# EXTRACCIÓN DE CONOCIMIENTO EN GRANDES BASES DE DATOS UTILIZANDO ESTRATEGIAS ADAPTATIVAS

Waldo Hasperué
Directores: Armando De Giusti y Laura Lanzarini
III-LIDI. Facultad de Informática. Universidad Nacional de La Plata
Fecha de exposición: 28 de marzo de 2012

## 1. Resumen

Hoy en día, dada la existencia de enormes volúmenes de información digital, resulta de interés contar con técnicas que permitan, en una primera etapa, analizar tal información y obtener conocimiento nuevo a partir de ella. Además, como se espera que la información disponible se modifique o incremente a lo largo del tiempo, en una segunda etapa, sería relevante poder adaptar el conocimiento adquirido a los cambios o variaciones que ocurran en el conjunto de datos original.
El objetivo general de esta tesis es la definición de una nueva técnica de minería de datos capaz de generar conocimiento útil, produciendo resultados provechosos para el usuario final.
El aporte central es la definición e implementación de una técnica adaptativa que permite obtener modelo dinámico, formado por reglas de clasificación, capaz de adaptarse a los cambios de la información.

## 2. Hiper-rectángulos

La técnica presentada en esta tesis basa su funcionamiento en el manejo de hiper-rectángulos como descriptores de las diferentes clases de datos presentes en la base de datos estudiada.
Los hiper-rectángulos son politopos *D*-dimensionales donde todas sus caras tienen forma de rectángulo. Así, si un problema dado se presenta en dos dimensiones se trabaja simplemente con rectángulos y si es en tres dimensiones se hace con los cuerpos en el espacio conocidos como ortoedros o cuboides.
Los límites de un hiper-rectángulo en cada una de las dimensiones del espacio son utilizados como descriptores de un determinado conjunto de datos. La principal ventaja de trabajar con hiper-rectángulos es que cada una de estas caras presenta un único valor en una dimensión determinada. En cada eje del espacio un hiper-rectángulo tiene dos caras ortogonales a dicho eje, por lo tanto cada cara tiene un valor distinto, así es posible utilizar estos dos valores para definir los límites del hiper-rectángulo en la dimensión correspondiente. De esta manera, cada dimensión queda definida dentro de un intervalo cerrado y los límites de todo el hiper-rectángulo queda definido por todos los intervalos de cada una de las dimensiones del espacio. A partir de un hiper-rectángulo con las características mencionadas es posible describir sus límites utilizando el lenguaje natural.
En el espacio *D*-dimensional dos o más hiper-rectángulos pueden presentar una superposición la cual no es deseable ya que, si ese caso ocurre, un mismo dato podría pertenecer a más de una clase al estar incluido en ambos hiper-rectángulos. El objetivo de la técnica propuesta en esta tesis es el de reducir las superposiciones existentes entre hiper-rectángulos de distintas clases y en el mejor de los casos eliminar todas las superposiciones existentes.

### 2.1. Índices de superposición

La eliminación de una superposición entre dos hiper-rectángulos se hace modificándolos en una dimensión en particular, ya sea por la división de uno de ellos o por la reducción de volumen de uno o

ambos. La elección de la dimensión en la cual llevar a cabo la tarea de ajuste se basa en el cálculo de una serie de índices de superposición, donde cada uno de ellos mide un aspecto diferente de la superposición. Estos índices se combinan en un único valor para cada dimensión de manera que el índice con el valor más alto determina la dimensión por la cual llevar a cabo el ajuste.

Cada índice es calculado para todas las dimensiones. Para llevar a cabo el cálculo del índice se utilizan los valores mínimos y máximos de los hiper-rectángulos y los valores de los datos de ambas clases presentes en la superposición. De esta manera se define una superposición de área que queda determinada por los límites de la superposición y una superposición de datos que queda determinada por los datos incluidos en la superposición.

En esta tesis se proponen seis índices los cuales miden diferentes aspectos de una superposición. Cada uno de estos índices, denominados índices *Z*, son calculados para los dos hiper-rectángulos presentes en una superposición determinada y se calcula en cada una de las dimensiones del espacio del problema.

