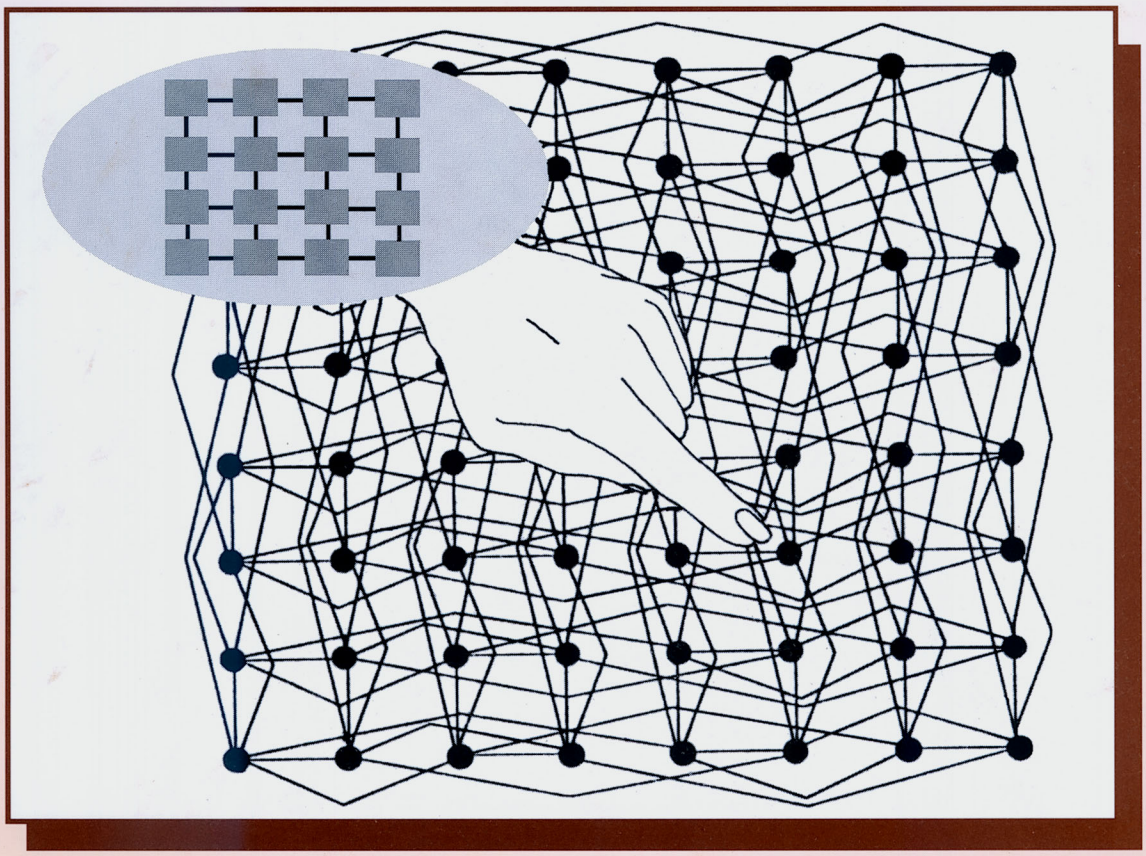
Ingeniería

Revista de la Universidad de Costa Rica
Julio/Diciembre 1996 VOLUMEN 6 Nº 2



INGENIERIA

Revista Semestral de la Universidad de Costa Rica
Volumen 6, Julio/Diciembre 1996 Número 2

CORRESPONDENCIA Y SUSCRIPCIONES

Editorial de la Universidad de Costa Rica
Apartado Postal 75
2060 Ciudad Universitaria Rodrigo Facio
San José, Costa Rica

CANJES

Universidad de Costa Rica
Sistema de Bibliotecas, Documentación e Información
Unidad de Selección y Aquisiciones-CANJE
Ciudad Universitaria Rodrigo Facio
San José, Costa Rica

Suscripción anual:
Costa Rica: ¢ 1 000,00
Otros países: US $ 25,00

Número suelto:
Costa Rica: ¢ 750,00
Otros países: $ 15,00

Edición aprobada por la Comisión Editorial de la Universidad de Costa Rica

Ciudad Universitaria Rodrigo Facio
San José, Costa Rica.

