G. Ramos, and J. Lopez, "ID3f+A. Algoritmo de Aprendizaje Inductivo Borroso con División Intervalar Automática de los Atributos", *VI Congreso Español Sobre Tecnologías y Lógica Fuzzy (ESTYLF96)*, pp. 225-230, 1996.
NICS Lab. Publications: https://www.nics.uma.es/publications

ID3*f*+A
ALGORITMO DE APRENDIZAJE INDUCTIVO BORROSO CON DIVISIÓN INTERVALAR AUTOMÁTICA DE LOS ATRIBUTOS

Gonzalo Ramos Jiménez
Dpto. Lenguajes y Ciencias de la Computación
E.T.S. de Ingeniería Informática
Universidad de Málaga
Aptdo. 4114, 29080-Málaga (SPAIN)
Tlfno: 95-2132725, FAX: 95-2131397
e-mail: ramos@lcc.uma.es

Javier López Muñoz
Dpto. Lenguajes y Ciencias de la Computación
E.T.S. de Ingeniería Informática
Universidad de Málaga
Aptdo. 4114, 29080-Málaga (SPAIN)
Tlfno: 95-2131327, FAX: 95-2131397
e-mail: jlm@lcc.uma.es

Resumen

Uno de los campos más prometedores dentro del estudio de la ambigüedad es el del aprendizaje, tanto por su importancia consustancial como por su relación con la Inteligencia Artificial. Esta relación se hace evidente cuando intentamos resolver, desde una perspectiva borrosa, el problema de la adquisición automática del conocimiento en los sistemas expertos. El algoritmo ID3, el más relevante de los utilizados para la inducción de árboles de decisión, no es utilizable tal cual con un enfoque borroso del concepto de pertenencia. Además se muestra ineficiente cuando no existe un experto humano que defina correctamente los subrangos de actuación para los atributos, que junto a las clases expresan las relaciones entre situaciones que este algoritmo de aprendizaje intenta descubrir. Proponemos como solución un nuevo algoritmo, el ID3f+A, que posee la capacidad de tratamiento borroso del concepto de pertenencia, gracias a una modificación del concepto de entropía, y además realiza la división intervalar automática de los atributos, merced al control del proceso inductivo por medio de la utilización de experiencias de control.

Palabras Clave: fuzzy, aprendizaje, inducción, ID3, ID3*f*+A, árbol de decisión.

1. INTRODUCCIÓN

El aprendizaje empírico a partir de experiencias ha recibido una considerable atención en términos de investigación y aplicaciones industriales. Los programas que aprenden de ejemplos preclasificados (experiencias) intentan evitar el cuello de botella que supone la adquisición de conocimiento en el desarrollo de sistemas expertos [3]. Los algoritmos inductivos de aprendizaje [7][9] intentan descubrir las relaciones (reglas) entre las situaciones, expresándolas en términos de un conjunto de atributos, y las clases. El inconveniente que surge es que la ambigüedad es inherente a la realidad [13][16], haciendo que sea difícil una representación correcta de las experiencias y de las reglas en términos no ambiguos.

El clásico algoritmo ID3 presentado por Quinlan [10][11] es un método simple y eficiente para inducir *Árboles de Decisión* a partir de conjuntos de experiencias previamente asignadas a determinadas clases. A pesar de las indiscutibles ventajas y de la facilidad de uso del ID3, este presenta dos limitaciones. La primera, que los posibles valores de todos los atributos que el algoritmo vaya a utilizar han de estar perfectamente definidos antes de que dicho algoritmo empiece a funcionar. Por ello es de suponer que existe un experto humano que basándose en su conocimiento y, a partir del conjunto de experiencias existente, va a delimitar los valores de cada uno de los atributos; es decir, va a indicar cuáles son los valores significativos de cada atributo dentro del rango de valores discretos que estos pueden adquirir. Pero no siempre es posible disponer de un experto humano que realice dicha valoración. La segunda, que el ID3 obliga a que cada experiencia pertenezca por completo a una sola clase, tanto las experiencias clasificadas de partida, como la predicción realizada

para una experiencia nueva sin clasificar; es decir, el ID3 no admite ambigüedad.