Los seis índices están diseñados para devolver un valor entre 0 y 1. El índice calculado en una dimensión vale 0 cuando en esa dimensión no exista un aporte importante para la decisión de dividir los hiper-rectángulos y vale 1 cuando esa dimensión es candidata para realizar la división.

Finalmente, se utilizan los valores calculados de los índices *Z* para el cálculo del índice Ω en cada una de las dimensiones. Este índice Ω será utilizado para determinar el grado de separabilidad de los hiper-rectángulos en dicha dimensión. Por lo tanto, la dimensión en la cual realizar el ajuste de hiper-rectángulos queda determinada por el índice Ω con mayor valor.

En la técnica de clasificación presentada en esta tesis se propone utilizar como índice Ω el promedio de los seis índices *Z*. Este índice Ω, al ser calculado como un promedio, también dará un único valor en el intervalo [0, 1] con la particularidad de que el valor 1 significa que es una dimensión candidata para llevar a cabo la división y 0 lo contrario.

De esta manera, se calcula un índice Ω*i* para cada dimensión *i* el cual es calculado además para cada uno de los dos hiper-rectángulos presentes en la superposición, calculando así pares de índices Ω en todas las superposiciones presentes en el modelo de datos. De todos los índices Ω*i* calculados, aquel con mayor valor determina el hiper-rectángulo a dividir y en que dimensión hacer la división.

## 3. CLUHR

La técnica propuesta en esta tesis, denominada CLUHR [1] (Clasificación utilizando hiper-rectángulos), permite, una vez que se consigue armar el modelo de datos, extraer como conocimiento reglas de clasificación, las cuales se obtienen de los hiper-rectángulos formados. Es posible encuadrar esta técnica en la categoría de los algoritmos de “divide y vencerás”, ya que dicho algoritmo comienza formando un único hiper-rectángulo para cada clase y luego mediante la división de estos logra armar el modelo de datos. Finalizado el modelo de datos es posible extraer un conjunto de reglas de clasificación, este conjunto representa a un conjunto de reglas por cobertura, es decir, no todo el espacio está cubierto por las reglas, e incluso y dependiendo de como se configure el algoritmo para el armado del modelo, algunas de ellas podrían cubrir una misma parte del espacio de los datos.

CLUHR es una técnica determinista y el resultado solo depende de la configuración elegida por el usuario al momento de armar el modelo. Esta es una de las principales ventajas de esta técnica ya que una misma entrada, con la misma configuración del algoritmo, produce siempre la misma salida.

La filosofía del algoritmo de CLUHR para el armado del modelo de datos es la siguiente: *mientras existan superposiciones de hiper-rectángulos, eliminarlas*.

El algoritmo tiene una etapa de inicialización en la cual crea, para cada clase de datos presente en la base de datos un hiper-rectángulo. De esta manera, el algoritmo comienza con tantos hiper-rectángulos como clases haya en la base de datos.

El proceso de clasificación y armado del modelo de datos consiste en un proceso iterativo en el cual se buscan todas las superposiciones existentes entre dos hiper-rectángulos de distintas clases. Luego se procede a realizar el cálculo de los índices Ω para cada uno de los hiper-rectángulos presentes en las superposiciones encontradas y en todas las dimensiones del espacio del problema, para luego buscar aquella superposición en la cual el índice Ω tenga el valor más alto. En la superposición hallada se realiza el ajuste de los hiper-rectángulos involucrados dividiendo al que tiene el valor Ω más alto. Finalmente se ajustan los hiper-rectángulos que fueron modificados y los nuevos hiper-rectángulos que fueran creados a sus correspondientes datos formando así nuevos hiper-rectángulos.

El algoritmo vuelve a buscar superposiciones entre todos los hiper-rectángulos y se repite el proceso anterior. Este proceso iterativo continúa hasta eliminar todas las superposiciones existentes.

Al finalizar el algoritmo, cada clase tendrá uno o más hiper-rectángulos representativos y cada uno de ellos describe una regla de clasificación para la clase.

Es interesante mencionar que la división de los hiper-rectángulos puede resultar en un hiper-rectángulo el cual represente a un único dato, en cuyo caso se obtiene una regla de clasificación para ese único dato. Para lidiar con este problema, el algoritmo posee un parámetro μ cuya tarea es la de determinar que superposiciones se analizan cuando la cantidad de datos involucrados de una o ambas clases es mayor que el valor especificado en μ.