Revisión Filológica: *Lorena Rodríguez*

Diagramación:
José R. Argüello V.

Control de Calidad:
Unidad Diseño Revistas. Oficina de Publicaciones

Impreso en la Oficina de Publicaciones
de la Universidad de Costa Rica

Revista
620.005
I-46i Ingeniería / Universidad de Costa Rica. ——
Vol. I, no. 1 (ene./jun. 1991)— . —— San José, C. R. : Editorial de la Universidad de Costa Rica, 1991— (Oficina de Publicaciones de la Universidad de Costa Rica)
v. : il

Semestral.

1. Ingeniería - Publicaciones periódicas.

CCC/BUCR—250

Ingeniería 6 (2): 9-18 1996 San José, Costa Rica

GENERACIÓN DE GRANDES BASES DE DATOS ARTIFICIALES PARA EVALUACIÓN DE HERRAMIENTAS DE MINERÍA DE DATOS

José Ronald Argüello V[1]

Resumen

La minería de datos o el proceso de extracción de conocimiento almacenado en bases de datos requiere de herramientas adecuadas que permitan procesar, eficientemente, grandes cantidades de datos. Este artículo expone por qué la evaluación de estas herramientas no se hace, usualmente con datos reales y muestra varios proyectos de extracción inductiva de árboles de decisión llevados a cabo con datos artificiales o sintéticos, cuyo comportamiento es similar a los datos naturales. Además, introduce el tema de generación de grandes bases de datos artificiales usando una versión modificada del programa DGP/2 de P. Benedict.

Summary

Data mining or the process of knowledge extraction from data bases requires of suitable tools to allow an efficient process of large amounts of data. This paper shows why tool evaluation is not commonly done with real data and describes several research projects for inductive decision tree extraction carry out with synthetic data bases, whose behavior is similar to natural data. In addition, the subject of large synthetic data base generation is introduced using a modified version of the DGP/2 program of P. Benedict.

1.- INTRODUCCIÓN

Durante el desarrollo de proyectos de programación de sistemas computacionales en los cuales se usen o elaboren herramientas que trabajen en una gran cantidad de datos se presentan varios inconvenientes: no disponibilidad de los datos en la forma y cantidad que se requiere, privacidad y seguridad de datos reales, ruido o inconsistencias en la presentación de estos que interfiere con la concepción de las herramientas, regularidades o irregularidades estadísticas presentes en los datos que impiden una evaluación justa de las características de las herramientas diseñadas. Un camino para solventar estos inconvenientes es la creación de datos de prueba que permitan una proyección sobre el posible desempeño de los programas o herramientas. El problema se complica pues en la mayor parte de los casos no solo se trata de evaluar la eficacia de los programas sino también su eficiencia [Savasere, 95][Srikant, 96]. Es aquí que requerimos de sistemas automáticos que permitan la generación adecuada de una gran cantidad de datos. Se discute en este artículo algunos aspectos relacionados con la generación automática de datos y cómo extender esto a bases de datos grandes y reales. Se discute luego la aplicación de tal sistema a la minería de datos basada en árboles de decisión , así

[1] Prof. Catedrático, Ph . D. Escuela de Ciencias de la Computación e Informática
Facultad de Ingeniería. Universidad de Costa Rica.

como ciertos resultados obtenidos. Las ventajas y desventajas de este enfoque son analizadas al final del artículo. El programa desarrollado se encuentra disponible gratuitamente en la Escuela de Ciencias de la Computación e Informática o en la máquina *anubis.ecci.ucr.ac.cr* en el directorio */public/gabda.gz.*

2.- MINERÍA DE DATOS Y BASES DE DATOS ARTIFICIALES

La minería de datos es el proceso de extracción de información significativa no estructurada previamente sobre datos almacenados con anterioridad. [Frawley , 91][Piatesky-Shapiro, 91] [Mathews, 93]. La información es significativa para el usuario, en la medida en que arroja un resultado útil para su función. No es previamente estructurada porque no existe un procedimiento previo para solicitar tal tipo de información, sino más bien una idea o meta, la cual dicha información debía satisfacer (aunque esto no descarta el obtener información de la que no se tenía del todo una idea preconcebida). Dicha información constituye un conocimiento útil para el usuario y por su carácter es generalmente nuevo, de aquí el término *knowledge discovery* o descubrir conocimiento utilizado indistintamente por los diferentes autores.

