

ALGORITMOS TDIDT

APLICADOS A LA MINERÍA DE DATOS INTELIGENTE

Servente, M.¹ & García-Martínez, R.²

1. Investigadora Asistente del Laboratorio de Sistemas Inteligentes. Facultad de Ingeniería. Universidad de Buenos Aires.
2. Director Adjunto del Programa de Magister en Ingeniería de Software. Escuela de Postgrado. Instituto Tecnológico de Buenos Aires.

1. Introducción

El Aprendizaje Automático (*Machine Learning*) es el campo dedicado al desarrollo de métodos computacionales para los procesos de aprendizaje y a la aplicación de los sistemas informáticos de aprendizaje en problemas prácticos [Michalski y otros, 1998]. La Minería de Datos (*Data Mining*) es la búsqueda de patrones e importantes regularidades en bases de datos de gran volumen [Michalski y otros, 1998]. La Minería de Datos utiliza métodos y estrategias de otras áreas o ciencias, entre las cuales podemos nombrar al Aprendizaje Automático. Cuando este tipo de técnicas se utilizan para realizar la minería, decimos que estamos ante una Minería de Datos Inteligente.

En este trabajo analizamos la aplicación de algunas técnicas de Aprendizaje Automático a la Minería de Datos. Nuestro interés se centró en una familia de métodos de inducción conocida como la familia TDIDT (*Top Down Induction Trees*), y en particular en los algoritmos ID3 y C4.5 desarrollados por Quinlan, pertenecientes a la misma. Se buscó determinar en qué medida los algoritmos de la familia TDIDT pueden usarse en minería de datos para generar modelos válidos en los problemas de clasificación.

Tanto el ID3 como el C4.5 generan árboles y reglas de decisión a partir de datos preclasificados. Para construir los árboles se utiliza el método de aprendizaje “divide y reinarás”, que particiona el conjunto de ejemplos en subconjuntos a medida que avanza; trabajar sobre cada subconjunto es más sencillo que trabajar sobre el total de los datos.

1.1. ID3

EL ID3 o *Induction Decision Trees*, desarrollado en los años ochenta por Quinlan, es un sistema de aprendizaje supervisado que construye árboles de decisión a partir de un conjunto de ejemplos. Estos ejemplos o tuplas están constituidos por un conjunto de atributos y un clasificador o clase. Los dominios de los atributos y de las clases deben ser discretos. Además, las clases deben ser disjuntas. Las primeras versiones del ID3 generaban

descripciones únicamente para dos clases, como ser positiva y negativa. En las versiones posteriores, se eliminó esta restricción, pero se mantuvo la restricción de clases disjuntas. El ID3 genera descripciones que clasifican a cada uno de los ejemplos del conjunto de entrenamiento.

Este sistema tiene una buena performance en un amplio rango de aplicaciones de diversos dominios, como el dominio médico, el artificial y el análisis de juegos de ajedrez. El nivel de precisión en la clasificación generalmente es alto. Sin embargo, el sistema tiene algunas desventajas. Recordemos que los atributos y las clases deben ser discretos y no pueden ser continuos. Además, aún cuando se cuente con conocimientos de dominio o conocimientos previos, el sistema no hace uso de ellos. A veces, los árboles son demasiado frondosos, lo cual conlleva una difícil interpretación. En esos casos pueden ser transformados en reglas de decisión para hacerlos más comprensibles.

1.2. C4.5

El C4.5 es una extensión del ID3 que acaba con muchas de sus limitaciones. Por ejemplo, permite trabajar con valores continuos para los atributos, separando los posibles resultados en dos ramas: una para aquellos $A_i \leq N$ y otra para $A_i > N$. Además, los árboles son menos frondosos porque cada hoja no cubre una clase en particular sino una distribución de clases, lo cual los hace menos profundos y menos frondosos. Este algoritmo fue propuesto por Quinlan en 1993. El C4.5 genera un árbol de decisión a partir de los datos mediante particiones realizadas recursivamente, según la estrategia de profundidad-primero (*depth-first*). Antes de cada partición de datos, el algoritmo considera todas las pruebas posibles que pueden dividir el conjunto de datos y selecciona la prueba que resulta en la mayor ganancia de información o en la mayor proporción de ganancia de información. Para cada atributo discreto, se considera una prueba con n resultados, siendo n el número de valores posibles que puede tomar el atributo. Para cada atributo continuo, se realiza una *prueba binaria* sobre cada uno de los valores que toma el atributo en los datos.

2. Construcción de los árboles de decisión

El ID3 y el C4.5 utilizan la estrategia de “divide y reinarás” para generar el árbol de decisión inicial a partir de un conjunto de datos de entrenamiento. La idea original de esta estrategia se basa en un trabajo de Hoveland y Hunt de los años 50, culminado en el libro *Experiments in Induction* [Hunt y otros, 1966].

El método “divide y reinarás” realiza en cada paso una partición de los datos del nodo según una prueba realizada sobre el “mejor” atributo. Cualquier prueba que divida a T en una manera no trivial, tal que al menos dos subconjuntos distintos $\{T_i\}$ no estén vacíos, eventualmente resultará en una partición de subconjuntos de una única clase, aún cuando la mayoría de los subconjuntos contengan un solo ejemplo. Sin embargo, el proceso de construcción del árbol no apunta meramente a encontrar cualquier partición de ese tipo, sino a encontrar un árbol que revele una estructura del dominio y, por lo tanto, tenga poder predictivo. Para ello, necesitamos un número

importante de casos en cada hoja y cada partición debe tener la menor cantidad de clases posibles. En el caso ideal, nos gustaría elegir en cada paso la prueba que genere el árbol más pequeño.