Con el uso de este parámetro, el algoritmo solo elimina aquellas superposiciones que tienen una cantidad mínima de datos participantes dejando sin modificar las que el experto o el usuario del modelo consideran insignificantes. Obviamente, el no eliminar estas superposiciones con pocos datos produce como resultado un modelo de datos menos preciso, por lo que el uso de este parámetro resulta clave para encontrar un equilibrio entre obtener un modelo de datos preciso versus una cantidad mínima de reglas de clasificación.

### 3.1. Extracción de las reglas

Cuando el algoritmo de clasificación finaliza su tarea, se obtiene como resultado un conjunto de hiper-rectángulos donde cada uno de ellos representa a un subconjunto de datos de una clase específica. De cada uno de los hiper-rectángulos se extrae una regla de clasificación del tipo IF-THEN, esta regla de clasificación tendrá la siguiente forma:

$$\text{IF } ((x_1 \geq Hn_1) \text{ AND } (x_1 \leq Hx_1) \text{ AND } \ldots \text{ AND } (x_D \geq Hn_D) \text{ AND } (x_D \leq Hx_D)) \text{ THEN } C$$

donde $Hn_i$ y $Hx_i$ son los valores mínimos y máximos en la dimensión $i$ del hiper-rectángulo $H$ respectivamente.

Esta regla estará formada por $2*D$ cláusulas y como es fácil imaginar la utilización de este tipo de reglas puede resultar muy engorroso para el usuario, en especial si la dimensión del problema a tratar es alta. Por lo tanto es necesario un proceso post-clasificación que simplifique al máximo la complejidad de estas estas reglas de clasificación. Este proceso consiste en eliminar las cláusulas que no son necesarias

para la descripción del problema. Así, se extrae un conjunto de reglas simplificadas que tienen la mínima cantidad de cláusulas necesarias para describir el conjunto de datos del propio hiper-rectángulo sin que se presente contradicción o ambigüedad con otra regla de otro hiper-rectángulo de otra clase.

### 3.2. Intervención del experto

El algoritmo de CLUHR, para llevar a cabo el armado del modelo de datos, debe tomar muchas decisiones. La decisión de que superposición resolver es decidida mediante el cálculo de los índices Ω. Es posible calcular este índice de mucha formas y una vez que se estableció que superposición eliminar surge la duda de si hay que dividir los hiper-rectángulos, disminuir su volumen, cual dividir o como llevar a cabo la disminución. En esta dirección, la técnica propuesta en esta tesis no determina ninguna decisión en particular, ya que esta depende fuertemente del problema.

De todas formas, es posible tomar todas estas decisiones previo al armado del modelo, configurar el algoritmo para que lleve a cabo el armado y ejecutarlo de manera totalmente automática para lograr el resultado. Aunque la técnica propuesta en esta tesis es completamente flexible en el sentido que un experto, previo a la ejecución de proceso, puede determinar que índices utilizar y como llevar a cabo las divisiones y así personalizar el proceso automático, en problemas reales nunca es posible armar un proceso automático que sea capaz de mejorar a las decisiones “on-line” que tomaría un humano, más cuando este es experto en el domino del problema.

En esta dirección, la técnica presentada en esta tesis presenta la particularidad de ser utilizada como un proceso totalmente automático o un proceso interactivo mediante la intervención de un experto en el dominio del problema

### 3.3. Adaptabilidad del modelo

Desde hace varios años se han propuesto varias técnicas en las cuales un modelo de datos ya armado actualiza su estructura interna ante los cambios producidos en la base de datos. Estas técnicas, incluidas en el área de la minería de datos incremental, se han aplicado a distintas tareas entre las que se incluyen clustering, frequent pattern mining y clasificación. La principal ventaja que presentan estas técnicas es que el modelo de datos armado es adaptado a los nuevos cambios sin necesidad de llevar a cabo todo el proceso desde cero, logrando así una técnica mucho más eficiente, ya que, en vez de re-entrenar el modelo de datos con la base de datos completa, sólo se modifica aquella parte de la estructura interna del modelo de datos que es afectada ante el cambio de la base de datos.