Uno de los problemas más comunes encontrados por los desarrolladores de herramientas en extracción de conocimientos o minería de datos es la falta de bases de datos adecuadas en donde aplicar sus herramientas o ponerlas a prueba contra otras herramientas [Benedict, 90a] [Imielinski, 96]. Aunque el problema ha existido siempre en el desarrollo de sistemas tradicionales de acceso a bases de datos, es tal vez agudizado en esta área debido a varios factores. El primer factor es el contenido de los datos. Mientras que en sistemas tradicionales el acceso eficiente a los datos es irrelevante a su contenido i.e., podemos generar un millón de registros y proceder a evaluar la eficiencia de acceso sin interés en el contenido del registro que accesamos; en minería de datos lo que interesa es la asociación entre los contenidos de los diferentes registros y como este se relaciona con otros en otras tablas [Imielinski, 96]. El contenido de los registros y la distribución de los datos influye significativamente en el desempeño último de las herramientas.

Un segundo factor es la importancia comercial que puede tener el conocimiento derivado para los dueños de la base de datos. Un cantidad significativa de datos almacenados tiene un costo alto no solo en lograrlo, sino en el potencial conocimiento que se pueda derivar. La naturaleza indefinida de las herramientas a utilizar y de la estructura y tipo de conocimiento que se pueda derivar provoca una actitud de recelo y desconfianza hacia el uso que se le pueda dar a los datos si estos se facilitan a un investigador o desarrollador externo a la empresa en cuestión. Aunque se ofrezca compartir resultados y respetar lo que se pueda obtener no existe una aptitud positiva para suministrar bases de datos a terceros.

Otros factores no menos importantes incluyen la privacidad de los datos e información derivada de ellos, la seguridad de los datos (en grandes bases de datos no se puede, simplemente, facilitar una copia de los mismos y el trabajo sobre ellos puede implicar problemas de seguridad para el acceso a los mismos).

3.- RAZONES PARA LA CREACIÓN DE BD ARTIFICIALES

Los comentarios anteriores sugieren buenas razones para la creación automática de grandes bases de datos artificiales:

- *Contenido:* Una base de datos debe poseer una alta calidad de información, no simplemente secuencia de hileras y símbolos sin relación alguna.

- *Estrategia:* Los datos artificiales no son de interés comercial y no reflejarán ninguna situación real , solo las hipótesis o características con la cual se generaron.
- *Privacidad*: Los datos artificiales no reflejan situación alguna que pudiera afectar la privacía de personas o instituciones.
- *Seguridad:* Un base de datos artificial puede hacerse pública sin que existan aspectos de seguridad que deban tomarse en cuenta. Es más, si los datos son corrompidos o alterados, el mismo sistema (automático) usado para crearla puede ser reutilizado.
- *Cantidad*: Automáticamente es posible generar la cantidad de los registros que se desean con el fin de evaluar la eficiencia de los métodos encontrados.
- *Forma:* Es posible dar a la base de datos la estructura sintáctica que se desee para ajustarse a las herramientas y métodos facilitando así la experimentación sobre los mismos.
- *Comparación*: Diferentes herramientas pueden evaluarse sobre el mismo conjunto de datos facilitando la comparación entre las mismas.
- *Portabilidad*: Un gran base de datos no puede ser transportada fácilmente (aún con los medios que *Internet* ofrece). Si los parámetros utilizados para la generación de la base de datos en un lugar, se mantienen, esta puede ser generada fácilmente en otro lugar y sin necesidad de transportarla.
- *Almacenamiento*: Si se tiene la facilidad de generarla es, eventualmente, más barato almacenar el programa y los parámetros utilizados (con excepción del tiempo necesario para generarla) que almacenar toda la base de datos (aún con métodos de compresión).

4.- EL PROGRAMA DE GENERACION DE DATOS DE P. BENEDICT

Powell Benedict et al., trabajando en la Universidad de Illinois at Urbana, crearon tal vez uno de los primeros programas en generación automática de datos y le llamaron DGP-2 ("Data Generation Program 2") [Benedict, 90b].

