UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
FACULTAD DE MATEMÁTICA, FÍSICA Y COMPUTACIÓN
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

Extensión de algoritmos representativos del aprendizaje automático al trabajo con datos tipo conjunto

Tesis presentada en opción al título de Master en Ciencia de la Computación

Autor: Lic. Mabel González Castellanos

Tutores: Dra. Yanet Rodríguez Sarabia
Dr. Carlos Morell Pérez

Consultante: Dr. Amilkar Puris Cáceres

Santa Clara, 2010

RESUMEN

La forma de representación de los ejemplos en un conjunto de entrenamiento resulta de extrema importancia para el concepto de aprendizaje automatizado. Este constituye uno de los primeros pasos que se sigue en el diseño de un sistema que se supere mediante la experiencia y consiste en la descripción del problema mediante un conjunto de atributos con un determinado nivel de medición.

Existen problemáticas con presencia de rasgos que pueden tomar varios valores de manera simultánea para un mismo ejemplo. Dichos rasgos se pueden representar de forma natural mediante conjuntos. Considerar conjuntos como tipo de dato es una problemática poco abordada en el contexto del aprendizaje automatizado. Las herramientas de aprendizaje automatizado disponibles no brindan facilidades para el tratamiento de rasgos con las características anteriormente descritas.

Considerando lo anterior esta investigación aborda el tratamiento de atributos de tipo conjunto en el contexto de tres enfoques fundamentales del aprendizaje automatizado: árboles de decisión, enfoque basado en instancias y el probabilístico. Este trabajo incluye la presentación de nuevas propuestas basadas en algoritmos de clasificación clásicos, que explotan los beneficios de utilizar conjuntos como tipo de dato. El análisis experimental demuestra la validez de las propuestas evaluadas.

ABSTRACT

The form of representation of the examples in a training set is of utmost importance to the concept of machine learning. This is one of the first steps to be followed in the design of a system that is exceeded by the experience and is the description of the problem through a set of attributes with a certain level of measurement.

There are problems with the presence of traits that can take several values simultaneously for the same example. These features can be represented naturally by sets. Consider sets as a data type is a little problem addressed in the context of machine learning. Machine learning tools available do not provide facilities for the treatment of features with the characteristics described above.

Considering the above, this research addresses the treatment of type attributes set in the context of three fundamental approaches to machine learning: focus lazy approach and probabilistic instances. This work includes the submission of new proposals based on traditional classification algorithms that exploit the benefits of using such type of data sets. The experimental analysis demonstrates the validity of the proposals evaluated.

CONTENIDO

FIGURAS

TABLAS

Introducción

Una de las partes fundamentales del diseño de un sistema de aprendizaje consiste en la experiencia suministrada al sistema, de la cual se va a aprender. En esta etapa del diseño resulta extremadamente importante la forma de representar los ejemplos en el conjunto de entrenamiento. Una adecuada representación comprende realizar una modelación del problema lo más cercana posible a la naturaleza del mismo [1].

Existen disímiles problemáticas donde aparecen rasgos que toman diferentes valores de forma simultánea, los cuales pudieran representarse de forma natural mediante conjuntos. La mayoría de las herramientas de aprendizaje automatizado no ofrecen una forma adecuada de tratamiento a datos con estas características, limitando los valores a tomar por los atributos a valores reales o nominales fundamentalmente.

Esto ha provocado que se tomen alternativas, como la descomposición de los conjuntos en n-uplas binarias, con el fin de poder describir el problema mediante los rasgos convencionales disponibles en las herramientas de aprendizaje automatizado. Esta solución provoca un aumento de la cantidad de atributos del problema proporcional a la cantidad de elementos que conforman el dominio del atributo. A lo anterior se añade un tratamiento que no tiene en cuenta características propias de los conjuntos.

Un enfoque más actual a este problema incluye la extensión de algoritmos de manera que permitan el tratamiento de atributos de tipo conjunto. En este sentido se encontraron varias propuestas en la literatura [2, 3] que tienen la desventaja de realizar operaciones sobre los elementos individuales que componen los conjuntos, en lugar de tratarlos como un todo.

Objetivo general

El objetivo de este trabajo consiste en extender algoritmos representativos del aprendizaje automatizado de modo que se garantice un adecuado tratamiento

de los atributos tipo conjunto, propiciando mayor naturalidad en la modelación computacional del problema.

Objetivos específicos

Para lograr este objetivo general se plantean los siguientes objetivos:

1. Extender algoritmos de clasificación pertenecientes a los siguientes enfoques:
    - árboles de decisión (ID3)
    - basado en instancias (k-NN)
    - probabilístico (Naive Bayes)
2. Incluir en el ambiente de aprendizaje automatizado Weka las facilidades necesarias para el trabajo con datos tipo conjunto:
    - Formato de entrada (*.arff*)
    - Filtros para las conversiones (de binario a conjunto y de conjunto a binario)
    - Algoritmos de clasificación
3. La evaluación de los algoritmos de clasificación extendidos en el ambiente de aprendizaje Weka utilizando varios conjuntos de datos.

Pregunta de investigación

¿Representar mediante conjuntos los atributos que de forma natural lo demanden, unido a un correcto tratamiento en el contexto de cada algoritmo, implica mejoras en la modelación computacional del problema?

Justificación de la investigación y su viabilidad

Representar y manejar conjuntos como un tipo de atributo resulta extremadamente útil ya que hace posible una representación más natural de los datos que así lo requieran. Además, resulta significativo conocer si a lo anterior se añade una mejora del desempeño de los algoritmos de clasificación.

Esta investigación es viable ya que se cuenta con los recursos necesarios para acceder a la información sobre el tema y es interés del centro que esta investigación se concluya de forma satisfactoria.

Tipo de investigación

Esta investigación abarca fines exploratorios, descriptivos y correlacionales. Es exploratoria en sus inicios ya que abarca un tema poco estudiado y que se ha tratado sin llegar a ser el tema central de investigaciones anteriores. Es de tipo descriptivo ya que en la literatura aparecen algunas variantes del tratamiento de conjuntos en el contexto de algunos algoritmos de aprendizaje automatizado. También es de tipo correlacional ya que se pretende medir el grado de relación entre la correcta representación y tratamiento de los atributos de tipo conjunto con el desempeño de los algoritmos de clasificación.

Hipótesis de investigación

El tratamiento mediante el tipo de dato conjunto de los atributos que de forma natural lo demanden, unido a un correcto tratamiento en el contexto de cada algoritmo, permite que el modelo computacional obtenido muestre un desempeño similar, mejore su interpretabilidad y reduzca la dimensionalidad de los conjuntos de los datos.

Capítulo 1: Los conjuntos como tipo de dato en el aprendizaje automatizado

En este capítulo se presentarán las bases de la teoría clásica de conjuntos. Se mostrará la relación que tiene el concepto de conjunto con rasgos de problemáticas actuales abordadas a través del aprendizaje automatizado. Posteriormente se hará énfasis en el tratamiento que se le ha dado a los atributos de tipo conjunto en este campo, considerando dos enfoques fundamentales: la descomposición de los conjuntos en atributos básicos y el tratamiento de los mismos como un tipo de atributo. En este último se hará énfasis en la forma en que se han manejado los conjuntos en el contexto de varios algoritmos pertenecientes a diversos enfoques. Finalmente se presentarán las características básicas de la herramienta de aprendizaje automatizado Weka.

1.1 Teoría básica de conjuntos

G. Cantor creó una nueva disciplina matemática entre 1874 y 1897: la teoría de conjuntos. Desde entonces disímiles debates han acontecido en el seno de la teoría de conjuntos, sin duda por hallarse estrechamente conectados con importantes cuestiones lógicas. Algunos años más tarde la teoría axiomática de Zermelo (1908) y refinamientos de ésta debidos a Fraenkel (1922), Skolem (1923), von Newman (1925) y otros, sentaron las bases para la teoría de conjuntos actual.

1.1.1 Concepto de conjunto

La definición inicial de Cantor [4] es totalmente intuitiva: un conjunto es cualquier colección C de objetos determinados y bien distintos x de nuestra percepción o nuestro pensamiento (que se denominan elementos de C), reunidos en un todo.

Dado un elemento a y un conjunto X utilizamos la expresión: $a \in X$, para declarar que a es un elemento o miembro de X. Para expresar lo contrario escribimos: $a \notin X$.

A partir de dos conjuntos X y Y decimos que X es un subconjunto de Y si y solo si cada elemento de X es un elemento de Y, esto se denota como: $X \subset Y$.

1.1.2 Operaciones sobre conjuntos

Existe un número de operaciones simples [4] que pueden ser aplicadas a conjuntos, formando nuevos conjuntos a partir de conjuntos dados. A continuación se consideran las más comunes:

La unión de dos conjuntos X y Y da como resultado un conjunto consistente en los miembros de X junto con los miembros de Y, esta operación se denota por $X \bigcup Y$.

La intersección de dos conjuntos X y Y da como resultado un conjunto formado por aquellos objetos que son miembros de ambos conjuntos y se denota por $X \bigcap Y$.

La diferencia de dos conjuntos X y Y es el conjunto formado por aquellos elementos de X que no son miembros de Y, esta operación se denota por $X - Y$.

La diferencia simétrica de dos conjuntos X y Y es el conjunto formado por la diferencia $X - Y$ en unión con la diferencia $Y - X$ y se denota mediante $X \Delta Y$.

En teoría de conjuntos es conveniente definir una colección sin objetos como un conjunto, el conjunto vacío o nulo. Este conjunto es usualmente denotado mediante el símbolo ϕ derivado de una letra escandinava.

Se dice que dos conjuntos son disjuntos si no tienen ningún miembro en común o lo que es lo mismo: $X \bigcap Y = \phi$.

1.1.3 Notaciones de conjuntos

Si un conjunto es finito podemos describirlo mediante la enumeración de sus miembros: si X consiste en los objetos $a_1, a_2, \ldots, a_n$, podemos denotar a X mediante $\{a_1, a_2, \ldots, a_n\}$. Esta definición es conocida como definición por extensión [4].

Cuando el número de objetos es infinito o demasiado numeroso se utiliza el método de definición por intensión [4], que consiste en la descripción de un conjunto mediante una o varias propiedades que caracterizan a los elementos de ese conjunto. En este trabajo solo abarcaremos conjuntos finitos, descritos a partir del listado de cada uno de sus elementos.

1.2 El aprendizaje automatizado y el tratamiento de atributos de tipo conjunto

El almacenamiento de datos tiende a incrementarse en el mundo entero. Las computadoras han facilitado esta tarea, haciendo posible el acopio de datos que de otra forma se hubieran despreciado. Oculta en estos datos existe información potencialmente útil, que resulta difícil aprovechar.

La minería de datos [5] se define como el proceso de descubrimiento de patrones a partir de datos almacenados previamente. Este proceso debe ser automático o semiautomático y los patrones descubiertos generalmente permiten obtener algún tipo de ventaja [6].

El aprendizaje automatizado abarca una gran variedad de técnicas para buscar y descubrir patrones estructurales a partir de datos. En otros términos según [7] podemos concluir que aprendizaje automatizado incluye a cualquier programa de computadora que mejore su desempeño en alguna tarea, mediante la experiencia.

El tipo de experiencia suministrada al sistema para el aprendizaje tiene un significativo impacto en el futuro desempeño del mismo. La experiencia puede ser proporcionada a través de ejemplos directos pertenecientes al dominio del problema que se desea resolver [8]. Un conjunto de instancias constituye la entrada de un algoritmo de aprendizaje automatizado y cada instancia individual constituye un ejemplo de un concepto que debe ser aprendido. La forma de representar los ejemplos en el conjunto de entrenamiento es de extrema importancia para el diseño del sistema de aprendizaje automatizado y es de las primeras tareas que se debe encarar en su construcción [7].

Las instancias están caracterizadas mediante un conjunto de rasgos o atributos. El valor de un atributo numérico, para una instancia en particular, es una medición de la cantidad a la cual el atributo se refiere. En caso de que el atributo tome valores en un conjunto finito, especificado con anterioridad, se define de tipo nominal. Estos dos tipos de atributos son los considerados tradicionalmente por los algoritmos de aprendizaje automatizado y cada rasgo del problema se define a partir de alguno de estos tipos [5].

Los dos tipos de atributos considerados con anterioridad tienen la característica de que cada instancia, para un atributo en específico, toma un único valor. Este esquema no es aplicable en situaciones donde un rasgo puede tomar más de un valor simultáneamente. Datos que cumplan las características anteriormente descritas aparecen de forma natural en disímiles problemas [9, 10].

En el campo de la medicina por ejemplo, para llegar a un diagnóstico generalmente se parte de conjuntos de síntomas y signos que presenta el paciente los cuales pueden aparecer de forma simultánea en una misma persona [11-13]. Usualmente en problemas de diagnóstico en general son comunes la existencia de múltiples síntomas y la ocurrencia de un subconjunto de estos en los casos a diagnosticar. De igual forma en el procesamiento de texto [14] es muy usual asociar un documento con un conjunto de palabras claves las cuales no tienen un número fijo y pueden pertenecer a un dominio extremadamente amplio. Todos los ejemplos anteriores tienen en común que este tipo de rasgo, puede expresarse de forma natural mediante un conjunto.

Por otra parte, ya que no existe un algoritmo apropiado para todos los tipos de problemas [15-17], se hace necesario obtener por medio de la experimentación, un modelo que se adecue a la estructura del dominio que se esté tratando. Esta tarea se facilita con el uso de herramientas de aprendizaje automatizado. Dichas herramientas permiten de una forma rápida probar métodos existentes del aprendizaje automatizado a nuevos conjuntos de datos.

En la mayoría de las herramientas de este tipo los ejemplos o instancias de entrenamiento se representan como un conjunto de rasgos de longitud fija. Los

atributos son restringidos a solo dos niveles de medición: numéricos o nominales [5]. Luego de analizar los tipos de datos soportados por varias herramientas de este tipo [5, 18-20] no se encontró ninguna que permitiera definir atributos de tipo conjunto.

1.2.1 Tratamiento de los conjuntos mediante su descomposición

La solución más común que se sigue ante la presencia de datos de tipo conjunto se reduce a crear una n-upla binaria formada por los n posibles valores del dominio del atributo. De esta forma se crean por cada atributo de tipo conjunto, tantos atributos binarios como elementos tenga el dominio del atributo. Para cada instancia, la presencia de cero o uno en una componente de la n-upla, representa la pertenencia o no de dicho valor al conjunto.

Para una mejor comprensión veamos el siguiente ejemplo tomado de una aplicación para el filtrado automatizado del correo electrónico [21]. Cada correo está caracterizado a partir de tres conjuntos: *Desde*, *Asunto* y *Cuerpo*. Cada conjunto está formado por las palabras claves presentes en dichos campos. El esquema de la Figura 1.1 muestra una instancia del problema y su respectiva descomposición en una n-upla de atributos binarios. Esta forma de lidiar con el problema genera instancias con un gran número de atributos, en ocasiones inmanejable como en el caso del procesamiento de texto. Este aumento de la dimensionalidad provoca efectos indeseables que afecta el desempeño de los algoritmos [22-24].

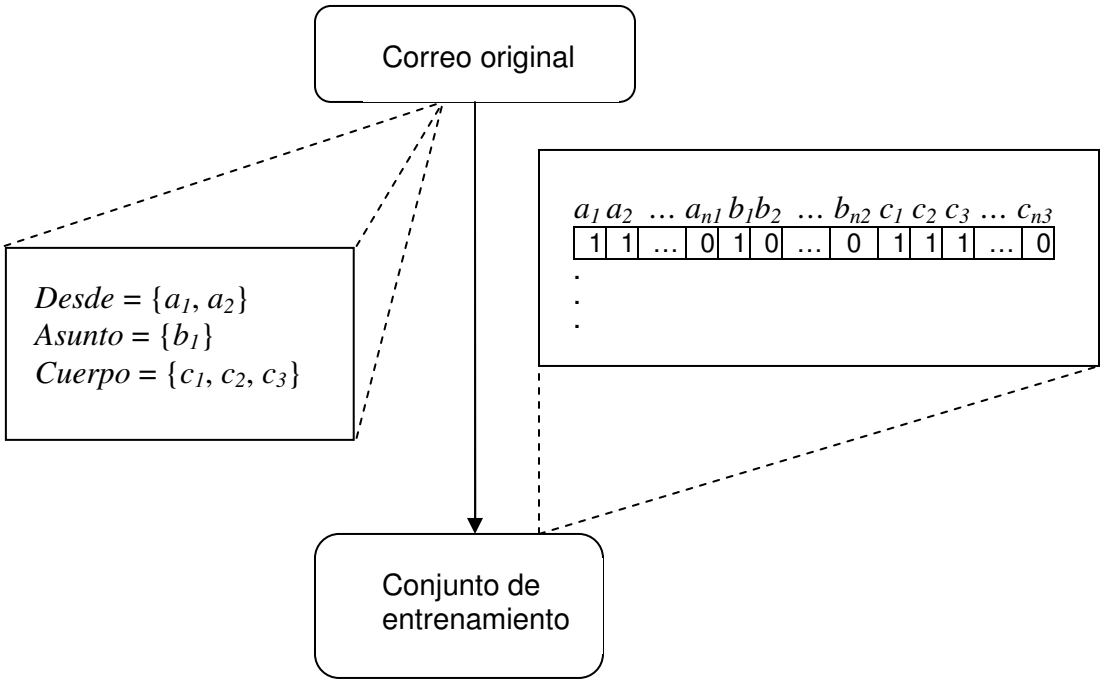

Figura 1 Ejemplo de una instancia con atributos de tipo conjunto y su descomposición en atributos binarios.