En esta dirección, la técnica propuesta en esta tesis es capaz de actualizar su estructura interna de manera eficiente ante la aparición de nuevos datos y ante la eliminación o modificación de los datos ya representados por el modelo.

Una técnica que es adaptable posee la capacidad de realizar con muy poco esfuerzo computacional las modificaciones pertinentes en su estructura interna para reflejar los cambios de los datos. La capacidad de adaptación logra un proceso de extracción de conocimiento más eficiente y transparente en la tarea de clasificación. Esto hace que resulte una técnica más eficiente y amigable para el usuario por las modificaciones mínimas necesarias que se realizan. Además es transparente, ya que el propio usuario puede ver qué parte del modelo sufre los cambios, y en el caso puntual de CLUHR que reglas de clasificación sufren modificación.

La adaptabilidad llevada a cabo por CLUHR es "on-line", realiza solo mínimas modificaciones en su estructura interna y, por lo tanto, el conocimiento adquirido previamente "sufre" pequeños cambios.

## 4. Resultados

CLUHR ha sido comparado contra otras técnicas de obtención de reglas utilizando bases de datos del repositorio UCI. Las técnicas contra las cuales se compararon los resultados fueron el clásico C4.5 [2], una estrategia que, al igual que la propuesta, utiliza hiper-rectángulos junto con un algoritmo evolutivo para obtener las reglas de clasificación [3] y una estrategia que utiliza PSO junto con ACO para obtener el conjunto de reglas de clasificación [4].

Se compara la precisión promedio alcanzada por el modelo de datos (tabla 1), la cantidad de reglas extraídas (tabla 2), la cantidad promedio de cláusulas de cada una de las reglas (tabla 3) y el número de veces que se recorre la base de datos durante el armado del modelo (tabla 4).

En la estrategia propuesta se cuenta como número de reglas a la cantidad de hiper-rectángulos del modelo de datos. En el algoritmo C4.5 de cada rama del árbol se extrae una regla de clasificación y la longitud de la rama determina el número de cláusulas. El método de PSO/ACO2 arroja como resultado el conjunto de reglas de clasificación.

Dado que la precisión de un modelo de datos y la cantidad de reglas son resultados inversamente proporcionales entre sí, ya que se puede lograr una muy buena precisión pero con una gran cantidad de reglas, y al mismo tiempo se puede obtener un número muy reducido de reglas pero con una mala precisión, en los ensayos realizados con CLUHR y con C4.5 se eligieron los parámetros correspondientes para lograr un equilibrio razonable entre precisión y cantidad de reglas y así poder comparar contra los resultados publicados en [4].

*Tabla 1. Exactitud del modelo logrados por cada una de las estrategias medidas y para cada una de las bases de datos ensayadas. Se presentan las medias y entre paréntesis el desvío estándar.*

| | *C4.5* | *EHS-CHC* | *PSO/ACO2* | *CLUHR* |
|---|---|---|---|---|
| Ecoli | 0,7964 (0,0141) | 0,7948 | - | 0,7891 (0,0160) |
| Glass | 0,6576 (0,0302) | 0,6287 | 0,7095 (0,075) | 0,6215 (0,0360) |
| Haberman | 0,7103 (0,0202) | 0,7122 | - | 0,7356 (0,0064) |
| Image | 0,8586 (0,0155) | - | 0,9667 (0,0117) | 0,8538 (0,0135) |
| Ionosphere | 0,9054 (0,0151) | - | 0,8806 (0,0491) | 0,8777 (0,0169) |
| Iris | 0,9420 (0,0077) | 0,9267 | 0,9467 (0,0526) | 0,9300 (0,0079) |
| Liver | 0,6418 (0,0300) | 0,6167 | - | 0,5918 (0,0211) |
| Pima | 0,7434 (0,0093) | 0,7384 | - | 0,5595 (0,0191) |
| Sonar | 0,7235 (0,0247) | 0,7650 | 0,7505 (0,0911) | 0,6666 (0,0283) |
| Vehicle | 0,7111 (0,0099) | - | 0,7305 (0,0445) | 0,6819 (0,0171) |
| Vowel | 0,6008 (0,0158) | - | 0,8616 (0,0347) | 0,7120 (0,0132) |
| Wine | 0,9141 (0,0145) | 0,9490 | - | 0,9530 (0,0113) |
| Wisconsin | 0,9446 (0,0047) | 0,9599 | 0,9487 (0,0253) | 0,9251 (0,0102) |
| Forest covertype | 0,7063 (0,0187) | - | - | 0,6928 (0,0149) |