La motivación para la creación de DGP-2 fue basada, en parte por las razones anteriores, pero principalmente para generar conjuntos de datos algo pequeños y para el estudio de técnicas de aprendizaje mecánico, inducción constructiva y optimización del sesgo [Mitchel, 80] [Mitchalski, 83] [Utgoff, 82, 89][Russell, 90].

P. Benedict se planteó las siguientes preguntas:

1. ¿ Qué características medibles del dominio del problema son más útiles de sintetizar i.e., para predecir la eficiencia de los algoritmos inductivos ?.

2. ¿ Hasta que punto algunas de estas características son no medibles sin conocimiento del concepto meta y por lo tanto ¿ Cómo podremos estimarlas ?.

3. ¿ Cuánto entrenamiento es realmente necesario i.e., cuánto debe explorarse de los datos sintéticos para mejorar significativamente la eficiencia en problemas de inducción naturales.?

4. ¿ Qué suposiciones de DGP/2 son válidas para problemas naturales ?

La respuesta a estas preguntas puede encontrarse parcialmente durante la práctica y la experimentación. La utilidad de generar datos artificialmente o sintéticamente ha sido objeto de otros estudios [Rendell 88,89] [Benedict 90a].

Un antecesor de DGP2 fue utilizado en un estudio hecho por Rendell [Rendell 88].

Su objetivo fue estudiar la eficiencia de varios sistemas de aprendizaje automático en una variedad de dominios. El primer DGP tenía la habilidad de generar datos basado en parámetros

de entrada. Como en la mayoría de dominios para aprendizaje automático, solo dos clases de instancias (registros) , positivos y negativos, era necesario especificar así como la proporción entre ellas. El programa genera puntos en el espacio (cada instancia podía ser vista como un punto en el espacio) y los clasificaba de acuerdo a su cercanía a uno o varios puntos que representaban la clase positiva, medida en términos de medias y varianzas que también podían ser especificados. Lo más importante de este estudio fue que las conclusiones obtenidas de dominios naturales y artificiales eran muy similares, indicando el comportamiento similar de los datos artificiales.

En otro estudio hecho también por Rendell y Cho [Rendell 89] se mostró que conceptos disjuntos en gran medida requerían más datos para aprenderse, lo cual era completamente consistente con los resultados teóricos de Ehrenfeuct, Haussler, Kearns, y Valiant. [Ehrenfeuct, 88].

Benedict encontró resultados similares i.e, eficiencia similar en dominios naturales y artificiales estudiando la validez de reglas de optimización del sesgo [Benedict, 90a].

Operación de DGP/2.

Como se mencionó anteriormente, DGP/2 genera un número *n* predeterminado de puntos el espacio (picos) que son los que determinan la clase positiva del conjunto de instancias por generarse. Ver Figura Nº 1.

Los picos se producen en un espacio m-dimensional donde m es el número de atributos por instancia (registro). El número total de instancias es dividido, proporcionalmente, entre los n picos.

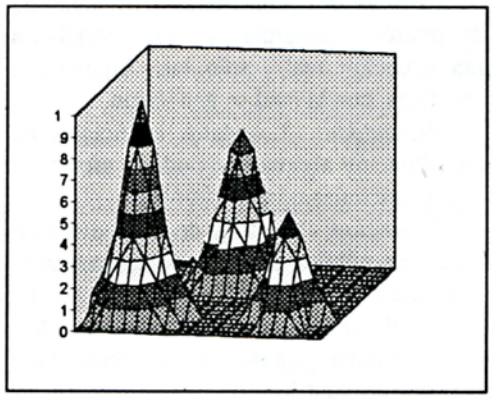


Figura Nº 1. Superficie de instancias positivas (2 dimensiones).

Para lograr una proporción adecuada de casos positivos y negativos, los valores de los atributos generados siguen una distribución normal con la media definida por cada pico y donde la varianza es ajustada de acuerdo a la proporción requerida. Ver Figura Nº 2. Así, una instancia (punto) que queda fuera del rango de acción de los picos es asignada a la clase negativa.