Entonces, estamos buscando un árbol de decisión compacto que sea consistente con los datos de entrenamiento. Podríamos explorar todos los árboles posibles y elegir el más simple. Desafortunadamente, este espacio de búsqueda o espacio de hipótesis tiene un número exponencial de árboles que deberían ser analizado. El problema de encontrar el árbol de decisión más pequeño consistente con un conjunto de entrenamiento es de complejidad NP-completa.

Para calcular cuál es el mejor atributo en cada partición, se utilizaron tanto la ganancia como la proporción de ganancia. Además, en el caso del C4.5 se podaron los árboles obtenidos utilizando el método descrito en [Quinlan, 1993d], evitando de esta manera, la sobregeneralización.

2.1. Transformación a Reglas de Decisión

Los árboles de decisión demasiado grandes son difíciles de entender porque cada nodo debe ser interpretado dentro del contexto fijado por las ramas anteriores. En cualquier árbol de decisión, las condiciones que deben satisfacerse cuando un caso se clasifica por una hoja pueden encontrarse analizando los resultados de las pruebas en el camino recorrido desde la raíz. Es más, si el camino fuese transformado directamente en una regla de producción, dicha regla podría ser expresada como una conjunción de todas las condiciones que deben satisfacerse para llegar a la hoja. Consecuentemente, todas los antecedentes de las reglas generadas de esta manera son mutuamente excluyentes y exhaustivos.

Para pasar a reglas de decisión, el ID3 recorre el árbol desde la raíz hasta las hojas en preorden (de raíz a hojas, de izquierda a derecha) y genera una regla por cada camino recorrido. El antecedente de cada regla estará compuesto por la conjunción de las pruebas de valor de cada nodo visitado y la clase será la correspondiente a la hoja.

2.3. Evaluación en la familia TDIDT

Para evaluar los árboles de decisión y las reglas de decisión obtenidas, se utilizó un método de evaluación cruzada. Cada conjunto de datos se dividió en dos partes al azar de proporciones 2:3 y 1:3. Se utilizaron entonces, dos tercios de los datos originales para realizar el entrenamiento, y el tercio restante para evaluar los resultados. Se generó una matriz de confusión, en donde, para cada clase se expresaron la cantidad de datos de prueba clasificados correctamente y la cantidad clasificados erróneamente.

3. Pruebas realizadas

Para estudiar los algoritmos propuestos, se desarrolló un sistema que integra el ID3 y el C4.5. El sistema recibe los datos de entrenamiento como entrada y permite que el usuario elija el algoritmo que desea aplicar. Si el usuario elige el ID3, el sistema genera el árbol y las reglas de decisión según dicho algoritmo. Si, en cambio, el usuario elige el C4.5, el desarrollo del sistema es similar: se genera el árbol de decisión según el C4.5, se lo poda y se construyen las reglas de decisión

Además, el usuario puede indicarle al sistema que realice una evaluación de los resultados sobre los datos de prueba. En el caso del ID3, esta evaluación se realiza a partir de las reglas de decisión cuya performance, dada su forma de construcción, es idéntica a la de los árboles. La evaluación de los resultados del C4.5, en cambio, se realiza por separado y se obtienen, por lo tanto, dos evaluaciones distintas, una para el árbol y otra para las reglas.

Se utilizó el sistema desarrollado en siete dominios de datos: Créditos, Cardiología, Votaciones, Elita: base de asteroides, un estudio sobre los hongos, Hipotiroidismo y Vidrios.

4. Resultados

4.1. Comparación de los resultados obtenidos con el ID3 y con el C4.5

La gran diferencia destacable que se encontró entre los resultados obtenidos con ambos algoritmos, fue el tamaño de los árboles de decisión. Como el C4.5 realiza una post-poda los árboles fueron generalmente de menor tamaño que los obtenidos con el ID3. Tomemos, por ejemplo, los árboles obtenidos para el dominio de Votaciones.

En dicho caso, los porcentajes de error obtenidos con el ID3 el porcentaje de rondaron el 5.20%, mientras que con el C4.5, los porcentajes de error se encontraron entre el 3% y el 3.7%, aún cuando los modelos obtenidos con el C4.5 fueron mucho menores en tamaño que los obtenidos con el ID3. Esta diferencia en tamaño se debe a que cada hoja del C4.5 cubre una distribución de casos (aún en los árboles sin simplificar), entonces el árbol resultante es más simple. A continuación se presentan ambos árboles de decisión y se pueden apreciar las simplificaciones hechas por el C4.5.

```
Cong_honorarios_medicos = a_favor
  Reduccion_corp_Synfuels = a_favor
    Export_sin_impuestos = a_favor
      democrata
    Export_sin_impuestos = desconocido
      republicano
  Export_sin_impuestos = en_contra
    Presupuesto_de_educacion = a_favor
      Der_demanda_Superfund = a_favor
        Particip_proy_agua = a_favor
          republicano
        Particip_proy_agua = en_contra
          Acta_sudaf_admin_export = a_favor
            republicano
```

```

Acta_sudaf_admin_export = desconocido
republicano
Acta_sudaf_admin_export = en_contra
Niños discapacitados = a_favor
republicano
Niños discapacitados = en_contra
democrata
Der_demanda_Superfund = en_contra
Democrata (1)
Presupuesto_de_educacion = desconocido
democrata
Presupuesto_de_educacion = en_contra
Acta_sudaf_admin_export = a_favor
Adop_resolucion_presup = a_favor
republicano
Adop_resolucion_presup = en_contra
Ayuda_a_El_Salvador = a_favor
republicano
Ayuda_a_El_Salvador = en_contra
democrata
Acta_sudaf_admin_export = desconocido
democrata
Acta_sudaf_admin_export = en_contra
Democrata (2)
Reduccion_corp_Synfuels = desconocido
republicano
Reduccion_corp_Synfuels = en_contra
Export_sin_impuestos = a_favor
Inmigracion = a_favor
republicano
Inmigracion = en_contra
Acta_sudaf_admin_export = a_favor
democrata
Acta_sudaf_admin_export = desconocido
Particip_proy_agua = a_favor
republicano
Particip_proy_agua = en_contra
democrata
Acta_sudaf_admin_export = en_contra
republicano
Export_sin_impuestos = desconocido
republicano
Export_sin_impuestos = en_contra
Adop_resolucion_presup = a_favor
Acta_sudaf_admin_export = a_favor
republicano
Acta_sudaf_admin_export = desconocido
Niños discapacitados = a_favor
republicano
Niños discapacitados = en_contra
democrata
Adop_resolucion_presup = en_contra
Republicano (3)
Cong_honorarios_medicos = desconocido
Misil_mx = a_favor
Prohib_pruebas_anti_satel = a_favor
democrata
Prohib_pruebas_anti_satel = desconocido
democrata
Prohib_pruebas_anti_satel = en_contra
Republicano (4)
Misil_mx = desconocido
republicano
Misil_mx = en_contra
democrata
Cong_honorarios_medicos = en_contra
Presupuesto_de_educacion = a_favor
democrata
Presupuesto_de_educacion = desconocido
Adop_resolucion_presup = a_favor
democrata
Adop_resolucion_presup = en_contra
republicano
Presupuesto_de_educacion = en_contra
Democrata (5)