Otra alternativa consiste en la creación de un atributo nominal por cada atributo de tipo conjunto [21]. Para el caso en que el valor que toma el atributo sea un conjunto de más de un elemento se repite cada instancia tantas veces como combinaciones diferentes existan de los valores que toman los atributos de tipo conjunto para dicha instancia. La Figura 1.2 muestra el resultado de aplicar esta variante en el ejemplo anterior.

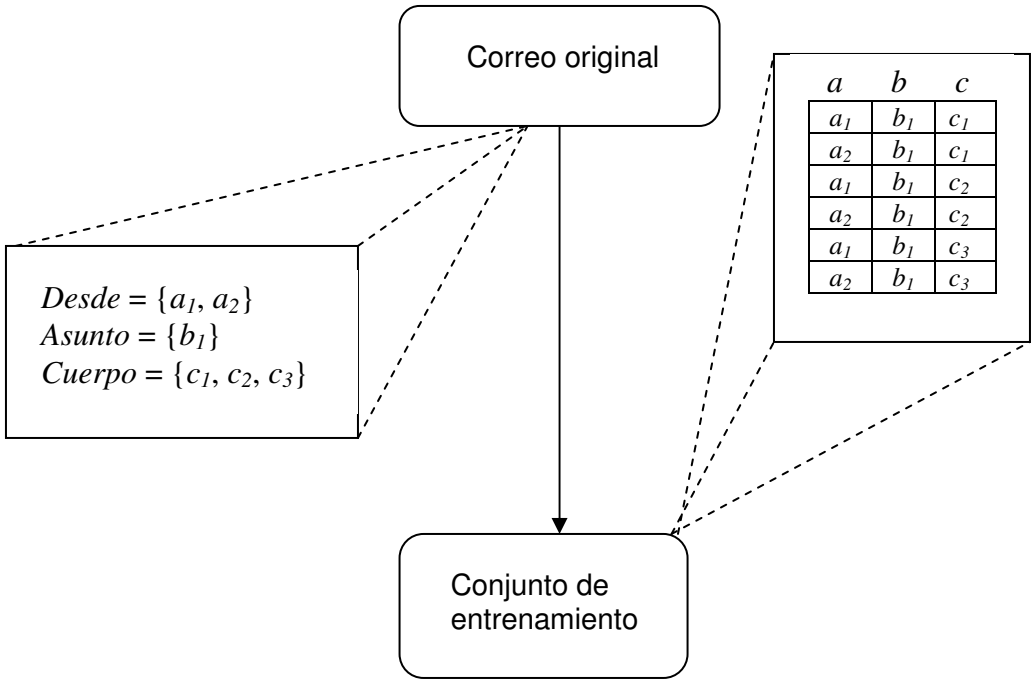

Figura 2 Ejemplo de una instancia con atributos de tipo conjunto y su descomposición en atributos nominales.

Este enfoque provoca una elevación considerable de la cantidad de instancias ya que por cada ejemplo se van a crear un número de ejemplos igual al producto del cardinal de cada conjunto presente en dicha instancia. Es por esto que en el ejemplo se obtienen seis nuevas instancias $(2*1*3)$. Este incremento del tamaño del conjunto de entrenamiento también afecta y en ocasiones imposibilita el desempeño de cualquier algoritmo. Vale destacar que el resto de los valores presentes en las instancias que no formen parte de los atributos de tipo conjunto, así como los valores de conjuntos de menor tamaño, estarían más representados en el conjunto de entrenamiento debido a la repetición de sus valores. Este es el caso del valor del atributo *Asunto* del ejemplo, el cual presenta un conjunto de cardinal uno cuyo elemento está presente en todas las instancias generadas.

Las dos soluciones anteriores tienen la desventaja de no ofrecer una representación natural a los atributos de tipo conjunto. Añádase a esto la posible pérdida o distorsión de la información provocada por la descomposición de los conjuntos originales que caracterizaban el atributo.

1.2.2 Tratamiento de los conjuntos como tipo de dato

Un enfoque más actual plantea el tratamiento de los conjuntos como un tipo de dato más, donde el reto consiste en la forma de manejarlos en el contexto de cada algoritmo. En la literatura se reportan varios resultados de este tipo aplicados a algoritmos pertenecientes a diversos enfoques. A continuación se analizarán las propuestas encontradas.

1.2.3.1 Algoritmo ID3

Un árbol de decisión [25] es un modelo de predicción perteneciente al enfoque "divide y vencerás" utilizado en el ámbito de la inteligencia artificial. Dado un conjunto de instancias se construyen diagramas de construcciones lógicas, muy similares a los sistemas de predicción basados en reglas, que sirven para representar y categorizar una serie de condiciones que ocurren de forma sucesiva, para la resolución de un problema.

La entrada del árbol consiste en una instancia descrita por medio de un conjunto de atributos y a partir de esta se devuelve una clasificación, la cual es una decisión que es tomada a partir de las entradas. Un árbol de decisión lleva a cabo un conjunto de pruebas a medida que este se recorre hacia las hojas para alcanzar así una decisión. Cada nodo interno contiene una prueba sobre algún valor de uno de los atributos. Los nodos hoja representan el valor que devolverá el árbol de decisión y finalmente las ramas brindan los posibles caminos que se tienen de acuerdo a la decisión tomada.

La Figura 3 muestra un ejemplo de un árbol de decisión, donde cada instancia a ser clasificada es enrutada a través del árbol hasta un nodo hoja apropiado, retornándose la clasificación asociada a dicha hoja. Este árbol de decisión clasifica las mañanas de sábado acorde a si son adecuadas o no para jugar tenis.

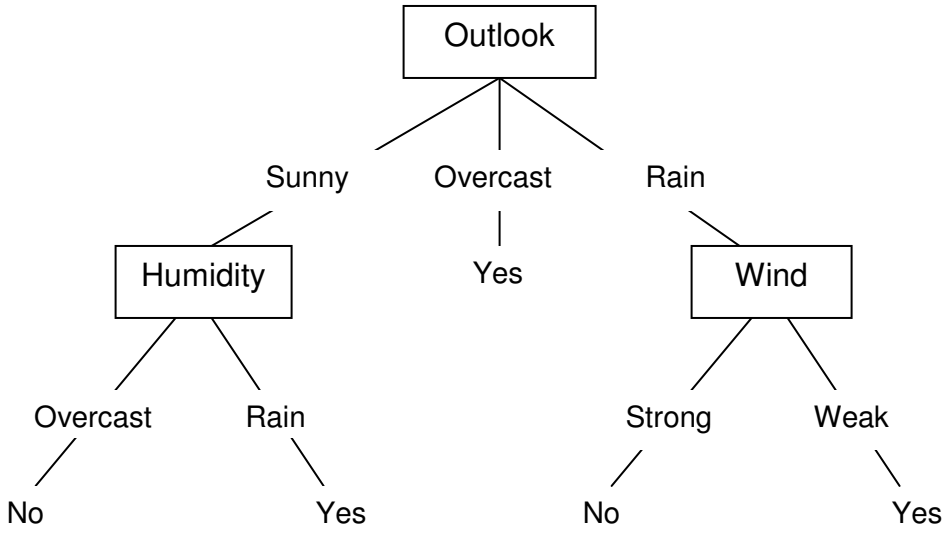

Figura 3 Un árbol de decisión para el concepto *PlayTennis.*

La construcción de un árbol de decisión puede expresarse de forma recursiva. Primeramente se selecciona el atributo a colocar en el nodo raíz y se crea una rama para cada posible valor del atributo. Posteriormente se repite este proceso recursivamente por cada rama utilizando solamente aquellas instancias que alcanzaron dicha rama. Si en un determinado momento se obtiene una rama donde todas las instancias tienen la misma clasificación se termina la generación de esta parte del árbol. Idealmente este proceso termina cuando todos los nodos hojas alcanzan el estado anteriormente descrito, también conocido como nodos puros. Lamentablemente esto no siempre es posible ya que dos instancias pueden tener iguales valores pero diferente clasificación y nunca podrán ser separadas mediante este método.

El algoritmo ID3 [26] es uno de los más conocidos en la construcción de árboles de decisión y fue desarrollado por J. Ross Quinlan de la Universidad de Sydney, Australia. Este modelo no admite atributos numéricos aunque mejoras realizadas posteriormente ya permiten su manejo. Este método se basa fundamentalmente en el uso del criterio de ganancia de información para la selección del atributo utilizado para realizar la bifurcación. El objetivo consiste en seleccionar en todo momento el atributo que maximice la ganancia de información de la bifurcación realizada. Este proceso equivale a una búsqueda ávida para un árbol de

decisión aceptable, en el cual el algoritmo nunca regresa atrás para reconsiderar decisiones ya tomadas con anterioridad.

La ganancia de información mide cuán bien un atributo dado separa los ejemplos de entrenamiento acorde al atributo clase. Esta medida se estima a partir del cálculo de la entropía [27]. Esta última caracteriza la impureza de una colección arbitraria de instancias. Dado un atributo objetivo, que puede tomar c valores diferentes, con frecuencia de aparición p_i (Fórmula 1.1), la entropía se calcula a partir de la Fórmula 1.2. Esta se calcula en base a la cantidad de instancias de cada clase que alcanzan un nodo determinado y tiene como significado el monto de información necesario para clasificar una instancia que alcance dicho nodo. Si todos los miembros del conjunto S pertenecen a una misma clase el valor de la entropía será cero. En otro caso tomará un valor mayor que cero pudiendo alcanzar como valor máximo $\log_2 c$.

$$p_i = \frac{|S_i|}{|S|} \tag{1.1}$$

$$Entropy(S) \equiv \sum_{i=1}^{c} - p_i \log_2 p_i \tag{1.2}$$

$$Gain(S, A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{1.3}$$

La ganancia de información $Gain(S,A)$ de un atributo A, relativa a una colección de ejemplos S se calcula a partir de la Fórmula 1.3. Esta medida es simplemente la reducción esperada de la entropía causada por la partición de los ejemplos acorde a un atributo determinado. $Values(A)$ es el conjunto de todos los posibles valores del atributo A, y S_v es el subconjunto de S para el cual el atributo A tiene valor v. El primer término de esta ecuación es justamente la entropía presente en la colección original S y el segundo término es el valor esperado de entropía después que S es particionado usando el atributo A. La entropía descrita mediante el segundo término está compuesta por la suma de las entropías de

cada subconjunto S_v, ponderadas por la fracción de los ejemplos $\frac{|S_v|}{|S|}$ que pertenecen a S_v.

Cohen plantea en [2] una extensión al ID3 que permite tratar además de los rasgos nominales tradicionales, conjuntos de cadenas. De esta forma se definen rasgos de tipo conjunto para expresar conceptos que se representan de forma natural mediante conjuntos.

En esta propuesta se definen primeramente pruebas primitivas para el caso de los atributos de tipo conjunto de la forma "$x_{ij} \in X_i$" y "$x_{ij} \notin X_i$", donde x_{ij} corresponde al elemento j-ésimo perteneciente al dominio del atributo i-ésimo (U_i) y X_i es el conjunto que toma como valor una instancia para dicho atributo. Esta prueba permite interrogar a un conjunto perteneciente a una instancia acerca de la pertenencia del elemento x_{ij} a dicho conjunto. Esta interrogante tiene lugar cuando la instancia alcanza un nodo etiquetado de esta forma y tiene solo dos posibles respuestas: verdadero o falso. El algoritmo ID3 es extendido de forma que permite etiquetar nodos internos con pruebas de la forma anteriormente descrita para el caso de los rasgos de tipo conjunto.

A partir de este punto basta con encontrar dado un conjunto de instancias y un atributo i de tipo conjunto, la prueba "$x_{ij} \in X_i$" o "$x_{ij} \notin X_i$" que maximice la ganancia de información. Con este objetivo se define una función $MaxGainTest$ que se le pasa como parámetro un conjunto de ejemplos y el índice de un atributo de tipo conjunto, devolviendo la mejor prueba encontrada. La Figura 4 muestra un pseudo código de esta función en su totalidad.

```
function MaxGainTest(S, i)
1       Visited := ϕ ;
2       TotalCount[+] := 0;       TotalCount[-] := 0;
3       for each example ⟨x⃗, y⟩ in the sample S do
4               for each element x_ij ∈ X_i do
5                       Visited := Visited ⋃ { x_ij };
6                       ElementCount[ x_ij , y] := ElementCount[ x_ij , y] + 1;
6               endfor
7               TotalCount[y] := TotalCount[y] + 1;
8       endfor
9       BestEntropy := -1;
10      For each x_ij ∈ Visited do
11              p := ElementCount[ x_ij ,+];
12              n := ElementCount[ x_ij , -];
13              if (Entropy(p, n) > BestEntropy) then
14                      BestTest := " x_ij ∈ X_i";
15                      BestEntropy := Entropy(p, n);
16              endif
17              p' := TotalCount[+] - ElementCount[ x_ij ,+];
18              n' := TotalCount[-] - ElementCount[ x_ij , - ];
19              if (Entropy(p', n') > BestEntropy) then
20                      BestTest := " x_ij ∉ X_i";
21                      BestEntropy := Entropy(p', n');
22              endif
23              ElementCount[ x_ij , +] := 0;
24              ElementCount[ x_ij , -] := 0;
25      endfor
26      return BestTest
```

Figura 4 Pseudo código de la función *MaxGainTest*.

Las líneas 1-9 iteran sobre el conjunto de ejemplos y guardan para cada cadena encontrada la distribución por clase en la variable *ElementCount*. Adicionalmente la variable *Visited* guarda un conjunto con todas las distintas cadenas presentes en los ejemplos. Las líneas 10-25 hacen uso de estas cantidades para encontrar la mejor prueba. Dado un elemento x_{ij}, el número de ejemplos de la clase y cubiertos por la prueba "$x_{ij} \in X_i$" está dado por $ElemCount[x_{ij},y]$.

El autor aplica el nuevo algoritmo a dos tipos de problemas: representaciones proposicionales de primer orden y categorización de texto. En los resultados experimentales presentados este algoritmo obtiene resultados iguales o mejores con respecto al error cometido en la clasificación al compararse con otros algoritmos. En todos los casos se obtienen mejoras sustanciales del tiempo empleado por el clasificador en el aprendizaje ya que el uso de atributos de tipo conjunto reduce drásticamente la memoria utilizada para almacenar los conjuntos de entrenamiento.

El algoritmo propuesto por Cohen presenta dos desventajas significativas:

- Al seleccionar la mejor prueba, no se toma en cuenta la entropía presente en el conjunto de instancias que respondieron de forma negativa y solo se realiza el cálculo con las que respondieron de forma positiva a la prueba.
- La estructura de la prueba es poco flexible ya que obliga a interrogar por la pertenencia de solo un elemento y por tanto la respuesta solo puede ser: afirmativa o negativa. Este tipo de prueba no explota ninguna ventaja de utilizar conjuntos como tipo de dato.

1.2.3.2 Algoritmo *k*-NN

El algoritmo k-NN (k-Nearest Neighbor) [28] es uno de los algoritmos tradicionales del aprendizaje basado en instancias perteneciente al enfoque perezoso. El aprendizaje en este tipo de algoritmo consiste simplemente en el almacenamiento del conjunto de datos de entrenamiento. Cuando se presenta una nueva instancia para ser clasificada son recuperadas desde la memoria las instancias similares relacionadas con esta y posteriormente utilizadas para clasificar la instancia. Una de las desventajas del enfoque basado en instancias es que el costo de clasificar una nueva instancia puede ser alto ya que todos los cálculos tienen lugar en el momento de la clasificación.