Note que la complejidad del concepto (clase positiva) viene determinada por el número de picos utilizados y la proporción de casos positivos en los datos artificiales.

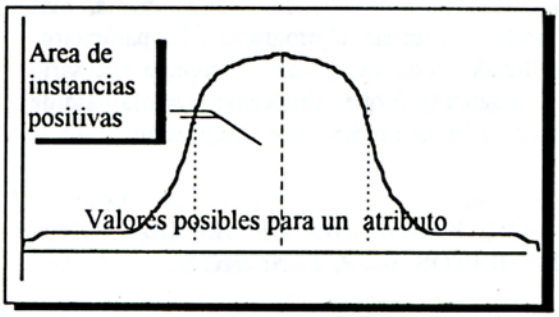


Figura Nº 2. Distribución normal de cada atributo.

5.- MODIFICACIONES PARA GRANDES BASES DE DATOS.

El programa DGP/2 tiene las siguientes desventajas para su aplicación a grandes bases de datos:

- Está orientado a sistemas de aprendizaje mecánico de conceptos descritos por instancias positivas y negativas. Esto lo hace inadecuado para generar espacios donde las instancias son clasificadas en varias clases.
- Usa una representación interna de instancias matricial lo que limita su capacidad de generar una base de datos a las instancias alojadas en memoria real. Aunque está limitante puede ser solventada a través de los sistemas de memoria virtual o la capacidad particular del computador especifico.
- El número de valores de cada atributo es exactamente el mismo, lo que no refleja una distribución de los datos en el mundo real.
- La creación de picos y asignación de clases a instancias generadas, toma en cuenta todas las características de las instancias, sin considerar que en bases de datos reales existen factores que son completamente irrelevantes para la determinación de las clase de cada instancia.
- No existe forma de determinar una instancia especifica, excepto sus contenidos. En sistemas administradores de bases de datos se hace necesario identificar cada instancia.
- No es flexible en la presentación y formato de los datos generados.

Debido a estos problemas DGP/2 fue modificado para adaptarlo a las necesidades de generación de bases de datos con largas colecciones de instancias. Al programa generado se le ha llamado DGP/3 en honor a su antecesor. La figura N° 3 es una lista de las características que se pueden especificar en DGP/3. Una descripción de las mismas va más allá del objetivo de este artículo. Al lector interesado lo referimos a la descripción del programa en */public* en *anubis.ecci.ucr.ac.cr.*

6.- UTILIZACIÓN DEL PROGRAMA DGP/3

El programa fue utilizado para generación de diferentes bases de datos en experimentos de clasificación por árboles de decisión usando el algoritmo ID5,pero para efectos de los experimentos sin las capacidades reorganizativas del mismo[Utgoff, 89, 95] Otros algoritmos reorganizativos pueden ser han sido estudiados por Schilmmer y Van de Velde [Schilmmer, 86][Van-de-Velde, 90].

EXPERIMENTOS CON GRANDES BASES DE DATOS ARTIFICIALES.

En esta serie de experimentos conducidos por el autor [Argüello, 96] para comparar la capacidad inductiva en grandes bases de datos de la entropía, usando una variante del algoritmo ID3 en su forma más simple , i.e., sin poda u optimización del árbol de decisión [Quinlan, 83, 86,89] [Mehta, 95] y a la vez analizar el comportamiento de una medida simple de clasificación , a la cual he llamado determinación y de como esta se compara con la entropía usando ambas el algoritmo ID3. Independiente de los resultados de la comparación, lo cual se escapa de los fines de este artículo, es interesante observar en las cuatro tablas de datos artificiales el comportamiento del algoritmo inductivo.

Generación de las bases de datos experimentales

Cuatro base de datos sintéticas con alrededor de cien mil casos (tuples) fueron generadas para los experimentos. Cada base de datos consistió de 20 atributos (A0 a A19), el atributo clase (que contiene la clase de cada instancia) y 10 valores por atributo aproximadamente (así evitando los efectos de atributos con muchos valores, los cuales tienden a afectar inconvenientemente y de muchas formas la capacidad inductiva del algoritmo [Quinlan, 86] [Argüello, 86] [Argüello, 96]).