```

```

cong_honorarios_medicos = en_contra: demócrata (168.0/1.0) (5)
cong_honorarios_medicos = a_favor:
  reduccion_corp_Synfuels = en_contra: republicano (97.0/3.0) (3)
  reduccion_corp_Synfuels = desconocido: republicano (4.0)
  reduccion_corp_Synfuels = a_favor:
    export_sin_impuestos = a_favor: demócrata (2.0)
    export_sin_impuestos = desconocido: republicano (1.0)
    export_sin_impuestos = en_contra:
      presupuesto_de_educación = a_favor: republicano (13.0/2.0) (1)
      presupuesto_de_educación = en_contra: demócrata (5.0/2.0) (2)
      presupuesto_de_educación = desconocido: demócrata (1.0)
cong_honorarios_medicos = desconocido:
  misil_mx = a_favor: demócrata (4.0/1.0) (4)
  misil_mx = en_contra: demócrata (3.0)
  misil_mx = desconocido: republicano (2.0)

```

En el caso (1), podemos observar que el subárbol de tamaño 10 generado por el ID3, se representó en el C4.5 con una hoja que cubre 13 casos, dos incorrectamente. En el caso (2), el C4.5 presenta una hoja que cubre 5 casos, dos de ellos erróneamente, mientras que el ID3 presenta un subárbol de tamaño 8. En el caso (3), el subárbol presentado por el ID3 es de tamaño 17 y la hoja presentada en el mismo caso por el C4.5 de los 97 casos que cubre, sólo tres son clasificados erróneamente. La diferencia en el caso (4) no es tan notable, ya que el C4.5 representa en una hoja con $N=4$ y $E=2$, lo que el ID3 presenta en un nodo de decisión con tres hojas hijas. Finalmente, en el caso (5), el C4.5 se equivoca una sola vez en los 168 casos que cubre la hoja, mientras que el ID3 los clasifica a todos correctamente con un subárbol de tamaño 6.

El ID3 no generaliza los resultados de una hoja, es decir, no permite que una hoja cubra casos de una clase distinta a la expresada. Por lo tanto, cubre exhaustivamente todos los casos de entrenamiento. Mientras que la generalización realizada por el C4.5 permite obtener árboles más pequeños a un precio no tan alto. Pensemos que, muchas veces es preferible tener una hoja con performance del 96.9%, como en el caso (3), que un árbol de tamaño 17. Este fenómeno que ocurre en los árboles y, como consecuencia lógica, en las reglas generadas con el ID3, se conoce como sobreajuste. Como su nombre lo indica, se origina en que el ID3 cubre absolutamente todos los casos de entrenamiento correctamente. Existen muchas maneras de solucionar el sobreajuste. Podríamos, por ejemplo, realizar una poda del árbol cuando un subárbol tenga una performance mayor a una cota predefinida, es decir, cuando $(E*100)/N$ sea superior a una cota mínima de performance. Otra opción sería realizar esta simplificación y adjuntarle al árbol las reglas de decisión con las excepciones.

Estas diferencias de tamaño entre los modelos obtenidos para el dominio Votaciones se observaron también en los otros dominios. En todos los casos, los árboles generados con el C4.5 fueron de menor tamaño que los generados con el ID3, esto también se cumplió para la cantidad de reglas obtenidas.

