

Centro Nacional de Investigación y Desarrollo Tecnológico

Subdirección Académica

Departamento de Ciencias Computacionales

TESIS DE MAESTRÍA EN CIENCIAS

**Definición y Aplicación de Heurística Monotónica para Evaluar el
Algoritmo "Branch & Bound" para Selección de Atributos**

presentada por
Ing. Omar Guzmán Ruiz

como requisito para la obtención del grado de
Maestro en Ciencias de la Computación

Director de tesis
Dr. Manuel Mejía Lavalle

Dedicatoria

A mis padres por su apoyo moral y económico.

Agradecimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por brindarme el apoyo económico para realizar mis estudios.

Al Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) por admitirme en su institución.

A mi director de tesis el Dr. Manuel Mejía Lavalle por el apoyo y asesoría para la realización de esta tesis.

A mis revisores el Dr. Raúl Pinto Elías y el Dr. José Ruiz Ascencio, por sus valiosos comentarios y consejos.

A mis compañeros: Alfredo, Alma, Andres, Balam, Bernardo, Marcos, Miguel, Ramiro, Selene, Vicente y Vitervo, gracias por su amistad.

Resumen

El trabajo realizado en este documento trata acerca de la reducción de dimensionalidad de las bases de datos para el aprendizaje supervisado. Uno de los problemas fundamentales en áreas como minería de datos, aprendizaje automático o reconocimiento de patrones es la selección de atributos o reducción de atributos en la cual se busca resolver el problema de encontrar los atributos más relevantes, ya que puede darse el caso en tareas de clasificación que aumente la precisión del clasificador con el tamaño reducido de atributos, en lugar de utilizar todo el conjunto original.

Existen en la literatura diversos artículos para el problema de selección de atributos, varios de ellos mencionan que uno de los mejores métodos de selección de atributos del subconjunto óptimo es el "*Branch & Bound*", si la heurística empleada es monótonica, asegura encontrar la mejor solución sin necesidad de realizar una búsqueda exhaustiva.

En este trabajo se compara la calidad del subconjunto encontrado por el *Branch & Bound* utilizando como medida de evaluación del subconjunto dos criterios de evaluación: uno es la incertidumbre simétrica que es una versión normalizada de información mutua y el otro criterio es la medida probabilística Bhattacharyya.

Se presenta un conjunto de pruebas sobre bases de datos reales que muestran el tiempo de procesado y la precisión del clasificador para la selección de atributos hecha por el *Branch & Bound* contra diferentes herramientas de selección de atributos.

Abstract

The work done in this paper deals with the dimensionality reduction of databases for supervised learning. One of the fundamental problems in areas such as data mining, machine learning or pattern recognition is feature selection or feature reduction in which it is sought to solve the problem of finding the most relevant features, as it may be the case in tasks that increase the accuracy of the classifier with reduced feature size, rather than using the entire original set.

In the literature there are some papers for the feature selection problem, several of them mention that one of the best feature selection methods for optimal subset is the "Branch & Bound", if the heuristic used is monotonic, it ensures you find the best solution without having to perform an exhaustive search

In this paper we compare the effectiveness of the subset found by Branch & Bound using as a measure of evaluation of the subset two criterion functions: one is the symmetric uncertainty which is a normalized version of mutual information and the other criterion is the Bhattacharyya probabilistic measure.

We present a set of tests on real-world databases that show the processing time and the classifier accuracy for feature selection made by Branch & Bound against different feature selection tools

Índice de contenido

Capítulo 1 Introducción	1
1.1. Motivación	1
1.2. Descripción del problema	2
1.2.1. Delimitación del problema específico.....	2
1.2.2. Complejidad del problema	2
1.3. Objetivo general y específico.....	3
1.3.1. Objetivo general	3
1.3.2. Objetivos específicos.....	3
1.4. Alcances y limitaciones del proyecto	3
1.5. Entregables	4
1.6. Justificación y beneficios	4
1.7. Metodología de la solución	4
1.8. Organización de la tesis	5
Capítulo 2 Marco Teórico	6
2.1 Reducción de dimensionalidad.....	6
2.2 Conceptos de selección de atributos	7
2.2.1 Atributo.....	7
2.2.2 Descriptor	8
2.2.3 Instancia.....	8
2.2.4 Conjunto de datos	8
2.2.5 Selección de atributos	9
2.2.6 Relevancia de atributos	10
2.3 Relevancia y optimalidad de atributos	11
2.4 Eficiencia y optimalidad de un algoritmo.....	12
2.5 Medidas de evaluación usadas para encontrar atributos relevantes.....	13
2.6 Metodos de selección univariados vs multivariados	16
2.7 Taxonomía de algoritmos de selección de atributos	17
2.7.1 Categorización por la naturaleza de los atributos resultantes	17
2.7.2 Categorización de acuerdo a su propósito (objetivo)	17
2.7.3 Categorización con respecto a la optimalidad.....	18
2.7.4 Categorización con respecto al criterio de selección	18

2.7.5	Caracterización con respecto al conocimiento del problema.....	20
2.8	Selección de atributos visto desde el enfoque lógico combinatorio	21
2.8.1	Calculando todos los testores típicos.....	23
2.8.2	Establecer los criterios de comparación de los atributos.....	23
2.8.3	Matriz de diferencias MD	24
2.8.4	Matriz básica MB	24
2.8.5	Algoritmo BT.....	25
2.8.6	Medida de la importancia informacional de los atributos (relevancia)	28
2.9	Branch & Bound (B&B).....	30
2.9.1	Heurística	31
2.9.2	Criterio de monotonía	31
2.9.3	Criterio de monotonía para el B&B	32
2.10	Discusión.....	32
Capítulo 3 Estado del arte y de la práctica		33
3.1	Estado del arte	33
3.1.1	Antecedentes	33
3.1.2	Trabajos relacionados	38
3.1.3	Estado de la práctica	42
3.2	Discusión	48
Capítulo 4 Análisis, diseño e implementación del sistema		49
4.1.	Análisis del B&B	49
4.2.	Observaciones, detalles y desventajas del B&B	56
4.3.	Diseño del Sistema	57
4.3.1.	Medidas de evaluación escogidas	58
4.3.2.	Diseño del criterio de evaluación para seleccionar atributos relevantes	59
4.4.	Implementación del sistema	61
4.4.1.	Base de datos de imágenes	61
4.4.2.	Representación de imágenes por sus descriptores	61
4.4.3.	Reducción de atributos por el B&B	62
4.4.4.	Subconjunto de atributos obtenido por el B&B.....	62
4.4.5.	Descripción de imágenes con el subconjunto obtenido por el B&B.....	62

4.4.6. Comparación del subconjunto de atributos encontrado por el B&B y otros sistemas.....	62
4.5. Discusión.....	63
Capítulo 5 Experimentación y resultados	64
5.1. Detalles y objetivos de la experimentación	64
5.1.1. Hardware.....	64
5.1.2. Bases de datos.....	64
5.1.3. Sistemas de selección de atributos.....	64
5.2. Pruebas de calidad y eficiencia	65
5.2.1. Pruebas de calidad.....	65
5.3. Pruebas de eficiencia	87
5.4. Discusión.....	90
Capítulo 6 Conclusiones	91
6.1. Objetivos logrados.....	91
6.2. Productos.....	93
6.3. Conclusiones.....	94
6.4. Trabajo futuro	95
Referencias	96
Referencias bibliográficas	96
Referencias en línea	100
Anexos.....	102
Anexo A.....	102
Base de datos.....	102
Anexo B.....	107
Herramientas de SA.....	107
Anexo C.....	113
Contenido del CD.....	113

Índice de figuras

Figura 1.1. Diagrama de la metodología de la solución.....	5
Figura 2.1. Reducción de dimensionalidad de BD [Ruiz, 2006].....	7
Figura 2.2. Jerarquía de tipos de atributos.....	8
Figura 2.3. Ejemplo Base de datos de aprendizaje supervisado [Quinlan, 1993].	9
Figura 2.4. Esquema tipo filtro para selección de atributos [Kohavi, 1997].....	19
Figura 2.5. Etapas en un <i>wrapper</i> para selección de atributos [Kohavi, 1997].	20
Figura 2.6. Árbol de enumeración [Narendra, 1978].	30
Figura 4.1. Árbol de búsqueda en donde $m=2$ atributos a seleccionar de un conjunto de $n=5$ atributos [Somol, 2004].	49
Figura 4.2. Árbol de búsqueda en donde $n=5$ atributos totales y $m=2$ atributos a seleccionar, nodos de primer grado mostrados sin relleno [Yu, 1993].	51
Figura 4.3. Árbol de solución mínima [Yu, 1993] [Wang, 2003].....	52
Figura 4.4. Análisis del número de nodos en un árbol de búsqueda con $n=30$ atributos. .	54
Figura 4.5. Análisis eficiencia temporal.	55
Figura 4.6. Ilustración del mecanismo de predicción del <i>Fast Branch & Bound</i> en un problema sintético donde $m=2$ y $n=6$ [Somol, 2004].....	56
Figura 4.7. Diagrama de la metodología de la solución.....	61
Figura 4.8. Lista de descriptores implementados	62
Figura 5.1. Herramienta Elvira.	65
Figura 5.2. Herramienta Reduclas.	66
Figura 5.3. Herramienta See5.....	66
Figura 5.4. Herramienta Weka.	67
Figura 5.5. Clasificador J48 aplicado a la BD <i>Image</i>	81
Figura 5.6. Clasificador J48 con validación cruzada aplicado a la BD <i>Image</i>	82
Figura 5.7. Clasificador NB aplicado a la BD <i>Image</i>	82
Figura 5.8. Clasificador NB con validación cruzada aplicado a la BD <i>Image</i>	83
Figura 5.9. Clasificador J48 aplicado a la BD <i>Segmentation</i>	83
Figura 5.10. Clasificador J48 con validación cruzada aplicado a la BD <i>Segmentation</i>	84
Figura 5.11. Clasificador NB con validación cruzada aplicado a la BD <i>Segmentation</i>	84
Figura 5.12. Clasificador NB con validación cruzada aplicado a la BD <i>Segmentation</i>	85
Figura 5.13. Clasificador J48 aplicado a la BD WDBC.	85