El k-NN asume que todas las instancias se corresponden a puntos en el espacio n-dimensional $\Re^n$. El vecino más cercano a una determinada instancia está definido en términos de la distancia Euclidiana estándar. De forma más precisa,

dada una instancia x descrita mediante el vector de rasgos $\langle x_1, x_2, ..., x_n \rangle$ donde x_i denota el valor del i-ésimo atributo de la instancia x. La distancia entre dos instancias x y y está definida por la expresión $d(x,y)$ (Fórmula 1.4).

$$d(x, y) \equiv \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (1.4)$$

En el tratamiento de datos heterogéneos muchos autores han propuesto distancias para utilizar el modelo del vecino más cercano. Esto no es un problema sencillo ya que si existen valores perdidos la comparación entre estos valores y con otros, no resulta tarea fácil. Esto ha traído como consecuencia que se violen propiedades que debe cumplir una función para que sea considerada una distancia. En [1] se hace un análisis profundo al respecto.

Una de estas soluciones reportadas en la literatura es la función denominada por sus autores Heterogeneous Euclidean-Overlap Metric (HEOM) [29]. Esta función permite calcular la distancia entre valores de atributos simbólicos y numéricos (Fórmula 1.5.1 y Fórmula 1.5.2). La función HEOM toma valores en el intervalo [0..1] devolviendo el grado de disimilitud entre las instancias que se comparan, donde uno representa la diferencia máxima y cero la mínima. Si al menos uno de los atributos a comparar tiene un valor perdido (?) se considera entonces la máxima disimilitud.

$$HEOM(x, y) \equiv \sqrt{\sum_{i=1}^{n} d_i^2(x_i, y_i)} \qquad (1.5.1)$$

$$d_i(x_i, y_i) = \begin{cases} 1 \text{ si } x_i = ? \text{ ó } y_i = ? \\ \delta_{\text{overlap}}(x_i, y_i) \text{ simbólico} \\ \delta_{norm1}(x_i, y_i) \text{ numérico} \end{cases} \qquad (1.5.2)$$

$$\delta_{overlap}(x_i, y_i) = \begin{cases} 0 \text{ si } x_i = y_i \\ 1 \text{ e.o.c.} \end{cases} \qquad (1.5.3)$$

$$\delta_{norm1}(x_i, y_i) = \frac{|x_i - y_i|}{l_i - u_i} \qquad (1.5.4)$$

Si los valores a comparar son simbólicos (Fórmula 1.5.3) la función solo toma dos valores posibles: cero si son iguales y uno si son diferentes. En caso de que sean numéricos (Fórmula 1.5.4) se calcula la diferencia entre los valores, normalizada.

Esta función no es definida positiva[1], pues bastaría que en la descripción de un objeto x apareciera un valor desconocido para que ocurriera que $HEOM(x,x) \neq 0$, lo que no sería una distancia.

Otro intento lo constituye la función Heterogeneous Value Difference Metric (HVDM) [30] la cual aunque no cumple los requerimientos para ser una métrica de distancia, ha recibido gran atención por parte de la comunidad científica. La función VDM [31] difiere de otras muchas métricas en que la distancia entre los valores de un atributo no se determina comparando los valores en sí, sino a partir de la comparación de las probabilidades condicionales de que la instancia pertenezca a una determinada clase dado el valor del atributo (Fórmula 1.6).

$$vdm(x,y) = \sum_{i=0}^{n} \delta(x_i, y_i).w(x_i) \qquad (1.6.1)$$

$$\delta(x_i, y_i) = \sum_{c \in C} \left| P(c \mid x_i) - P(c \mid y_i) \right|^2 \qquad (1.6.2)$$

$$w(x_i) = \left[\sum_{c \in C} P(c \mid x_i)^2 \right]^{0.5} \qquad (1.6.3)$$

$$P(c \mid x_i) = \frac{\left| (\text{instancias que contienen} \quad x_i) \ \wedge \ (clase = c) \right|}{\left| \text{instancias que contienen} \quad x_i \right|} \qquad (1.6.4)$$

Dadas dos instancias x y y, un conjunto de n atributos y siendo C el conjunto de todas las clases presentes en el conjunto de entrenamiento, la probabilidad condicional $P(c|x_i)$ es la probabilidad de que el valor x_i ocurra en el conjunto de datos, para el atributo i, conjuntamente con la clase c (Fórmula 1.6.4). La componente pesada de VDM (Fórmula 1.6.3) provee información acerca de cuán bien el valor de un atributo discrimina entre las diferentes clases. Dicha

[1] Una función es definida positiva cuando toma el valor cero solo en el caso en que se compara un elemento, un objeto, consigo mismo. Sólo en este caso la función toma el valor cero.

componente puede variar entre un mínimo y uno, el cual representa un discriminador ideal.

Payne propone en [3] tres algoritmos basados en el modelo del vecino más cercano, que utilizan representación basada en conjuntos para los atributos: IBPL1, IBPL2 y PIBPL. Los dos primeros difieren en la forma en que se comparan los conjuntos, en el momento en que una nueva instancia va a ser clasificada. El tercero investiga un método para la poda de términos irrelevantes en los conjuntos.

IBPL1 es una instancia del k-NN que compara dos conjuntos a partir de la similitud promedio entre los elementos de dichos conjuntos. La similitud promedio entre una nueva instancia X_i y una instancia almacenada en el conjunto de entrenamiento Y_i, para el atributo i, es calculada de la siguiente forma:

1. Se calcula la distancia entre un elemento x en el conjunto X_i y cada elemento y del conjunto Y_i mediante $\sum_{y \in Y_i} \delta(x, y)$.
2. Se calcula la distancia para cada elemento x en X_i y se suma para generar una distancia total, es decir $\sum_{x \in X_i} \sum_{y \in Y_i} \delta(x, y)$.
3. Esta distancia total se divide por la cantidad total de las distancias calculadas, es decir, el producto del cardinal de los conjuntos (Fórmula 1.7.1).

$$ibpl1(X_i, Y_i) = \frac{\sum_{x \in X_i} \sum_{y \in Y_i} \delta(x, y)}{sizeof(X_i) \times sizeof(Y_i)} \qquad (1.7.1)$$

$$D(x, y) = \sum_{i=0}^{n} ibpl1(X_i, Y_i) \qquad (1.7.2)$$

$$P(c \mid x) = \frac{\left|\text{instancias que cumplen} \quad x \in X_i \ \wedge \ clase = c\right|}{\left|\text{instancias que cumplen} \quad x \in X_i\right|} \qquad (1.7.3)$$

La distancia total entre dos instancias, se calcula mediante la suma del promedio de la distancia entre cada atributo de tipo conjunto que conforma la instancia

(Fórmula 1.7.2). Para calcular la expresión $\delta(x, y)$ en la Fórmula 1.7.1 se utiliza la distancia VDM, de esta forma se garantiza que sea pequeña la distancia entre elementos con similar distribución por clase. La probabilidad condicional de un elemento x perteneciente a un atributo i de tipo conjunto, se reescribe (Fórmula 1.7.3) para mayor claridad. IBPL1 asume máxima distancia cuando en un conjunto de la instancia a ser clasificada aparece algún símbolo desconocido ($\delta(x_q, y) = 1$, cuando x_q es un símbolo desconocido).

La Figura 5 muestra un ejemplo del uso del IBPL1 en el filtrado del correo electrónico. Los atributos predictores de tipo conjunto son: *Subject* (*Título*) y *Body* (*Cuerpo*), obtenidos a partir de un mensaje electrónico. El atributo clase es el buzón de destino. La instancia superior pertenece al conjunto de entrenamiento y la inferior es una nueva instancia de la cual se desconoce su clase. Los términos "*apprentice*" y "*plan*" son términos desconocidos por lo que se asigna valor máximo a las comparaciones que los involucran. Las líneas indican las comparaciones realizadas dentro de cada conjunto. Estas distancias son entonces sumadas y divididas por el número total de mediciones, seis para el caso del atributo *Subject*.

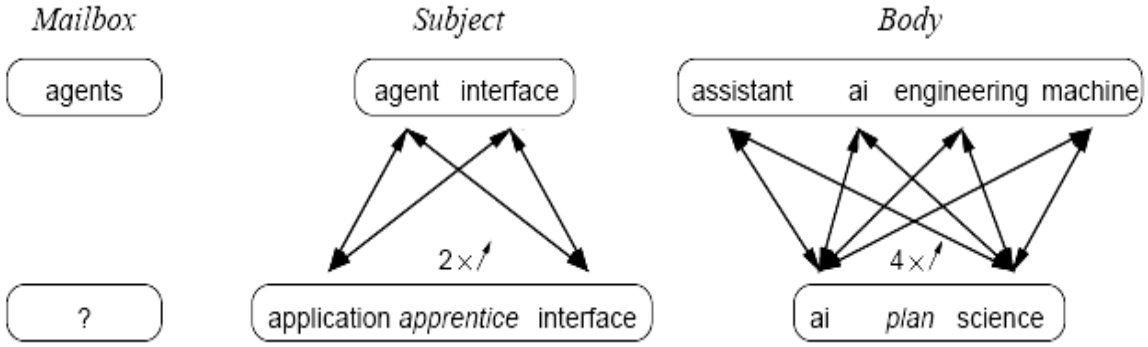

Figura 5 Algoritmo IBPL1 aplicado en la comparación de dos instancias.

El IBPL2 constituye una mejora al IBPL1, dado que en este último la distancia entre dos conjuntos idénticos puede resultar diferente de cero. El IBPL2 supera esta limitación considerando solo la distancia entre cada elemento del conjunto X_i y el elemento más cercano del conjunto Y_i correspondiente al conjunto de entrenamiento (Fórmula 1.8).

$$ibpl2(X_i, Y_i) = \frac{\sum_{x \in X_i} \sum_{y \in Y_i} Min\delta(x, y)}{sizeof(X_i) \times sizeof(Y_i)} \quad (1.8)$$

De esta forma se garantiza que la distancia entre dos conjuntos idénticos siempre sea cero. La Figura 6 retoma el ejemplo de la Figura 5 pero aplicado al IBPL2. Las líneas sólidas denotan distancias mínimas encontradas. Las líneas discontinuas denotan distancias alejadas del mínimo.

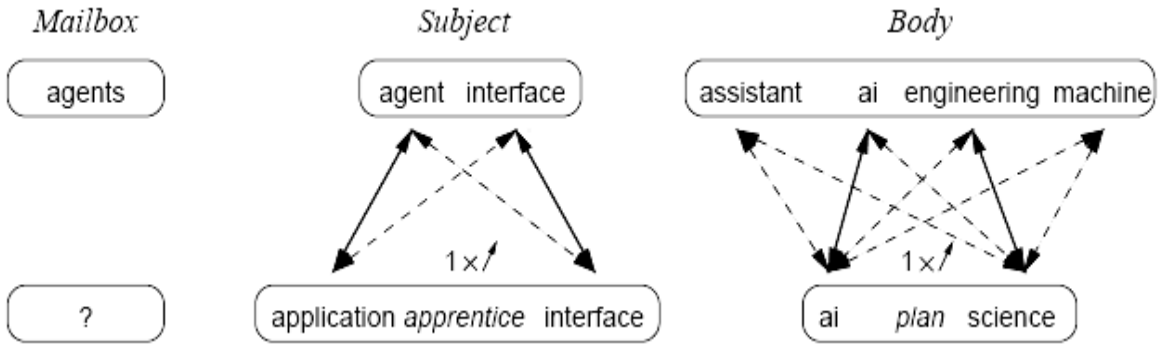

Figura 6 Algoritmo IBPL2 aplicado en la comparación de dos instancias.

El algoritmo PIBPL utiliza los pesos de ecuación (1.6.3) para filtrar términos de los conjuntos. Los términos cuyos pesos no alcancen un determinado umbral son removidos de los conjuntos presentes en las instancias de entrenamiento. Los términos desconocidos siempre son rechazados de las comparaciones.

La principal desventaja de los algoritmos propuestos por Payne es que realiza una descomposición de los conjuntos para calcular la distancia entre ellos, en lugar de utilizarlos como un todo. Además de la ventaja que implica la reducción de la dimensionalidad lograda a partir del uso de los conjuntos, esta propuesta no aprovecha ninguna otra ventaja de su utilización.

1.2.3.3 Algoritmo Naive-Bayes

El razonamiento Bayesiano provee a la inferencia de un enfoque probabilístico. Este se basa en la suposición de que los valores de interés son gobernados por distribuciones probabilísticas y que estas pueden ser utilizadas en la toma de decisiones. Este tipo de enfoque sienta las bases para algoritmos del aprendizaje automatizado que manipulan probabilidades directamente.

El teorema de Bayes brinda una forma de calcular la probabilidad de una hipótesis basado en: la probabilidad a priori de la hipótesis $P(h)$, la probabilidad de los datos observados dada la hipótesis $P(D \mid h)$ y la probabilidad de los datos observados en sí $P(D)$. En los problemas de aprendizaje automatizado el interés recae en la probabilidad $P(h \mid D)$, de que el evento h ocurra dado un conjunto de entrenamiento observado D. El teorema de Bayes es la piedra angular de los métodos de aprendizaje probabilísticos, debido a que provee una forma de calcular dicha probabilidad (Fórmula 1.9).

$$P(h \mid D) = \frac{P(D \mid h).P(h)}{P(D)} \qquad (1.9)$$

En el contexto del aprendizaje automatizado los datos observados se asocian con ejemplos de entrenamiento y la hipótesis h con un candidato de la función objetivo.

El clasificador Naive Bayes [32] es uno de los principales métodos de aprendizaje pertenecientes al enfoque probabilístico. En varios dominios de desempeño ha sido comparable al de las redes neuronales y los árboles de decisión [33]. El algoritmo Naive Bayes es aplicado en tareas de aprendizaje donde cada instancia x es descrita mediante una conjunción de valores y existe una función objetivo $f(x)$ que toma uno de los valores de un conjunto finito V.

El clasificador es entrenado a partir de un conjunto de instancias y posteriormente le es presentada una nueva instancia a ser clasificada, descrita por una tupla de valores $\langle x_1, x_2, ..., x_n \rangle$. El clasificador es interrogado para predecir el valor objetivo o clasificación de la nueva instancia. Esto se resuelve asignando el valor objetivo más probable, dados los valores de los atributos que describen la instancia (Fórmula 1.10).

$$v_{MAP} = \underset{v_j \in V}{\operatorname{argmax}} \frac{P(x_1, x_2, ..., x_n \mid v_j).P(v_j)}{P(x_1, x_2, ..., x_n)} \qquad (1.10)$$

La probabilidad del denominador puede ser eliminada de los cálculos ya que es constante para todos los valores v_j. Las dos probabilidades restantes son

estimadas a partir del conjunto de entrenamiento. $P(v_j)$ se calcula a partir de la frecuencia con que cada valor objetivo v_j aparece en el conjunto de entrenamiento.

Naive Bayes se basa en el supuesto de que los valores de los atributos son independientes condicionalmente dado el valor objetivo. De esta forma la probabilidad de que ocurra la conjunción $x_1, x_2, ..., x_n$ no es más que el producto de las probabilidades de cada atributo individualmente: $P(x_1, x_2, ..., x_n \mid v_j) = \prod_i P(x_i \mid v_j)$. A partir de esta sustitución se obtiene la expresión final para el clasificador Naive Bayes (Fórmula 1.11).

$$v_{NB} = \underset{v_j \in V}{\operatorname{argmax}} P(v_j).\prod_i P(x_i \mid v_j) \qquad (1.11)$$

En [34] se proponen dos algoritmos que extienden el clasificador Naive Bayes para la clasificación de datos estructurados: 1BC y 1BC2. Los datos estructurados abarcan un concepto mucho más amplio que el de conjuntos y comprenden dominios tan diversos como: la biología molecular, la bioinformática, la recuperación de información, el procesamiento de imágenes y muchos otros. El primer enfoque 1BC realiza una proyección del individuo o estructura en rasgos de primer orden que se obtienen a partir de la estructura. El segundo considera distribuciones de probabilidades sobre objetos estructurados y estima estas distribuciones a partir de las probabilidades de los elementos que los componen.

