza y soporte.

Sin embargo, más allá de las diferencias de los métodos, es importante realizar la comparativa entre las reglas generadas. Como se observa las reglas en ambas experiencias

Regla	Soporte	Confianza	Lift
[1]	0.01079137	0.8750000	36.22872
[2]	0.01079137	0.8400000	34.77957
[3]	0.01233299	0.8275862	34.26559
[4]	0.01233299	0.8000000	33.1234
[5]	0.01079137	0.7777778	32.20331
[6]	0.01079137	0.7500000	31.05319
[7]	0.01233299	0.7500000	31.05319
[8]	0.01233299	0.7272727	30.11219

Cuadro 5.1: Reglas generadas en la experiencia anterior ordenadas según lift

resultan similares en cuanto a los niveles hormonales para clasificar al paciente en una clase u otra. Esto se repite en la reglas tres y seis de la experiencia actual con respecto a las reglas ocho y nueve de la experiencia anterior, en que los niveles de FTI correspondientes son un factor determinante a la hora de crear los antecedentes de las reglas.

Por lo contrario, es interesante observar como los niveles de FTI son la raíz del árbol, es decir, el árbol generado a partir de los niveles de esta hormona es el que entrega mayor cantidad de variabilidad y este hecho puede verse reflejado en las reglas de asociación gracias a que, como se mencionó, el atributo aparece en casi todos los antecedentes.

Una desventaja que tuvo el método del árbol de asociación es que eliminó una variable que agrega riesgo a sufrir la enfermedad, que corresponde al sexo y que se de hecho es esencial en las reglas de asociación generadas en la experiencia anterior.

La otra desventaja presentada por este método, es que las reglas generadas tienen un bajo valor Lift, como se observa en la tabla 3.1, pero como se observa en la tabla 3.2 la capacidad de clasificación del modelo queda patente. Ahora si se toman en consideración las reglas de asociación generadas en la experiencia anterior se observa como son altamente redundantes y que por lo tanto el valor del Lift de cada una de ellas es consecuencia de que son solo derivación de la regla principal hecho que no ocurre en esta experiencia, ya

que el algoritmo C50 elimina las redundancias.

CAPÍTULO 6. CONCLUSIÓN

En base a los resultados obtenidos se observa como los árboles de decisión son capaces de generar clasificadores altamente efectivos, ya que utilizan como criterio de elección de variables los niveles de información que aportan a la decisión final.

En la experiencia existía la desventaja de que parte de los datos utilizados pertenecen al dominios de los reales, sin embargo, la implementación del algoritmo C50 es capaz de tratar automáticamente este tipo de variable, a través de la generación de intervalos que evitan la redundancia, hecho que fue una desventaja en la experiencia anterior, ya que se crearon muchas reglas redundantes.

Si bien las reglas generadas en la experiencia, pueden resultar débiles, consecuencia de que el nivel de Lift es uno en cinco de las seis reglas, la única regla generada es altamente robusta y es capaz de clasificar correctamente un gran porcentaje de los pacientes pertenecientes a la base de datos.

Una situación que es mejorable de esta experiencia es que se podría llegar a utilizar la base de datos puesta a disposición de manera directa, es decir, sin la eliminación de los valores nulos, ya que es probable que la generación del árbol se haya visto sesgada gracias a la eliminación de los registros con atributos nulos, cuestión que se apoya con la idea de que este algoritmo es capaz de trabajar con esta situación de manera automática.

Otra característica de la experiencia que puede ser revisada para mejorar el rendimiento de esta técnica corresponde a la variación de las formas en que es generada la discretización del problema, es decir, tener la capacidad de crear los intervalos de los niveles hormonales a partir de los niveles considerados normales o altos desde el punto de vista de los laboratorios.

También, es interesante considerar que esta información es de hace más de 20 años, por lo tanto, la información recabada para generar las reglas de asociación ya no es necesaria, ya que existen nuevos exámenes más exactos que entregan más información. Esta situación es ejemplificada con el FTI, este examen, ya no es realizado por lo tanto las reglas creadas en estas experiencias no son válidas o generalizables a nuestros tiempo, ya que la información tiene valor en el tiempo.