Figura 5.14. Clasificador J48 con validación cruzada aplicado a la BD WDBC.	86
Figura 5.15. Clasificador NB aplicado a la BD WDBC.	86
Figura 5.16. Clasificador NB con validación cruzada aplicado a la BD WDBC.	87
Figura 6.1. Artículo publicado en el CIINDET 2014.	93
Figura B.1. Sistema Elvira.	107
Figura B.2. Sistema See5.	111
Figura B.3. Sistema Weka.	111
Figura B.4. Sistema Reduclas.	112

Índice de tablas

Tabla 2.1. Ejemplo atributos continuos y discretos.	22
Tabla 2.2. Matriz de aprendizaje.	23
Tabla 2.3. Matriz de diferencia.	24
Tabla 2.4. Matriz básica.	25
Tabla 2.5. Matriz básica.	26
Tabla 2.6. Ejemplo del algoritmo BT.	27
Tabla 3.1. Resumen de los antecedentes al proyecto de tesis.	36
Tabla 3.2. Resumen de trabajos relacionados.	40
Tabla 3.3. Resumen del estado de la práctica.	45
Tabla 4.1. Análisis del número de nodos en un árbol de búsqueda con n=30 atributos. ..	53
Tabla 4.2. Valores SU de cada par de atributos.	59
Tabla 4.3. Valores SU modificados para la métrica monotónica.	60
Tabla 5.1. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD Image, de los rankings obtenidos por la herramienta Elvira.	68
Tabla 5.2. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD Image, de los subconjuntos del B&B obtenidos con la medida de información mutua.	69
Tabla 5.3. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD Image, de los subconjuntos del B&B obtenidos con la distancia Bhattacharyya.	70
Tabla 5.4. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD Image Segmentation, de los rankings obtenidos por la herramienta Elvira.	72
Tabla 5.5. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD Image Segmentation, de los subconjuntos del B&B obtenidos con la medida de información mutua.	73

Tabla 5.6. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD Image Segmentation, de los subconjuntos del B&B obtenidos con la distancia Bhattacharyya.....	74
Tabla 5.7. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD WDBC, de los rankings obtenidos por la herramienta Elvira.....	75
Tabla 5.9. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD WDBC, de los subconjuntos del B&B obtenidos con la medida de información mutua.	77
Tabla 5.10. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD WDBC, de los subconjuntos del B&B obtenidos con la distancia Bhattacharyya.	78
Tabla 5.10. Resultados eficiencia y calidad See5 vs B&B.....	88
Tabla 5.11. Resultados eficiencia y calidad Reduclas vs B&B	89
Tabla 6.1. Objetivos en la investigación.....	91
Tabla 6.2. Alcances y limitaciones.	92

Capítulo 1

Introducción

En este capítulo se plantea el problema de selección de atributos y cómo se tratará de resolver; se muestra el objetivo general de la tesis y los objetivos específicos, los alcances y limitaciones del proyecto, la justificación y beneficios, la metodología para resolver el problema y una breve descripción del contenido de los capítulos de la tesis.

1.1. Motivación

En la actualidad los Sistemas de Bases de Datos manipulan grandes volúmenes de información. Las dificultades más habituales que se les presentan son: la captura, el almacenamiento, la búsqueda, compartición, análisis y visualización de datos.

Es claro que los conjuntos de datos están superando la capacidad del *software* (algoritmos) y *hardware* (agotamiento de la memoria, tiempos de procesamiento excesivos) y que éstos están al límite. Existe así una tendencia clara en las bases de datos que es su crecimiento, tanto en atributos como en instancias [Guyon, 2003].

Por tal motivo se necesitan algoritmos que ayuden a reducir la información (sin pérdida de su esencia) y contribuyan a descubrir los datos potencialmente útiles en algún repositorio de información.

De una manera amplia, el área de la Reducción de Datos surge debido al crecimiento de las bases de datos, lo que vuelve intratables los datos para los algoritmos de descubrimiento de conocimiento tradicionales.

La reducción de datos aplica a las instancias (normalmente los registros o renglones de la base de datos, dando origen al Muestreo de Instancias, *Sampling* ó *Sub-sampling* [Domingos, 2001]), y los atributos (las columnas de la base de datos, dando origen a la selección de atributos y transformación de atributos).

En este sentido, la Selección de Atributos (SA) juega un papel muy importante, donde hay que encontrar algoritmos adecuados de SA que tomen en cuenta el costo y eficiencia computacional para evitar la explosión combinatoria (la cual se produce cuando se busca el subconjunto de atributos que mejor representa a los datos), sin afectar la calidad de la solución.

1.2. Descripción del problema

1.2.1. Delimitación del problema específico

En el área de Visión Artificial (VA) para llevar a cabo la descripción y representación de los objetos se generan grandes volúmenes de datos, por esta razón se debe de trabajar con las características que mejor describan los objetos; una buena descripción de una imagen tiene una enorme influencia en el éxito o fracaso del reconocimiento o clasificación de ellos, es por eso que una vez obtenidas las características que describen un objeto (descriptores) es necesaria la reducción de esa información, para obtener sólo los atributos relevantes al problema en cuestión.

1.2.2. Complejidad del problema

Encontrar el subconjunto óptimo de atributos es una tarea que computacionalmente es muy costosa, ya que se tiene que probar todas las posibles combinaciones de subconjuntos y evaluarlos para saber cuál es el que mejor describe al conjunto, la solución es la búsqueda exhaustiva. Si se tiene n atributos el número de posibles combinaciones de subconjuntos sería $2^n - 1$ es una búsqueda de tipo exponencial, es fácil verificar si una respuesta es correcta, pero no se conoce mejor solución que explorar exhaustivamente los subconjuntos posibles hasta encontrar el óptimo al criterio de evaluación; incluso si se conociera el tamaño m del subconjunto óptimo, en el peor de los casos seguiría siendo una búsqueda de tipo exponencial, si se tiene un conjunto total de atributos de 50 y se quiere encontrar el mejor subconjunto que contenga 25 atributos se tendrían que generar y evaluar [Walpole, 2012] $\binom{n}{m} = \frac{n!}{m!(n-m)!} = \binom{50}{25} = 1.2641 \times 10^{14}$ posibles combinaciones sin repetición de subconjuntos; la búsqueda del subconjunto óptimo de menor tamaño posible es un problema NP-completo como lo muestra [Davies, 1994], ya que para encontrarlo se necesita la generación de todos los posibles subconjuntos de atributos; existen métodos heurísticos que encuentran el subconjunto óptimo de atributos de tamaño m , según [Fukunaga, 1990], [Mejía, 2007], [Molina, 2002], [Narendra, 1977], [Nakariyakul, 2009], [Somol, 2004] miembros de la comunidad especializada en el tema de selección de atributos, marcan que el algoritmo de “*Branch & Bound*” encuentra la solución óptima si se escoge la heurística correcta (criterio de evaluación del subconjunto monótono) garantizando encontrar el subconjunto óptimo de atributos respecto a la medida de evaluación monótona utilizada.

1.3. Objetivo general y específico

1.3.1. Objetivo general

Implementar el algoritmo de “*Branch & Bound*” (*B&B*) de selección de atributos para problemas de VA, abordando el problema de Selección de Atributos (SA) con la clasificación como principal objetivo.

1.3.2. Objetivos específicos

- Estudiar e Implementar el algoritmo *B&B* propuesto por [Narendra, 1977].
- Definir una heurística para seleccionar atributos relevantes de bases de datos de aprendizaje supervisado.
- Implementar el algoritmo *B&B* con la heurística definida.
- Comparar el subconjunto de atributos obtenido por el algoritmo *B&B* con la heurística definida con diferentes herramientas de selección de atributos.

1.4. Alcances y limitaciones del proyecto

- Estudiar el área de selección de atributos y el *B&B* original.
- Revisar los problemas típicos de Visión Artificial relacionados con selección de atributos, se van a revisar las tesis del Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) para ver los diferentes descriptores usados en ellas.
- Las bases de datos podrán tener atributos discretos con clase discreta, y atributos continuos con clase discreta.
- Implementar el *B&B* propuesto por [Narendra, 1977].
- Implementar la adaptación del *B&B* con la heurística monótona para Visión Artificial y experimentar con bases de datos de imágenes.
- Realizar un estudio comparativo.
- Utilización de un conjunto total de atributos similar a la que usa [Somol, 2004] el cual utiliza un conjunto de 30 atributos.
- Se requiere definir previamente el tamaño del subconjunto deseado de atributos.
- No se puede saber si el tamaño escogido por el usuario es el correcto ya que el algoritmo *B&B* original, sólo encuentra el óptimo del tamaño seleccionado, (siempre y cuando se escoja la heurística correcta de tipo métrica monotónica).
- En esta tesis no se modificará la restricción de *B&B* en el sentido de que en la cardinalidad del subconjunto de Atributos debe ser predefinido “*a priori*” por el usuario.