La representación utilizada para describir las estructuras es conocida con el nombre de representación ISP (*Individual, Structural predicates and Properties*). Las declaraciones de tipo ISP constan de tres partes fundamentales: declaración del individuo, declaración de predicados estructurales y declaración de propiedades. La Figura 7 muestra un ejemplo de declaración ISP que describe una molécula. La primera parte declara al individuo como una molécula. La segunda involucra los aspectos estructurales, describiendo el tipo de relación que se establece entre los objetos que conforman la estructura; una molécula está compuesta por varios átomos y a su vez un átomo puede estar enlazado

con otro. Las propiedades siempre asignan valores como por ejemplo: tipo del átomo, carga, tipo de enlace, etc.

```
--INDIVIDUAL
molecule 1 mol
--STRUCTURAL
mol2atom 2 1:mol *:atom
fr_atom2bond 2 1:atom *:bond
to_atom2bond 2 1:atom *:bond
--PROPERTIES
class 2 mol #class
lumo 2 mol #lumo
logp 2 mol #logp
element 2 atom #element
atomType 2 atom #atomType
charge 2 atom #charge
bondType 2 bond #bondType
```

Figura 7 Representación ISP de una estructura que representa una molécula.

1BC [35] es una variante del clasificador Naive Bayes que trabaja con rasgos elementales de primer orden. Este tipo de rasgo involucra conjunciones de literales que contienen una propiedad. Por ejemplo "$mol2atom(M,A),element(A,c)$" averigua si la molécula contiene un átomo de carbono. El rasgo puede tomar como valor verdadero o falso, en dependencia de si el individuo cumple o no dicha propiedad. El sistema 1BC en la etapa de entrenamiento genera rasgos de primer orden y cuenta cuántos individuos de cada clase satisface cada valor de cada rasgo. Específicamente por cada rasgo i, cada valor del rasgo x_i y cada clase c, se obtienen las cantidades $n(x_i \wedge c)$ y $n(c)$. La función $estP$ calcula la probabilidad estimada a partir de estas cantidades usando la corrección de Laplace (Fórmula 1.12), donde ϕ_i es el número de valores distintos de x_i.

$$estP(x_i \mid c) = \frac{n(x_i \wedge c)+1}{n(c)+\phi_i} \qquad (1.12)$$

La etapa de clasificación dada una nueva instancia comprende: el cálculo de la probabilidad por cada clase, multiplicar esta por la probabilidad a priori de la clase y determinar cuál clase maximiza este producto. La Figura 8 detalla el

algoritmo 1BC para determinar las probabilidades $P(i \mid c)$ de un individuo i, dada la clase c, para todos los posibles valores de c.

```
getLikelihood(i) {
/* initialise likelihoods */
FOR each class c
    CL[c] = 1;
/* iterate through features */
FOR each feature F
    let f be the value of F for i
    FOR each class c
        CL[c] = CL[c] x estP(f|c);
Return CL}
```

Figura 8 Algoritmo de clasificación 1BC.

1BC2 modela directamente un espacio probabilístico sobre tipos de datos estructurados. Esto requiere de la definición de la distribución de las probabilidades sobre este tipo de dato, la cual se define como la unión de la probabilidad de sus propiedades y de sus objetos relacionados. En la fase de entrenamiento se estima la probabilidad de cada valor de una propiedad para cada clase y cada contexto. La Figura 9 detalla el algoritmo que calcula la contribución de un objeto *o*, a partir de un contexto *ctxt* y una clase *c*. El contexto es una especie de camino formado por las propiedades estructurales y que relacionan al individuo con propiedades de sus objetos relacionados.

```
addContribution(o,ctxt,c) {
/* handle properties of the object */
FOR all properties prop of o
    let v be the corresponding parameter value;
    propValueCounts[prop,v,ctxt,c] += 1;
/* handle related objects */
FOR all structural predicates struc involving o
    ctxt' = add struc to ctxt;
    IF struc is functional given o THEN
        let o' be the related object;
        addContribution(o',ctxt',c);
    ELSE /* non-determinate structural predicate */
        let O' be the set of related objects;
        parentCounts[struc,ctxt,c] += 1;
        childrenCounts[struc,ctxt,c] += |O'|;
        FOR each object o' of O'
            addContribution(o',ctxt',c);
}
```

Figura 9 Algoritmo que estima la probabilidad de cada valor de una propiedad.

Este algoritmo mantiene una matriz de cuatro dimensiones *propValueCounts* la cual es actualizada en el recorrido por todas las instancias de entrenamiento. Posteriormente estas cantidades son utilizadas para estimar las probabilidades condicionales por clase. Para calcular el aporte probabilístico de estructuras que contengan conjuntos de objetos, se utiliza la distribución geométrica sobre listas (Fórmula 1.13).

$$P_{li}([x_{j1},...,x_{jl}]) = \tau(1-\tau)^{l}\prod_{i=1}^{l}P_D(x_{ji}) \qquad (1.13)$$

El símbolo τ se refiere a la probabilidad de la lista vacía, l es la cantidad de elementos de la lista y $P_D(x_{ji})$ es la probabilidad de aparición en la lista de un elemento x_{ji} en particular. La probabilidad de la lista vacía para 1BC2 se calcula como: $\tau_c = \frac{1}{1+\bar{l}_c}$, donde $\bar{l}_c$ es la longitud promedio de las colecciones dada la clase c, un contexto *ctxt* y una estructura. La Figura 10 muestra el pseudo código que devuelve las probabilidades por clase, dado un objeto y un contexto.

```
getLikelihood(o,ctxt) {
/* initialise likelihoods */
FOR each class c
    CL[c] = 1;
/* handle properties of the object */
FOR all properties prop of o
    let v be the corresponding parameter value;
    FOR each class c
        CL[c] = CL[c] x estP2(prop,v,ctxt|c);
/* handle related objects */
FOR all structural predicates struc involving i
    ctxt' = add struc to ctxt;
    IF struc is functional given o THEN
        let o' be the related object;
        CL' = getLikelihood(o',ctxt');
        FOR each class c
            CL[c] = CL[c] x CL'[c];
    ELSE /* non-determinate structural predicate */
        let O' be the set of related objects;
        FOR each object o' of O'
            CL''[o'] = getLikelihood(o',ctxt');
        FOR each class c
            tau = estTau(struc,ctxt|c);
            CL[c] = CL[c] x composeP(CL'',tau,c);
Return CL}
```

Figura 10 Cálculo de las probabilidades por clase en el algoritmo 1BC2, dado un objeto y un contexto.

Los resultados experimentales concluyeron que 1BC2 supera 1BC, excepto en conjuntos de entrenamiento pequeños donde la calidad de 1BC2 se ve afectada.

1.3 Herramienta de aprendizaje automatizado Weka

Waikato Environment for Knowledge Analysis (Weka[2]) es una herramienta de aprendizaje automatizado [5], fue Ian Witten, profesor del Departamento de Ciencias de la Computación de la Universidad de Waikato, Nueva Zelanda (1992) quien la creó. La primera versión publicada contaba con algunos algoritmos en C, posteriormente se decidió rescribir Weka en Java y a partir de la versión 3 ya Weka quedó escrita totalmente en Java. Desde este momento se

[2] WEKA is a Java-written open source. It is available at http://www.cs.waikato.ac.nz/~ml/weka/ under the GNU General Public License.

le comenzaron a hacer mejoras y se le agregaron nuevas facilidades apareciendo así diferentes versiones. Weka superó los límites del lugar de creación y hoy día en varios lugares del mundo diferentes personas se esfuerzan en la ampliación y perfeccionamiento de esta herramienta.

La herramienta Weka es un ambiente de trabajo para la prueba y validación de algoritmos de la Inteligencia Artificial. Tiene implementada una colección de algoritmos conocidos, varias maneras para preprocesar los archivos de datos a utilizar por dichos algoritmos; así como facilidades para validar los mismos. Posee interfaces gráficas de usuario (*GUI: Graphical User Interface*) y cuenta con herramientas para realizar tareas de regresión, clasificación, agrupamiento, asociación y visualización.

1.4 Conclusiones parciales

A partir del análisis de la bibliografía encontrada se concluye:

- ✓ En el aprendizaje automatizado existen múltiples problemas con rasgos que se pueden expresar de forma natural mediante conjuntos.
- ✓ Los algoritmos que se reportan en la literatura al igual que las herramientas de aprendizaje automatizado, no suelen manejar atributos de tipo conjunto.
- ✓ Las propuestas analizadas que sí tratan los conjuntos como atributos no explotan totalmente los beneficios de su utilización.

Capítulo 2: Algoritmos propuestos para el tratamiento de los conjuntos como atributos

En este capítulo se abordarán varias propuestas de algoritmos que manejan atributos de tipo conjunto. Primeramente se hacen varias precisiones sobre cuándo nos encontramos en presencia de un dato de tipo conjunto en el aprendizaje automatizado. Posteriormente se describen los algoritmos propuestos, algunos están inspirados en los algoritmos encontrados en la literatura y otros son totalmente nuevos. En este capítulo, aunque solo se hace énfasis en los atributos de tipo conjunto, no se excluye la presencia de atributos de otra naturaleza en el conjunto de aprendizaje como numéricos o nominales.

2.1 Los conjuntos como tipo de dato

En el Capítulo 1 se presentaron argumentos sobre la necesidad de definir un tipo de dato, que permitiera representar conjuntos en el contexto del aprendizaje automatizado. Debido a que el concepto de conjunto resulta sumamente amplio se hace necesario acotarlo para su tratamiento como atributo. En este trabajo se limita la noción de conjunto, a datos que cumplen las siguientes propiedades:

- ✓ Existe una colección de elementos bien definida, $X = \{x_1, x_2, ..., x_k\}$
- ✓ Los elementos pertenecientes a dicha colección no se repiten, $\forall i \neq j : x_i \neq x_j$
- ✓ El orden en que aparecen los elementos en la colección carece de importancia

Además solamente se tratarán conjuntos que contengan un número finito de elementos. La colección formada por todos los posibles valores, que pueden pertenecer a los conjuntos de un determinado atributo i, deben conocerse de antemano y se hará referencia a este conjunto U_i, como el dominio del atributo i. Para tener una idea gráfica de cómo quedaría el conjunto de entrenamiento cuando contiene atributos de tipo conjunto, consultar Figura 11.

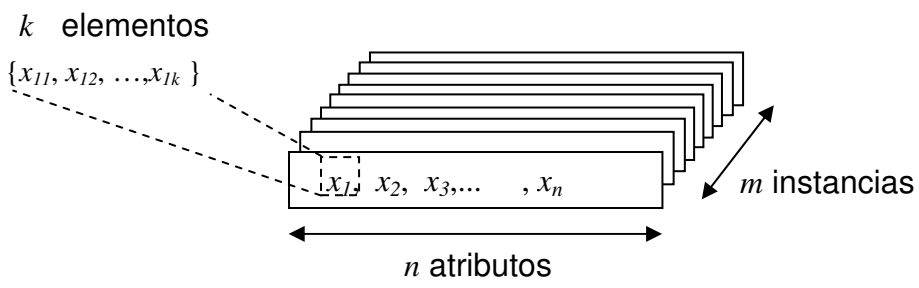

Figura 11 Conjunto de entrenamiento donde el atributo x_1 es de tipo conjunto.

La sintaxis[3] para la declaración de los atributos de tipo conjunto es la siguiente: @attribute <nombre del atributo> SET $\{< x_1 >,< x_2 >,...,< x_n >\}$

En la declaración del atributo se define también el dominio del mismo. El valor que toma cada instancia para este atributo será un conjunto X_i definido por extensión, el cual podrá tomar cualquier combinación de los valores definidos en el dominio e incluso el conjunto vacío ($X_i \in P_i$, donde P_i es el conjunto potencia que se obtiene a partir de U_i). Para más información acerca de la implementación de este tipo de atributo consultar [36]. En el Anexo 1 se muestra un ejemplo del uso de este nuevo tipo de dato en un conjunto de entrenamiento.

2.2 Extensión de varios algoritmos para el tratamiento de conjuntos

En este epígrafe se hacen varias propuestas del tratamiento de datos de tipo conjunto. Algunas constituyen mejoras a variantes de los algoritmos ya analizados y otras consisten en una nueva perspectiva, basada en funciones de distancias entre conjuntos. Con estas propuestas se pretende dar un tratamiento apropiado al atributo de tipo conjunto en tres enfoques diferentes del aprendizaje automatizado: el enfoque perezoso, el enfoque basado en árboles de decisión y el probabilístico.

[3] La sintaxis que se propone ya se encuentra definida en la herramienta de aprendizaje automático Weka.

2.2.1 Extensión al ID3

En esta sección se retoma la variante propuesta por Cohen que fue analizada anteriormente en el Capítulo 1. Una de las desventajas de esta variante es lo rígido de la estructura de la prueba, la cual para datos de tipo conjunto solo tenía dos posibilidades: "$x_{ij} \in X_i$" y "$x_{ij} \notin X_i$", donde x_{ij} representa un elemento del dominio del atributo y X_i es el valor que toma una determinada instancia para un rasgo i de tipo conjunto.

La propuesta que se presenta a continuación consiste en generalizar esta prueba de forma tal que el elemento de la izquierda sea un conjunto en lugar de un elemento. Entonces la nueva prueba para el atributo i-ésimo de tipo conjunto tendría la forma: "$V_i \subseteq X_i$" y "$V_i \not\subset X_i$", donde tanto V_i como X_i pertenecen al conjunto potencia del atributo i $(V_i, X_i \in P_i)$. La prueba original propuesta por Cohen sería un caso particular de esta generalización, cuando el conjunto V_i contiene un solo elemento. De esta manera es posible encontrar subconjuntos que discrimen mejor entre las clases.

El pseudo código del algoritmo que busca la mejor prueba, la cual calcula a partir de un conjunto de ejemplos S, y un atributo i la prueba de mejor desempeño, tendría varias modificaciones (Figura 12). La función *PowerSet*, dado un conjunto de elementos del dominio *Visited* y una longitud máxima l_u que pueden alcanzar los conjuntos, devuelve el conjunto potencia restringido. El conjunto potencia se restringe de esta forma para controlar la complejidad temporal de la función *MaxGainTest*, la cual depende del tamaño de *Sets* y de la muestra S. La complejidad del método anterior para la propuesta original de Cohen pertenece a un $O(máx(m, n))$, siendo m la cantidad de instancias en S y n el número de elementos pertenecientes al dominio del atributo. Al realizar la modificación que se propone, esta función eleva su complejidad a un $O(máx(m, n^{l_u}))$.

Otro de los cambios consiste en la modificación de la matriz que anteriormente guardaba las apariciones de cada elemento x_{ij} por clase, dicha matriz ahora

tendrá como longitud por filas la misma que el conjunto *Sets* y se llamará *SetCount*.

```
function MaxGainTest(S, i, l_u)
1       Visited := φ;
2       TotalCount[+] := 0;       TotalCount[-] := 0;
3       for each example ⟨x⃗, y⟩ in the sample S do
4               for each element x_ij ∈ X_i do
5                       Visited := Visited ⋃ {x_ij};
6               endfor
7               TotalCount[y] := TotalCount[y] + 1;
8       endfor
9       Sets := PowerSet(Visited, l_u);
10      for each set V_i ∈ Sets do
11              for each example ⟨x⃗, y⟩ in the sample S do
12                      if (V_i ⊆ X_i) then
13                              SetCount[V_i, y] := SetCount[V_i, y] + 1;
14                      endif
15              endfor
16      endfor
17      BestEntropy := -1;
18      for each V_i ∈ Sets do
19              p := SetCount[V_i, +];
20              n := SetCount[V_i, -];
21              EntropyPositive := Entropy(p,n) * (p + n) / (TotalCount[+] + TotalCount[-]);
22              p' := TotalCount[+] – SetsCount[S,+];
23              n' := TotalCount[-] – SetsCount[S,-];
24              EntropyNegative := Entropy(p',n') * (p' + n') / (TotalCount[+] + TotalCount[-]);
25              Entropy := EntropyPositive + EntropyNegative;
26              if (Entropy > BestEntropy) then
27                      BestTest := "V_i ⊆ X_i)";
28                      BestEntropy := Entropy;
29              endif
30              SetCount[V_i,+] := 0;
31              SetCount[V_i, -] := 0;
32      endfor
33      return BestTest
```

Figura 12 Pseudo código de la función *MaxGainTest* extendida.

Una de las deficiencias detectadas al algoritmo propuesto por Cohen, está dada por la ausencia de un análisis sobre el conjunto de instancias que respondieron de forma negativa a la prueba. Por este motivo también se incluye el cálculo de la entropía presente en este conjunto de instancias en el momento de seleccionar la mejor prueba (líneas 22-25). En cada caso la entropía se pondera

por la fracción de los ejemplos que cumplen (no cumplen) la prueba. Al incluir este análisis carece de sentido mantener la prueba alternativa "$V_i \not\subset U_i$", por lo que estas líneas del algoritmo son removidas. Al efectuar esta modificación para el caso de l_u = 1 el algoritmo coincide con el ID3 clásico variante binaria debido a que se selecciona en cada momento el elemento de mejor entropía.

