Algoritmo ID3 mixto(quórum e incertidumbre)

Jesús Amador Garrocho Jiménez
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
jesgarjim5@us.es jesusgarrocho1998@gmail.com

Juan Manuel Garrocho Jiménez
dpto. Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla
Sevilla, España
juagarjim3@us.es juanmagarrocho@gmail.com

I. INTRODUCCIÓN

¿Qué es un árbol de decisión? Antes de comenzar directamente en cuestión sobre el algoritmo desarrollado, intentaremos situarnos en el contexto adecuado de los distintos elementos implicados. Podemos definir un árbol de decisión como un modelo de predicción utilizado en distintos ámbitos. En nuestro caso nos centraremos en el campo de la inteligencia artificial. Dado un conjunto de datos se fabrica dicho árbol cuya función no es otra que clasificar o categorizar una serie de sucesos o condiciones que suceden de un modo continuado. [1]

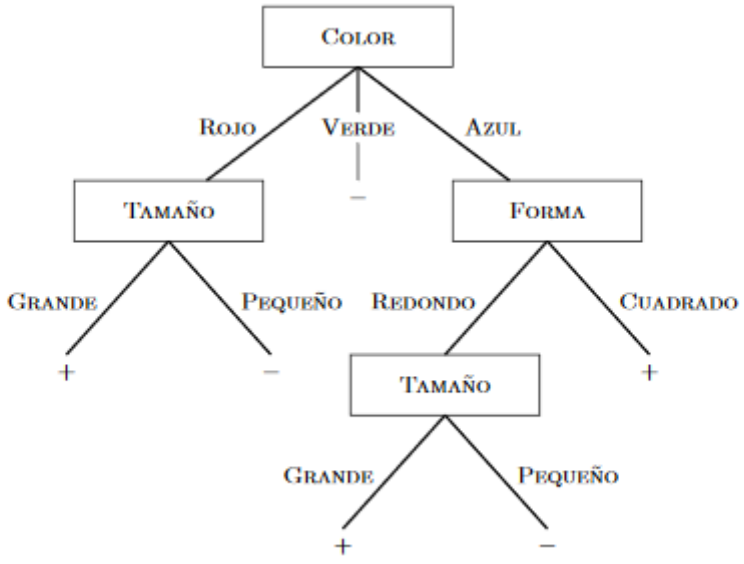

Figura 1: Ejemplo de árbol de decisión

Nuestro algoritmo desarrollado tendrá como resultado un árbol de decisión. Para la construcción de ese árbol, se utilizará el algoritmo ID3 con ciertas modificaciones. ID3 es un algoritmo inventado por Ross Quinlan, utilizado para generar árboles de decisiones dado un conjunto de datos, algo que ya hemos definido anteriormente. Dicho algoritmo es usado de manera habitual en el campo de machine learning o aprendizaje automático y el procesamiento del lenguaje

natural. ID3 está basado en la elección de mejor atributo, que será establecido mediante la entropía [2]. Se elegirá aquel atributo que proporcione una mejor ganancia de información. La fórmula de la entropía es:

$$Ent(D) = -\frac{|P|}{|D|}\log_2\frac{|P|}{|D|} - \frac{|N|}{|D|}\log_2\frac{|N|}{|D|}$$

Donde P y N son, respectivamente, los subconjuntos de ejemplos positivos y negativos de D. La ganancia de información se calculará de la siguiente manera:

$$Ganancia(D, A) = Ent(D) - \sum_{v \in Valores(A)} \frac{|Dv|}{|D|} Ent(Dv)$$

Donde Dv es el subconjunto de ejemplos de D con valor del atributo A igual a v [3].

Nuestro algoritmo contará con un quórum de entrada. Es decir, una cantidad mínima de ejemplos que queramos imponer para considerar que una rama de nuestro árbol de decisión sea fiable. En aquellos casos que tras una ramificación el número de ejemplos restantes sea inferior a nuestro quóroum de entrada, se añadirá al árbol un nodo que contemplará el cálculo realizado por Naive Bayes [11].