1.5. Entregables

- Documentos de revisión del estado del arte.
- Presentaciones de avance de la tesis.
- *Software* de la implementación de la herramienta de selección de atributos.
- Bases de datos usadas para el estudio.
- Estudio comparativo de resultados.
- Tesis de maestría.
- Artículo relacionado con el tema de tesis.

1.6. Justificación y beneficios

Para describir un objeto, en especial una imagen, en algunas ocasiones se tiene demasiados descriptores, esto puede hacer que los algoritmos de aprendizaje de máquinas se muestren lentos o degraden su rendimiento, algunos algoritmos de aprendizaje de máquina tal como el ID3 o C4.5 se sabe que disminuyen su precisión del clasificador cuando se utilizan muchos atributos que no son necesarios para predecir la salida deseada, otros algoritmos como el Naïve Bayes (NB) son robustos con respecto a atributos irrelevantes (es decir, su rendimiento se degrada muy lentamente cuando atributos irrelevantes son agregados), pero su rendimiento puede degradarse rápidamente si se añaden atributos correlacionados, incluso si los atributos son relevantes [Kohavi, 1997]. Si se reduce el número de descriptores y se encuentran cuáles son los que mejor representan al objeto en cuestión, puede ayudar en una mejor respuesta del algoritmo de aprendizaje automático. Los métodos heurísticos como el “*Branch & Bound*” han sido poco abordados a pesar de encontrar la solución óptima sin realizar una búsqueda exhaustiva; lo que se pretende realizar con la adaptación de este método a problemas de Visión Artificial, es tener una herramienta que ayude en la selección de atributos, mejorando así el rendimiento de los algoritmos utilizados en el área de visión artificial.

1.7. Metodología de la solución

Se realizará una herramienta que permita dado un conjunto de n atributos, obtener el subconjunto de m atributos proporcionado por el usuario, usando el algoritmo de *Branch & Bound* [Narendra, 1977] y una medida de evaluación del subconjunto de atributos que cumpla con la propiedad de monotonía.

Para la realización de esta herramienta se utilizará el diagrama de la Figura 1.1 y los siguientes puntos.

- Recopilar información de los diferentes descriptores usados en las tesis de maestría de Visión Artificial del CENIDET.

- Implementar el algoritmo *B&B* propuesto por [Narendra, 1977].
- Definir una heurística monotónica (o cercana a monotónica) para seleccionar atributos relevantes.
- Implementar el algoritmo *B&B* con la heurística definida para seleccionar el subconjunto óptimo de atributos y obtener el subconjunto de atributos relevantes.

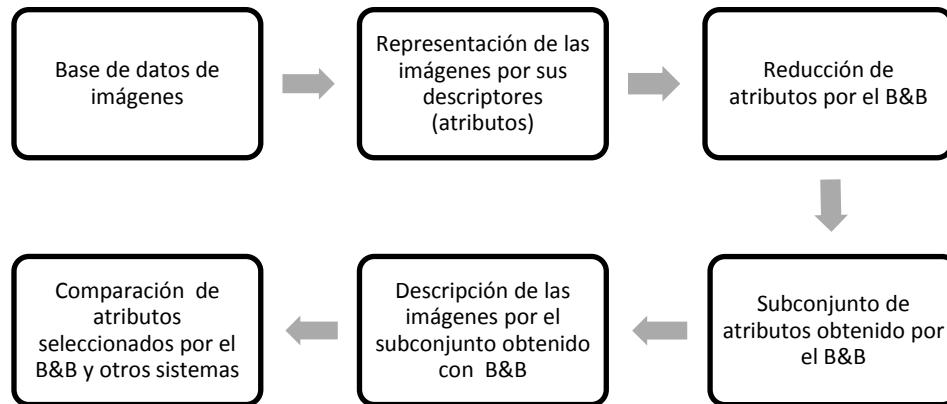


Figura 1.1. Diagrama de la metodología de la solución.

1.8. Organización de la tesis

Este documento de tesis está estructurado de la siguiente manera:

El Capítulo 2 nombrado estado de arte y de la práctica, presenta algunos trabajos y sistemas existentes de área de selección de atributos; el Capítulo 3 presenta el marco teórico para el estudio del tema de selección de atributos en aprendizaje supervisado; en el Capítulo 4 se presenta el análisis, diseño e implementación del algoritmo del *Branch & Bound* así como los criterios de evaluación propuestos; el Capítulo 5 experimentación y resultados engloba una serie de pruebas y sus resultados de diferentes sistemas de selección de atributos y finalmente el Capítulo 6 conclusiones, se muestran las conclusiones de esta investigación.

Al final se presenta las referencias bibliográficas, referencias en línea y los anexos, en el cual se muestran las bases de datos utilizadas, las herramientas de SA y el contenido de CD.

Capítulo 2

Marco Teórico

Se presentan algunos conceptos relacionados con la reducción de dimensionalidad que serán útiles para abordar el tema de selección de atributos para bases de datos de aprendizaje supervisado.

2.1 Reducción de dimensionalidad

El problema de reducción de dimensionalidad ha sido abordado desde diferentes áreas interdisciplinarias como minería de datos, aprendizaje automático, estadística, Bases de datos [Barbara, 1997], e implícitamente en procesamiento de señales [Liu, 2001].

Existen dos formas principales de reducir las columnas en una base de datos dependiendo de sus atributos resultantes, estas son, reducción de atributos por selección de atributos (SA) y reducción de atributos por extracción de instancias (EA); para reducir las filas se usa principalmente el muestreo.

[Somol, 2010] dice que la reducción de la dimensionalidad se refiere a la tarea de encontrar una representación de baja dimensión para datos de alta dimensión. La reducción de la dimensión es un paso importante en diferentes áreas, en aprendizaje automático algunas veces se da el caso que en tareas de clasificación o aproximación de los datos representados por el llamado vector (tupla) de atributos, puede ser más preciso en el espacio (tamaño) reducido que en el espacio (tamaño) original.

[Han, 2006] Las técnicas de reducción de datos se pueden aplicar para obtener una representación reducida del conjunto total de datos, se puede eliminar variables (selección de atributos) y registros (muestreo de instancias) poco relevantes de la base de datos original, pero sin deteriorar la calidad de respuesta. El conjunto de datos reducidos podría ser más eficiente y producir los mismos (o casi los mismos) resultados analíticos.

La reducción de dimensiones puede ser lograda de muchas maneras [Liu, 1998], por selección de atributos, se reduce el número de columnas en una base de datos; discretizando los valores de atributos, se reduce el número de posibles valores de los atributos discretizados y por selección de instancias se reduce el número de renglones en una base de datos, Figura 2.1.

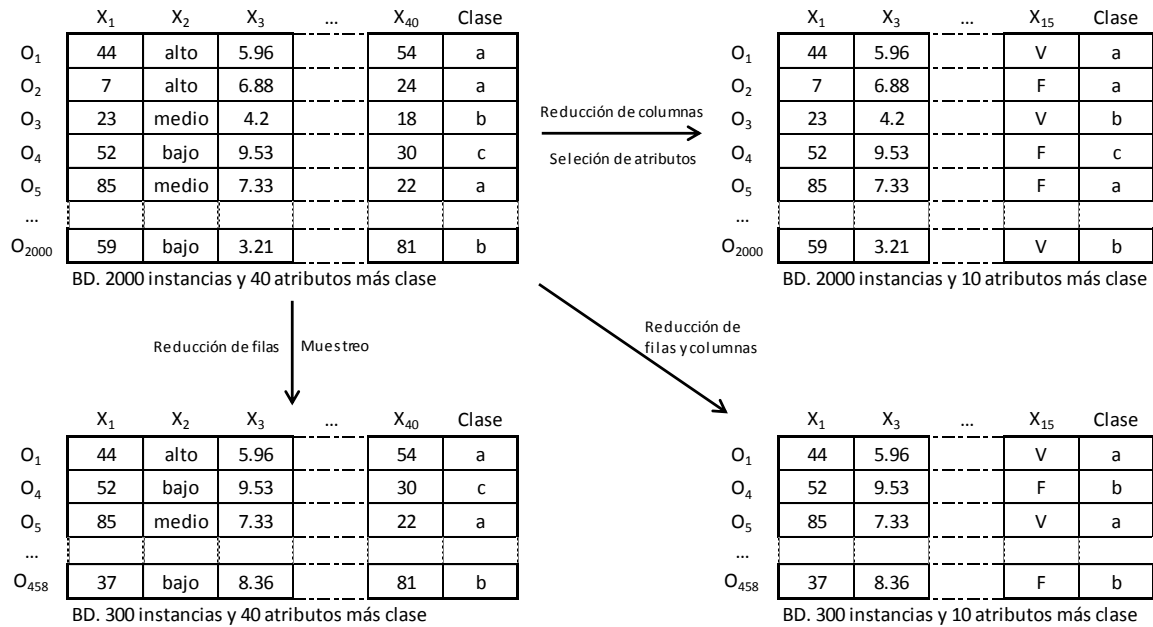


Figura 2.1. Reducción de dimensionalidad de BD [Ruiz, 2006].

Las bases de datos de aprendizaje automático están compuestas por columnas y renglones, a las columnas se les denomina atributos y a los renglones instancias, existen diferentes formas para mencionar el término atributo como son: característica, rasgo, variable, propiedad o factor; y a los renglones aparte de instancia se les suele llamar también: muestra, objeto, patrón, caso, ejemplo. En ocasiones algunos autores hacen ciertas diferencias como en [Guyon, 2003], el término variable se refiere a los datos de entrada originales (*raw variable*), en tanto que atributo (*feature*) se refiere a las entradas construidas a partir de las variables o también en [Ruiz, 1999] se establece una diferencia entre un objeto a clasificar y uno ya clasificado y en este último caso se le llama patrón; en este documento se emplearán los diferentes términos para columnas o renglones sin distinción.