A continuación se muestra un ejemplo para una mejor comprensión de los cambios realizados. Supóngase que dado un conjunto de instancias *Visited* obtuvo valor: $\{x_1, x_2, x_3\}$. Si la longitud máxima considerada es l_u = 2, entonces el conjunto potencia restringido *Sets* obtiene como valor: $\{\{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}\}$ y la matriz *SetCount* tendrá seis filas. Suponiendo que el subconjunto que mejor valor de entropía obtuvo fue $\{x_1, x_2\}$, el nuevo nodo sería etiquetado con dicho valor (Figura 13).

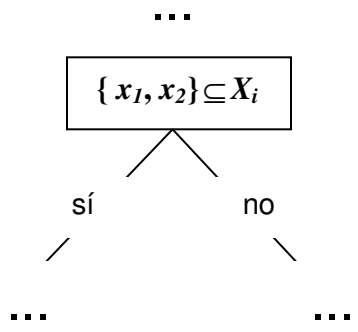

Figura 13 Nodo etiquetado con el conjunto de mejor valor de entropía.

Esta es una forma de aprovechar las potencialidades de la utilización de conjuntos en el algoritmo ID3. Aunque la extensión realizada al algoritmo de Cohen presenta una mayor complejidad computacional, esta se ve compensada en la etapa de clasificación debido a que se generan árboles con menor cantidad de nodos.

2.2.2 Extensión al *k*-NN

Para extender el *k*-NN con el manejo de atributos de tipo conjunto primeramente se selecciona la función de distancia a modificar, ya que esta constituye un elemento clave en este tipo de algoritmo. La función seleccionada fue la HEOM, debido a que esta considera datos de distinta naturaleza, de una forma sencilla.

La propuesta presentada [37, 38] se basa en añadir una nueva rama a la distancia HEOM (Fórmula 2.1) que permita el cálculo de la distancia entre conjuntos, la cual llamaremos $\delta_{set}(X,Y)$.

$$d_i(x_i,y_i)=\begin{cases}1 \text{ si } x_i=? \text{ ó } y_i=? \\ \delta_{\text{overlap}}(x_i,y_i) \text{ simbólico} \\ \delta_{norm1}(x_i,y_i) \text{ numérico} \\ \delta_{set}(X_i,Y_i) \text{ conjunto}\end{cases} \quad (2.1)$$

La expresión $\delta_{set}(X,Y)$ va a estar definida por la distancia entre conjuntos que se utilice, en este caso se tuvieron en cuenta siete variantes utilizadas en el análisis de datos [39], específicamente datos formados por secuencias binarias. En la Tabla 1 se muestran dichas distancias. La mayoría de estas funciones aparecían en realidad como similitudes a las cuales fue necesario transformar en distancias (Fórmula 2.2, dada una similitud s se obtiene una distancia d).

$$d=1-s \quad (2.2)$$

Tabla 1 Distancias entre conjuntos.

Jaccard: $\delta_{set}(X,Y)=\frac{\lvert X\Delta Y\rvert}{\lvert X\cup Y\rvert}$	Czakanowsky-Dice: $\delta_{set}(X,Y)=\frac{\lvert X\Delta Y\rvert}{\lvert X\rvert+\lvert Y\rvert}$
Sokal-Sneath: $\delta_{set}(X,Y)=1-\frac{\lvert X\cap Y\rvert}{\lvert X\cup Y\rvert+\lvert X\Delta Y\rvert}$	Cosine: $\delta_{set}(X,Y)=1-\frac{\lvert X\cap Y\rvert}{\sqrt{\lvert X\rvert\cdot\lvert Y\rvert}}$
Braun-Blanquet: $\delta_{set}(X,Y)=1-\frac{\lvert X\cap Y\rvert}{máx\{\lvert X\rvert,\lvert Y\rvert\}}$	Simpson: $\delta_{set}(X,Y)=1-\frac{\lvert X\cap Y\rvert}{mín\{\lvert X\rvert,\lvert Y\rvert\}}$
Kulczynski: $\delta_{set}(X,Y)=1-\frac{\lvert X\cap Y\rvert\cdot(\lvert X\rvert+\lvert Y\rvert)}{2\cdot\lvert Y\rvert\cdot\lvert X\rvert}$	

Todas estas distancias se basan en operaciones clásicas entre conjuntos como la intersección ($X\cap Y$), la unión ($X\cup Y$), la diferencia simétrica[4] ($X\Delta Y$) y el

[4] $X\Delta Y=(X-Y)\cup(X-Y)$

cardinal ($|X|$). Las distancias anteriores tienen como argumento dos conjuntos y toman valores de salida en el intervalo [0,1]. En caso de que los dos conjuntos sean vacíos, las funciones de distancia se indefinen, por lo que se toma valor cero ya que son iguales. La complejidad temporal de estas funciones de distancia pertenecen a un orden de $O(n)$, donde n es igual al número de elementos del dominio del atributo.

La función HEOM así definida permite calcular la distancia entre conjuntos, dando lugar a siete funciones diferentes en dependencia de la función de distancia que se utilice para comparar conjuntos [40]. Todas estas variantes de HEOM permiten el tratamiento de datos de tipo conjunto de una forma sencilla y natural.

2.2.3 Extensión al Naive Bayes

En este epígrafe primeramente se retoman los algoritmos analizados en el Capítulo 1 referentes al enfoque probabilístico: 1BC y 1BC2. Estos algoritmos calculan la probabilidad por clase de un dato estructurado. Un dato estructurado permite describir individuos complejos y en este epígrafe vamos a adaptar este concepto a datos de tipo conjunto.

Al considerar un conjunto como una estructura se hace necesaria su representación mediante una declaración ISP (Figura 14). En dicha declaración se describe al individuo como un conjunto, el cual va a estar formado por múltiples elementos, los cuales a su vez tienen un valor. Los valores de estos elementos son propiedades de la estructura. De esta forma se conserva la estructura de un conjunto, ya que solo se formaliza a través de una declaración ISP.

```
-- INDIVIDUO
conjunto 1 set
-- ESTRUCTURA
set2elem 2 1:set *:elem
-- PROPIEDADES
value 2 elem #value
```

Figura 14 Declaración ISP de un conjunto.

Algoritmo 1BC

Según lo analizado anteriormente, este algoritmo considera rasgos elementales de primer orden y cuenta la cantidad de instancias que satisfacen cada rasgo. A partir de estas cantidades calcula la probabilidad condicional $P(x_{ij} \mid c)$, donde para el caso particular de un conjunto, x_{ij} se refiere al valor del rasgo representado por el j-ésimo elemento del dominio correspondiente al atributo i-ésimo. Esto es debido a que en la estructura mencionada anteriormente los rasgos siempre van a tener la forma "$set2elem(X, E)$,$value(E, x_j)$" y su significado es: si un elemento de valor x_j pertenece al conjunto X. La cantidad de rasgos va a coincidir con el número de elementos presentes en el dominio del atributo. En este punto se hace posible definir la probabilidad condicional de un conjunto X_i dada una clase c, un atributo i de tipo conjunto (Fórmula 2.3), y siendo n es la cantidad de rasgos considerados. Los valores que puede tomar el rasgo x_{ij} son: verdadero, si el valor x_{ij} pertenece al conjunto X_i y falso en caso contrario.

$$P(X_i \mid c) = \prod_{j=1}^{n} P(x_{ij} \mid c) \qquad (2.3)$$

$$P(x_{ij} \mid c) = \frac{n(x_{ij} \wedge c) + 1}{n(c) + \phi_i} \qquad (2.4)$$

La expresión $P(x_{ij} \mid c)$ es la probabilidad de que se tome el valor x_{ij}, dada la clase c (Fórmula 2.4), donde $n(x_{ij} \wedge c)$ es la cantidad de conjuntos con presencia del rasgo x_{ij} (para el caso de x_{ij} = verdadero) simultáneamente con la clase c.

$n(c)$ es la cantidad de instancias con clase c y ϕ_i es el número de valores diferentes que puede tomar el rasgo x_{ij}, dos en este caso.

Ahora consideremos el caso en que un conjunto esté expresado mediante una n-upla binaria. En el capítulo 1 se analizó cómo calcular la probabilidad de una conjunción de atributos (Fórmula 2.5).

$$P(x_1, x_2, ..., x_n \mid v_j) = \prod_i P(x_i \mid v_j). \qquad (2.5)$$

Si se aplica esta ecuación para el caso en que los x_i son los elementos del dominio de un conjunto S y los valores de x_i son cero ($x_i \in S$) o uno ($x_i \notin S$), se obtiene sin duda la misma expresión de la Fórmula 2.3. Por tanto es posible concluir, que al aplicar el algoritmo 1BC a un dato de tipo conjunto, se le está dando el mismo tratamiento que si descompusiéramos el conjunto en una n-upla binaria. Este algoritmo no aporta una nueva variante para el tratamiento de datos de tipo conjunto.

Algoritmo 1BC2

En el caso del algoritmo 1BC2, este tiene en cuenta la estructura del individuo en el cálculo de las probabilidades. Como los conjuntos están compuestos por colecciones de elementos, se utiliza la distribución geométrica sobre listas (Fórmula 2.6) para calcular el aporte probabilístico de dicha colección.

$$P_{li}([x_{j1}, ..., x_{jl}]) = \tau(1-\tau)^l \prod_{i=1}^{l} P_D(x_{ji}) \qquad (2.6)$$

En [34] se plantea el uso de la ecuación 2.6 incluso en casos donde la colección es un conjunto y no una lista, por lo que es válida su utilización en este caso. Al aproximar un conjunto X mediante una lista, esta última se construye a partir de X, seleccionando un orden arbitrario para sus elementos. A partir de estas consideraciones se plantea la ecuación de la probabilidad condicional de un conjunto X_i dada la clase c y un atributo i de tipo conjunto (Fórmula 2.7). Es necesario aclarar que los valores x_{ij} son exclusivamente los elementos presentes en el conjunto X_i. No ocurre así en 1BC, donde en el cálculo de la probabilidad condicional se consideran todos los valores pertenecientes al domino del

atributo. En esta ecuación l representa la cantidad de elementos del conjunto X_i y τ_c representa la probabilidad condicional de la lista vacía (Fórmula 2.8). En esta última ecuación $\bar{l}_c$ representa el promedio de los cardinales de los conjuntos dada la clase c.

$$P(X_i \mid c) = P_{li}\left([x_{i1}, x_{i2}, ..., x_{il}] \mid c\right) = \tau_c (1-\tau_c)^l \prod_{j=1}^{l} P_D(x_{ij} \mid c) \quad (2.7)$$

$$\tau_c = \frac{1}{1+\bar{l}_c} \quad (2.8)$$

La probabilidad $P_D(x_{ij} \mid c)$ aunque se define mediante la misma ecuación que en 1BC tiene un significado diferente (Fórmula 2.9). La expresión $n(x_{ij} \wedge c)$ representa la cantidad de elementos con valor x_{ij} en los conjuntos, dada la clase c. Debido a que en un conjunto no se repiten elementos, en realidad se están contando los conjuntos que tienen al elemento x_{ij} como miembro. Por tanto el numerador de $P_D(x_{ij} \mid c)$ coincide en 1BC y 1BC2.

$$P(x_{ij} \mid c) = \frac{n(x_{ij} \wedge c) + 1}{n(c) + \phi_i} \quad (2.9)$$

No ocurre así con el denominador, debido a que ϕ_i es el número de elementos diferentes del dominio del atributo y $n(c)$ es la cantidad de elementos presentes en todos los conjuntos, dada la clase c. Esto último equivale a la suma de los cardinales de todos los conjuntos que aparecen simultáneamente con la clase c (Fórmula 2.10, m es el total de instancias presentes en la muestra de entrenamiento).

$$n(c) = \sum_{X_i \in \, Clase\, c}^{m} \left|X_i\right| \quad (2.10)$$

Esta forma de calcular la probabilidad condicional de un conjunto X_i dada una clase c, tiene como ventaja que se toman en cuenta características estructurales de los conjuntos como su tamaño. Además solamente se consideran los elementos presentes en el conjunto, lo cual es un enfoque más natural que el

utilizado en 1BC. Considerando lo anterior, el costo estimado del cálculo de la probabilidad condicional de un conjunto X_i a partir del algoritmo 1BC2, pertenece a un $O(l)$, siendo l es la longitud del conjunto X_i. La principal desventaja que tiene 1BC2 aplicado a conjuntos, es que se descomponen los mismos para calcular el aporte probabilístico de los elementos que lo componen.

Propuesta basada en la similitud entre conjuntos

Retomemos la probabilidad condicional $P(x_i \mid c)$ de que el valor de un atributo sea x_i (Fórmula 2.11).

$$P(x_i \mid c) = \frac{n(x_i \wedge c) + 1}{n(c) + \phi_i} \qquad (2.11)$$

Esta probabilidad es interpretada como la frecuencia de aparición del valor x_i en las instancias con clase c. El cálculo es sencillo, consiste en contar todas las instancias con valor: x_i en el atributo i y clase c simultáneamente. Posteriormente se divide este número por la cantidad de instancias de la clase c. La frecuencia se calcula de esta forma cuando el atributo i es nominal y por tanto existe un número finito de valores que puede tomar. En el caso de los atributos numéricos, contar la cantidad de veces que aparece un valor ya no tiene sentido, puesto que pueden existir infinitos valores. En lugar de contar los valores estos se listan y se calculan estadísticos descriptivos que posteriormente son usados en funciones de densidad probabilística [41].

En el caso de los atributos de tipo conjunto tampoco tiene sentido contar la frecuencia de aparición de cada conjunto diferente en la muestra de entrenamiento, debido a la gran cantidad de conjuntos distintos que pueden aparecer (2^n, donde n es la cantidad de elementos presentes en el dominio del atributo). Además, puede ocurrir que el conjunto presente en una instancia a ser clasificada no haya aparecido nunca en el conjunto de entrenamiento.

Considerando la problemática anterior, se puede dar a los conjuntos un tratamiento que tiene elementos en común con la forma en que se manejan los atributos numéricos y nominales. El mismo consiste en listar todos los conjuntos

presentes en el conjunto de aprendizaje. Esto es todo lo que se hace durante la fase de entrenamiento, el resto ocurre cuando se va a clasificar una nueva instancia. En ese instante se calcula el promedio de las similitudes del nuevo conjunto V_i con los conjuntos presentes en la muestra de entrenamiento, dada la clase c y un atributo i de tipo conjunto (Fórmula 2.11).

$$P(V_i \mid c) = \frac{\sum_{X_i \in \, Clase\, c} \gamma(V_i, X_i) + 1}{n(c) + \phi_i} \qquad (2.11)$$

La función $\gamma(V_i, X_i)$ que calcula la similitud entre dos conjuntos, tiene como valor mínimo cero y máximo uno. Al adicionar estas similitudes se está efectuando una suma de la cantidad de veces que el conjunto V_i aparece en las instancias de entrenamiento, ponderada por la similitud. El pseudo código de esta función se muestra en la Figura 15. El valor de ϕ_i para esta propuesta es el mismo que el utilizado en 1BC2.

```
function ProbCondConjunto(Vi) {
        /*Inicializo las probabilidades */
        For each class c do
                CL[c] = 0;
        /*Itero sobre el conjunto de entrenamiento*/
        For each instance value <Xi, y> do
                CL[y] = CL[y] + γset(Vi, Xi) ;
        /*Itero sobre las clases*/
        For each class c do
                CL[c] = CL[c] / count(c);
        Return CL;
}
```

Figura 15 Pseudo código de la función que calcula la probabilidad condicional $P(V_i \mid c)$.

Volviendo a la expresión de similitud, esta se puede calcular mediante las funciones presentadas en el epígrafe 2.2.2. Anteriormente se utilizaron como distancias y en la Tabla 2 se muestran en forma de similitudes.

Tabla 2 Similitudes entre conjuntos.