En el aprendizaje automático, Naive Bayes es un clasificador simple de la familia de clasificadores probabilísticos. Naive Bayes está basado en la aplicación del teorema de Bayes al que se le añaden ciertas modificaciones de simplificación. En estas modificaciones adicionales se asume que la presencia o ausencia de una característica particular no está relacionada con la ausencia de otra de las características [10]. El cálculo de las distintas probabilidades aplicando Naive Bayes, se explicarán con mayor profundidad una vez expliquemos como se ha desarrollado nuestro algoritmo.

Una vez puesto en contexto, nuestro algoritmo devolverá un árbol de decisión basado en el algoritmo ID3. Cada rama del árbol podrá ser: una rama sin incertidumbre o devolverá un valor de clasificación con una probabilidad asociada en caso de que no se haya alcanzado el quórum de entrada.

II. METODOLOGÍA

A. Representación de un árbol mixto

Un árbol de decisión está formado por nodos y arcos, en nuestro caso para facilitar la implementación el árbol mixto estará compuesto solamente por nodos, de esta forma el árbol consistirá de una lista de nodos.
Los nodos representarán tanto atributos del conjunto de datos, clasificaciones o llamadas al método Naive Bayes, así tendremos los nodos interiores, nodos-hoja-categoría y nodos-hoja-truncada en un mismo tipo de dato.
Los nodos pueden tener varios atributos:

- Nombre: es el nombre del atributo al que representa, en caso de ser nodo-hoja-categoria tiene como nombre el valor del atributo objetivo correspondiente, por último en caso de ser nodo-hoja-truncada en nombre será “Naive Bayes”.
- Hijos: lista de los nodos que están relacionados con este, en caso de no ser nodo interior esta lista está vacía.
- Arista: Valor del atributo al que representa el nodo padre de este que habria que tomar. El nodo raíz no tiene arista.
- Padre: Nodo superior a él en el árbol.

El atributo padre no es necesario para la creación del árbol o clasificación de nuevos ejemplos, pero ha sido añadido para facilitar la representación gráfica del árbol.

Además de los nodos el árbol mixto contiene otros atributos los cuales son necesarios para la creación del árbol, clasificación de nuevos ejemplos o medición del rendimiento, estos son:

- Atributos: lista de los atributos del conjunto de datos sin el atributo objetivo.
- Datos de entrenamiento: representados como un dataframe de la librería Pandas [6].
- Datos de evaluación: representados como un dataframe de la librería Pandas [6].
- Clasificaciones: lista con las clasificaciones para los datos de evaluación, en caso de haberse realizado llamada Naive Bayes además de la clasificación también tiene la probabilidad obtenida.

B. Datos

Como se puede observar en el punto anterior para la representación de los datos de entrenamiento y evaluación se ha usado la librería Pandas, concretamente el tipo de dato Dataframe. Esta representación de los datos nos permite manejarlos con facilidad ya que trae consigo varios métodos implementados para filtrar los cuáles son simples de usar.
Para el correcto funcionamiento del programa los conjuntos de datos deberán estar en formato .csv, en cuanto a los atributos deberán estar colocados de forma que la última columna contenga el atributo objetivo. Además se le deberá indicar al programa si los datos contiene en la primera línea del archivo los nombres de los atributos, en caso de tenerlos se usarán como nombres de los nodos interiores, en caso de no tenerlos se usarán índices, por lo que es recomendable añadir los nombres si se quiere visualizar el árbol.

C. Algoritmo ID3 modificado

A la hora de crear el árbol mixto la mayor parte es muy similar al funcionamiento del algoritmo ID3. Para que la creación del árbol se ha implementado un método para que el usuario solo tenga que indicar si el archivo con los datos tiene en la primera línea los nombres de los atributos, la ruta del archivo y el valor del quórum. Este método inicializa los atributos necesarios del árbol y realiza la llamada inicial al método que realmente ejecuta el algoritmo ID3 modificado para construir el árbol.

Procedimiento algoritmo ID3 modificado:
Entrada:

- Datos
- Quórum
- Nodo anterior (nulo en la primera llamada) se usan en la creación de nuevos nodos para asignar el padre
- Valor anterior(nulo en la primera llamada) se usan para asignar la arista en la creación de nuevos nodos
- Atributos

Salida:

- No tiene ya que va creando los nodos y los va añadiendo al atributo nodos del árbol mixto.