2.2 Conceptos de selección de atributos

Se describen algunos términos que serán empleados en el documento.

2.2.1 Atributo

Un atributo algunas veces llamado característica es la descripción de alguna medida de un objeto. Los atributos tienen un dominio definido por el tipo de atributo, determinando dicho dominio los valores que puede tomar el atributo [Ruiz, 2006], estos pueden ser discretos, continuos o complejos, los atributos continuos pueden tener valores en el dominio de los números reales, los atributos discretos se dividen en ordinales y nominales, si entre los valores se puede establecer una relación de orden se denominan ordinales, los atributo nominales pueden tomar un número finito de valores como por ejemplo el color, los atributos

complejos pueden estar estructurados de la forma $x + iy$ en donde i es la unidad imaginaria $\sqrt{-1}$ y x, y números reales cualesquiera, Figura 2.2 [Liu, 1998]; se denomina vector de atributos $X = \{x_1, x_2, \dots, x_n\}$ al conjunto de valores correspondientes de cada uno de los atributos. En algunos casos se puede tener una descripción de un objeto incompleta es decir no se conoce el valor de cierto atributo por lo que se denomina valor perdido o ausencia de información.

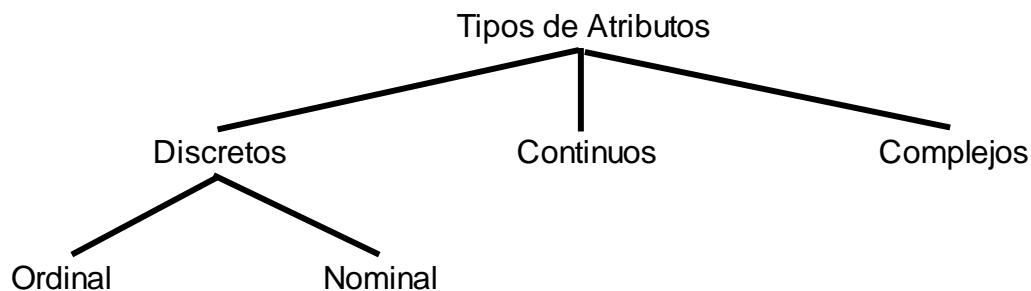


Figura 2.2. Jerarquía de tipos de atributos.

2.2.2 Descriptor

Los descriptores son valores o medidas de las propiedades cuantificables de los objetos [Maravall, 1994]. El problema de la descripción en la Visión Artificial consiste en extraer características de un objeto para reconocerlo. Por lo general, los descriptores deben ser independientes del tamaño, la localización y orientación del objeto y deben contener suficiente información de discriminación para distinguir un objeto de otro [Pajares, 2008].

2.2.3 Instancia

Una muestra, instancia, objeto, ejemplo o caso, es una tupla representada por un conjunto de valores de atributos, en el caso particular de aprendizaje supervisado se dispone de un atributo especial denominado clase que indica la pertenencia a un determinado grupo de casos [Ruiz, 2006].

2.2.4 Conjunto de datos

Un conjunto de datos o base de datos, se caracteriza por el número de objetos que tiene y por el número de atributos y su tipo. En la Figura 2.3 se muestra un ejemplo de una base de datos de aprendizaje supervisado que consta de 14 objetos y 4 atributos más el atributo de clase.

		Atributos				Clase
		x_1	x_2	x_3	x_4	x_c
		Ambiente	Temperatura(°F)	Humedad (%)	¿Ventoso?	Golf
Instancias	O_1	soleado	75	70	Sí	Jugar
	O_2	soleado	80	90	Sí	No jugar
	O_3	soleado	85	85	No	No jugar
	O_4	soleado	72	95	No	No jugar
	O_5	soleado	69	70	No	Jugar
	O_6	nublado	72	90	Sí	Jugar
	O_7	nublado	83	78	no	Jugar
	O_8	nublado	64	65	Sí	Jugar
	O_9	nublado	81	75	No	Jugar
	O_{10}	lluvioso	71	80	Sí	No jugar
	O_{11}	lluvioso	65	70	Sí	No jugar
	O_{12}	lluvioso	75	80	No	Jugar
	O_{13}	lluvioso	68	80	No	Jugar
	O_{14}	lluvioso	70	96	No	Jugar

Figura 2.3. Ejemplo Base de datos de aprendizaje supervisado [Quinlan, 1993].

2.2.5 Selección de atributos

La selección de atributos consiste en pasar de un conjunto de n variables o características a un subconjunto de m características que dé lugar a un rendimiento igual o mejor a uno preestablecido (normalmente considerado como una tasa de aciertos del clasificador) [Maravall, 1994].

La SA, también llamado selección del subconjunto de atributos, selección de variables, selección de características o *feature selection*, tiene como propósito identificar los atributos relevantes de acuerdo a una definición de relevancia, el concepto de relevancia en aprendizaje automático no ha sido rigurosamente establecido, un algoritmo de SA podría ser visto como un enfoque computacional a una definición de relevancia [Molina, 2002].

Sea X el conjunto original de atributos, con cardinalidad $|X| = n$. El problema de selección de atributos continuo se refiere a la asignación de pesos w_i para cada atributo $x_i \in X$ de tal manera que el orden correspondiente a su relevancia se conserva. El problema de selección de atributos binario se refiere a la asignación de pesos binarios, los atributos se usan o no, puede entenderse como una búsqueda en un espacio de soluciones posibles, siendo el número de

subconjuntos potenciales a evaluar 2^n donde n es el número total de atributos [Almuallin, 1991].

En el caso continuo se conservan todos los atributos en el proceso de aprendizaje, por el contrario en el caso binario se está interesado en conservar sólo el subconjunto de atributos relevantes para el proceso de aprendizaje.

Una definición general de SA según [Kudo,1997]:

Sea $J(X')$ un criterio de evaluación para ser optimizado (supóngase a maximizar) definido como $J: X' \subseteq X \rightarrow \mathbb{R}$. En donde J es la medida de evaluación, X' es el subconjunto a evaluar, X el conjunto total de atributos y \mathbb{R} el resultado de aplicar la medida de evaluación.

La SA atributos puede ser vista bajo tres consideraciones:

- Fijando $|X'| = m < n$. Encontrar aquel $X' \subset X$ tal que $J(X')$ sea máximo.
- Fijar un valor J_o , esto es, el mínimo J que se esté dispuesto a tolerar. Encontrar el $X' \subseteq X$ de menor $|X'|$, tal que $J(X') \geq J_o$.
- Encontrar un compromiso entre minimizar $|X'|$ y maximizar $J(X')$ (caso general).

2.2.6 Relevancia de atributos

El propósito de un algoritmo de SA es identificar los atributos relevantes de acuerdo a una definición de relevancia [Molina, 2002]. Los atributos relevantes son asociados con el subconjunto óptimo de atributos [Liu, 1998].

Para [Kohavi, 1995] un **subconjunto óptimo de atributos**, es aquel que dado un inductor, un conjunto de instancias etiquetadas, atributos X_1, X_2, \dots, X_n , es aquel subconjunto de atributos tal que la precisión del clasificador es máxima. Se puede ver que Kohavi, define el subconjunto óptimo de atributos con respecto a un algoritmo particular de inducción, tomando en cuenta sus heurísticas, suposiciones, ventajas y desventajas.

Un subconjunto óptimo de atributos puede no ser único, posiblemente se puede lograr la misma precisión con diferentes subconjuntos como cuando existen atributos perfectamente correlacionados puede ser reemplazado uno por otro o cuando existen diferentes subconjuntos que predicen bien el atributo de clase, si se busca un solo subconjunto óptimo de atributos entra uno en el dilema de que si existieran atributos correlacionados, cual escoger o que grado de relevancia tienen los atributos, es por esto que [John, 1994] y [Kohavi, 1995] introducen dos nuevos tipos de relevancia, que son la relevancia débil y la relevancia fuerte.

Las siguientes definiciones de relevancia son sacadas de [Molina, 2002] que a su vez fueron tomadas de [Kohavi, 1995].

Relevancia respecto a la clase: Un atributo $x_i \in X$ es relevante respecto a una clase c si existen dos ejemplos A, B en el conjunto de instancias E tal que A y B solo difieren en sus valores de x_i y $c(A) \neq c(B)$, este concepto según [Molina, 2002] se queda corto para redundancia.

Relevancia fuerte respecto al conjunto de instancias: Un atributo $x_i \in X$ es fuertemente relevante respecto a un conjunto muestreado de instancias S si existen dos ejemplos $A, B \in S$ tal que A y B sólo difieren en sus valores de x_i y $c(A) \neq c(B)$, es la misma definición respecto a la clase pero ahora $A, B \in S$ y la definición es respecto un conjunto muestreado de instancias.

Relevancia débil respecto al conjunto de instancias: Un atributo $x_i \in X$ es débilmente relevante respecto al conjunto de instancias S si no es fuertemente relevante, existe al menos un subconjunto de atributos $X' \subset X$ ($x_i \in X'$) donde x_i sea fuertemente relevante respecto a S . Un atributo es débilmente relevante cuando al ser removido otro atributo puede ocupar su lugar, cuando el atributo es redundante tiene una relevancia débil. Un atributo es relevante si es fuertemente relevante o débilmente relevante de otra manera es irrelevante.