Jaccard: $\gamma_{set}(X,Y)=\frac{\lvert X\cap Y\rvert}{\lvert X\cup Y\rvert}$	Czakanowsky-Dice: $\gamma_{set}(X,Y)=1-\frac{\lvert X\Delta Y\rvert}{\lvert X\rvert+\lvert Y\rvert}$
Sokal-Sneath: $\gamma_{set}(X,Y)=\frac{\lvert X\cap Y\rvert}{\lvert X\cup Y\rvert+\lvert X\Delta Y\rvert}$	Cosine: $\gamma_{set}(X,Y)=\frac{\lvert X\cap Y\rvert}{\sqrt{\lvert X\rvert\cdot\lvert Y\rvert}}$
Braun-Blanquet: $\gamma_{set}(X,Y)=\frac{\lvert X\cap Y\rvert}{máx\{\lvert X\rvert,\lvert Y\rvert\}}$	Simpson: $\gamma_{set}(X,Y)=\frac{\lvert X\cap Y\rvert}{mín\{\lvert X\rvert,\lvert Y\rvert\}}$
Kulczynski: $\gamma_{set}(X,Y)=\frac{\lvert X\cap Y\rvert\cdot(\lvert X\rvert+\lvert Y\rvert)}{2\cdot\lvert Y\rvert\cdot\lvert X\rvert}$	

Esta forma de calcular la probabilidad condicional de un conjunto tiene varias ventajas, entre ellas su sencillez y naturalidad. No es necesario descomponer el conjunto para realizar cálculos a partir de sus elementos por separado. El uso de la similitud entre conjuntos ofrece una medida del grado en que el conjunto V_i está presente en el conjunto X_i, tomando valor uno cuando son iguales y cero cuando son disjuntos. Además se pueden utilizar distintas funciones de similitud entre conjuntos. En este trabajo solo se presentan siete pero es posible utilizar otras.

La desventaja del método propuesto es que el procesamiento ocurre en el momento de la clasificación, lo cual es característico de los enfoques perezosos. La complejidad computacional en el momento de la clasificación es un $O(n)$, donde n es el número de instancias del conjunto de aprendizaje.

Algoritmo 1BC2 y propuesta basada en similitud, aplicados a un ejemplo

Se parte de un conjunto de entrenamiento que contiene seis instancias distribuidas de la siguiente forma: tres ejemplos positivos y tres negativos, de una determinada clase. Las instancias describen una representación simplificada de moléculas mediante conjuntos. Cada conjunto está compuesto por los elementos químicos que componen la molécula. El dominio del atributo está

formado por el conjunto: $\{oxígeno, carbono, hidrógeno, nitrógeno\}$. El conjunto de aprendizaje es el siguiente:

$\{o, c, n, h\}$ +

$\{h, o, n\}$ +

$\{o, c, h\}$ +

$\{c\}$ −

$\{h, o\}$ −

$\{o\}$ −

La nueva instancia a ser clasificada tiene la forma: $\{o, c, h\}$?.

Algoritmo 1BC2

En este algoritmo primeramente se construye la tabla de probabilidades por cada rasgo (Tabla 3).

Tabla 3 Probabilidades condicionales de cada rasgo (1BC2).

Rasgos	+	−
mol2atom(M,A),element(A,c)	3/14	2/8
mol2atom(M,A),element(A,h)	4/14	2/8
mol2atom(M,A),element(A,o)	4/14	3/8
mol2atom(M,A),element(A,n)	3/14	1/8
τ	3/13	3/7

El cálculo de $P(h|+) = \frac{n(h \wedge +)+1}{n(+)+\phi_i}$ se realiza del siguiente modo:

$n(h \wedge +)$ = 3, ya que las tres instancias de la clase positiva cumplen que un elemento del conjunto es el hidrógeno.

$n(+)$ = 10, ya que la suma del cardinal de los conjuntos presentes en las instancias positivas arroja dicho valor.

ϕ_i = 4, el cardinal del dominio del atributo.

Finalmente $P(h|+) = \frac{3+1}{10+4} = \frac{4}{14}$. De esta forma se calcula el resto de los rasgos.

$\tau_- = \frac{1}{1+\bar{l}_-}$ se calcula de la siguiente forma:

$\bar{l}_- = \frac{1+2+1}{3} = \frac{4}{3}$, es el promedio del cardinal de los conjuntos presentes en los ejemplos negativos.

Finalmente $\tau_- = \frac{1}{1+\frac{4}{3}} = \frac{3}{7}$, de igual forma se calcula τ_+.

Ahora ya es posible calcular:

$$P_{li}([o,c,h]\mid +) = \tau_+(1-\tau_+)^3 \prod_{i=1}^{3} P_D(s_i \mid +) = \frac{3}{13} * \left(\frac{10}{13}\right)^3 * \frac{4}{14} * \frac{3}{14} * \frac{4}{14} = 0.001837405$$

$$P_{li}([o,c,h]\mid -) = \tau_-(1-\tau_-)^3 \prod_{i=1}^{3} P_D(s_i \mid -) = \frac{3}{7} * \left(\frac{4}{7}\right)^3 * \frac{3}{8} * \frac{2}{8} * \frac{2}{8} = 0.001874219$$

Como la probabilidad a priori de cada clase es la misma, debido a que hay igual cantidad de instancias en ambas clases, no es necesario tenerla en cuenta en este ejemplo. Los resultados obtenidos clasifican la nueva instancia en la clase negativa, lo cual puede parecer ilógico debido a que en la clase positiva existe una instancia idéntica a la instancia clasificada. La explicación está en la forma de calcular $P(x_{ij} \mid c)$, específicamente en la expresión $n(c)$ la cual toma un valor mayor para las instancias positivas debido a la presencia de conjuntos más grandes en este tipo de instancias. Esto redunda en probabilidades más pequeñas en la clase positiva.

Propuesta basada en similitud

La función de similitud que se va a utilizar en este ejemplo es la de Jaccard:

$$\gamma_{set}(X,Y) = \frac{|X \cap Y|}{|X \cup Y|}$$

Como no se estiman probabilidades de antemano, se calcula directamente la probabilidad de la nueva instancia por cada clase:

$$P(\{o,c,h\}\mid +) = \frac{\gamma_{set}(\{o,c,h\},\{o,c,n,h\}) + \gamma_{set}(\{o,c,h\},\{h,o,n\}) + \gamma_{set}(\{o,c,h\},\{o,c,h\})}{n(+) + \phi_i}$$

$$= \frac{\frac{3}{4} + \frac{2}{4} + 1}{3} = 0.75$$

$$P(\{o,c,h\} \mid -) = \frac{\gamma_{set}(\{o,c,h\},\{c\}) + \gamma_{set}(\{o,c,h\},\{h,o\}) + \gamma_{set}(\{o,c,h\},\{o\})}{n(+) + \phi_i}$$

$$= \frac{\frac{1}{3} + \frac{2}{3} + \frac{1}{3}}{3} = 0.44\overline{4}$$

La propuesta que calcula la probabilidad condicional basada en distancias clasifica la instancia en la clase positiva, ya que encontró mayor similitud con los miembros de esa clase.

2.3 Conclusiones parciales

- ✓ Se presenta una propuesta basada en el algoritmo ID3 que permite el tratamiento de atributos de tipo conjunto. Dicha propuesta basada en una modificación al algoritmo de Cohen, ofrece un concepto de test más flexible, que además explota las ventajas de la utilización de conjuntos.
- ✓ Se proponen varias distancias en el contexto del algoritmo k-NN, que manejan atributos de tipo conjunto de una forma sencilla y natural, basadas en funciones de distancia entre conjuntos.
- ✓ Se adapta el algoritmo basado en probabilidades 1BC2 para datos de tipo conjunto y se realiza una propuesta del cálculo de la probabilidad basada en similitudes entre conjuntos.

Capítulo 3: Resultados experimentales

En este capítulo se realiza una validación estadística de las propuestas desarrolladas en comparación con los algoritmos clásicos. Las pruebas se aplican utilizando varios de los estadísticos obtenidos a partir de los experimentos. Los conjuntos de datos utilizados en la experimentación proceden de diversas fuentes y tanto el número de atributos, como su tipo y demás características varían de uno a otro lo que permite obtener resultados más fiables. En el diseño del experimento se utiliza el propuesto por [42].

3.1 Diseño del experimento

A partir de los experimentos se desea conocer el grado de relación entre la forma en que se tratan los atributos de tipo conjunto, con el desempeño de varios algoritmos de clasificación del aprendizaje automatizado. Con este objetivo se evaluarán los algoritmos propuestos sobre varios conjuntos de entrenamiento, incluyendo en el análisis muestras de aprendizaje donde los atributos de tipo conjunto están representados mediante n-uplas binarias. De esta forma se realiza una comparación entre dos variantes referentes al tratamiento de datos de tipo conjunto: la descomposición de los conjuntos en atributos binarios y el tratamiento de los mismos como un tipo de dato.

En esta investigación de tipo experimental se realizarán pruebas horizontales para medir el estímulo de utilizar datos de tipo conjunto [43]. Los individuos de la muestra consisten en conjuntos de entrenamiento correspondientes a diferentes problemas. Este diseño se repetirá para cada enfoque donde se trate el tipo de dato conjunto. Las pruebas utilizadas serán no paramétricas [44] debido a que la cantidad de conjuntos de entrenamiento que conforman la muestra, donde se requiera el uso de conjuntos, no será lo suficientemente grande como para probar normalidad y aplicar pruebas paramétricas.

Las variables del experimento son de tipo cuantitativo ya que se van a manejar los principales estadísticos descriptivos derivados de la aplicación de un algoritmo de clasificación. Las medidas que se utilizan para caracterizar el

desempeño de los algoritmos son: porciento de clasificación correcta, F-Measure y área bajo la curva ROC [45]. Los algoritmos propuestos se implementaron en la herramienta de aprendizaje automatizado Weka. Las variables utilizadas se obtuvieron al calcularse la media de los resultados obtenidos, al efectuar 10 repeticiones con validación cruzada de 10 particiones, para cada algoritmo y cada conjunto de entrenamiento.

Con respecto a las pruebas estadísticas idóneas para este tipo de experimento, en [42], se recomienda utilizar en el caso de la comparación de dos algoritmos sobre una colección de conjuntos de entrenamiento, la prueba no paramétrica de signos de Wilcoxon [46]. En caso que se desee realizar una comparación sobre varios algoritmos simultáneamente [47] entonces se sugiere utilizar la prueba de Friedman [48] o una mejora de esta conocida como prueba de Iman-Davenport [49]. Si se detectan diferencias significativas en el grupo de algoritmos entonces se deben utilizar pruebas pareadas como: el test de Holm [50] y el de Shaffer [51], con el objetivo de detectar dónde se encuentran las diferencias.

3.2 Caracterización de los conjuntos de datos

Se utilizaron en las pruebas 10 conjuntos de datos obtenidos de distintas fuentes. Todos los conjuntos de datos seleccionados tienen al menos un atributo que se puede expresar de forma natural como un conjunto. Los atributos encontrados son de variada naturaleza: algunos representas síntomas, signos, colores, patrones y otros.

En los conjuntos de entrenamiento originales los atributos de tipo conjunto aparecían representados como n-uplas binarias, por lo que fue necesario el uso de filtros para su conversión en conjuntos (*SetToBinary*, *BinaryToSet*). Estos filtros fueron implementados e incluidos en la herramienta de aprendizaje automatizado Weka. Al realizarse esta conversión se redujo sensiblemente la cantidad de atributos presentes en los conjuntos de entrenamiento. La Tabla 4 muestra la cantidad de atributos antes y después de realizarse este proceso.

Tabla 4 Reducción del número de atributos mediante el uso de datos tipo conjunto.

# atributos / Conjuntos de datos	Cantidad original	Cantidad posterior al uso de atributos tipo conjunto
Primary Tumor	18	6
Hepatitis	20	10
Flags	29	17
SPECT	23	2
Voting	17	2
Leptospirosis	93	10
Pneumonia	88	16
Pediatric	193	32
Arrhythmia	280	220
Audiology	70	60

Un grupo importante de estos conjuntos de datos se derivaron de aplicaciones médicas desarrolladas en el Centro de Estudios Informáticos (CEI). El conjunto de datos *Leptospirosis* es una base de conocimiento que permite el diagnóstico de la leptospirosis. *Pneumonia* también se utiliza para el diagnóstico médico en este caso de la neumonía. *Pediatric* permite pronosticar la gravedad de un paciente pediátrico a su llegada al hospital [52]. El resto de los conjuntos de datos se tomaron de la UCIMLR [53]. Para mayor información sobre las características de estos conjuntos de datos consultar Tabla 5.

Tabla 5 Características de los conjuntos de datos.

Conjuntos de datos	Atributos nominales	Atributos numéricos	Atributos de tipo conjunto	Número de clases	Dominio de los atributos
Primary Tumor	5	0	1 (órganos)	22	{médula-ósea, pleura, peritoneo, hígado, cerebro, piel, supraclavicular, axilar, pulmón, óseo, mediastino, abdominal, cuello}
Hepatitis	4	5	1 (síntomas)	2	{fatiga, malestar, anorexia, hígado-aumentado, hígado-endurecido, vaso-palpable, arañas, ascitis, varices, esteroides, antivirales}
Flags	15	2	2 (colores, formas)	6 10 7[5]	{rojo, verde, azul, oro, blanco, negro, naranja} {media-luna, triángulo, icono, animado, texto}
SPECT	1	0	1 (patrones)	2	{F1, F2, F3, F4, F5, F6, F7, F8,…,F22}
Voting	1	0	1 (votos)	2	{niños-discapacitados, proyectos-agua,…, Ley-administración-exportaciones}
Leptospirosis	2	1	7 (síntomas, animales, agua, casa, baño, pruebas, actividades)	2	{síntomas$_1$,…, síntomas $_{26}$} {animales$_1$,…,animales$_{15}$} {agua $_1$,…, agua $_{12}$} {casa$_1$,…, casa $_6$} {baño$_1$,…, baño$_7$} {pruebas$_1$,…, pruebas$_{10}$} {actividades$_1$,…, actividades$_{14}$}
Pneumonia	8	1	7 (síntomas, tiraje, físico, factores, riesgo, antecedentes, frecuencia)	3	{síntomas$_1$,…,síntomas$_{28}$} {tiraje$_1$,…, tiraje$_9$} {físico$_1$,…,físico$_{15}$} {factores$_1$,…, factores$_{10}$} {riesgo$_{1,…,}$ riesgo$_5$} {antecedentes$_1$,…, antecedentes$_{10}$} {frecuencia$_1$,…,frecuencia$_2$}
Pediatric	23	3	6 (estado, diagnóstico, tiraje, estertores, coloración, púrpuras)	2	{sano, acidosis,…, tumor} {adenoiditis, anemia,…, varicela} {subcostal,…, generalizado} {roncos,…, ruidos} {normal, pálida,…, íctero} {petequias, vasculitis}
Arrhythmia	2	206	12 (chDI, chDII, chDIII, chAVR, chAVL, chAVF, chV1, chV2, chV3, chV4, chV5, chV6)	16	{RRwave, DD_RRwave, RPwave, DD_RPwave, RTwave, DD_RTwave}

[5] El número de clases depende del atributo objetivo.

Conjuntos de datos	Atributos nominales	Atributos numéricos	Atributos de tipo conjunto	Número de clases	Dominio de los atributos
Audiology	59	0	1 (history)	24	{buzzing, dizziness, fluctuating, fullness, heredity, nausea, noise, recruitment, ringing, roaring, vomiting}

3.3 Validación de las propuestas realizadas

Los experimentos van a realizarse en el marco de cada enfoque de forma separada. En cada experimento, se comparan los algoritmos que manejan conjuntos como tipo de dato, incluyendo en las comparaciones el algoritmo clásico. Este último no maneja atributos de tipo conjunto por lo cual se le suministra los conjuntos descompuestos en atributos binarios.

3.3.1 ID3

En esta sección se comparan tres versiones del algoritmo ID3: la variante clásica, la extensión propuesta por Cohen y la variación realizada a la propuesta de Cohen donde se generaliza la concepción de la prueba para los atributos de tipo conjunto. Esta última para el caso de l_u = 1 coincide exactamente con la variante clásica de ID3, también se hicieron pruebas para l_u = 2 y l_u = 3.

Debido a que el algoritmo ID3 solo considera atributos nominales y en el caso del ID3 extendido se añaden los de tipo conjunto, fue necesario discretizar los atributos numéricos en aquellos conjuntos de datos donde estuvieran presentes. Este fue el caso de: *flags*, *hepatitis*, *infantil*, *neumocbe* y *arrhythmia* (Para más información consultar Tabla 6).

Tabla 6 Conjunto de atributos numéricos que fueron discretizados.

Conjunto de datos	Atributos numéricos	Cantidad
Flags	Area Population	2
Hepatitis	Age bilirubin alk_phosphate sgot albumin Protime	6
Infantil	Edad TAS TAD	3
Neumocbe	grupo_etario	1
Arrhythmia	height weight … chV6_QRSTA	26

También fue necesario filtrar las instancias con el objetivo de descartar las que contuvieran valores ausentes, ya que el ID3 presenta también esta restricción. Con este fin se utilizó una modificación del filtro *RemoveWithValues* presente en Weka, la cual fue nombrada *RemoveWithMissingValues* y tiene como función eliminar todas las instancias que tengan al menos un valor perdido en uno de sus atributos. Para los conjuntos de datos que al remover las instancias con valores perdidos perdían la representatividad de alguna de sus clases se les aplicó el filtro *ReplaceMissingValues*. Este reemplaza los valores perdidos por la media para los atributos numéricos y por la moda para el caso de los nominales.