*Tabla 2. Cantidad de reglas extraídas por cada una de las estrategias medidas y para cada una de las bases de datos ensayadas. Se presentan las medias y entre paréntesis el desvío estándar.*

| | *C4.5* | *EHS-CHC* | *PSO/ACO2* | *CLUHR* |
|---|---|---|---|---|
| Ecoli | 12,1 (1,45) | 11,1 | - | 12,62 (1,44) |
| Glass | 14,8 (0,79) | 12,2 | 20,4 (1,35) | 15,17 (1,30) |
| Haberman | 10,7 (3,62) | 4,4 | - | 4,29 (0,33) |
| Image | 10,6 (0,70) | - | 21,9 (0,99) | 10,93 (0,47) |
| Ionosphere | 10,2 (2,04) | - | 3,6 (0,97) | 3,98 (0,37) |
| Iris | 4,0 (0,47) | 3,4 | 3,0 (0,00) | 3,21 (0,12) |
| Liver | 23,9 (4,46) | 9,8 | - | 17,79 (2,21) |
| Pima | 13,2 (1,40) | 11 | - | 10,45 (0,91) |
| Sonar | 10,9 (1,60) | 10,3 | 4,4 (1,58) | 4,14 (0,20) |
| Vehicle | 31,0 (2,31) | - | 37,8 (1,2) | 32,35 (2,03) |
| Vowel | 32,8 (2,20) | - | 29,0 (0,82) | 31,74 (0,78) |
| Wine | 5,1 (0,57) | 3,6 | - | 3,18 (0,11) |
| Wisconsin | 11,9 (1,79) | 3,8 | 10,2 (1,87) | 9,63 (1,39) |
| Forest covertype | 39,7 (2,35) | - | - | 41,25 (2,05) |

Debido a que las ejecuciones del algoritmo C4.5 fueron llevadas a cabo para elaborar esta tesis y que los autores de [4] publican en sus resultados la media, desvío estándar y n para cada base de datos, se realiza una prueba *t*-student de doble cola con nivel de confianza del 95% para determinar si las diferencias logradas entre CLUHR y C4.5 y PSO/ACO2 son estadísticamente significativas o no. La tabla 5 muestra los resultados obtenidos. En dicha tabla se marca con un signo "+" cuando CLUHR es mejor estadísticamente, con un signo "-" cuando CLUHR es peor estadísticamente y con un signo "=" cuando no hay diferencias significativas.

En la tabla 4 las estrategias EHS-CHC y PSO/ACO2 no aparecen por desconocer la verdadera cantidad de veces que se recorre. Aunque, como se detalla en esta tesis, la estrategia EHS-CHC se estima que la recorre 10000 veces y en PSO/ACO2 se estima un promedio de 3000 veces. De esta misma tabla puede observarse que CLUHR recorre menos veces la base de datos comparado con C4.5. Se realizó un test t-student de doble cola al 95% y en la tabla figura un símbolo "+" cuando CLUHR es mejor estadísticamente, un símbolo "-" cuando CLUHR es peor estadísticamente y con un símbolo "=" cuando la diferencia no resulta estadísticamente significante. Hay casos donde el número de veces es similar, pero en otras se recorre una, dos y hasta tres veces menos que las recorridas por C4.5. Solo en dos bases de datos CLUHR necesitó recorrer más veces la base de datos que C4.5.

Haciendo un estudio de los distintos resultados obtenidos no es posible determinar que CLUHR se destaque sobre el resto, tampoco lo contrario, que CLUHR sea una técnica mala comparada contra el resto. Comparado con C4.5 los resultados han sido muy similares mientras que PSO/ACO2 parecería lograr un número reducido no solo de reglas, sino también de cláusulas por regla. Esto último se debe a dos factores claves que presenta la propia estrategia. 1) al tener ordenado el conjunto de reglas permite eliminar muchas cláusulas de las mismas. 2) al ser una estrategia de optimización, las partículas de PSO recorren todo el espacio de búsqueda encontrando una solución óptima.