Relevancia como medida de complejidad: Dado un conjunto de instancias S y una clase c , se define un algoritmo inductor $r(S, c)$ como el menor número de atributos relevantes respecto a la clase c , usando la definición de relevancia respecto a la clase c solo en S , tal que el error en S es el menos posible sobre el conjunto de aprendizaje. Esta relevancia es respecto al inductor, si se retira un atributo, el error aumentará o disminuirá.

Relevancia como utilidad incremental: Dado un conjunto de instancias S , un algoritmo de aprendizaje L y un subconjunto de atributos X' , el atributo x_i es incrementalmente útil para L respecto a X' , si la precisión de la hipótesis que L produce, usando el grupo de atributos $\{x_i\} \cup X'$ es mejor que la precisión alcanzada usando sólo el subconjunto de atributos X' .

Relevancia entrópica: Se denota la entropía de Shannon $H(x)$, $H(x|y)$ la entropía de x generada por el conocimiento de y y la Información Mutua por $I(x; y) = H(x) - H(x|y)$, la relevancia entrópica de x para y es definida como $r(x, y) = \frac{I(x; y)}{H(y)}$.

2.3 Relevancia y optimalidad de atributos

[Liu, 1998] Dice que un atributo relevante, es aquel que al ser removido, la medición de los atributos restantes se deteriorara, sea la medida de precisión, distancia, consistencia, información o dependencia, si el atributo removido no deteriora la medida escogida, entonces el atributo es irrelevante. Esta definición de relevancia no es basada en la suposición que la optimalidad es en términos de precisión, sólo cuando la medida de evaluación es escogida, puede ser

determinada la relevancia del atributo, esto es, la relevancia de un atributo depende de la medida escogida.

Un subconjunto óptimo de atributos no necesariamente tiene que ser único, existe la posibilidad de lograr la misma precisión usando diferentes subconjuntos de atributos como cuando existen atributos perfectamente correlacionados o cuando hay varios subconjuntos de atributos que pueden diferenciar a la clase.

El subconjunto óptimo de atributos es siempre relativo a una medida de evaluación, por lo que al cambiar la medida puede cambiar el subconjunto [Ruiz, 2006]. En el libro del enfoque lógico combinatorio al reconocimiento de patrones [Ruiz, 1999] muestra a través del algoritmo de Testores Típicos como encontrar los subconjuntos óptimos que representan a toda la base de datos de aprendizaje supervisado, estos subconjuntos óptimos, no fueron obtenidos por una medida de evaluación específica o por un algoritmo inductor en particular por lo que aventaja a todos los diferentes enfoques existentes.

2.4 Eficiencia y optimalidad de un algoritmo

La eficiencia y optimalidad, pueden servir para comparar algoritmos entre sí. La eficiencia de un algoritmo suele medirse en función de dos parámetros, espacio y tiempo. El espacio, es decir la memoria que utiliza, y el tiempo, lo que tarda en ejecutarse, Ambos representan los costos que supone encontrar la solución al problema planteado mediante un algoritmo [Guerequeta, 2000]. La optimalidad de un algoritmo de SA es dada por la calidad de la solución [Mejía, 2014].

La eficiencia espacial de un algoritmo indica la cantidad de espacio requerido para ejecutar el algoritmo; es decir, el espacio en memoria que ocupan todas las variables propias del algoritmo. Para calcular la memoria estática de un algoritmo se suma la memoria que ocupan las variables declaradas en el algoritmo. Para el caso de la memoria dinámica, el cálculo no es tan simple ya que, este depende de cada ejecución del algoritmo.

El tiempo de ejecución de un algoritmo tiene 2 formas posibles de medirse:

- Uno que proporciona una medida teórica *a priori*, que consiste en obtener una función que acote (por arriba o por abajo) el tiempo de ejecución del algoritmo para unos valores de entrada dados, ejemplo notación *Big O*.
- Otro que ofrece una medida real *a posteriori*, consistente en medir el tiempo de ejecución del algoritmo para unos valores de entrada dados y en una computadora en concreto.

Ambas medidas son importantes, la primera ofrece estimaciones del comportamiento de los algoritmos de forma independiente de la máquina en donde serán implementados sin necesidad de ejecutarlos, la segunda representa las medidas reales del comportamiento del algoritmo.

Dentro de los aspectos importantes para elegir algún método de selección de atributos está el tiempo de procesado y la calidad de la solución, el tiempo de procesado o la complejidad computacional del algoritmo usado, dice qué tan eficiente es el método y la calidad de la solución da la optimalidad del subconjunto encontrado [Mejía, 2014].

Para evaluar la optimalidad de la solución generalmente se hace de una manera indirecta, usando un clasificador y evaluando la precisión con ese subconjunto de atributos; si se usa la manera directa que es, dada una medida de evaluación y un algoritmo de SA, el algoritmo encuentra el subconjunto óptimo, esta forma pareciera no recomendable ya que cualquier medida buena o mala por definición será óptima, otra forma de medir directamente la optimalidad es mediante bases de datos sintéticas en donde se conoce cuáles son los atributos relevantes, se puede establecer una comparación de optimalidad. Es por esto que revisando la literatura de SA se recomienda usar la Teoría de Testores explicada en capítulos posteriores, la cual no se presta a ambigüedades de relevancia y optimalidad, tomando la optimalidad como los subconjuntos de atributos que representen a toda la base de datos de aprendizaje supervisado sin confundir la clase.

2.5 Medidas de evaluación usadas para encontrar atributos relevantes

[Liu, 1998] Existen varias maneras de evaluar la calidad de un subconjunto de atributos, es claro al observar que la relevancia de un atributo es sólo en función de esta medida y no de las características propias del atributo, otra consideración importante es el hecho de que el rango y escala de una medida de evaluación es inmaterial. Lo que cuenta es que los valores relativos asignados a diferentes subconjuntos reflejan su mayor o menor relevancia de la función objetivo.

El subconjunto óptimo de atributos es elegido de acuerdo a una medida de evaluación, algunas medidas de evaluación según [Liu, 1998], [Molina, 2002], son las siguientes:

- Distancia (Euclidiana, Mahalanobis, Minkowsky..., [Molina, 2002]).
- Información (Entropía de Shannon, de Renyi, Daroczy..., [Ben-Bassat]).
- Dependencia (Correlación, Bhattacharrya , Patrick-Fisher..., [Yu, 2004]).
- Consistencia (Hall, Liu..., [Liu, 1998]).
- Divergencia (Kullback-Liebler, Chernoff, Kolmogorov..., [Molina, 2002]).
- Precisión (Error de generalización).

Información o incertidumbre: Las medidas de información miden la incertidumbre de las fuentes de información, una medida de información J se conoce como la función de incertidumbre relativa a la clase, los valores más grandes para J representan mayores niveles de incertidumbre.

Una medida de incertidumbre muy usada es la Entropía [Shanon, 1948].

$$H(X) = - \sum_i P(x_i) \log_2 P(x_i) \quad (\text{Ec. 1})$$

La entropía de X después de observar los valores de la variable Y se define como:

Entropía condicional [Shanon,1948].

$$H(X|Y) = - \sum_j P(y_j) \sum_i P(x_i|y_j) \log_2 P(x_i|y_j) \quad (\text{Ec. 2})$$

Donde $P(x_i)$ es la probabilidad *a priori* de todos los valores de X , y $P(x_i|y_j)$ es la probabilidad *a posteriori* de X dado el valor de Y .

La ganancia de información mide la reducción de la incertidumbre (entropía) de una variable aleatoria, X , debido al conocimiento del valor de otra variable aleatoria Y .

Ganancia de información (información mutua) [Quinlan, 1993].

$$IG(X|Y) = H(X) - H(X|Y) \quad (\text{Ec. 3})$$

Distancia: Se denominan también medidas de separabilidad, divergencia o de discriminación. Estas medidas estiman la capacidad de un subconjunto de atributos en separar las clases. Suponiendo que instancias de la misma clase forman una región compacta en el espacio, el conjunto de atributos a seleccionar es aquel cuya separación entre estas regiones sea máxima. Utilizando este tipo de medidas se intenta seleccionar aquellos atributos que hacen que los ejemplos de la misma clase estén más juntos y los de diferentes clases más separados. Estas medidas son basadas en la asunción que las instancias de las diferentes clases son distantes en el espacio de instancias. Esto es suficiente para definir una métrica entre clases y usar ésto como medida.

$$D(w_i, w_j) = \frac{1}{N_i N_j} \sum_{K_1}^{N_1} \sum_{k_2=K_1+1}^{N_j} d(x_{(i,k_1)}, x_{(j,k_2)}) \quad (\text{Ec. 4})$$

Siendo $x_{(i,j)}$ la instancia j de la clase w_i y N_i el número de instancias de la clase w_i . Las distancias más usuales d son las euclidianas. Estas mediciones no requieren el modelado de alguna función de densidad, pero su relación con la probabilidad de error puede ser muy pobre.

Dependencia: Este tipo de medidas también se conoce como medidas de asociación o de correlación. Están diseñadas para cuantificar qué tan fuertemente se asocian dos variables o están correlacionadas entre sí de modo que si se conoce el valor de una variable se puede predecir el valor de la otra. En la

evaluación del atributo, en lugar de comprobar cómo un atributo cambia la ganancia de información o una clase de probabilidad, se busca qué tan fuertemente está asociado el atributo con la clase. Las medidas de dependencia están muy relacionadas con las medidas de información y distancia.