El algoritmo ID3*f*+A que presentamos amplía al ID3 de forma que soluciona estas dos limitaciones ("*f* ", "+A"). Para conseguir que las experiencias iniciales utilizadas en el proceso de aprendizaje inductivo puedan pertenecer simultáneamente, con cierto grado borroso [14], a distintas clases; y además las reglas obtenidas tras la inducción sean borrosas, es decir, realicen predicciones sobre el grado de pertenencia a las distintas clases para experiencias no clasificas, utilizamos una modificación del concepto de entropía [12]. Por otra parte, respecto a la segunda limitación, el ID3*f*+A además de identificar el árbol de clasificación a partir de las citadas experiencias borrosas es capaz, también, de encontrar los valores que delimitan los subrangos en los que dividir el espectro de valores de cada uno de los atributos del conjunto de experiencias. Ello es posible gracias a la división (exclusiva de nuestro algoritmo) del conjunto de experiencias en otros dos subconjuntos, un subconjunto de *clasificación* y un subconjunto de *control*. El primero lo usaremos para construir el árbol de decisión y el segundo para controlar dicho proceso de construcción.

En un trabajo anterior [1] L.I. Arranz presentó una modificación al ID3, pero sin alterar el concepto de entropía, por lo que sus grados de pertenencia debían sumar uno, lo cual era un enfoque más probabilístico que borroso. Además no realizaba la división intervalar automática de los atributos propia del ID3*f*+A.

En la Sección 2 se realiza la definición del problema de búsqueda de un árbol de clasificación y se definen los conceptos que necesitaremos, y en la Sección 3 se presenta nuestro algoritmo ID3*f*+A que da solución a las limitaciones mencionadas.

2. DEFINICIÓN DEL PROBLEMA

Un árbol de decisión [2][15] es una representación de un procedimiento de decisión para determinar la clase de una determinada instancia. Cada nodo hoja de dicho árbol de decisión tiene asociada una regla, definida por el recorrido por el árbol desde el nodo raíz hasta el nodo hoja que estamos considerando. Un árbol de este tipo es el que deseamos construir, para lo cual primero debemos delimitar el problema.

Disponemos de un conjunto de experiencias, E, cada una de las cuales viene definida por unos valores para unos atributos, $A, B, C, \ldots$, y un grado de pertenencia [4][17] a cada una de las p clases posibles a priori. De esta forma, para una experiencia concreta j perteneciente a E, tendremos:

$$j: A_m, B_n, C_s, \ldots / X_1, X_2, \ldots, X_p$$

donde $A_m, B_n, C_s, \ldots$ son los valores de los atributos $A, B, C, \ldots$ en la experiencia j, y $X_i \in [0,1]$ es el grado de pertenencia de la experiencia j a la clase i.

Los valores de los atributos han de ser discretos y finitos, y el conjunto de las clases ha de ser finito. El problema consiste en formular un conjunto de reglas que tengan como antecedente valores o intervalos de valores de los atributos y como consecuente grados de pertenencia a cada una de las clases, de forma que este conjunto de reglas sea un "buen" predictor de a qué clases pertenecen futuras experiencias no clasificadas.

Para nuestro algoritmo ID3*f*+A dividiremos el conjunto de experiencias E en dos de tamaño similar, experiencias de clasificación, ECL, y experiencias de control, ECO.

También necesitamos dos índices de control, I_A e I_R, absoluto y relativo respectivamente, que nos indican, para el árbol de clasificación que en cada momento estamos construyendo, el grado de pronóstico acertado que dicho árbol tiene asociado para el conjunto ECO. Para obtener los índices, calculamos, para el absoluto, cuantas experiencias de ECO se clasificarían bien con dicho árbol si consideráramos que cada experiencia es de la clase de mayor grado de pertenencia y tomáramos como pronóstico la clase con mayor grado de pertenencia de la regla asociada a cada experiencia; para el relativo, también consideramos que cada experiencia es de la clase de mayor grado de pertenencia pero en lugar de contabilizar experiencias bien clasificadas sumamos los grados de pertenencia asociados a la clase de cada experiencia que indican sus reglas correspondientes; y en ambos casos dividimos por el cardinal de ECO. Así:

Sea CEB = | {experiencias de ECO bien clasificadas por mayor grado de pertenencia} |

Sea SG = Σ (grados de pertenencia de ECO asociados a reglas)

$$I_A = \frac{CEB}{|ECO|}$$

$$I_R = \frac{SG}{|ECO|}$$

Por supuesto, al ser el ID3*f*+A un algoritmo que engloba al ID3, utiliza la idea de la entropía de la información para la creación del árbol de decisión, pero usando un concepto modificado de entropía. Hartley (1928) [6] fue el antecedente del concepto de

entropía de Shannon (1948) [12], y desde entonces este concepto de entropía ha sido modificado en mayor o menor grado con el fin de obtener distintas medidas de la incertidumbre (funciones U, E, C, V, etc...) [8]. Nuestra modificación, aún manteniendo la forma general de una medida de incertidumbre, es distinta de otras anteriores y permite la utilización de grados de pertenencia.

Para su cálculo usaremos la función Σg que aplicada a un conjunto de experiencias nos devuelve la suma de todos los grados de pertenencia de las experiencias del conjunto. Esta función la utilizaremos como factor de normalización. Así, nuestra entropía H , se define como:

$$H = -\sum_i P_i \log_2 P_i$$

donde

S1 = Σg({conjunto de experiencias de clase i asociadas al nodo})

S2 = Σg({conjunto de experiencias asociadas al nodo})

$$P_i = \frac{S1}{S2}$$

Si se clasifica en ese nodo según un atributo A, se producirán h subconjuntos de experiencias asociadas a los h nodos hijo, en cada uno de los cuales podemos calcular la entropía parcial H_i $(i=1,..,h)$ y a partir de aquí calcular la nueva entropía en el nodo padre gracias a la media ponderada:

$$H(A) = \sum_{i=1}^{h} prob(A_i) \cdot H_i$$

por tanto, el incremento de información aportado por la clasificación según A se evalúa por la disminución producida de nuestra entropía respecto de la inicial del nodo, es decir:

$$\Delta(A) = H - H(A)$$

Por cada nodo hoja del árbol obtendremos una regla que tendrá como antecedente el recorrido por el árbol desde el nodo raíz hasta dicho nodo y como consecuente grados de pertenencia a cada clase que serán la media de los grados respectivos de las experiencias asociadas al nodo.

3. EL ALGORITMO ID3*f*+A

Con la modificación realizada al concepto de entropía hemos incorporado la ambigüedad a nuestro algoritmo, solucionando una de las limitaciones mencionadas. Para resolver la segunda no se puede aplicar directamente la generación del árbol de decisión dividiendo los valores de los atributos, ya que esto provoca una gran cantidad de reglas con muy pocas experiencias (a veces solo una) que la respalden, es decir, reglas muy poco significativas. Por ello surge la necesidad de un método de control del proceso de división intervalar que impida la generación de las reglas poco significativas antes mencionadas. Se propone como método de control la separación en dos conjuntos distintos de experiencias, utilizando uno de ellos para la generación del árbol y el otro para supervisar dicha generación. Al algoritmo surgido a partir de la utilización conjunta de estas dos metodologías, la modificación mostrada del concepto de entropía y el control del proceso de división intervalar de los atributos, lo hemos denominado ID3*f*+A.

Entrada: Un conjunto E de experiencias borrosas dividido en dos de tamaño similar, ECL y ECO.
Salida: Un conjunto de reglas para la predicción borrosa de futuras experiencias no clasificadas.

ID3*f*+A:
- 1/ El nodo raíz tiene asociado los conjuntos ECL y ECO completos.
- 2/ Para cada nodo del árbol no marcado como hoja hacer:
- 2-1/ ELEGIR_ATRIBUTO
- 2-2/ EXPANDIR_NODO

ELEGIR_ATRIBUTO:
- 1/ Para cada atributo libre del nodo hacer:
  1-1/ ELEGIR_PARTICIÓN
- 2/ Si no hay ningún atributo-partición generado marcar nodo como hoja.
- 3/ Si hay algún atributo-partición generado elegimos el que más disminuye la entropía (ECL).