Algoritmo:

1. Si longitud(datos) < quorum
   a. Crear nodo para llamada Naive Bayes
2. Si longitud(valores de ultima columna) == 1
   a. Crear nodo con el valor que contenga la ultima columna
3. Si longitud(atributos) <= 0
   a. Crear nodo con el valor más común de la última columna en datos
4. En otro caso

a. Obtener mejor atributo
b. Crear nodo con mejor atributo
c. nodo = nodo creado en el paso anterior
d. Para valor en los valores del mejor atributo
e. nuevos datos = Obtener datos con el valor del mejor atributo
f. Copiar atributos
g. Eliminar el mejor atributo de la copia de atributos
h. Si longitud(nuevos datos) == 0 y quorum == 0
i, Crear nodo con el valor más común en datos
j. En otro caso
k. Hacer llamada recursiva con datos = nuevos datos, quórum, nodo,valor, copia atributos

D. Clasificación de nuevos datos

Para la clasificación de datos al igual que en la creación del árbol se ha creado un método que recibe la información sobre los datos (si los atributos están especificados en la primera línea del archivo y la ruta del archivo) y es este el que realiza la llamada al método que obtiene la clasificación del nuevo ejemplo. El primer método obtiene los datos de la ruta indicada, inicializa las variables correspondientes y para cada fila en los datos llama al método que obtiene la clasiciación.

Procedimiento para obtener la clasificación de una fila:
Entrada:

- Fila a clasificar
- Nodo actual, en la llamada que realiza el método mencionado anteriormente es el nodo raíz.

Salida:

- Clasificación de la fila, en caso de que se realice la llamada al método Naive Bayes devuelve una lista con la clasificación en la primera posición y la probabilidad en la segunda(Posiciones 0 y 1 en la lista).

Algoritmo:

1. Si el nodo es nulo
    a. Devolver nulo
2. Si el nombre del nodo == “Naive Bayes”
    a. Realizar llamada al método Naive Bayes y devolver lo que devuelve este
3. Si longitud(hijos del nodo) == 0
    a. Devolver el nombre del nodo
4. En otro caso
    a. Encontrar el hijo del nodo actual que tiene como arista el valor de la columna que tiene el nombre igual al nombre del nodo actual en la fila.
    b. Realizar llamada recursiva con el nodo encontrado en el paso anterior

Aclaración punto 4a:
Si se alcanza el punto 4 quiere decir que el nodo acutal es un nodo interior, es decir, tenemos un atributo y necesitamos saber el valor de este para poder alcanzar el siguiente nodo. Para obtener correctamente el siguiente nodo tenemos que mirar en la fila el valor de dicho atributo y encontrar dentro de los hijos del nodo actual el que tiene como arista dicho valor.

Las clasificaciones de las filas se van añadiendo a una lista que es la que se devuelve finalmente con las clasificaciones de todos los datos del conjunto de entrenamiento.

E. Naive Bayes

Para las llamadas Naive Bayes en lugar de usar alguna librería para calcular la clasificación se ha decidido realizar un método que haga los cálculos. El método calcula la probabilidad con suavizado de Laplace para evitar las probabilidades nulas. Entonces la formula usada para obtener la clasificación sería:

$$\text{argmax}_{vj \in V} P(vj) \prod_i P(ai|vj)$$

Y las probabilidades serían:

$$P(vj) = \frac{\#(V = vj)}{N}$$

$$P(ai|vj) = \frac{\#(Ai = ai, V = vj) + k}{\#(V = vj) + k|Ai|}$$

Donde N es el número total de ejemplos, #(V = vj) es el número de ejemplos clasificados como vj y #(Ai = ai, V = vj) es el número de ejemplos clasificados como vj cuyo valor en el atributo Ai es ai.
Hemos escogido k = 1.