Existen en general dos tipos de medidas para la correlación entre dos atributos: lineal y no lineal [Yu, 2004].

De correlación lineal, la medida más conocida es el coeficiente de correlación lineal de Pearson, el coeficiente de correlación r_{ij} entre dos atributos i e j es:

$$r_{ij} = \frac{\sum_p (x_i - \bar{x}_i)(x_j - \bar{x}_j)}{\sqrt{\sum_p (x_i - \bar{x}_i)^2} \sqrt{\sum_p (x_j - \bar{x}_j)^2}} \quad (\text{Ec. 5})$$

Donde x_i y x_j son los valores de atributos i e j respectivamente. Las variables \bar{x}_i y \bar{x}_j representa la media de los valores x_i y x_j , promediado sobre p ejemplos. Si los atributos i y j son completamente correlacionados por ejemplo si existe una dependencia lineal exacta, entonces r_{ij} podría ser 1 o -1. Si i y j son completamente des-correlacionados entonces r_{ij} podría ser 0.

Entre las medidas de correlación no lineal, muchas medidas se basan en el concepto de la entropía de la información, que es una medida de la incertidumbre de una variable aleatoria.

Coeficiente de incertidumbre [Witten, 2011].

$$SU(X, Y) = 2 \frac{IG(X|Y)}{H(X) + H(Y)} \quad (\text{Ec. 6})$$

El coeficiente de incertidumbre (*Symmetric Uncertainty* SU), compensa la ganancia de información de atributos con muchos valores, y restringe su valor a un rango de [0,1]. Un valor de 1 indica que el conocimiento de cualquier atributo predice completamente el valor del otro. Un valor de 0 indica que X y Y son independientes.

Consistencia: Las tres medidas anteriores comparten algo en común, tratan de encontrar los mejores atributos que pueden decir la mejor clase entre todas. Un problema en común es que no pueden desempatar 2 atributos igualmente de buenos, por lo tanto estas medidas no pueden detectar si uno de ellos es redundante. Las medidas de consistencia tratan de encontrar el número mínimo de atributos que separen las clases consistentemente como el conjunto completo de atributos. Una inconsistencia es definida cuando dos instancias tienen el mismo valor en los atributos pero diferente en el atributo de clase. En otras palabras las medidas de consistencia tienen como fin conseguir lo siguiente: $P(C|FullSet) = P(C|SubSet)$. Las reglas de evaluación de atributos derivadas del estado de

medidas de consistencia, deben de seleccionar el subconjunto mínimo de atributos que pueden mantener la consistencia de datos mantenido por el conjunto total de atributos. Usando medidas de inconsistencia, los atributos redundantes e irrelevantes pueden ser removidos.

Precisión: Este tipo de medidas se basa en los clasificadores. Para un clasificador dado, entre todas las posibles combinaciones de subconjuntos de atributos, se escoge el subconjunto que puede dar la mejor precisión predictiva. Obviamente las medidas de precisión se destacan por arriba de las cuatro medidas antes mencionadas. La razón es que, dado que se quiere tener un buen subconjunto de atributos para mejorar la precisión del clasificador, ¿Por qué no se usa la precisión del clasificador como una medida? Con este tipo de medidas vienen algunas consideraciones, una consideración es como estimar verdaderamente la precisión del clasificador y evitar el sobreentrenamiento. Otra consideración es que al clasificador le toma tiempo para aprender los datos del algoritmo de aprendizaje, la primera consideración a menudo reduce la velocidad del clasificador.

Naturalmente, se quieren conocer cómo éstas medidas se complementan unas con otras y qué tan bien o mal les va en la eliminación de atributos redundantes e irrelevantes y como mejorar la precisión del clasificador. Intuitivamente un algoritmo de SA usando la medida de precisión puede producir la mejor precisión para un clasificador específico sin preocuparse si los atributos son consistentes o dependientes. Usando una medida de consistencia, se pueden remover atributos redundantes, esto no puede ser tan claro con una medida de información, distancia o dependencia que pueden o no hacerlo (por ejemplo un empate entre dos atributos podría significar que un atributo es redundante).

Si la tarea de selección de atributos es para clasificación el objetivo de la selección de atributos será maximizar la exactitud de predicción. Es por lo tanto razonable que la precisión del clasificador sea la medida principal de evaluación, generalmente esta medida es la aceptada por los investigadores [Liu, 1998].

2.6 Métodos de selección univariados vs multivariados

Los métodos univariados consideran para su evaluación un atributo a la vez, evalúan cada atributo individualmente, asumen independencia de atributos. El inconveniente que tienen los métodos univariados es que en algunos problemas los atributos tomados independientemente no proveen información de la clase, mientras en unión con otro aporta información suficiente, como en el caso de la or-exclusiva.

Los métodos multivariados hacen uso de subconjuntos o de todos los atributos para su evaluación (predecir su objetivo), focus es uno de los primeros algoritmos multivariados [Almuallin, 1991], Relief también es un método multivariado [Kira, 1992], redes neuronales, máquinas de soporte vectorial, etc.

2.7 Taxonomía de algoritmos de selección de atributos

Existen diferentes formas de agrupar los métodos de selección de atributos, estas son según [Somol, 2010]:

- De acuerdo a la naturaleza de los atributos resultantes.
- Por su propósito (Objetivo)
- Con respecto a la optimalidad.
- Con respecto al criterio de selección.
- Con respecto al conocimiento del problema.

2.7.1 Categorización por la naturaleza de los atributos resultantes

Hay dos distintas maneras de ver la reducción de dimensiones de acuerdo a la naturaleza de los resultados obtenidos:

- Por **Selección de Atributos** (SA) '*Feature selection (FS)*'.
- Por **Extracción de Atributos** (EA) '*Feature Extraction (FE)*'.

El enfoque SA no procura generar nuevos atributos, selecciona los “mejores” del conjunto original de atributos. (Nota: en algunos campos, por ejemplo en análisis de imágenes, el término selección de atributos puede ser interpretado como extracción de atributos, este no es el caso). Dependiendo de la salida del proceso de SA, el resultado puede ser o bien un conjunto de ponderación de pesos, una ponderación (ranking) o un subconjunto de atributos. El enfoque EA define un nuevo espacio de vector de atributos, en el cual cada atributo nuevo es obtenido por combinaciones o transformaciones de los atributos originales. SA permite ahorros en el costo de mediciones ya que algunos de los atributos son descartados y los atributos escogidos conservan su interpretación física original; además, los atributos conservados pueden ser importantes para la comprensión del proceso físico que generó el vector de atributos. Por otro lado, los atributos transformados generados por la extracción de atributos pueden proporcionar una mejor capacidad discriminativa que el mejor subconjunto de atributos dados, pero estos nuevos atributos pueden no tener un significado físico claro.

2.7.2 Categorización de acuerdo a su propósito (objetivo)

- Para la representación óptima de datos.
- Para clasificación.

El primer propósito conserva la estructura topológica de los datos en un espacio de dimensiones bajo tanto como sea posible, el segundo propósito tiene como objetivo mejorar la capacidad de discriminación del subconjunto de atributos. Aunque las mismas herramientas pueden ser utilizadas a menudo para ambos propósitos, es necesario tener precaución. Un ejemplo es PCA, una de las

principales herramientas para representación de datos en un espacio de baja dimensión, que puede descartar fácilmente información importante si se usa para clasificación.

2.7.3 Categorización con respecto a la optimalidad

Los métodos de selección de atributos según su optimalidad pueden dividirse en familias básicas en:

- Métodos óptimos
- Métodos sub-óptimos

Los métodos óptimos incluyen, por ejemplo, métodos de búsqueda exhaustivos que son factibles para problemas de tamaño pequeño y métodos acelerados, la mayoría construidos bajo el principio de *Branch & Bound* [Somol, 2004]. Todos los métodos óptimos se pueden esperar considerablemente lentos para problemas de alta dimensionalidad.

Métodos sub-óptimos Lo principal en la búsqueda del subconjunto de atributos seleccionado es la eficiencia computacional. Estos métodos producen resultados sub-óptimos a costa de ser computacionalmente factibles. Estos incluyen por ejemplo, métodos flotantes [Pudil, 1994], métodos oscilantes [Somol, 2008], aleatorios como las vegas [Liu, 1998], entre otros.

Aunque la búsqueda exhaustiva garantiza la optimalidad de la solución, en muchos problemas reales es computacionalmente prohibitivo. El algoritmo B & B garantiza encontrar el subconjunto óptimo de atributos de tamaño m sin la evaluación explícita de todas las posibles combinaciones de m mediciones. Sin embargo, el algoritmo sólo es aplicable bajo el supuesto de que el criterio de selección de atributos utilizado satisface la condición de monotonía. Esta suposición impide el uso de la tasa de error del clasificador como el criterio de evaluación en los *wrappers*, [Kohavi, 1997]. Este es un inconveniente importante, ya que la tasa de error puede ser considerado superior a otros criterios de evaluación, [Kohavi, 1997], [Liu, 1998]. Por otra parte, todos los algoritmos óptimos son computacionalmente prohibitivos para los problemas de alta dimensionalidad. En la práctica, por lo tanto, se tiene que confiar en los procedimientos computacionalmente factibles que realizan la búsqueda rápidamente, pero puede producir resultados sub-óptimos.

2.7.4 Categorización con respecto al criterio de selección

Basados en la elección del criterio de selección, los métodos de SA en términos generales se dividen en:

- Métodos *filter*
- Métodos *wrapper*
- Métodos *embedded*

- Metodos híbridos

Métodos *filter* o filtro: Se basan en medidas de evaluación de desempeño como distancia, información, dependencia y consistencia calculadas directamente de los datos de entrenamiento, seleccionan el subconjunto de atributos sin involucrar algún algoritmo de aprendizaje.