La Tabla 7 muestra la media del porciento de clasificación correcta obtenida por cada algoritmo. Para el caso de la extensión al algoritmo de Cohen se muestra además el valor de l_u para el cual se evaluó el algoritmo. En el Anexo 2 se puede consultar esta misma información pero presentada de distinta forma (se muestra solamente el mejor valor obtenido, especificándose para qué valor del parámetro l_u). En el Anexo 3 se pueden consultar otros estadísticos.

Tabla 7 Porciento de clasificación correcta para el enfoque basado en árboles.

Variante ID3 / Datos	ID3 clásico	Cohen	Cohen extendido con $l_u = 2$	Cohen extendido con $l_u = 3$
Flags-landmass	62.129	61.037	62.129	**62.232**
Flags-language	41.795	41.747	**42.624**	**42.624**
Flags-religion	57.924	56.8	**57.976**	**57.976**
Hepatitis	**75.25**	73.125	73.75	73.5
Infantil	88.345	**88.891**	88.364	88.364
Leptospira	78.618	78.564	**79.055**	78.855
Neumocbe	**65.667**	60.083	59.917	59.083
Primary	**34.044**	31.258	33.505	33.505
Spect	74.098	**77.218**	76.33	76.67
Vote	94.357	94.442	93.746	**95.822**
Arrhythmia	31.286	31.286	31.286	31.286
Audiology	**78.049**	67.496	77.96	77.96

En el proceso de obtención de árboles de decisión son preferibles los árboles de tamaño reducido. Considerando lo anterior, otro parámetro que se tuvo en cuenta fue la interpretabilidad resultante de los árboles generados, evaluada a partir de la cantidad de nodos presentes en los mismos. En la Tabla 8 se muestra la media de la cantidad de nodos obtenida para cada conjunto de dato y algoritmo. Es observable cómo el algoritmo de Cohen tiende a incrementar el tamaño de los árboles.

Tabla 8 Enfoque basado en árboles. Cantidad de nodos.

Variante ID3 / Datos	ID3 clásico	Cohen	Cohen extendido con $l_u = 2$	Cohen extendido con $l_u = 3$
Flags-landmass	224.04	239.09	221.07	**219.97**
Flags-language	323.16	330.64	**321.45**	**321.45**
Flags-religion	246.61	273.75	**246.56**	**246.56**
Hepatitis	33.1	43.58	31.95	**28.51**
Infantil	116.45	149.85	**116.23**	**116.23**
Leptospira	38.36	47.26	**29.94**	29.98
Neumocbe	16.49	24.29	**16.13**	16.15
Primary	141.33	171.02	**140.56**	**140.56**
Spect	108.02	124.66	92.72	**91.52**
Vote	26.28	42.42	20.94	**20.12**
Arrhythmia	126.26	126.26	126.26	126.26
Audiology	122.37	156.86	122.37	122.37

La Figura 16 muestra un fragmento del árbol obtenido para el conjunto de datos *Vote.* El conjunto de entrenamiento está compuesto por un único atributo predictor de tipo conjunto que representa el conjunto de votos a favor de cada congresista. En el fragmento del árbol mostrado se observan nodos etiquetados con pruebas que presentan dos elementos en el conjunto S. La existencia de nodos de este tipo justifica la obtención de árboles más pequeños para el algoritmo de Cohen extendido.

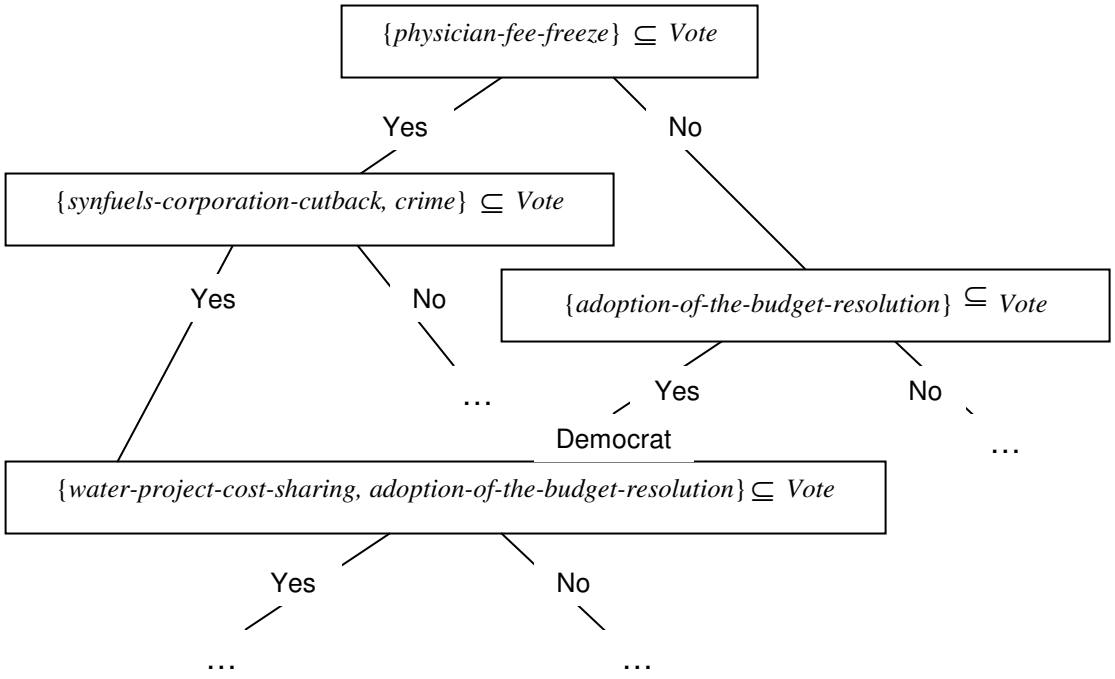

Figura 16 Un árbol de decisión para la clasificación de los congresistas en demócratas y republicanos.

En el análisis estadístico primeramente se aplica un test de Iman-Davenport a los valores de exactitud obtenidos por los algoritmos, no detectándose diferencias significativas en el grupo (valor del test 0.393135). También se aplica este test a los valores obtenidos por el estadístico F-Measure sin detectarse diferencias significativas (valor del test 0.925767). En ambos casos se utiliza un valor de $\alpha = 0.05$.

Con el objetivo de determinar diferencias significativas en el tamaño de los árboles generados por cada algoritmo se aplica el test de Iman-Davenport a la cantidad de nodos promedios de los árboles obtenidos. El test encuentra

diferencias significativas en el grupo de algoritmos (valor del test 3.9895E-9), para un valor de $\alpha = 0.05$. La Tabla 9 muestra los ranking medios asociados a los algoritmos que están siendo evaluados.

Tabla 9 Ranking medios de la cantidad promedio de nodos generados.

Algorithm	Ranking
Cohen extendido (l_u = 3)	2.166667
Cohen extendido (l_u = 2)	2.375
ID3	2.416667
Cohen	3.041667

Posteriormente para detectar específicamente entre qué algoritmos existen diferencias significativas se aplica un test de comparaciones múltiples de Holm, donde se utiliza como muestra de control el algoritmo de Cohen extendido (l_u = 3). Como resultado de este test se encontraron diferencias significativas con respecto al algoritmo de Cohen y al ID3 clásico, para un valor de $\alpha = 0.05$. La Tabla 10 muestra estos resultados.

Tabla 10 Resultados del test de Holm para un valor de confianza de 0.05.

i	*Algoritmo*	$z = (R_0 - R_i)/SE$	P	α / i	*hipótesis*
3	Cohen	4.42718872	9.546919845E-6	0.016667	**R**
2	ID3	2.529822128	0.011412036386	0.025	**R**
1	CohenExt(2)	0.316227766	0.751829634046	0.05	**A**

Como se puede apreciar en la Tabla 10 la hipótesis de igualdad es rechazada (**R**) en las comparaciones del algoritmo de control Cohen extendido (l_u = 3) contra el algoritmo ID3 y el de Cohen. En ambos casos se puede apreciar como el valor-p ajustado (columna P) es menor que su respectivo valor crítico (columna α / i). De lo anterior se deduce que el algoritmo de control obtiene árboles con una cantidad de nodos significativamente menor que los otros dos algoritmos. No siendo así con respecto algoritmo de Cohen extendido (l_u = 2) donde se acepta la hipótesis de igualdad (**A**).

Para complementar lo anterior se realizaron comparaciones múltiples de todos los pares de algoritmos utilizando las pruebas de Holm y Shaffer. La Tabla 11 muestra los resultados obtenidos. Ambas pruebas coinciden en que para el caso del algoritmo de Cohen extendido con (l_u = 2) en comparación con el algoritmo original de Cohen, se rechaza la hipótesis de igualdad.

Tabla 11 Test de Holm y test de Shaffer para un valor de confianza de 0.05.

i	*Algoritmo*	$z = (R_0 - R_i)/SE$	*P*	*Holm*	*Shaffer*	*Hipótesis*
6	Cohen vs. CohenExt(3)	4.4271887242	9.5469198E-6	0.008333	0.008333	**R**
5	Cohen vs. CohenExt(2)	4.1109609582	3.9401582E-5	0.01	0.016666	**R**
4	ID3 vs. CohenExt(3)	2.5298221281	0.0114120363	0.0125	0.016666	**R**
3	ID3 vs. CohenExt(2)	2.2135943621	0.0268566955	0.016666	0.016666	**A**
2	ID3 vs. Cohen	1.8973665961	0.0577795711	0.025	0.025	**A**
1	CohenExt(2) vs. CohenExt(3)	0.3162277660	0.7518296340	0.05	0.05	**A**

A partir de los resultados anteriores se concluye que no existen diferencias significativas en cuanto a la exactitud de la clasificación en el grupo de algoritmos evaluados. Por lo tanto podemos afirmar que las extensiones propuestas tienen un comportamiento competitivo en comparación con el ID3 y el algoritmo de Cohen.

En relación al tamaño de los árboles generados sí se encontró diferencias significativas a favor de las extensiones al algoritmo de Cohen propuestas con respecto al algoritmo de Cohen original. Dichas extensiones obtienen árboles con una mayor interpretabilidad que el algoritmo de Cohen y para (l_u = 3) el algoritmo llega incluso a superar al ID3 clásico.

3.3.2 *k*-NN

En esta sección se validan las propuestas de distancias basadas en la función HEOM. La diferencia entre cada propuesta es la función de distancia que se utiliza para comparar datos de tipo conjunto. Como grupo de control se utiliza la función HEOM clásica. Estos conjuntos de datos se clasificaron mediante el algoritmo k-NN utilizando en todos los casos un vecino más cercano. En la Tabla

12 se muestra la media del porciento de clasificación correcta obtenida por cada conjunto de datos para cada una de las distancias. En el Anexo 4 se muestran otros estadísticos. La Tabla 13 contiene las medias de las áreas bajo la curva ROC obtenidas.

Tabla 12 Porciento de clasificación correcta para el enfoque perezoso.

Conjuntos De datos	Jaccard	Braun-Blanquet	Cosine	Dice	Kulczynski	Simpson	Sokal-Sneath	HEOM clásica
Landmass	64.59	**64.7**	39.67	64.64	39.62	39.46	64.49	57.84
Language	49.37	48.85	30.92	49.47	30.92	30.62	**49.73**	46.29
Religión	**58.26**	57.99	38.59	**58.26**	38.59	39.01	58.16	55.33
Hepatitis	83.84	**84.8**	83.57	83.45	82.53	80.05	83.98	80.25
Infantil	89.15	89.15	86.47	**89.27**	86.57	87.02	89	88.61
Leptospira	**92.25**	91.95	91.86	91.07	91.08	89.89	91.26	92.06
Neumocbe	66.5	**67.6**	66	67	65.1	63.8	65.1	58.3
P. tumor	37.64	36.49	37.43	37.64	38.23	**45.97**	37.64	39.91
Spect	**78.78**	77.54	71.41	**78.78**	70.8	74.89	**78.78**	76.86
Vote	93.41	93.27	93.45	93.41	**93.91**	93.5	93.41	92.58
Arrhythmia	54.087	54.087	54.197	54.197	54.197	**54.219**	54.021	54.087
Audilogy	76.35	76.35	76.35	76.35	76.35	76.881	76.35	**78.431**

Los resultados de la Tabla 13 muestran cómo las variantes ganadoras son en la mayoría de los casos las que consideran los conjuntos como un tipo de dato. La distancia HEOM clásica, la cual se utiliza con los conjuntos descompuestos en n-uplas binarias, obtuvo un mejor resultado del área bajo la curva ROC solamente en 4 conjuntos de datos.

Tabla 13 Área bajo la curva ROC para el enfoque perezoso.

Conjuntos de datos	Jaccard	Braun-Blanquet	Cosine	Dice	Kulczynski	Simpson	Sokal-Sneath	HEOM
Landmass	0.749	0.749	0.752	0.749	0.752	0.738	0.749	**0.797**
Language	0.785	0.785	0.785	0.785	0.785	0.784	0.795	**0.834**
Religión	0.769	0.766	0.769	0.769	0.768	**0.774**	0.769	0.763
Hepatitis	0.734	**0.761**	0.76	0.757	0.744	0.685	0.735	0.694
Infantil	0.89	0.89	0.892	0.892	0.893	**0.901**	0.889	0.887
Leptospira	0.823	0.806	0.799	0.794	0.783	0.73	0.799	**0.865**
Neumocbe	0.713	**0.715**	0.702	0.712	0.691	0.684	0.707	0.632
P. tumor	0.787	0.789	0.793	0.787	0.795	**0.86**	0.787	0.802
Spect	0.699	0.733	0.689	0.699	0.696	0.523	0.699	**0.737**
Vote	0.961	0.974	0.961	0.961	0.964	**0.98**	0.961	0.96
Arrhythmia	**0.619**	**0.619**	**0.619**	**0.619**	**0.619**	**0.619**	**0.619**	0.616
Audilogy	0.615	0.615	0.615	0.615	0.615	**0.637**	0.615	0.631

A continuación se presenta un análisis estadístico para validar los resultados obtenidos. Para este enfoque la cantidad de conjuntos de entrenamiento no es suficiente para obtener resultados confiables, al aplicar un test de comparaciones múltiples de k muestras relacionadas, considerando la cantidad de algoritmos que se evalúan. Por este motivo se decide aplicar un test de signos de Wilcoxon con el objetivo de detectar diferencias significativas entre el algoritmo HEOM clásico y cada una de las propuestas presentadas.

La Tabla 14 muestra los resultados del test Wilcoxon aplicado a los valores de la exactitud. Como resultado de este test se encontró diferencias significativas con respecto a la distancia de Jaccard, para un $\alpha = 0.05$. El test de Wilcoxon aplicado a otros estadísticos como área ROC y F-Measure no arrojó diferencias significativas.

Tabla 14 Comparación de la distancia HEOM con el resto (test de Wilcoxon).

	HEOM – Sokal	HEOM – Simpson	HEOM – Kulczynski	HEOM – Dice	HEOM – Cosine	HEOM – Braun	HEOM – Jaccard
Z	-1.726(a)	-1.412(b)	-1.490(b)	-1.726(a)	-1.490(b)	-1.689(a)	-1.956(a)
Asymp. Sig. (2-tailed)	.084	.158	.136	.084	.136	.091	.050

a Based on positive ranks.

B Based on negative ranks.

C Wilcoxon Signed Ranks Test

Los resultados experimentales muestran que la distancia que mejor desempeño obtuvo es la que utiliza la distancia de Jaccard para la comparación de conjuntos. Considerando los resultados alcanzados al utilizar los tres estadísticos mencionados anteriormente (porciento de clasificación correcta, área ROC y F-Measure), se concluye que las siete distancias propuestas tienen un comportamiento competitivo con respecto a la distancia HEOM clásica.

3.3.3 Naive Bayes

En esta sección se evalúan las propuestas obtenidas para el cálculo de la probabilidad para atributos de tipo conjunto. La primera consiste en la adaptación del algoritmo 1BC2 para conjuntos. La segunda es la propuesta basada en la similitud. En esta última se utiliza para calcular la similitud entre conjuntos el coeficiente de similitud de Jaccard, por ser la medida que mejor desempeño obtuvo en el enfoque perezoso. En esta evaluación se incluye además el Naive Bayes clásico.