F. Rendimiento

Como medida de rendimiento hemos escogido la tasa de aciertos dado que es una medida simple de realizar e indica si el clasificador está funcionando correctamente o no. En nuestro caso para calcular el rendimiento basta con obtener las lista de evaluaciones del árbol que ha sido creado y la última columna de los datos de entrenamiento e ir comparando uno a uno. En caso de que el valor de la evaluación sea una lista siempre tomamos el valor en la posición cero ya que esa es la clasificación.

III. RESULTADOS

Se han realizado diversos experimentos con varios conjuntos de datos y distintas medidas de quórum, dichos conjuntos de datos han sido divididos en conjuntos de entrnamiento y conjuntos de evaluación.
Para los valores del quórum se han usado valores desde 1 hasta el mayor numero possible de forma que el árbol mixto generado contenga mayormente nodos que realicen Naïve Bayes, es por ello que el mayor valor del quórum no se ha decidido de forma arbitraria para los distintos conjuntos de datos sino que dependerá del tamaño del mismo. Con esto se espera observar los cambios en el rendimiento de los árboles generados conforme el quórum va aumentando de valor.

A. Experimento 1

El primer experimento ha sido realizado con un conjunto de datos extraído del boletín de ejercicios de la asignatura [9].
Nombre del conjunto de entrenamiento: ejemplo2.csv
Nombre del conjunto de evaluación: evaluación2.csv
Número de entradas en el conjunto de entrenamiento: 15
Número de entradas en el conjunto de evaluación: 8
Resultados: Representados en la siguiente gráfica de forma que en el eje Y se encuentra el rendimiento y en el eje X los distintos valores para el quórum.

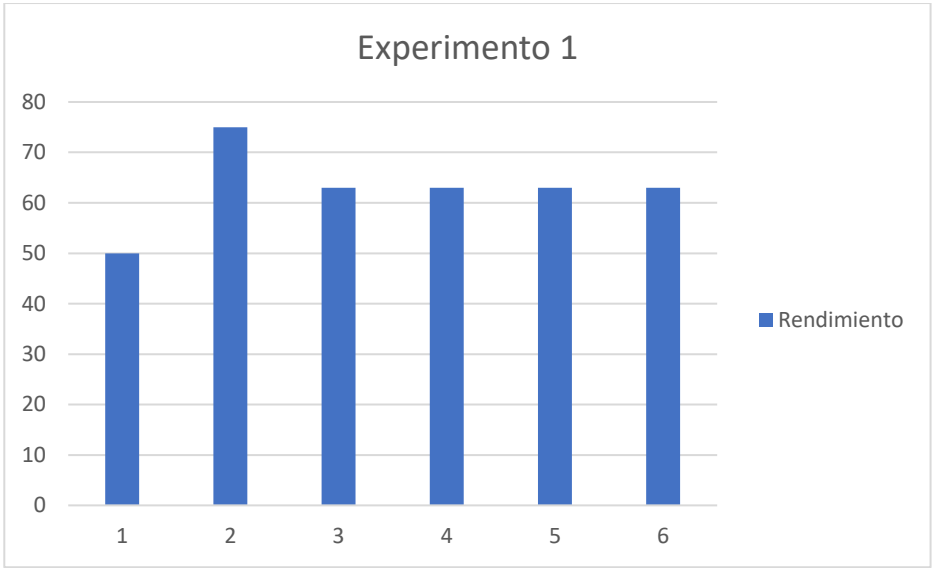

Figura 2: Resultado del experiment número 1

En los resultados obtenidos se puede observar que al imponer un quórum mayor que uno aumenta el rendimiento del árbol mixto generado, alcancanzo 75% de aciertos como mayor rendimiento y estabilizando el rendimiento en torno al 60% de aciertos para valores mayores que 2, debido a que los árboles generados para los valores de quórum 3,4,5,6 son similares.

B. *Experimento 2*

El segundo experimento ha sido realizado con un conjunto de datos mayor obtenido en un repositorio de github [5].
Nombre del conjunto de entrenamiento: car-data-train.csv.
Nombre del conjunto de evaluación: car-data-test.csv.
Número de entradas en el conjunto de entrenamiento: 1296
Número de entradas en el conjunto de evaluación: 432
Resultados: Representados en la siguiente gráfica de forma que en el eje Y se encuentra el rendimiento y en el eje X los distintos valores para el quórum.