En la Figura 2.4 se muestra el proceso genérico que se realiza para la SA en un esquema tipo filtro. Básicamente el método filtro es alimentado con el conjunto original de datos y atributos (atributos de entrada) y a su salida se obtiene un conjunto reducido de atributos relevantes (debido al algoritmo de selección del subconjunto de atributos), los cuales serán procesados a continuación por el algoritmo minero o algoritmo de inducción. Generalmente la relevancia de cada atributo se obtiene aplicando una métrica que calcula cuán discriminante es el atributo con respecto a la clase. Los métodos de SA tipo filtro, para aprendizaje supervisado, se sub-dividen en dos tipos: los que evalúan un atributo aislado dada la clase (*ranking*), y los que evalúan subconjuntos de atributos dada la clase.



Figura 2.4. Esquema tipo filtro para selección de atributos [Kohavi, 1997].

Métodos *wrapper*: Requiere un algoritmo de aprendizaje predeterminado y usado para estimar el rendimiento como un criterio de evaluación. Tratan de encontrar los mejores atributos para el algoritmo de aprendizaje con el fin de mejorar el rendimiento. Generalmente los métodos *wrapper* logran un mejor rendimiento que los métodos filtro, pero tienden a ser más costosos computacionalmente que los métodos filtro, además los métodos *wrapper* producen subconjuntos de atributos optimizados sólo para el algoritmo de aprendizaje dado, el mismo subconjunto puede resultar malo en otro contexto.

Siguiendo la Figura 2.5 propuesta por [Kohavi, 1997], la forma tradicional en que opera un *wrapper* inicia con la obtención de un subconjunto de datos de aprendizaje. Se cuenta con una máquina de búsqueda que se encarga de ir generando subconjuntos de atributos $n - m$ (donde n es el total de atributos original, y m es el número de atributos excluidos). El subconjunto de datos alimenta al algoritmo de aprendizaje, el cual realiza la evaluación del subconjunto. Posteriormente se vuelve a ejecutar el algoritmo de aprendizaje, pero ahora para un subconjunto de atributos diferente, por ejemplo $n - m - 1$, o quizás $n - m + 1$,

obteniéndose como resultado otra calificación para este nuevo subconjunto. Este proceso se repite tantas veces como un criterio de paro determine. Al final se elige al subconjunto de atributos que mejor evaluación haya obtenido. Una vez elegido el subconjunto de atributos mejor evaluado, se usa el modelo de clasificación (que se ha aprendido en la etapa anterior) sobre el conjunto de datos de prueba (por ejemplo, el 30% de instancias del conjunto original que no fueron empleadas en la etapa de aprendizaje) para hacer la evaluación final.

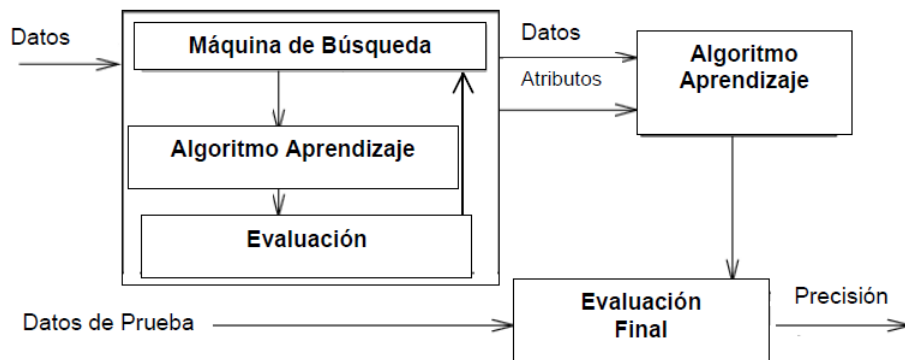


Figura 2.5. Etapas en un *wrapper* para selección de atributos [Kohavi, 1997].

Métodos *embedded*: Integran el proceso de selección de atributos dentro del propio algoritmo de inducción. El modelo y la SA es un proceso de aprendizaje inseparable, puede ser visto como una forma especial de *wrappers*. Los métodos *embedded* ofrecen un rendimiento competitivo a los *wrappers*, permiten el proceso de aprendizaje más rápido, pero producen resultados fuertemente acoplados con un modelo particular.

Metodos híbridos: Combinan las ventajas de más de un enfoque mencionados anteriormente, recientemente se han propuesto algoritmos híbridos para hacer frente a datos de alta dimensionalidad. Estos algoritmos se centran principalmente en la combinación de métodos filtro y *wrappers* para lograr el mejor rendimiento posible con un algoritmo de aprendizaje particular con la complejidad de tiempo comparable a la de los algoritmos tipo filtro.

2.7.5 Caracterización con respecto al conocimiento del problema

Desde otro punto de vista, hay quizás dos clases básicas de situaciones con respecto a un conocimiento *a priori* de las estructuras fundamentales de probabilidad:

- **Algún conocimiento *a priori* está disponible:** Es al menos conocida la función de densidad de probabilidad. En ese caso una de las medidas probabilísticas de distancias Mahalanobis, Bhattacharyya, etc., pueden ser apropiadas como una medida de evaluación.

- **Ningún conocimiento *a priori* está disponible:** No se puede asumir que la función de densidad de probabilidad es unimodal, para estas situaciones los métodos de búsqueda basados en *wrapper* sub-óptimos pueden ser adecuados, siempre que el tamaño de los datos de entrenamiento sea suficiente.

Varios autores comentan la necesidad de desarrollar un buen marco de trabajo (*Framework*) de selección de atributos, ya que los que existentes son ambiguos, existen una gran abundancia de métodos de selección de atributos y ningún consenso ha emergido de todos [Guyon, 2003], [Somol, 2011].

2.8 Selección de atributos visto desde el enfoque lógico combinatorio

La selección de atributos vista desde el enfoque lógico combinatorio presenta cuestiones muy interesantes con conceptos bien definidos que no se prestan a ambigüedades.

En el tema de SA existen términos fundamentales, estos son, SA, relevancia, redundancia y optimalidad, estos términos son contemplados desde el enfoque lógico combinatorio en el libro [Ruiz, 1999] como se describen a continuación.

La selección de atributos visto desde el enfoque lógico combinatorio puede ayudar a resolver 2 problemas, para clasificación y para representación, por lo que al realizar la reducción de atributos (si se pudiera), debe de cumplir con lo siguiente:

Al reducir la dimensión del número de atributos, los atributos escogidos no deben de cambiar la pertenencia a la clase.

La calidad de cierto clasificador no debe de disminuir con el proceso de selección de atributos.

Con SA se solucionan problemas fundamentales como la reducción de atributos en términos de los cuales se describen los objetos y se pueden encontrar los atributos con mayor influencia (relevantes), para esto existen diferentes algoritmos dentro de la Teoría de Testores, en este documento sólo se hará mención del algoritmo *Bottom-Top (BT)* de la Teoría de Testores descrita en [Ruiz, 1999] y [Ochoa, 2006].

Se puede trabajar con cualquier tipo de dato, cada atributo x_i puede tomar el valor admisible dentro del atributo ya sean continuos o discretos, incluso acepta ausencia de información utilizando "*" como símbolo especial de la descripción incompleta del objeto, Tabla 2.1.

Tabla 2.1. Ejemplo atributos continuos y discretos.

Objeto número	Profesión	Edad	Sexo	Altura	CLASE
1	ingeniero	26	M	1.75	buen pagador
2	médico	33	F	1.73	buen pagador
3	abogado	50	M	1.62	buen pagador
4	ingeniero	48	M	1.70	buen pagador
5	abogado	46	F	1.55	buen pagador
6	médico	26	F	1.61	mal pagador
7	médico	30	F	1.63	mal pagador
8	abogado	24	M	1.79	mal pagador
9	ingeniero	26	M	1.64	mal pagador

La **confusión** (error) inducida por un atributo en una tabla de aprendizaje supervisado, es el número de pares de valores iguales que pertenecen a objetos que están en clases distintas.

La confusión inducida por el rasgo profesión es 6, porque hay seis pares de objetos que tienen la misma profesión y pertenecen a clase diferente, estos objetos son, (1,9), (2,6), (2,7), (3,8), (4,9), (5,8); la confusión inducida por {edad, sexo} es 1, porque el par (1,9) es el único par de objetos que tienen los mismos valores de edad y sexo pero tienen clases distintas. En este ejemplo se usó la igualdad como criterio de comparación pero existen diferentes criterios de comparación para cualquier tipo de dato o se puede establecer una comparación establecida de manera *ad hoc*, para más detalles revisar [Ruiz, 1999].

Mientras menor sea la confusión de un conjunto de atributos, mejor sirve para discriminar o diferenciar. Los testores producen confusión cero.

Un testor es un conjunto de rasgos (atributos, columnas) que permite diferenciar entre dos clases. Dado un testor se le podría quitar una columna y ver si sigue siendo testor, hasta llegar a un testor especial llamado testor típico que al suprimirle cualquiera de las columnas confunde la clase de algún objeto.

En este sentido un testor típico es un subconjunto óptimo de atributos no necesariamente único, puede haber varios testores típicos para una base de datos de aprendizaje supervisado.