4.2. Porcentaje de error

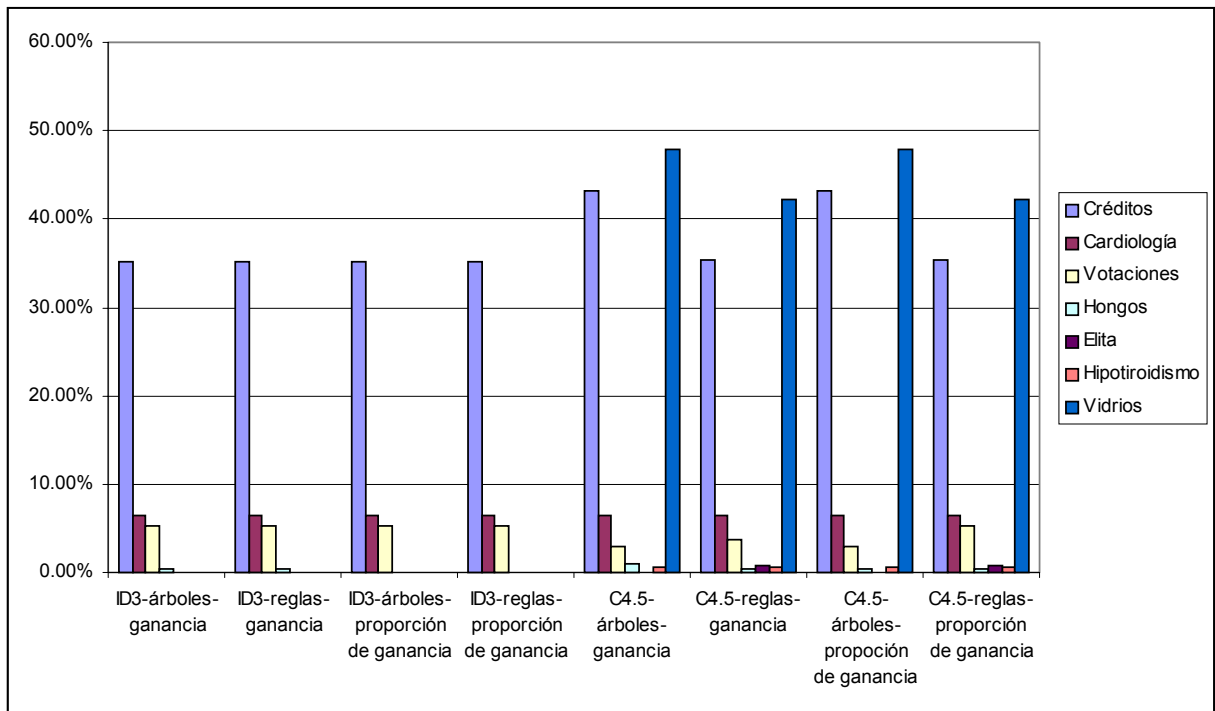


Figura 4.2.1: Porcentaje de error obtenido para cada uno de los métodos en cada dominio de datos

La figura 4.2.1 muestra el porcentaje de error obtenido con cada uno de los métodos, para cada dominio. A simple vista nos damos cuenta que en los dominios de Créditos y Análisis de Vidrios los porcentajes de error son altos para todos los métodos. A partir de este gráfico, podríamos decir que no hay un método¹ que genere un modelo claramente superior al resto para todos los dominios. Al contrario, podríamos decir que el porcentaje de error en todos los casos no parece depender del método utilizado sino del dominio analizado. Es decir, el rango de porcentajes de error dentro de cada uno de los dominios parecería estar acotado.

5. Conclusiones Generales

5.1. Conceptos destacables

A partir del estudio realizado sobre la Minería de Datos y el Aprendizaje Automático en general, y de los métodos de la familia TDIDT en particular, podemos concluir que hay varios puntos claves a tener en cuenta al realizar Minería de Datos con algoritmos inteligentes. Entre ellos, podemos destacar:

¹ Por *método* nos referimos a la aplicación de un algoritmo (ID3 o C4.5) utilizando ganancia o proporción de ganancia y generando un árbol o reglas de decisión.

- El Aprendizaje de Conceptos puede pensarse como una búsqueda en un gran espacio de hipótesis predefinidas [Mitchell, 1997]. En el caso de la familia TDIDT, este espacio de hipótesis está constituido por todos los árboles de decisión posibles para los datos que se están analizando.
- Los datos ruidosos y faltantes pueden influir en la performance del algoritmo, y depende de cada método en particular ser robusto o no ante estas situaciones.
- Los algoritmos de aprendizaje son capaces de clasificar nuevos casos, nunca vistos para ellos, porque tienen un sesgo inductivo implícito, es decir, realizan alguna suposición que les permite construir el modelo. En el caso de los algoritmos de la familia TDIDT, esta suposición implícita se divide en dos:
 1. Los datos sobre los que se construye el problema son representativos del dominio sobre el que se aplicará el modelo obtenido.
 2. Las hipótesis (árboles de decisión en este caso) más simples se prefieren sobre las hipótesis más complejas, es decir, se aplica la Afeitadora de Occam.

Si el espacio de hipótesis se extendiera hasta cubrir todos los casos posibles, se eliminaría este sesgo inductivo. Sin embargo, trabajar con todos los árboles de decisión posibles para un conjunto de datos, no permitiría realizar la clasificación de un caso no presente en los datos de entrenamiento, ya que no sería clasificado por ningún árbol. Es decir, un modelo totalmente insesgado, no podría clasificar nuevos casos [Mitchell, 1997].

5.2. Espacio de hipótesis

Como se destacó en la sección anterior, tanto el ID3 como el C4.5 realizan una búsqueda en un espacio de hipótesis constituido por los árboles de decisión posibles. El espacio de hipótesis para estos algoritmos es un espacio completo según los atributos disponibles. Como cualquier función de prueba de valor de atributos puede representarse como un árbol de decisión, estos métodos evitan uno de los mayores riesgos de los métodos inductivos que trabajan con un espacio de hipótesis reducido: que la función resultado, en nuestro caso el árbol de decisión, no se encuentre en el espacio de hipótesis analizado.

A medida que exploran el espacio de hipótesis, los algoritmos analizados mantienen una sola hipótesis actual y no todas aquellas consistentes con los datos analizados. Esto ocasiona que estos métodos no sean capaces de representar todos los árboles consistentes con los datos de entrada.