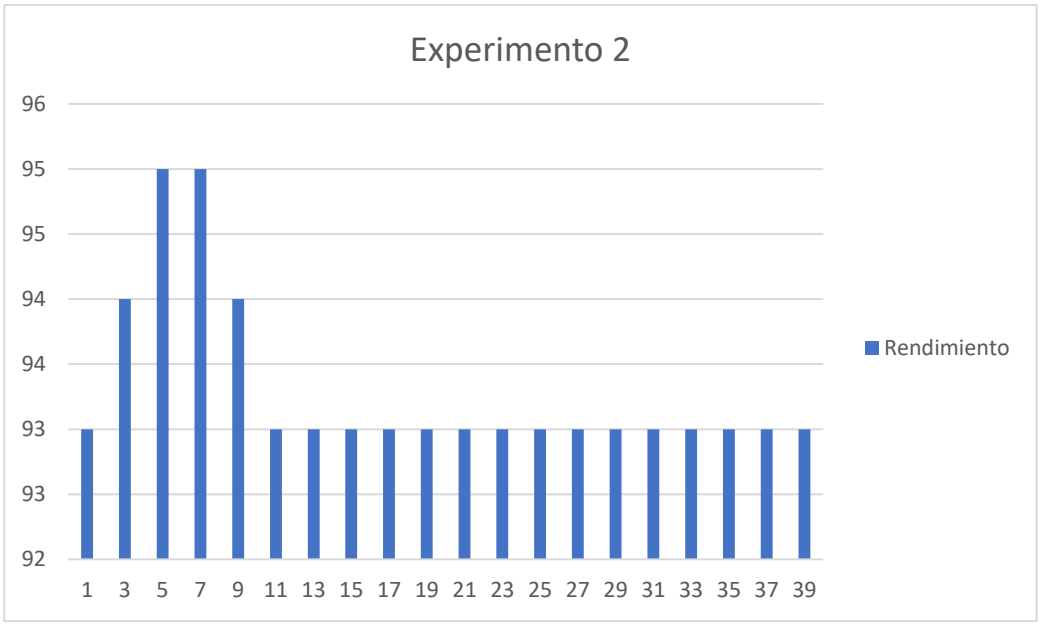

Figura 3: Resultados de experimento número 2

Tras imponer un quórum mayor que uno el rendimiento va aumentando hasta alcanzar su mayor rendimiento con los valores de quórum 5 y 7, tras seguir aumentando el valor del quórum el rendimiento de los árboles generados va disminuyendo hasta alcanzar un rendimiento cercano al árbol generado para el valor de quórum 1.

C. *Experimento 3*

El último experimento ha sido realizado con un conjunto de datos aún mayor que el anterior obtenido de un repositorio de gihub [5].
Nombre del conjunto de entrenamiento: kr-vs-kp-train.csv.
Nombre del conjunto de evaluación: kr-vs-kp-test.csv
Número de entradas en el conjunto de entrenamiento: 2556
Número de entradas en el conjunto de evaluación: 640
Resultados: Representados en la siguiente gráfica de forma que en el eje Y se encuentra el rendimiento y en el eje X los distintos valores para el quórum.