No se hizo necesario realizar ningún tipo de preprocesamiento sobre los conjuntos de entrenamiento. En la Tabla 15 se muestran las medias del porciento de clasificación correcta obtenida en cada conjunto de datos para cada una de las propuestas. En el Anexo 5 se muestran otros estadísticos como resultado de los algoritmos.

Tabla 15 Porciento de clasificación correcta para el enfoque probabilístico.

Naive Bayes / Datos	Clásico	1BC2	Basado en distancias
Flags-landmass	**63.311**	62.342	63.168
Flags-language	53.942	**57.184**	56.779
Flags-religion	63.082	**64.568**	64.247
Hepatitis	83.808	**84.258**	82.758
Infantil	92.093	93.108	**93.379**
Leptospira	88.527	**89.236**	86.191
Neumocbe	75.6	77.8	**79**
Primary	**49.707**	48.906	46.399
Spect	78.684	77.915	**84.289**
Vote	**90.024**	89.498	88.487
Arrhythmia	**62.397**	62.37	62.109
Audiology	**72.638**	70.567	71.395

Para el análisis estadístico se aplica un test de Iman-Davenport a los valores de exactitud obtenidos por los algoritmos, no detectándose diferencias significativas en el grupo (valor del test 0.793273). También se aplica este test a los valores obtenidos por el estadístico F-Measure sin detectarse diferencias significativas (valor del test 0.981068). En ambos casos se utiliza como valor de confianza $\alpha = 0.05$.

Los resultados anteriores muestran que no existen diferencias significativas en el desempeño de estos algoritmos por lo que podemos considerar las propuestas que manejan conjuntos competitivas respecto al Naive Bayes clásico.

3.4 Conclusiones parciales

A partir de los experimentos y el análisis estadístico realizado se arriban a las siguientes conclusiones:

- ✓ Las extensiones realizadas al algoritmo de Cohen alcanzan una exactitud en la clasificación, competitiva respecto al algoritmo original y al ID3 clásico. Este resultado se obtiene a partir de generar árboles con una

cantidad de nodos significativamente menor que el algoritmo de Cohen y el ID3.

- ✓ Las distancias propuestas basadas en conjuntos obtienen un desempeño similar a la distancia HEOM clásica, obtenido a partir de un número de dimensiones menor.
- ✓ El algoritmo 1BC2, la propuesta probabilística basada en la similitud y el Naive Bayes clásico, obtienen un desempeño similar.

Conclusiones

Los resultados obtenidos permiten concluir que:

- ✓ La extensión de algoritmos representativos del aprendizaje automatizado al tratamiento de los atributos tipo conjunto, permite una considerable reducción de la complejidad espacial del conjunto de entrenamiento y mayor naturalidad en la modelación computacional de este tipo de dato, garantizando al menos un desempeño similar al tratamiento clásico (descomposición en n-uplas binarias).
- ✓ Se obtiene una extensión del algoritmo de Cohen que además mejora significativamente el algoritmo original en cuanto a complejidad de los árboles generados, manteniendo un desempeño competitivo.
- ✓ Se definen siete nuevas funciones de distancia a partir de extender la función HEOM, que utilizadas en el contexto del k-NN muestran un desempeño similar.
- ✓ Se extiende el algoritmo de Naive Bayes al tipo de dato conjunto, a partir del cálculo de la probabilidad condicional basada en la similitud entre conjuntos sin necesidad de estimar probabilidades individuales de los elementos que los componen, el cual alcanza resultados competitivos.
- ✓ Se extiende el ambiente de aprendizaje automatizado Weka al permitir utilizar el tipo de dato conjunto tanto en algoritmos reportados en la literatura como en las nuevas propuestas.

Recomendaciones

Derivadas del estudio realizado, así como de las conclusiones de este trabajo, se recomienda:

1. Mostrar las facilidades de las extensiones propuestas para resolver problemas reales del entorno.
2. Extender el uso del dato tipo conjunto a otras tareas del aprendizaje automatizado, tales como: selección de rasgos y reglas de asociación.

Referencias Bibliográficas

1. **Ruiz-Shulcloper, J., *Reconocimiento Lógico Combinatorio de Patrones: teoría y aplicaciones*, in *Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV)*. 2009: Ciudad de La Habana, Cuba.**
2. **Cohen, W.W. *Learning Trees and Rules with Set-valued Features*. in *Proceedings of the Thirteenth National Conference on Artifcial Intelligence. (AAAI-96)*. 1996.**
3. **Payne, T.R., *Dimensionality Reduction and Representation for Nearest Neighbour Learning*. 1999, University of Aberdeen.**
4. **Devlin, K.J., *The Joy of Sets*. 1993, Springer-Verlag. p. 1-7.**
5. **Witten, I. and D. Frank, *Data Mining Practical Machine Learning Tools 2*. 2005: Elsevier.**
6. **Berry, M.J.A. and G.S. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. second ed. 2004, Indianapolis, Indiana: Wiley.**
7. **Mitchell, T., *Machine Learning*. 1997: McGraw Hill.**
8. **Larose, D.T., *Discovering knowledge in data : an introduction to data mining*. 2005, Hoboken, New Jersey.: John Wiley & Sons,.**
9. **Ruiz-Shulcloper, J. and A. Fuentes-Rodríguez, *Un modelo cibernético para el análisis de la delicuencia juvenil.* Revista Ciencias Matemáticas 1981. II(1): p. 141-153.**
10. **Ruiz-Shucloper, J., et al., *Modelación Matemática del Problema de Discriminación de Anomalías AGE Perspectivas para Rocas Fosfóricas de Génesis Sedimentaria* Revista Ciencias Matemáticas, 1992. 13(2): p. 159-171.**
11. **Soñora, E., Y. Rodríguez, and M.M. García, *Sistema Experto para el pronóstico de Leptospirosis*. 1998, UCLV: Santa Clara.**
12. **Hernández, D., Y. Rodríguez, and M.M. García, *Sistema Experto para el diagnóstico de Pneumonía en niños*. 1998, UCLV: Santa Clara.**
13. **Rodríguez, Y. and M.M. García, *Sistema Basado en Casos para el diagnóstico de malformaciones cardiovasculares en recién nacidos*. 1996, UCLV: Santa Clara.**
14. **Lewis, D. and M. Ringuette. *A comparison of two learning algorithms for text categorization*. in *Symposium on Document Analysis and Information Retrival*. 1994. Las Vegas, Nevada.**
15. **Wolpert, D.H. and W.G. Macready, *No Free Lunch Theorems for Search.* IEEE Transactions on Evolutionary Computation, 1997. 1.**
16. **Schaffer, C. *A Conservation Law for Generalization Performance*. in *Proceedings of the Eleventh International Conference on Machine Learning (ICML-94)*. 1994.**
17. **Vilalta, R., D. Oblinger, and I. Rish, *What works well where in inductive learning?* 2000.**

18. **Alcalá-Fdez, J., et al., *KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems.* Soft Computing, 2009. 13(3): p. 307-318.**
19. **Demsar, J., B. Zupan, and G. Leban, *Orange: From Experimental Machine Learning to Interactive Data Mining*. 2004, Faculty of Computer and Information Science, University of Ljubljana.**
20. **Mierswa, I., et al. *YALE: Rapid Prototyping for Complex Data Mining Tasks*. in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06)*. 2006.**
21. **Payne, T. and P. Edwards, *Interface Agents that Learn: An Investigation of Learning Issues in a Mail Agent Interface.* Applied Artificial Intelligence, 1997. 11(1): p. 1-32.**
22. **Beyer, K., et al., *When is "nearest neighbor" meaningful?* Lecture Notes in Computer Science, 1999. 1540: p. 217–235.**
23. **Hughes, G., *On the mean accuracy of statistical pattern recognizers.* Information Theory, IEEE Transactions, 1968. 14(1): p. 55–63.**
24. **Scott, D. and J. Thompson. *In:Computer Science and Statistics: Probability density estimation in higher dimensions*. in *Fifteenth Symposium on the Interface*. 1983. North-Holland.**
25. **Corporation, T.C., *Introduction to Data Mining and Knowledge Discovery*. Third Edition ed. 1999.**
26. **Quinlan, J.R., *Induction of Decision Trees.* Machine Learning, 1986. 1: p. 81-106.**
27. **Shannon, C., *Mathematical Theory of communication.* Bell System Technical Journal, 1948. 27: p. 379-423 y 623-656.**
28. **Cover, T.M. and P.E. Hart, *Nearest Neighbor Pattern Classification.* IEEE Transactions on Information Theory, 1967. 13: p. 21-26.**
29. **Aha, D. and D. Kibler, *Instance-based learning algorithms.* Machine Learning, 1991. 6: p. 37-66.**
30. **Randall Wilson, D. and T.R. Martinez, *Improved Heterogeneous Distance Functions.* Journal of Artificial Intelligence Research, 1997. 6(1): p. 1-34.**
31. **Stanfill, C. and D. Waltz, *Toward memory-based reasoning.* Communications of the ACM, 1986. 29(12): p. 1213-1238.**
32. **Zhang, H. *The Optimality of Naive Bayes*. in *The 17th Internacional FLAIRS Conference*. 2004. Miami Beach, Florida.**
33. **Michie, D. and D.J. Spiegelhalter, *Machine learning, neural and statistical classification*. 1994, New York: Ellis Horwood.**
34. **Flach, P.A. and N. Lachiche, *Naive Bayesian Classification of Structured Data.* Machine Learning, 2004. 57: p. 233-269.**

35. **Flach, P.A. and N. Lachiche, *Confirmation-guided discovery of first-order rules with tertius.* Machine Learning, 2001. 42: p. 61-95.**
36. **González, M. and M. Espinosa, *Sistema pronóstico del paciente pediátrico*, in *Centro de Estudios Informáticos*. 2007, UCLV: Santa Clara.**
37. **González, M., Y. Rodríguez, and C. Morell. *Estudio del algoritmo k-NN en presencia de atributos de tipo conjunto*. in *COMPUMAT 2009*. 2009. La Habana.**
38. **González, M., Y. Rodríguez, and C. Morell. *KNN behavior with set-valued attributes*. in *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. 2010. Bruges, Belgium.**
39. **Deza, E. and M.M. Deza, *Dictionary of Distances*. 1 ed. 2006: Elsevier.**
40. **González, M., Y. Rodríguez, and C. Morell. *Estudio comparativo de distancias entre conjuntos en el contexto de algoritmo k-NN*. in *Jornada Científica Juvenil ICIMAF-2010*. 2010. La Habana.**
41. **John, G.H. and P. Langley. *Estimating Continuous Distributions in Bayesian Classifiers*. in *Eleventh Conference on Uncertainly in Artificial Intelligence*. 1995. San Mateo.**
42. **Demsar, J., *Statistical Comparisons of Classifiers over Multiple Data Sets.* Journal of Machine Learning Research, 2006. 7: p. 1-30.**
43. **Grau, R., C. Correa, and M. Rojas, *Metodología de la Investigación*. Segunda Edición ed. 2004, Ibagué, Tolima: El POIRA Editores S. A.**
44. **Sheskin, D.J., *Handbook of parametric and nonparametric statistical procedures*. Third Edition ed. 2004: Chapman & Hall/CRC.**
45. **Ye, N., *The handbook of data mining*. 2003: Lawrence Erlbaum Associates, Inc.**
46. **Wilcoxon, F., *Individual comparisons by ranking methods.* Biometrics, 1945. 1: p. 80-83.**
47. **García, S. and F. Herrera, *An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons.* Journal of Machine Learning Research, 2008. 9: p. 2677-2694.**
48. **Friedman, M., *A comparison of alternative test of significance for the problem of m rankings.* Annals of Mathematical Statistics, 1940. 11: p. 86-92.**
49. **Iman, R.L. and J.M. Davenport, *Approximations of the critical region of the Friedman statistic.* Communications in Statistics, 1980: p. 571-595.**
50. **Holm, S., *A simple sequentially rejective multiple test procedure.* Scandinavian Journal of Statistics, 1979. 6: p. 65-70.**
51. **Shaffer, J.P., *Multiple hypothesis testing.* Annual Review of Psychology, 1995. 46: p. 561-584.**

52. **Rodríguez, Y., et al., *Prediction of Pediatric Risk Using a Hybrid Model Based on Soft Computing Techniques*, in *MICAI 2008: Advances in Artificial Intelligence*. 2008, Springer Berlin / Heidelberg. p. 472-481.**

53. **Asuncion, A. and D.J. Newman. *UCI Machine Learning Repository*. 2007 [cited; Available from: http://www.ics.uci.edu/~mlearn/MLRepository.html.**

Anexos

Anexo 1: Ejemplo de declaración de un atributo de tipo conjunto en un fichero de entrada del Weka, de tipo arff.

```
@relation colores

@attribute colores set {rojo, azul, amarillo}
@attribute class {0, 1}

@data
rojo | amarillo, 0
rojo, 1
Ø, 0
```

Anexo 2: Comparación del porciento de clasificación correcta para el enfoque basado en árboles.

Variante ID3 / Datos	ID3 clásico	Cohen	Cohen extendido	Valor de l_u
Flags-landmass	62.129	61.037	**62.232**	3
Flags-language	41.795	41.747	**42.624**	2
Flags-religion	57.924	56.8	**57.976**	2
Hepatitis	**75.25**	73.125	**75.25**	1
Infantil	88.345	**88.891**	88.364	2
Leptospira	78.618	**78.564**	79.055	2
Neumocbe	**65.667**	60.083	**65.667**	1
Primary	**34.044**	31.258	**34.044**	1
Spect	74.098	**77.218**	76.67	3
Vote	94.357	94.442	**95.822**	3
Arrhythmia	31.286	31.286	31.286	1
Audiology	**78.049**	67.496	**78.049**	1

Anexo 3: Comparación del enfoque basado en árboles (F-Measure).

Variante ID3 / Datos	ID3 clásico	Cohen	Cohen extendido con $I_u = 2$	Cohen extendido con $I_u = 3$
Flags-landmass	0.704	**0.706**	0.705	0.705
Flags-language	0.639	0.636	**0.658**	**0.658**
Flags-religion	0.666	0.65	**0.667**	**0.667**
Hepatitis	**0.236**	0.217	0.202	0.188
Infantil	0.939	**0.946**	0.939	0.939
Leptospira	**0.535**	0.45	0.479	0.473
Neumocbe	0.178	**0.211**	0.1	0.083
Primary	0.686	0.492	**0.691**	**0.691**
Spect	0.449	0.475	0.483	**0.487**
Vote	0.946	0.947	0.94	**0.961**
Arrhythmia	0.413	0.413	0.413	0.413
Audiology	0	0	0	0

Anexo 4: Comparación del enfoque perezoso (F-Measure).

Conjuntos de datos	Jaccard	Braun-Blanquet	Cosine	Dice	Kulczynski	Simpson	Sokal-Sneath	HEOM
Landmass	0.567	0.567	0.576	0.567	0.576	0.546	0.567	**0.622**
Language	0.651	0.651	0.651	0.651	0.651	0.639	0.663	**0.717**
Religión	**0.626**	0.62	**0.626**	**0.626**	0.624	0.624	**0.626**	0.599
Hepatitis	0.57	**0.609**	0.601	0.595	0.569	0.48	0.574	0.497
Infantil	0.9	0.9	0.901	0.901	0.902	**0.905**	0.898	0.895
Leptospira	0.757	0.728	0.725	0.707	0.697	0.628	0.716	**0.799**
Neumocbe	0.649	**0.655**	0.638	0.645	0.624	0.62	0.64	0.553
P. tumor	0.618	0.605	0.623	0.618	0.633	**0.683**	0.618	0.643
Spect	0.475	0.478	0.472	0.475	0.455	0.421	0.475	**0.487**
Vote	0.946	0.944	0.946	0.946	**0.95**	0.946	0.946	0.938
Arrhythmia	**0.703**	**0.703**	**0.703**	**0.703**	**0.703**	**0.703**	**0.703**	0.701
Audilogy	0	0	0	0	0	0	0	0

Anexo 5: Comparación del enfoque probabilístico (F-Measure).

Naive Bayes / Datos	Clásico	1BC2	Basado en distancias
Flags-landmass	0.597	0.597	**0.621**
Flags-language	0.678	0.708	**0.721**
Flags-religion	0.708	0.733	**0.738**
Hepatitis	0.641	**0.647**	0.579
Infantil	0.922	0.932	**0.935**
Leptospira	0.624	**0.632**	0.408
Neumocbe	0.749	0.767	**0.782**
Primary	**0.701**	0.698	0.65
Spect	**0.595**	0.015	0.442
Vote	**0.916**	0.912	0.907
Arrhythmia	**0.773**	0.772	0.77
Audiology	0	0	0