Por otro lado, recordemos que estos métodos no tienen vuelta atrás. Es decir, una vez que se seleccionó un atributo como nodo del árbol, este nunca se cambiará; los algoritmos no vuelven atrás para reconsiderar sus elecciones. Esto ocasiona que los algoritmos sean susceptibles de caer en un máximo local y que converjan a una solución que no es globalmente óptima [Mitchell, 1997]. El C4.5 agrega un cierto grado de reconsideración de sus elecciones en la postpoda que realiza.

Por último, cabe destacar que el ID3 y el C4.5 utilizan todos los datos de entrenamiento en cada paso para elegir el “mejor” atributo; esta elección se realiza estadísticamente. Esto es favorable frente a otros métodos de aprendizaje automático que analizan los datos de entrada en forma incremental. El hecho de tener en cuenta todos los datos disponibles en cada paso, resulta en una búsqueda mucho menos sensible a errores en casos individuales.

5.3. Análisis de los Resultados Obtenidos

Del análisis de los resultados obtenidos podríamos concluir que no hay ningún método que sea predominante frente a los otros. Es decir, no podemos decir, por ejemplo, que el C4.5 que utiliza la ganancia es claramente superior. Sin embargo, podemos afirmar que los resultados muestran que la proporción de error parecería ser función del dominio. En cada dominio, la proporción de error para los cuatro métodos analizados varía es similar: si la proporción de error es grande para alguno de los métodos en un dominio, seguramente lo sea también para el resto de los métodos. Si la proporción de error para alguno de los cuatro métodos en un dominio es pequeña, probablemente también sea pequeña con los otros tres métodos.

Como línea futura de trabajo, se propone analizar los datos de entrada con los cuatro métodos (ID3 utilizando ganancia, ID3 utilizando proporción de ganancia, C4.5 utilizando ganancia y C4.5 utilizando proporción de ganancia) y elegir para el nuevo dominio, el modelo que presenta la menor proporción de error. Teniendo en cuenta que si con el primer método la proporción de error es inaceptable, probablemente también sea inaceptable para el resto de los métodos. En cuyo caso, convendría analizar el problema con otros métodos de aprendizaje que enfoquen la resolución del mismo desde otro ángulo.

La cantidad de datos presentada como entrada de los algoritmos debe ser lo mayor posible, ya que los casos analizados parecen mostrar que proporción de error disminuye a medida que la cantidad de datos de entrenamiento aumenta.

5.4. Una mirada al futuro

Los algoritmos analizados no clasifican perfectamente a todos los nuevos casos, a pesar de que los modelos de clasificación presentados son entendibles y aceptables. Quedan cuestiones sin resolver, posibles mejoras y futuras líneas de trabajo que se plantean a continuación

5.4.1. Atributos multivaluados en el ID3 y el C4.5

Cuando alguno de los algoritmos realiza la partición de los casos de entrenamiento según los valores de los atributos siguiendo el método de divide y reinaras, los resultados son útiles en la medida que los valores del atributo según el cual se particiona no sean demasiados. Si existen demasiados valores para el atributo se presentan básicamente dos inconvenientes:

1. Una de las consecuencias de particionar un conjunto de entrenamiento en numerosos subconjuntos es que cada subconjunto es pequeño. Por lo tanto, aquellos patrones útiles del subconjunto pueden tornarse indetectables por insuficiencia de datos.
2. Si los atributos discretos varían en forma notable en sus valores, ¿podemos estar seguros de que un criterio como la proporción de ganancia los está evaluando de la mejor manera? La proporción de ganancia mide la proporción de información relevante a la clasificación que provee la división sobre la información producida por la división en sí. El denominador crece rápidamente a medida que la cantidad de subconjuntos se incrementa, por lo cual, el estimador deja de ser efectivo al existir muchos valores para un atributo.

Si deseamos reducir el número de resultados de un atributo multivaluado, debemos asociar uno o más de sus valores en una colección de valores de atributos o *grupo de valores*. En los primeros trabajos sobre el tema [Hunt et al., 1966] la única forma de agrupar valores era mediante la división binaria o binarización, como la realizada por el C4.5 para los atributos continuos.

En lugar de realizar este tipo de división, los algoritmos podrían asociar cada grupo de valores con una de las ramas en cantidad variable. En algunos dominios, la agrupación de valores podría determinarse de acuerdo a los conocimientos sobre el dominio. De esta manera, además de mejorar el manejo de atributos multivaluados, estaríamos incorporando información previa al sistema. De no existir agrupaciones determinables de acuerdo al dominio, debe seguirse otro método. Si un atributo tiene n valores, entonces existen $2^{n-1}-1$ divisiones binarias no triviales de esto valores, entonces para un valor de n grande se hace imposible explorar todas estas combinaciones.

En cuanto al ID3, que no maneja atributos continuos, podría incorporársele la binarización utilizada por el C4.5, o un método similar, para que pueda trabajar con atributos de este tipo. El ID3 tal como fue presentado, no puede aplicarse a todos los dominios, además de descartarse los dominios con clases continuas, como en el C4.5, se descartan los dominios con cualquier atributo continuo.

5.4.2. El futuro de la Minería de Datos

La Ley de Conservación sostiene que ningún algoritmo puede superar a otro cuando la medida de performance es la precisión de generalización esperada, sobre la suposición de que todos los resultados posibles son igualmente probables. El hecho de promediar la performance de un algoritmo sobre todos los casos posibles, asumiendo que todos son igualmente probables, sería como evaluar la performance de un auto en todos los terrenos posibles, asumiendo que todos son igualmente probables. Esta afirmación es falsa para la práctica, ya que en un dominio en particular, es claro que no todos los casos son igualmente probables.