ELEGIR_PARTICIÓN:
- 1/ Calcular I_A e I_R (ECO).
- 2/ Para cada posible partición dicotómica del intervalo del atributo calcular nueva entropía (ECL).
- 3/ Si no hay partición dicotómica que disminuya la entropía no se genera atributo-partición.
- 4/ Si hay, elegir la partición dicotómica que más disminuye la entropía (ECL).
- 5/ Si hay varias particiones con la misma disminución máxima, elegir la partición intermedia.
- 6/ Calcular I_A e I_R con dicha partición (ECO).
- 7/ Si ninguno aumenta o alguno disminuye no se genera atributo-partición.
- 8/ Si ninguno disminuye y alguno aumenta hacer E_P de los intervalos de la partición dicotómica.

E_P:
- 1/ Calcular I_A e I_R (ECO).
- 2/ Para cada posible partición dicotómica del intervalo calcular nueva entropía (ECL).
- 3/ Si no hay partición dicotómica que disminuya la entropía, fin de E_P que devuelve el intervalo.
- 4/ Si hay, elegir la partición dicotómica que más disminuye la entropía (ECL).
- 5/ Si hay varias particiones con la misma disminución máxima, elegir la partición intermedia.
- 6/ Calcular I_A e I_R con dicha partición (ECO).
- 7/ Si ninguno aumenta o alguno disminuye, fin de E_P que devuelve el intervalo.
- 8/ Si ninguno disminuye y alguno aumenta hacer E_P de los intervalos de la partición dicotómica.

EXPANDIR_NODO:
- 1/ Si nodo no marcado como hoja, GENERAR_NODOS
- 2/ Si nodo marcado como hoja, GENERAR_REGLA

GENERAR_NODOS: Para el atributo-partición elegido generamos tantos nodos hijos como intervalos tenga la partición, asociándoles a cada uno los subconjuntos correspondientes de ECL y ECO del nodo.

GENERAR_REGLA: Generamos una regla que tiene como antecedentes los pares atributo-intervalo que llevan desde el nodo raíz hasta el nodo actual, y como consecuente la media de los grados de pertenencia a cada clase del conjunto de experiencias ECL asociadas al nodo. También generamos asociado a la regla el cardinal del conjunto ECL del nodo, que nos indica cuantas experiencias respaldan la regla generada.

En el algoritmo todas las referencias al concepto de entropía se refieren a nuestro concepto de entropía modificada explicada en la sección anterior. Además se indica entre paréntesis qué subconjunto de experiencias, de las asociadas al nodo, utilizamos en cada caso, si las *ECL* o las *ECO*. La división de *E* en estos dos subconjuntos, *ECL* y *ECO*, es, como indicamos, exclusiva del algoritmo ID3*f*+A ; así como en el nivel de procedimientos el de ELEGIR_PARTICIÓN, con todas las operaciones que engloba, también es propio del ID3*f*+A, siendo en dicho procedimiento donde realizamos la división intervalar inteligente de los atributos.

Siguiendo el algoritmo se observan claramente las funciones diferenciadas que tienen *ECL* y *ECO* dentro del ID3*f*+A. Las primeras se utilizan en la creación del árbol de decisión, siempre basada esta creación en nuestro concepto de entropía de la información. El problema que surgiría si utilizásemos el ID3, además de la fundamental imposibilidad de utilizar grados de pertenencia, es que tiende a generar reglas con pocas experiencias que las respalden, ya que busca la completa ordenación de las experiencias disponibles, sin preocuparse de hasta que punto las reglas generadas son realmente abstracciones útiles para la predicción futura o simplemente rozan los casos puntuales sin validez predictiva. Dicho problema se acentúa, como comentamos previamente, cuando intentamos realizar una división intervalar automática de los atributos sin el control que proponemos. En este caso el ID3 tendería a realizar un división de los atributos prácticamente total, lo que no supone realmente ninguna aportación en lo que a la división intervalar automática de los atributos se refiere.