La forma en la cual se asignan los valores al atributo clase determina el tipo de base de datos como se describe a continuación. La primera base de datos consistió de una tradicional con solo dos clases.

La segunda consistió en diez posibles clases de manera que se complicó ligeramente la tarea de inducción (10 clases).

Para complicar aún más, se escogió uno de los 20 atributos como el designador de la clase. Dado que este atributo tiene valores aleatorios en cada instancia, no existía relación alguna con los otros atributos y la clase de cada instancia (clases aleatorias).

La última designación de clases se hizo con un árbol de decisión creado previamente. Los datos fueron filtrados para que la clase de cada instancia coincidiera con el árbol de decisión. De esta forma el árbol quedó inmerso en la base de datos (árbol inmerso).

```
;*************************************************************
; Sample parameter file for DGP Version 3.0
; Author: Powell Benedict
; Modify by: Jose R. Argüello, Nov 95
;*************************************************************
; This is a sample parameter file for DGP/3 V1.0. Notice first that comments
; begin with a semi-colon. Next, blank lines are allowed anywhere. No line
; in this file may be more than 80 characters long. Finally, comments may
; be placed at the end of an actual parameter line.
; Parameter format is:
;    parm_name = parm_value
; You may have any number of intervening spaces between the fields of a
; parm line, provided it doesn't add up to more than 80 characters total.
; You may specify the parameters in any order. Parameters which are absent,
; or which are incorrectly formed will be pointed out by DGP/3
; No more than 50 features are accepted
; Binary attributes are assumed except if specified otherwise
; See max_feature_value_99 parameter
num_features              = 20      ; Number of features
num_irrelevant_features   = 10      ; Number of irrelevant features (last 10)
max_feature_value_0       = 700     ; Maximum feature value for feature 0
max_feature_value_1       = 700     ; Maximum feature value for feature 1
num_peaks                 = 1       ; Number of peaks in the final space
num_instances             = 100000  ; Number of instances generated
proto_seed                = 2398    ; Initial random seed
range                     = 0.1     ; Range for positive class membership
percentage                = 66      ; Percentage of positive instances
trunc_flag                = 0       ; Out of range instance disposition flag
out_file_name             = test.out ; Filename for instances
stat_file_name            = stat.out ; Filename for run statistics
header                    = yes     ; optional, header line: +key *Class A1 ..
instance_key              = yes     ; optional, key numbered instances.
value_separator           = space   ; optional, comma by default
out_of_range_class_value= 0         ; optional, integer value for out of range instances.
; End of parm file
```

Figura N° 3. Lista de parámetros necesarios para DGP/3.

Los experimentos.

Dos experimentos fueron realizados con cada una de las bases de datos artificiales anteriores, uno con el criterio de selección/clasificación entropía y otro con el otro criterio determinación.

Cada experimento consistió de una serie de inducciones de árboles. Cada inducción de un árbol fue determinada por la muestra inicial de la base de datos (de un 2% a un 17%). Una vez que el árbol fue extraído de la muestra , la base de datos fue filtrada con ese árbol y el error final calculado. Luego un porcentaje de las excepciones (instancias clasificadas incorrectamente por el árbol) fue añadido a la muestra original y el árbol reconstruido. De nuevo, la base de datos debe ser sometida al nuevo árbol.

Este proceso inductivo continua hasta que el error final sea suficientemente pequeño , hasta un predefinido número de iteraciones o hasta que el mejoramiento sobre la tasa de error sea no significativa.

Resultados

Las figuras Nº 4 y Nº 5 muestran las tasas de error obtenidas con cada experimento correspondiente a base de datos artificial a la vez. Cada línea en los gráficos representan un experimento.

Es notorio que altas tasas de error se obtienen para ambos criterios de selección de atributos cuando las clases son asignadas aleatoriamente.

Por otra parte, tasas muy bajas de error, aún considerando que la muestra es sensiblemente pequeña con respecto al tamaño de la base de datos (0.2% del total) se obtienen si el árbol está inmerso en la base de datos.