Comparando la precisión, el número de reglas y el promedio de cláusulas por regla podemos afirmar que CLUHR es equivalente al resto de las técnicas estudiadas produciendo resultados similares a los que arrojan C4.5 y PSO/ACO2. Sin embargo, si se analiza el costo computacional de cada método para alcanzar el resultado, puede afirmarse que CLUHR es el mejor.

En resumen, CLUHR demostró resolver todos los problemas estudiados ofreciendo resultados similares a las estrategias comparadas, pero con una utilización de recursos similar a la que presenta C4.5 y mucho menor a los utilizados por EHS-CHC y PSO/ACO2.

*Tabla 3. Número promedio de cláusula por regla por cada una de las estrategias medidas y para cada una de las bases de datos. Se presentan las medias y entre paréntesis el desvío estándar.*

| | *C4.5* | *PSO/ACO2* | *CLUHR* |
|---|---|---|---|
| Ecoli | 4,32 (0,30) | - | 4,65 (0,15) |
| Glass | 5,68 (0,75) | 3,11 (0,18) | 5,37 (0,18) |
| Haberman | 4,54 (1,27) | - | 2,54 (0,06) |
| Image | 4,31 (0,58) | 2,8 (0,27) | 3,74 (0,10) |
| Ionosphere | 5,36 (0,89) | 3,33 (0,79) | 5,17 (0,18) |
| Iris | 2,25 (0,27) | 0,93 (0,14) | 2,08 (0,05) |
| Liver | 6,80 (1,30) | - | 5,01 (0,06) |
| Pima | 4,55 (0,27) | - | 5,27 (0,12) |
| Sonar | 3,99 (0,43) | 2,6 (0,63) | 16,27 (0,72) |
| Vehicle | 7,10 (0,34) | 3,85 (0,18) | 7,38 (0,33) |
| Vowel | 5,69 (0,18) | 4,2 (0,25) | 8,13 (0,27) |
| Wine | 2,46 (0,17) | - | 4,08 (0,09) |
| Wisconsin | 4,31 (0,39) | 1,21 (0,07) | 3,59 (0,11) |
| Forest covertype | 6,67 (0,82) | - | 6,49 (0,48) |

*Tabla 4. Número de veces que se recorre la base de datos por CLUHR y C4.5.*

| | *C4.5* | *CLUHR* | *Significancia* |
|---|---|---|---|
| Ecoli | 4,19 (0,39) | 3,53 (0,33) | + |
| Glass | 5,64 (1,10) | 3,97 (0,37) | + |
| Haberman | 3,61 (1,26) | 5,28 (0,32) | - |
| Image | 3,84 (0,35) | 1,67 (0,06) | + |
| Ionosphere | 5,78 (0,73) | 2,47 (0,14) | + |
| Iris | 2,02 (0,13) | 1,5 (0,06) | + |
| Liver | 6,59 (1,48) | 5,20 (0,50) | + |
| Pima | 3,74 (0,24) | 4,97 (0,39) | - |
| Sonar | 4,03 (0,49) | 2,41 (0,17) | + |
| Vehicle | 5,98 (0,24) | 5,06 (2,56) | = |
| Vowel | 5,54 (0,13) | 2,99 (0,11) | + |
| Wine | 2,34 (0,10) | 1,20 (0,01) | + |
| Wisconsin | 3,19 (0,35) | 3,02 (0,32) | = |
| Forest covertype | 5,71 (0,72) | 5,24 (0,45) | = |
| Total | | | +7 |

*Tabla 5. Resultados de la prueba t-student para determinar si hay diferencias significativas entre los resultados obtenidos por CLUHR y los obtenidos por C4.5 y PSO/ACO2.*

| | *Exactitud* | | *Número de reglas* | | *Promedio de cláusulas por regla* | |
|---|---|---|---|---|---|---|
| | C4.5 | PSO/AC O2 | C4.5 | PSO/AC O2 | C4.5 | PSO/AC O2 |
| Ecoli | = | | = | | - | |
| Glass | - | - | = | + | = | - |
| Haberman | + | | + | | + | |
| Image | = | - | = | + | + | - |
| Ionosphere | + | = | + | = | = | - |
| Iris | - | = | + | - | = | - |
| Liver | - | | + | | + | |
| Pima | - | | + | | - | |
| Sonar | - | - | + | + | - | - |
| Vehicle | - | - | = | = | - | - |
| Vowel | + | - | + | - | - | - |
| Wine | + | | + | | - | |
| Wisconsin | - | - | + | + | + | - |
| Forest covertype | = | | - | | = | |
| Total | -3 | -6 | +8 | +2 | -2 | -8 |