Quinlan, quien ha identificado familias de dominios paralelos y secuenciales, sostiene que las redes neuronales son más eficientes en los dominios paralelos, mientras que los algoritmos que construyen árboles de decisión obtienen mejores resultados en los dominios secuenciales. Por lo tanto, aunque un único algoritmo de inducción puede no ser óptimo en todas las situaciones posibles, debe analizarse el mejor algoritmo para cada situación en particular.

El campo de la Minería de Datos es un campo en pleno desarrollo, donde la mayoría de las herramientas utilizadas provienen de otros campos relacionados como el reconocimiento de patrones, la Estadística o la teoría de complejidad. Dada la novedad de las investigaciones en esta área quedan todavía varios problemas por afrontar, como ser el tamaño de los datos y el ruido en los mismos.

En los últimos años se han desarrollado muchos sistemas de minería de datos y se espera que este desarrollo continúe floreciendo dada la enorme cantidad de datos que son almacenados día a día, que requiere algún tipo de análisis, entendimiento o clasificación. La diversidad de los datos, y de las técnicas y enfoques de la minería de datos, son un desafío para el crecimiento de este área de la tecnología.

6. Referencias

- [Babic et al, 1998] Babic, A., Mathiesen, U., Hedin, K., Bodemar, G., Wigertz, O. 1998. *Assessing an AI Knowledge-Base for Asymptomatic Liver Diseases*. Department of Medical Informatics, Department of Internal Diseases, Department of Infectious Diseases, Linköping University Hospital, Suecia. Faculty of Electrical and Computer Engineering, University of Ljubljana, Eslovenia. Department of Internal Diseases, Oskarshamn County Hospital, Suecia.
- [Baldwin et al, 2000] Baldwin, J.F., Lawry, J., Martin, T.P. 2000 *Mass Assignment Induction of Decision Trees on Words*. A.I. Group, Department of Engineering Mathematics, University of Bristol, Reino Unido, {jim.baldwin, j.lawry, trevor.martin@bristol.ac.uk}
- [Bergadano et al, 1992] Bergadano, F., Matwin, S. Michalski, R. S., Zhang, J. (1992) *Learning Two-Tiered Descriptions of flexible Concepts: the POSEIDON System*. En Machine Learning, Volumen 8, páginas 5-43, DBLP, www.dblp.uni-tier.de, Dinamarca.
- [Blockeel y De Raedt, 1997] Blockeel, H., De Raedt, L., 1997 *Top-Down Induction of Logical Decision Trees*. Katholieke Universiteit Leuven, Department of Computer Science, Celestijnenlaan, Bélgica
- [Blum, Langley, 1997] Blum, A., Langley, P. 1997 *Selection of Relevant Features and Examples in Machine Learning*. School of Computer Science, Carnegie Mellon University, Pittisburgh, Pennsylvania, Institute for the Study of Learning and Expertise, Palo Alto, California, EE.UU.
- [Blurock, 1996] Edward S. Blurock, 1996 *The ID3 Algorithm*, Research Institute for Symbolic Computation, www.risc.uni-linz.ac.at/people/bulrock/ANALYSIS/manual/document, Austria
- [Cabena et al, 2000] Cabena, P., Choi, H.H., Kim, S., Otsuka, S., Reinschmidt, J., Saarenvirta, G. 2000. *Intelligent Miner for Data Applications Guide*, International Technical Support Organization, IBM, <http://www.redbooks.ibm.com>
- [Callahan, B., Coombs, 1998] Callahan, B., Coombs, J. 1998 *Training Decision Trees with ID3*, <http://www.css.tayloru.edu/~jcoombs/proj/ml/id3.htm>
- [Chen, 1994] Chen, H. 1994. *Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms*. JASIS, <http://ai.bpa.arizona.edu/papers/mlir93/mlir93.html>
- [CiteSeer, 1999] S/A 1999. *What is Data Mining?*, www.citeseer.nj.nec.com/69212.html.
- [Davidsson, 1995] Davidsson, P. 1995. *ID3-SD: An Algorithm for Learning Characteristic Decision Trees by Controlling the Degree of Generalization*. Department of Computer Science, Lund University, Suecia
- [DeJong, Mooney, 1986] DeJong, G.F., Mooney, R.J. 1986. *Explanation-Based Learning. An Alternative View*, en Machine Learning, Volumen 1, páginas 145-176. Kluwer Academic Publishing
- [Elomaa, 1993] Elomaa, T. 1993. *In Defense of C4.5: Notes on Learning One-Level Decision Trees*. Department of Computer Science, University of Helsinki, Finlandia
- [Espasa-Calpe, 1974] 1974 *Diccionario Enciclopédico Abreviado*. Espasa-Calpe S.A., Madrid. Tomo I, Séptima Edición, España.
- [Fayad et al, 1996] Fayad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uhturudamy, R. (eds). 1996 *Advances in Knowledge Discovery and Data Mining*, San Mateo, AAAI Press, EE.UU.
- [Fjara, 2000] Fjara, 2000. *A Decision Tree Algorithm*. www.cs.uml.edu/~fjara/mineset/id3/id3_example/id3_algorithm.html
- [Frank y Witten, 1999] Frank, E., Witten, I.H. 1999. *Making Better Use of Global Discretization*, Proceedings 16th International Conference on Machine Learning, páginas 115-123. Department of Computer Science, University of Waikato, Nueva Zelanda
- [Gallion et al, 1993] Gallion, R., St Clair, D., Sabharwal, C., Bond, W.E. 1993. *Dynamic ID3: A Symbolic Learning Algorithm for Many-Valued Attribute Domains*. Engineering Education Center, University of Missouri-Rolla, St. Luis, EE.UU.