Todos estos problemas, la generación de reglas con pocas experiencias que las respalden, así como la excesiva división de los atributos, se solucionan en el ID3*f*+A mediante el uso de las *ECO* que controlan tanto la división de los atributos como la generación de reglas no significativas (esto último gracias al control de la división) permitiendo solamente dividir cuando realmente ganamos poder predictivo, e impidiendo dicha división si no se produce la mencionada ganancia. Por lo tanto para nuestro algoritmo no importa que tengamos muchos valores posibles para cada atributo, lo que nos lleva a poder utilizarlo con atributos continuos [5], siempre que estén acotados y discretizados con la precisión que deseemos.

A todo esto hay que unir la capacidad de tratamiento borroso del concepto de pertenencia por parte del ID3*f*+A gracias a nuestro concepto de entropía modificada.

Destacar el hecho de que, a diferencia del ID3, en el ID3*f*+A un mismo atributo en distintos nodos puede tener divisiones intervalares diferentes, lo que aumenta su flexibilidad y por tanto su capacidad de inducción.

A destacar también que las reglas que generamos, en GENERAR_REGLA, llevan asociadas el cardinal del conjunto ECL de experiencias que respaldan dicha regla. Esto nos proporciona un "peso" de la regla, un indicativo de su "fiabilidad", que usamos cuando intervienen en una predicción más de una regla. Un ejemplo de esto último se produce cuando el valor de alguno los atributos de una instancia que queremos clasificar nos es desconocido. En este caso ignoramos dicho atributo suponiendo que puede tener cualquier valor, lo que generalmente conlleva que más de una regla se corresponda con nuestra experiencia a clasificar. Cada una de estas reglas nos da un pronóstico, y para obtener un único pronóstico a partir de ellas realizamos una media ponderada de dichos

pronósticos en base a los pesos ya comentados de las reglas. Otro caso en el que necesitamos los pesos asociados a las reglas es si aplicamos el algoritmo ID3*f*+A más de una vez a un conjunto inicial *E* de experiencias. Esto puede ser conveniente debido a que la división de *E* en *ECL* y *ECO* es aleatoria, lo que puede introducir algún sesgo. Hasta que punto se produce dicho sesgo, cuanto puede influir en variar el árbol de decisión generado, y cuantas veces convendría repetir el ID3*f*+A para corregir dicho sesgo, dependerá del tipo de problema tratado y del tamaño de *E*. En las simulaciones que estamos realizando pretendemos obtener respuestas heurísticas a estas cuestiones.

Debido al espacio disponible, no es posible presentar aquí un ejemplo concreto de aplicación paso a paso de nuestro algoritmo. Las pruebas que hemos hecho han sido con muestras de 20 a 100 experiencias, con el objeto de verificar el correcto funcionamiento del algoritmo, y dichas pruebas han resultado satisfactorias. Un resultado relevante a destacar de dichas pruebas es la importancia que adquiere, cuando hay pocas experiencias, el hecho de elegir la partición intermedia cuando hay varias particiones con la misma disminución máxima de entropía. Esto hace que los intervalos seleccionados sean mucho más coherentes que si la elección fuera azarosa.

Respecto a la aplicación del ID3*f*+A a conjuntos grandes de experiencias hay dos aspectos importantes a señalar. El primero es referente al costo, en tiempo de cómputo, de la división intervalar automática, ya que si bien este costo existe, hay que tener en cuenta que dicho costo se produce solamente una vez, cuando el sistema está aprendiendo, o sea, induciendo las reglas. Es claro que este costo no afecta a lo que después será el uso del sistema una vez que ha aprendido, por lo que es perfectamente asumible dicho costo si sólo se ha de producir una vez, durante el aprendizaje inductivo borroso. El segundo aspecto hace referencia a como se produce la división intervalar automática conforme vamos bajando por el árbol de decisión generado cuando la muestra de experiencias es grande. Así, para los atributos superiores del árbol, al tener muchas experiencias asociadas, se tenderá a producir una partición más fina, con intervalos menores, mientras que en los últimos atributos del árbol, estos intervalos serán mayores, agrupando si es posible las pocas experiencias que queden. Esto es razonable ya que los atributos superiores del árbol indican que son más importantes, y por lo tanto una variación mínima en sus valores puede influir mucho, mientras que los atributos menos importantes han de variar sensiblemente para que afecten.