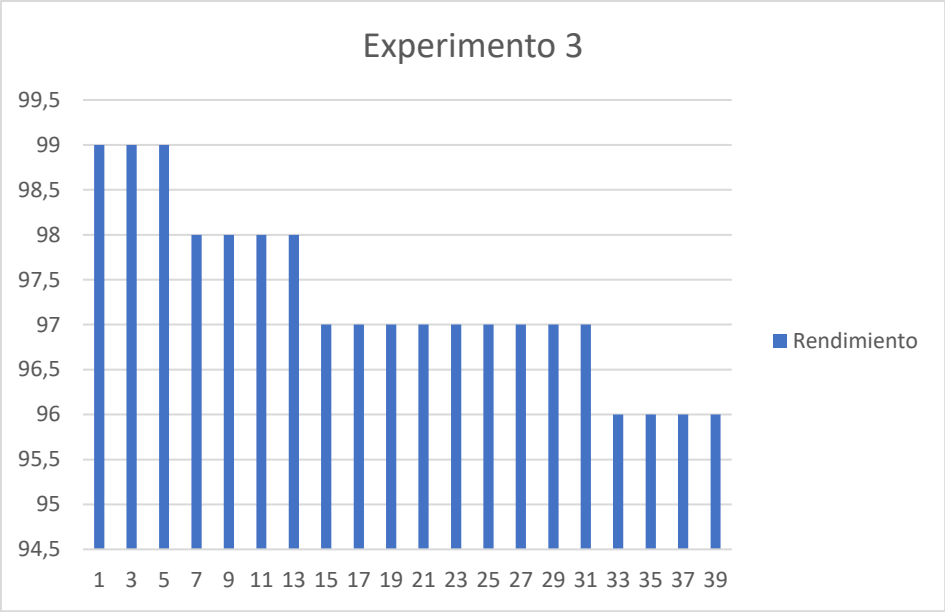

Figura 4: Resultados experiment número 3

El rendimiento de los árboles generados para valores de quórum menor que 5 son similares, conforme se va aumentando el valor del quórum el rendimiento de los árboles decae en los valores 15 y 33 hasta alcanzar finalmente un 96% de aciertos.

IV. CONCLUSIONES

Una vez llevado a cabo los distintos experimentos expuestos anteriormente, observamos de manera general que el algoritmo diseñado o desarrollado mejora el rendimiento en algunos casos. Como se puede observar en el experimento número tres, el rendimiento de nuestro algoritmo y el algoritmo original ID3 es similar, siendo este último algo superior (algunas décimas más). En cuanto a los conjuntos de datos en los que sí mejora el rendimiento, se aprecia que esto ocurre para valores de quórum comprendidos entre uno y siete, como sucede en los experimentos realizados, en caso de seguir aumentando el valor de este por encima de estos valores se aprecia una bajada en el rendimiento. En caso de que el conjunto de datos no sea demasiado grande hay que tener en cuenta que el valor del quórum hay que bajarlo aún más, como se puede apreciar en el experimento uno.

En cuanto posibles futuras mejoras, sería interesante aplicar algoritmos de podas al árbol mixto o aplicar otros algoritmos en lugar de Naives Bayes, como k-NN o k-medias. También podrían tenerse en cuenta la realización de distintas medidas de rendimiento, ya que la implementada no sería aceptable en ciertos problemas de clasificación por ejemplo en problemas relacionados con enfermedades en los cuales los falsos positivos y falsos negativos tienen distinta importancia [12].

REFERENCIAS

[1] Página web de Wikipedia sobre árboles de decisión. https://es.wikipedia.org/Árbol_de_decisión

[2] Página web de Wikipedia sobre el algoritmo ID3. https://es.wikipedia.org/wiki/Algoritmo_ID3

[3] Tema 1 del curso de IA de Ingeniería del Software. https://www.cs.us.es/cursos/iais-2018/temas/Aprendizaje.pdf, pp 6-43
[4] Repositorio de github. https://github.com/arunaugustine/ID3
[5] Repositorio de github .https://github.com/Silversmithe/ID3/tree/master/tests
[6] Documentación de la librería Pandas. https://pandas.pydata.org/pandas-docs/stable/
[7] Documentación de la librería AnyTree. https://anytree.readthedocs.io/en/latest/
[8] Página web de stackoverflow para consultar operaciones con Python: https://stackoverflow.com/
[9] Boletín de ejercicios 1 del curso de IA Ingenierís del Software. https://www.cs.us.es/cursos/iais-2018/ejercicios/R1.pdf página 6 ejercicio 7
[10] Página web de Wikipedia sobre clasificadores del tipo Naive Bayes. https://es.wikipedia.org/wiki/Clasificador_bayesiano_ingenuo
[11] Documentación del trabajo propuesto: Algoritmo ID3 mixto. https://www.cs.us.es/cursos/iais-2018/trabajos/ID3_quorum.pdf
[12] Medidas de rendimiento en problemas de clasificación. https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b