- [García Martínez et al, 1987] García Martínez, R., Fritz, W., y Blanqué, J. 1987. *Un algoritmo de aprendizaje de conceptos para sistemas inteligentes*. Anales del V Congreso Nacional de Informática y Teleinformática. Páginas 91-96. Buenos Aires. Junio. Argentina
- [García Martínez, 1994] García Martínez, R. 1994. *Adquisición de Conocimiento*. En Abecasis, S. y Heras, C. Metodología de la Investigación. Prologado por el Dr. L. Santaló. Editorial Nueva Librería. Argentina
- [García Martínez, 1997] García Martínez, R. 1997 *Sistemas Autónomos: Aprendizaje Automático*. Nueva Librería, Buenos Aires, Argentina
- [Gestwicki, 1997] Gestwicki, P. 1997 *ID3: History, Implementation, and Applications*, citeseer.nj.nec.com/398697.html
- [Grossman et al, 1999] Grossman, R., Kasif, S., Moore, R., Rocke, D., Ullman, J. 1999. *Data Mining Research: Opportunities and Challenges, A Report of three NSF Workshops on Mining Large, Massive, and Distributed Data*, January 1999, Chicago, EE.UU.
- [Hall, 1998] Hall, P.W. 1998. *Machine Learning – ID3 Classification*. Philipv@apk.net, <http://junior.apk.net/~philiv/rschmid.htm>
- [Holsheimer, Siebes, 1994] Holsheimer, M., Siebes, A.P.J.M. 1994. *Data Mining: the search for knowledge in databases*. Computer Science/Departament of Algorithmics and Architectire, Centrum voor Wiskunde en Informatica, CS-R9406, Amsterdam, Holanda.
- [Holte, 1993] Holte, R.1993. *Very Simple Classification Rules Perform Well on Most Commonly Used Datasets*. Computer Science Department, University of Ottawa, Canada.
- [Hunt et al, 1966] Hunt, E.B., Marin, J., Stone, P.J. 1966. *Experiments in Induction*. New York: Academic Press, EE.UU.
- [Hunt, 1975] Hunt, E.B. 1975. *Artificial Intelligence*. New York: Academic Press, EE.UU.
- [Joachims et al, 1995] Joachims, T., Freitag, D., Mitchell, T. 1997 *Web Watcher: A Tour Guide for the World Wide Web*, School of Computer Science, Carnegie Mellon University, EE.UU.
- [Joachims et al, 1997] Joachims, T., Mitchell, T., Freitag, D., Armstrong, R. 1995. *Web Watcher: Machine Learning and Hypertext*, School of Computer Science, Carnegie Mellon University, EE.UU.
- [Joshi, 1997] Joshi, K.P. 1997. *Analysis of Data Mining Algorithms*, http://userpages.umbc.edu/~kjoshi1/data-mine/proj_rpt.htm
- [Kerns, Mansour, 1996] Kearns, M., Mansour, Y. 1996. *On the Boosting Ability of Top-Down Decision Tree Learning Algorithms*, AT&T Research, Tel-Aviv University, Israel.
- [Klemettinen et al, 1994] Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H., Verkamo, A. 1994. *Finding Interesting Rules from Large Sets of Discovered Association Rules*, University of Helsinki, Department of Computer Science, Helsinki, Finlandia.
- [Korab, 1997] Korab, H. 1997. *Rule Induction: Decision Trees and Rules*, <http://www.ncsa.uiuc.edu/News/Access/Stories/97Stories/KUFRIN.html>
- [Mannila et al, 1994] Mannila, H., Toivonen, H., Verkamo, A. 1994. *Efficient Algorithms for Discovering Association Rules*, University of Helsinki, Department of Computer Science, Helsinki, Finlandia.
- [Michalski et al, 1998] Michalski, R.S., Bratko, I., Kubat M. 1998. *Machine Learning and Data Mining. Methods and Applications*. Wiley & Sons Ltd., EE.UU.
- [Michalski et al, 1982] Michalski, R. S., Baskin, A. B., Spackman, K. A. 1982. *A Logic-Based Approach to Conceptual Database Analysis*, Sixth Annual Symposium on Computer Applications on Medical Care, George Washington University, Medical Center, Washington, DC, EE.UU.
- [Michalski, 1983] Michalski, R. S. 1983. *A Theory and Methodology of Inductive Learning*. En Michalski, R. S., Carbonell, J. G., Mitchell, T. M. (eds.). (1983) *Machine Learning: An Artificial Intelligence Approach*, Vol. I. Morgan-Kaufman, EE.UU.
- [Michalski, 1991] Michalski, R. S. 1991. *Towards an Unified Theory of Learning: An Outline of Basic Ideas*, Proceedings of the 3rd World Conference on the Fundamentals of Artificial Intelligence, Paris, Julio 1-5, 1991
- [Michalski, Tecuci, 1994] Michalski, R. S., Tecuci, G. (eds) 1994. *Machine Learning: A Multistrategy Approach*, Vol.IV, Morgan Kauffman, EE.UU.
- [Michie, 1986] Michie, D. 1986. *On Machine Intelligence* (2nd ed), Ellis Horwood, Chichester, Reino Unido
- [Michie, 1998] Michie, D. 1988 *Machine Learning in the next five years*, EWSL-88, 3rd European Working Session on Learning, Pitman, Glasgow, Londres, Reino Unido.
- [Mitchell, 1997] Mitchell, T. 1997. *Machine Learning*. MCB/McGraw-Hill, Carnegie Mellon University, EE.UU.
- [Mitchell, 2000a] Mitchell, T. 2000 *Decision Trees*. Cornell University, www.cs.cornell.edu/courses/c5478/2000SP, EE.UU.
- [Mitchell, 2000b] Mitchell, T. 2000 *Decision Trees 2*. Cornell University, www.cs.cornell.edu/courses/c5478/2000SP, EE.UU.
- [Montalvetti, 1995] Montalvetti, Mario 1995. *Sistemas de adquisición automática de conocimientos*, Tesis de grado en Ingeniería en Computación. Universidad Católica de Santiago del Estero, Argentina.
- [Monter, 2001] Monter, C. 2001. *Equiparación de Marcos*. Notas de Seminario. Escuela de Posgrado, Instituto Tecnológico de Buenos Aires, Argentina
- [NIST, 1998] S/A 1998. *Confidence intervals for small sample sizes*. En Engineering Statistics Handbook, Information Technology Laboratory, NIST, <http://www.itl.nist.gov/div898/handbook/prc/section2/prc242.htm>, EE.UU.
- [Quinlan y Cameron-Jones, 1995] Quinlan, J.R., Cameron-Jones, R.M. 1995. *Oversearching and Layered Search in Empirical Learning*. Basser Departament of Computer Science, University of Science,