Así pues, hemos resuelto satisfactoriamente los problema: de como incorporar la ambigüedad dentro del proceso inductivo y de como incluir dentro del proceso de aprendizaje la división intervalar automática de los atributos, o dicho en otras palabras, hemos conseguido un algoritmo, el ID3*f*+A, que tiene como objetivo principal, no clasificar mejor un conjunto de experiencias no borrosas dadas, sino predecir mejor experiencias futuras (borrosas o no), que es al fin y al cabo lo que buscamos, predecir.

4. CONSIDERACIONES FINALES

Creemos que la capacidad de tratamiento borroso del concepto de pertenencia, gracias a la modificación introducida del concepto de entropía, y la división intervalar automática de los atributos, merced al control del proceso inductivo por medio de la utilización del conjunto ECO, convierten a nuestro algoritmo ID3*f*+A en el primer algoritmo del tipo *TDIDT (top-down induction of decision trees)* con estas características, lo que le confieren una capacidad predictiva y flexibilidad superiores.

El ID3*f*+A ha sido probado satisfactoriamente con pequeñas muestras de experiencias con objeto de comprobar su correcto funcionamiento. Actualmente estamos trabajando en aplicarlo a varios grupos grandes de experiencias reales para realizar una comparación de resultados entre el ID3*f*+A, el ID3 y otros algoritmos.

Agradecimientos

Queremos agradecer a los Dres. Rafael Morales Bueno y José Luis Pérez de la Cruz Molina sus consejos y apoyo durante la realización de este artículo.

Referencias

[1] Arranz, L. I. (1992). Modificaciones sobre ID3 para el tratamiento de incertidumbre. *II Congreso Español sobre Tecnologías y Lógica Fuzzy:* 193-202. Boadilla del Monte , Madrid.

[2] Buntine, W. y Nibblett, T. (1992). A Further Comparison of Splitting Rules for Decision-Tree Induction . *Machine Learning* 8: 75-85

[3] Cuena, J. (1987). Inteligencia Artificial: Sistemas Expertos. Alianza Editorial.

[4] Dubois, D., Prade, H. (1980). Fuzzy Sets and Systems: Theory and Applications. Academic Press.

[5] Fayyad, U. M. and Irani, K. B. (1992). On the Handling of Continuous-Valued Attributes in Decision Tree Generation. *Machine Learning* 8: 87-102.

[6] Hartley, R. V .L. (1928). Transmission of information. *The Bell Systems Technical Journal* 7: 535-563.
[7] Hunt, E. B., Marin, J. and Stone, P. T. (1996). Experiments in induction. Academic Press.
[8] Klir, G. J. and Folger, T. A. (1988). Fuzzy Sets, Uncertainty, and Information. Prentice-Hall International Limited.
[9] Michalski, R. (1983). Unifying principles and a methodology of Inductive learning. *Artificial Intelligence.*
[10] Quinlan, J. R. (1979). Discovering rules from large collections of examples: a case study. Expert Systems in the Micro Electronic Age. Edinburgh University Press.
[11] Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning* 1: 81-106.
[12] Shannon, C. E. (1948). The mathematical theory of communication. *The Bell Systems Technical Journal* 27: 379-423, 623-656.
[13] Trillas, E.(1980). Conjuntos Borrosos. Vicens-Vives.
[14] Trillas, E. y Gutiérrez Ríos, J. (Editores) y otros (1992). Aplicaciones de la Lógica Borrosa. Consejo Superior de Investigaciones Científicas.
[15] Utgoff, P. E. (1989). Incremental Induction of Decision Trees. *Machine Learning* 4: 161-186.
[16] Zadeh, L. A.(1965). Fuzzy Sets. En Yager, R. y col. (Eds.), 29-44 (1987).
[17] Zadeh, L. A. Fuzzy Sets and Applications.(1987). Selected Papers, editado por R. R. Yager, S. Dvchinnikov, R. M. Tong, H. T. Nguyen Yohn Wiley, Nueva York.