CAPÍTULO 7. BIBLIOGRAFÍA

Inducción de Árboles de Decisión. (2012). Recuperado desde https://ccc.inaoep.mx/~emorales/Cursos/NvoAprend/node6.html

Creación de árboles de decisión. (2016). Recuperado desde http://www.ibm.com/support/knowledgecenter/es/SSLVMB_22.0.0/com.ibm.spss.statistics.help/spss/tree/idh_idd_treegui_main.htm

Sistema endócrino. (2016). Recuperado desde http://med.unne.edu.ar/enfermeria/catedras/fisio/sistema%20endocrino.pdf

Wisse, B. (2016). Hipertiroidismo. Recuperado desde https://medlineplus.gov/spanish/ency/article/000356.htm

CAPÍTULO 8. ANEXO: CÓDIGO FUENTE EN R

```
require(stats)
require(C50)
require(gmodels)
data <- read.table("allhyper.data",quote="\"",
                   comment.char="|", na.strings="?", sep = ",")

colnames(data)<- c("Age","Sex","OnThyroxine","QueryOnThyroxine",
                   "OnAntithyroidMedication","Sick","Pregnant","ThyroidSurgery",
                   "I131Treatment","QueryHypothyroid","QueryHyperthyroid","Lithium",
                   "Goitre","Tumor","Hypopituitary","Psych","TSHMeasured",
                   "TSH","T3Measured","T3","TT4Measured",
                   "TT4","T4UMeasured","T4U","FTIMeasured",
                   "FTI","TBGMeasured","TBG","referralSource","Class")


na.columns.before <- colSums(is.na(data))
na.total.before <- sum(is.na(data))

#FUnciara cortar variables y agregar atributos al sistema de informaci
#por cada uno de los intervalos creados.
createAtt <- function(var,numberOfCuts,dataFrame,name){
  cutVar <- cut(var, breaks = numberOfCuts)
  name.cut.var <- paste(name," ",levels(cutVar))
  return(name.cut.var)
}

#Se calcula el porcentaje de Na acumulado por cada atributo, de modo tal de poder
#determinar si alguno de ellos es posible de eliminar
na.columns.relative <- 100*na.columns.before/na.total.before

na.columns.relative2 <- 100*na.columns.before/nrow(data)
#Como se observa la mayorde los valores perdidos se encuentran en la columna TBG con
#la totalidad de sus valores como Na

data$TBG <- NULL

outlier <- data$Age>120
data<- data[!outlier,]

#Finalmente se eliminan todos aquellos ejemplos que contengan
#algn valor perdido.
```

```
data.clean <- na.omit(data)

#Se calcula la cantidad de registros perdidos
lost.examples.total <- (nrow(data.clean)-nrow(data))/nrow(data)

#Hecho esto, se posible observar que las columnas que informan si un examen
#fue realizado o no, pierden importancia y por tanto puede ser eliminadas
#del dataset que estiendo utilizado, ya que todas entregarel valor V

data.clean$TBGMeasured  <- NULL
data.clean$FTIMeasured  <- NULL
data.clean$T4UMeasured  <- NULL
data.clean$TT4Measured <- NULL
data.clean$T3Measured <- NULL
data.clean$TSHMeasured <- NULL


#Por otro lado la fuente del ejemplo no aporta en el estudio de agrupamiento,
#entonce se utiliza

data.clean$referralSource <- NULL

  hiper.model <- C5.0(data.clean[-22],data.clean$Class, rules = TRUE)

  data.pred <- predict(hiper.model,data.clean)

  print(summary(hiper.model))
  confusion.matrix <- CrossTable(data.clean$Class, data.pred,prop.chisq = FALSE,
               prop.c = FALSE, prop.r = FALSE,dnn = c('Actual Class', 'predicted Class'))
```