En cuanto a la característica de estrategia incremental CLUHR es comparado contra la técnica ITI [5].
El principal problema que presenta la técnica ITI, como cualquier técnica basada en árboles de decisión es que la acumulación de datos y la re-evaluación de la función de decisión de los nodos provoca que tarde o temprano se requiera una re-estructuración de un sub-árbol. Cuando el sub-árbol a rehacer tiene como raíz un nodo de los primeros niveles, entonces esta re-estructuración es importante, ya que la re-construcción de uno de estos sub-árboles representa recorrer un importante porcentaje de la base de datos.
En CLUHR, la aparición de nuevos datos, sólo causa que se modifique un hiper-rectángulo. Y si este tiene superposiciones con otros hiper-rectángulos entonces, se modifican los hiper-rectángulos involucrados.
Para comparar estos métodos se mide la cantidad de veces que se recorre la base de datos para la modificación del modelo. Se realizan 10 ejecuciones independientes para ambas técnicas, CLUHR e ITI. En cada ejecución se divide la base de datos en dos conjuntos al azar, el primero con el 70% de los datos y el segundo con el 30% restante. Con el primer sub-conjunto de datos se arma un modelo, tanto para CLUHR como para ITI y, con los datos del segundo-subconjunto se presentan uno por uno al modelo armado. Ante cada dato presentado se cuenta la cantidad de veces que se recorre la base de datos. Por lo tanto, para una ejecución, la cantidad total de veces que se recorre la base de datos es la suma de las veces que se recorre para cada dato.
La tabla 6 muestra los resultados comparativos entre el esfuerzo computacional que necesita CLUHR y el que necesita la técnica ITI. Se realizó un test t-student de doble cola al 95% y en la tabla figura un

símbolo "+" cuando CLUHR es mejor estadísticamente, un símbolo "-" cuando CLUHR es peor estadísticamente y con un símbolo "=" cuando la diferencia no resulta estadísticamente significante.
Como puede observarse CLUHR requiere mucho menos esfuerzo computacional que el que necesitan los árboles de decisión basados en el algoritmo ITI. CLUHR fue muy superior en 10 bases de datos.

*Tabla 6. Recursos utilizados (cantidad de veces que se recorre la base de datos) por CLUHR e ITI en cada una de las bases de datos.*

| | *ITI* | *CLUHR* | *Significancia* |
|---|---|---|---|
| Ecoli | 5,19 (0,95) | 0,59 (0,45) | + |
| Glass | 14,50 (2,56) | 0,44 (0,11) | + |
| Haberman | 12,84 (2,24) | 0,99 (0,25) | + |
| Image | 2,37 (0,44) | 0,19 (0,15) | + |
| Ionosphere | 1,71 (0,30) | 1,59 (0,43) | = |
| Iris | 0,25 (0,05) | 0,90 (0,50) | - |
| Liver | 14,58 (2,70) | 0,61 (0,16) | + |
| Pima | 21,53 (3,69) | 1,31 (0,37) | + |
| Sonar | 5,11 (0,90) | 0,06 (0,05) | + |
| Vehicle | 14,42 (2,48) | 1,07 (0,32) | + |
| Vowel | 31,20 (5,30) | 0,11 (0,05) | + |
| Wine | 1,32 (0,26) | 0,26 (0,41) | + |
| Wisconsin | 1,65 (0,30) | 8,31 (2,11) | - |
| Total | | | +8 |

## 5. Conclusiones y Trabajo a futuro

CLUHR ha demostrado ser una poderosa estrategia para clasificación y extracción de conocimiento en donde se destacan las siguientes características:

- Ofrece similares resultados en cuanto a precisión, cantidad de reglas y cantidad de cláusulas por regla que se obtienen con otras estrategias de clasificación.
- Es una estrategia determinista que permite obtener el mismo resultado con la misma entrada.
- Consume mucho menos recursos que las estrategias de optimización y un número ligeramente menor a otras estrategias deterministas como C4.5.
- El algoritmo puede ser ejecutado de manera totalmente automática o totalmente supervisada pudiendo un experto en el dominio del problema participar en todas las decisiones que debe tomar la estrategia para lograr un modelo de datos.
- La estrategia es completamente personalizable pudiéndose elegir que índices de superposición utilizar a lo largo del procedimiento de armado del modelo.
- El algoritmo tiene un único parámetro lo que lo hace sencillo de utilizar. Además, mediante sus cálculos es posible obtener diferentes resultados con diferentes valores de este parámetro con el mismo costo computacional de una sola ejecución del algoritmo.
- Es una estrategia adaptativa, esto permite modificar su estructura interna con poco esfuerzo sin llevar a cabo una ejecución completa del armado del modelo. Esta adaptación puede ser usada de manera automática o supervisada por un experto.

Aún con todas las ventajas que presenta la estrategia propuesta quedan varios aspectos en los cuales se puede seguir investigando. Los mismos son detallados a continuación.

- En cuanto a los índices de superposición es posible seguir investigando el desarrollo de nuevos índices que midan otros aspectos de una superposición, como por ejemplo índices que midan características de más de dos clases al mismo tiempo o que midan aspectos de más de una dimensión a la vez.
- En ciertas situaciones es posible unir los hiper-rectángulos más pequeños de una clase en un único hiper-rectángulo, logrando así menos reglas en el modelo de datos. Es interesante investigar si las uniones de los hiper-rectángulos de una misma clase pueden ser llevadas a cabo durante el proceso de armado del modelo o es un tratamiento extra que debería hacerse en la etapa final de extracción de reglas simplificadas. El lograr mejorar este aspecto favorecería al procedimiento de extracción de reglas simplificadas, ya que este procedimiento podrá trabajar con una menor cantidad de reglas.
- Una de los puntos débiles de CLUHR es que el proceso final de extracción del conjunto de reglas simplificadas utiliza un procedimiento greedy del orden $O(n^2)$. Resulta interesante investigar de qué manera "marcar" aquellas caras de los hiper-rectángulos que representen un límite en el espacio de los datos. Si es posible identificar aquellas caras que representan los límites del espacio de datos se producirá la exclusión directa de la cláusula para la correspondiente regla sin necesidad de llevar a cabo el método de simplificación post-procesamiento.
- Otro aspecto débil en la estrategia propuesta es que solo es capaz de trabajar en dominios donde todos los atributos son numéricos. En esta dirección sería interesante poder trabajar de manera más confiable con atributos numéricos de pocos valores y con la posibilidad de adaptar la estructura de los hiper-rectángulos al domino de datos nominales. Esto podría obtenerse quizás trabajando con conjuntos de datos nominales asociados a los hiper-rectángulos.
- Un experto en el dominio del problema puede supervisar el armado del modelo de datos participando en las decisiones que debe tomar la propia estrategia al momento de eliminar una superposición. Resulta útil contar con una herramienta que visualice las características de las superposiciones para que el experto pueda tomar una decisión acorde al problema.

**Bibliografía**

[1] Hasperué, W.; Lanzarini, L.; De Guisti. 2012. Rule Extraction on Numeric Datasets Using Hyper-rectangles. A Computer and Information Science. Vol. 5, No 4, pp. 116 131.

[2] Quinlan, J. R. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc., 1993. ISBN 1-55860-238-0.

[3] Garcia, S., Derrac, J., Luengo, J., & Herrera, F. 2009. A First Approach to Nearest Hyperrectangle Selection by Evolutionary Algorithms. Intelligent Systems Design and Applications Ninth International Conference on, 517-522. http://dx.doi.org/10.1109/ISDA.2009.238

[4] Holden, N., & Freitas, A. A. 2008. A hybrid PSO/ACO algorithm for discovering classification rules in data mining. Journal of Artificial Evolution and Applications, 2008. 1-11.

[5] Utgoff P. E. 1996. Decision Tree Induction Based on Efficient Tree Restructuring. Machine Learning. Vol. 29. - págs. 5-44.