- Australia.
- [Quinlan, 1986] Quinlan, J.R. 1986. *Induction of Decision Trees*. En Machine Learning, Capítulo 1, p.81-106. Morgan Kaufmann, 1990
- [Quinlan, 1987] Quinlan, J.R. 1987. *Generating Production Rules from Decision trees*. Proceeding of the Tenth International Joint Conference on Artificial Intelligence, páginas. 304-307. San Mateo, CA., Morgan Kaufmann, EE.UU.
- [Quinlan, 1988b] Quinlan, J.R. 1988. *Decision trees and multi-valued attributes*. En J.E. Hayes, D. Michie, and J. Richards (eds.), Machine Intelligence, Volumen II, páginas. 305-318. Oxford University Press, Oxford, Reino Unido
- [Quinlan, 1989] Quinlan, J.R. 1989. *Unknown Attribute Values in Induction*. Basser Department of Computer Science, University of Science, Australia.
- [Quinlan, 1990] Quinlan, J. R. 1990. *Learning Logic Definitions from Relations*. En Machine Learning, Vol 5, páginas 239-266. Oxford University Press, Oxford, Reino Unido
- [Quinlan, 1993a] Quinlan, J.R. 1993. *The Effect of Noise on Concept Learning*, En R. S. Michalski, J. G. Carbonell, & T. M. Mitchells (Eds.) Machine Learning, The Artificial Intelligence Approach. Morgan Kaufmann, Vol. I, Capítulo 6, páginas 149-167. San Mateo, CA: Morgan Kaufmann, EE.UU.
- [Quinlan, 1993b] Quinlan, J.R. 1993. *Learning Efficient Classification Procedures and Their Application to Chess Games*, En R. S. Michalski, J. G. Carbonell, & T. M. Mitchells (Eds.) Machine Learning, The Artificial Intelligence Approach. Morgan Kaufmann, Vol. II, Capítulo 15, páginas 463-482, EE.UU.
- [Quinlan, 1993c] Quinlan, J.R. 1993. *Combining Instance-Based and Model-Based Learning*. Basser Department of Computer Science, University of Science, Australia.
- [Quinlan, 1993d] Quinlan, J.R. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, California, EE.UU.
- [Quinlan, 1995] Quinlan, J.R. 1995. *MDL and Categorical Theories*. Basser Department of Computer Science, University of Science, Australia.
- [Quinlan, 1996a] Quinlan, J.R. 1996. *Improved Use of Continuous Attributes in C4.5*. Basser Department of Computer Science, University of Science, Australia.
- [Quinlan, 1996b] Quinlan, J.R. 1996. *Learning First-Order Definitions of Functions*. Basser Department of Computer Science, University of Science, Australia
- [Riddle, 1997] Riddle, P.J. 1997. *ID3 Algorithm*. www.cs.auckland.ac.nz/~pat/706_99/ln/node75.html, Nueva Zelanda
- [Rissanen, 1983] Rissanen, J. 1983. *A universal prior for integers and estimation by minimum description length*. En Annals of Statistics 11, Vol 2, p. 416-431
- [Thakore, 1993] Thakore, M., St Clair, D. 1993. *Effect of the χ^2 test on the Construction of ID3 decision trees*, Sun Microsystems, University of MO-Rolla, Engineering Education Center, St. Louis, EE.UU.
- [Thrun et al, 1991] Thrun, S., Bala, J., Bratko, I., Cestnik, B., Cheng, J., De Jong, K., Dzeroski, S., Fahlman, S.E., Fisher, D., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Michalski, R.S., Mitchell, T., Pachowicz, P., Reich, Y., Vafaie, H., Van de Welde, W., Wenzel, W., Wnek, J, Zhang, J. 1991 *The MONK's Problems. A Performance Comparison of Different Learning Algorithms*, Carnegie Mellon University, Pittsburgh, EE.UU.
- [Thrun et al, 1998] Thrun, S., Faloutsos, C., Mitchell, T., Wasserman, L. 1998 *Automated Learning and Discovery: State-Of-The-Art and Research Topics in a Rapidly Growing Field*. CMU-CALD-98-100, Center for Automated Learning and Discovery, Carnegie Mellon University, Pittsburgh, EE.UU.
- [Witten y Frank, 2000] Witten, I.H., Frank, E. 2000. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Diego, EE.UU.
- [Yoda, 19950] S/A 1995. *Building Classification Models: ID3 and C4.5*, yoda.cis.temple.edu:8080/UGAIWWW/lectures/C45, Pensilvania, EE.UU.