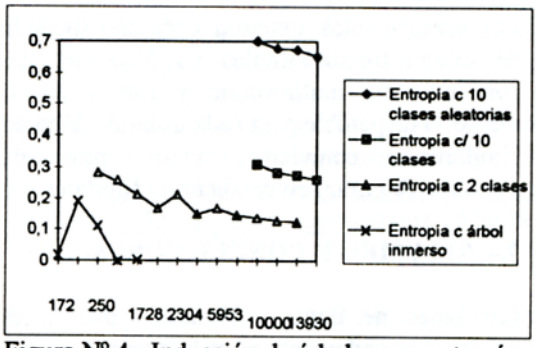


Figura Nº 4. Inducción de árboles con entropía.

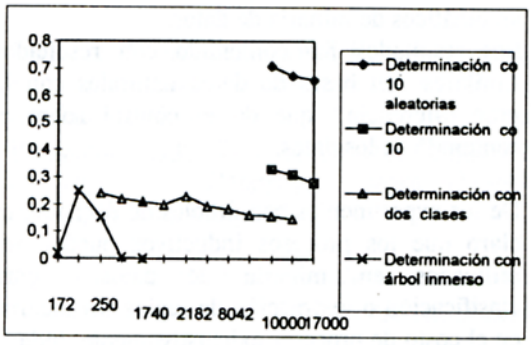


Figura Nº 5. Inducción de árboles con determinación.

El árbol predefinido no era sensiblemente complicado, i.e., su altura era 2 y tenía 42 nodos con 39 hojas, lo que deja solo dos nodos para el segundo nivel. Aún así, los resultados son significativos ya que una muestra no muy adecuada lleva a una convergencia rápida en este caso. Cuando menos clases están presentes el proceso de inducción se facilita y las tasas de error son mucho más bajas que cuando diez posibles clases están presentes.

Detalles adicionales pueden ser encontrados en [Argüello, 96].

Debe notarse que estas bases de datos artificiales representan un buen contexto para inducción con ID3 modificado pues no hay atributos con muchos valores presentes.

Los experimentos tomaron cada uno alrededor de veinte a treinta minutos en *Estaciones Sun* con ambiente multiusuario y con 8 mb de memoria disponible para cada usuario. Tiempos similares se obtuvieron en un procesador *Pentium* (60 mhz) en condiciones similares.

7.- RESUMEN Y CONCLUSIONES

Las bases de datos artificiales pueden ser usadas adecuadamente para la experimentación controlada y para probar la capacidad de extracción de conceptos de sistemas automáticos de minería de datos.
Los resultados han coincidido con resultados similares en bases de datos naturales con la gran diferencia que da el control sobre el contenido de los datos.

De los experimentos con árboles de decisión, es claro que los procesos inductivos pueden ser utilizados en minería de datos para clasificación o extracción de reglas sin incurrir en el costo de procesar exhaustivamente toda la bases de datos, usualmente en tamaños que las bases de datos de cien mil casos utilizadas en los experimentos mostrados aquí, pueden parecer de juguete. Bases de datos con millones de registros son necesarias para el proceso de descubrir conocimiento, tales como las usadas por Agrawal, Srikant, Han y Grossman. [Agrawal et al, 93] [Srikant, 96] [Han, 95] [Savasere,95][Grossman, 96].

En grandes bases de datos, una pequeña diferencia en el algoritmo puede ser muy significativa en términos de su eficiencia. Tal es el caso de procesar únicamente una muestra de las excepciones cuando se obtiene el árbol de decisión.

Por otra parte, si un simple concepto existe una pequeña muestra de la base de datos debe ser capaz de detectarlo (experimento con el árbol inmerso) ahorrando significativamente tiempo de procesamiento.

8.- BIBLIOGRAFIA

Agrawal, R. and Imielinski, T. and Swami, A.. *Mining Association Rules between set of items in largedatabases*, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, Usa 1993.

Argüello, José Ronald. *Decision Trees: Tools of AI and Data Modeling.* MSc Thesis, University of Denver, 1986.

Argüello, José Ronald. *On Decision Tree Induction for Knowledge Discovery in Very Large Databases* Tesis Doctoral. University of Florida. Junio 1996.

Benedict, P.A. *The Use of Synthetic Data in Dynamic Bias Selection,* Proc. Of the 6th Aerospace Applications of Artificial Intelligence. Conference, Dayton, Ohio, October, 1990.

Benedict, P. and Rendell, L. *Data Generation Program/2 V1.0,* Inductive Learning Group. University of Illinois. Urbana, 1990.

Ehrenfeucht, A., Haussler, D., Kearns, M, Valiant, *L. A general lower bound on the number of examples needed for learning.* Proc. Computational Learning. Theory, 1988, 139-154.

Frawley, W. J. and Piatetsky-Shapiro, G. and Mathews, C. J. *Knowledge Discovery in Databases. an Overview,* AAAI Pres/The MIT Press Massachusetts 1991.

Grossman, R. and Bodek, H. and Northcutt, D. *Early Experience with a System for Mining, Estimating, and Optimizing large collections of objects managed using an object wharehouse,* SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal 1996.

Han, J. and Fu, Y.. *Discovery of Multiple-level Association Rules from Large Databases,*

Proceedings of the 21st International Conference on Very Large Data Bases, Switzerland 1995.

Imielinski, T. *From File Mining to Database Mining*, SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal 1996.

Mathews, C. J. and Chan, P. K. and Piatetsky-Shapiro, G. *Systems for Knowledge Discovery in Databases*, IEEE Transactions on Knowledge and Data Engineering, New York 5(6), 93.

Mehta, M. and Agrawal, R. and Rissanen, J.. *MDL based decision tree pruning*, Proceedings of Int'l Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, Canada 1995.

Michalski, R. S. and J. G. Carbonell and T. M. Mitchell. Machine Learning. An Artificial Intelligence Approach, I, Tioga Publishing Company Palo Alto, California 1983.

Mitchell, T. M. *The need for biases in learning generalizations*. Technical Report CBM-TR-117, May 1980.

Piatetsky-Shapiro, G. and Frawley, W. J. Knowledge Discovery in Databases, AAAI Pres/ The MIT Press Massachussets 1991.

Quinlan, J. R. *Induction of Decision Trees*. Machine Learning. II An Artificial Intelligence Approach, I, Tioga Publishing Company Palo Alto, California 1983.

Quinlan, J. R. *Induction of Decision trees*, Machine Learning, 1, 1986.

Quinlan., J. R.. *Inferring decision trees using the minimum description length principle*, Information Computer, 80, 1989.

Rendell, L. A., Benedict, P. A., Cho, H. H., Seshu, *Improving the design of rule-learning systems*, Proceedings of the Seventh International Conference on Expert Systems and their Applications, June, 1988.

Rendell, L. A., Cho, H. H. *The effect of data character on empirical concept learning* in Proc. Fifth International Conference on Artificial Intelligence Applications, March, 1989.

Russell, S., Grosof, B. *Declarative bias: An overview*, in P. Benjamin (Ed.), Change of Representation and Inductive Bias. Kluwer Academic Press, 1990.

Savasere, A. and Omiecinski, E. and Navathe., S. *An Efficient Algorithm for Mining Association Rules in Large Databases*, Proceedings of the 21st International Conference on Very Large Databases, pages 432-444, Zurich, Switzerland, 1995.

Schlimmer, J. C. *A case study of incremental concept induction*, Proceedings of AAAI, Philadelphia 1986.

Srikant, R. and Agrawal, R. *Mining Generalized Association Rules*, Proceedings of the 21st International Conference on Very Large Data Bases, 1995.

Srikant, R. and Agrawal, R. *Mining Quantitative Association Rules in Large Relational Tables*, ACM SIGMOD96 International Conference on Management of Data, 1996.

Utgoff, P. E., Mitchell, T. M., *Acquisition of appropriate bias for inductive concept learning*, Proc. National Conference on Artificial Intelligence, 1982.

Utgoff, P. E. *Shift of bias for inductive concept learning*. Machine Learning: An Artificial Intelligence Approach, 1986, III.

Utgoff, P. E. *Incremental Induction of Decision Trees*, Machine Learning, 4, 1989.

Utgoff, P. E. *Decision Tree Induction based on efficient tree restructuring*, Technical Report. 95-18 Department of Computer Science. University of Massachussetts, 1995.

Van de Velde, W. *Incremental Induction of Topologically Minimal Trees*, Machine Learning: Proceedings of the Seventh International Conference, University of Texas, Austin , Texas 1990. /