2.8.1 Calculando todos los testores típicos

Para encontrar todos los testores típicos existen dos grandes estrategias, recorrer el árbol binario que es equivalente a recorrer el conjunto de todos los subconjuntos (algoritmos de escala exterior) o encontrar las condiciones que garanticen que determinadas columnas conforman un testor, en particular un testor típico (algoritmos de escala interior).

El algoritmo *Bottom-Top (BT)* de escala exterior, encuentra todos los testores típicos (subconjuntos óptimos) siguiendo una lista de pasos que se listan a continuación.

- Establecer los criterios de comparación de los atributos.
- Calcular la matriz de diferencias.
- Obtener la matriz básica.
- Aplicar el algoritmo *BT*.

2.8.2 Establecer los criterios de comparación de los atributos

Se establecen las medidas de semejanza para cada columna (atributo) de la matriz de aprendizaje, cuando en la Matriz de Aprendizaje (MA) tiene solo valores nominales o booleanos se puede usar como medida de semejanza la igualdad o dependiendo de problema conversar con el experto o la persona que tomo los datos de los objetos para así establecer los criterios de similaridad de los atributos.

Sea dada una matriz de aprendizaje MA_{nmr} formada por n atributos, m filas correspondientes a las descripciones de los objetos, agrupadas en r clases, $K'_1, \dots, K'_r, r \geq 2$.

Ejemplo Tabla 2.2, sea la Tabla 2.2 una matriz formada por 5 atributos, por la descripción de 5 objetos distribuidos en dos clases, los 3 primeros objetos (sombreados) pertenecen a la primera clase y los restantes a la segunda.

Tabla 2.2. Matriz de aprendizaje.

MA	X ₁	X ₂	X ₃	X ₄	X ₅
O ₁	1	1	0	1	1
O ₂	0	1	0	1	1
O ₃	1	0	1	0	1
O ₄	1	1	1	1	1
O ₅	0	0	0	1	1

2.8.3 Matriz de diferencias MD

Se le llamará matriz de diferencia (de comparación) de MA_{nmr} a la matriz booleana (que tiene sólo ceros y unos) MD_{nm} formada por las filas, como se muestra a continuación

$$S_{ij} = (\alpha_1^{ij}, \dots, \alpha_n^{ij}), i = j \quad i, j = 1, \dots, m \quad (Ec. 7)$$

Donde $\alpha_p^{ij} = C_p(\alpha_p^i, \dots, \alpha_p^j)$, $p = 1, \dots, n$, es el resultado de la comparación entre los valores $\alpha_p^i = X_p(O_i)$ (que denota el valor de la variable X_p en el objeto O_i) y $\alpha_p^j = X_p(O_j)$ (que denota el valor de la variable X_p en el objeto O_j) y donde la cantidad de elementos de MD_{nm} viene dada por la siguiente expresión.

$$m' = \sum_{i=1}^{r-1} \sum_{t=i+1}^r |K'_i| |K'_t| \quad (Ec. 8)$$

Siendo $|K'_i|$ la cantidad de elementos de MA_{nmr} que pertenece a K'_i $i = 1, \dots, r$.

A continuación se muestra la matriz de diferencia Tabla 2.3 obtenida de la matriz de aprendizaje Tabla 2.2.

Tabla 2.3. Matriz de diferencia.

MD	X_1	X_2	X_3	X_4	X_5
S_{14}	0	0	1	0	0
S_{15}	1	1	0	0	0
S_{24}	1	0	1	0	0
S_{25}	0	1	0	0	0
S_{34}	0	1	0	1	0
S_{35}	1	0	1	1	0

2.8.4 Matriz básica MB

Buscar todos los testores típicos en la MD implica ciertas ventajas, la comparación entre dos valores cualesquiera de MA_{nmr} se realizara sólo una vez; la búsqueda se realizará en una matriz booleana.

Sin embargo, aparece una dificultad práctica: la cantidad de filas de la MD tiene un crecimiento de orden cuadrático con respecto a las filas de la matriz MA_{nmr} por lo que la búsqueda en la MD de los testores típicos, aunque es más ventajosa que la búsqueda en MA_{nmr} por las razones antes mencionadas, presenta el inconveniente de que la comparación entre dos vectores es un número grande.

Para eliminar esta dificultad, se introduce un proceso que reduce considerablemente tanto las dimensiones de la MD como su complejidad en cuanto a la cantidad de 1's a considerar.

Sean p y t dos filas de la MD . Se dice que p es subfila de t ssi

$\forall j(a_{pj} = 1 \Rightarrow a_{tj} = 1)$ (en todas las columnas donde p tiene 1, t también)

$\exists k(a_{tk} = 1 \wedge a_{pk} = 0)$ (existe al menos una columna en la que t tiene 1 y p no lo tiene). También se dice que t es superfila de p .

Sea t una fila de la MD . La fila t es básica ssi en la MD no existe fila p alguna que sea subfila de t , como se muestra un ejemplo en la Tabla 2.4 matriz básica obtenida de la Tabla 2.3 matriz de diferencia.

Tabla 2.4. Matriz básica.

MB	X_1	X_2	X_3	X_4	X_5
a_1	0	0	1	0	0
a_2	0	1	0	0	0

2.8.5 Algoritmo BT

El n -uplo α se llamará lista testor ssi el conjunto de columnas $\{X_{j_1}, \dots, X_{j_s}\}$ tales que $\alpha_{j_i} = 1, i = 1, \dots, s$ constituye un testor en la matriz inicial. Si el testor es típico, α se denominará lista testor típico.

PROPOSICIÓN A La lista α no es lista testor cuando y sólo cuando en MB existe al menos una fila $a = (a_1, \dots, a_n)$ tal que se cumple la condición:

$$\forall j \ 1, \dots, n \quad (\alpha_j \wedge a_j) = 0 \quad (\text{Ec. 9})$$

siendo \wedge el operador booleano de conjunción lógica.

PROPOSICIÓN B Sea α una lista testor típico y k el subíndice del último 1 en α entonces los siguientes $2^{n-k} - 1$ n -uplos son lista testores pero no son típicos.

PROPOSICIÓN C Sea $a = (a_1, \dots, a_n)$ una fila de MB y k el subíndice del último 1 en a . Suponga además que $\alpha = (\alpha_1, \dots, \alpha_n)$ no es lista testor y que para α y a se satisface (ec. 9). Sea $\alpha' = (\alpha'_1, \dots, \alpha'_n)$ tal que

$$\alpha_j \begin{cases} \alpha_j & \text{si } j < k \\ 1 & \text{si } j = k \\ 0 & \text{si } j > k \end{cases} \quad (\text{Ec. 10})$$

Entonces ninguna lista comprendida estrictamente entre α y α' es una lista testor.

El algoritmo *BT* se basa en ir generando n-uplos booleanos a partir de $(0, \dots, 0, 1)$, el que como ya se vio anteriormente se corresponde con el conjunto $\{X_n\}$ hasta llegar al $(1, 1, \dots, 1, 1)$, correspondiente al conjunto total de atributos. En cada caso, el algoritmo verifica si el conjunto de columnas que se corresponden con las coordenadas unitarias del n-uplo generado es un testor haciendo uso de la proposición *A*. Para pasar de un n-uplo a otro y con vistas a producir eficientes “saltos” en el orden de generación de los n-uplos α , se utilizan las proposiciones *B* y *C*.

Paso 1.- Se genera la primera lista α no nula de longitud n .

Paso 2.- Se determina (proposición *A*) si la lista generada α es una lista testor en la MB.

Paso 3.- Si es lista testor, se aplica la proposición *B* y si es lista testor típico, se almacena α . Si no es lista testor, se determina la fila a de *MB* que provoca este hecho (de no ser la única se toma la que tenga el último 1 más a la izquierda) y se aplica la proposición *C*.

Paso 4.- Se genera la lista siguiente a las descartadas en virtud del paso 3 y se regresa al paso 2, en caso de que la lista de paso 3 no sea posterior a $(1, 1, \dots, 1, 1)$.

Considerando la matriz básica de la Tabla 2.4.

Tabla 2.5. Matriz básica.

MB	X_1	X_2	X_3	X_4	X_5
a_1	0	0	1	0	0
a_2	0	1	0	0	0

Las listas marcadas con * en la Tabla 2.6 son las únicas listas α que son necesarias examinar para calcular todos los testores típicos de *MA*. Observe que $\alpha = (0, 0, 0, 0, 1)$ implica considerar el conjunto de columnas $\{X_5\}$. Al aplicar la proposición *A* las dos filas de *MB* hacen que $(\alpha_j \wedge a_j) = 0$ para $j = 1, 2, \dots, 5$

Para nuestro propósito se usa la fila a_2 ya que la misma tiene el último 1 más a la izquierda, lo que garantizará un mayor salto, esto es, las siguientes $2^{5-2} - 1$ listas no son testores, en virtud de la proposición *C*.

Análogamente se procede en las restantes listas marcadas con * y como ya se sabe el único testor típico de *MA* se obtiene a partir de la lista $(0, 1, 1, 0, 0)$ esto es el conjunto $\{X_2, X_3\}$.

Los testores que tiene MA están sombreados. Se observa que desde que se comprueba que la lista $(0,1,1,0,0)$ era testor, se podría haber afirmado que las listas $(1,1,1,0,0)$, $(1,1,1,0,1)$, $(1,1,1,1,0)$, $(1,1,1,1,1)$, al corresponderse a superconjuntos de $\{X_2, X_3\}$, eran testores pero no típicos. Sin embargo el algoritmo BT no es capaz de hacerlo. A pesar de ello se puede apreciar la eficiencia de los saltos en la Tabla 2.6.

Tabla 2.6. Ejemplo del algoritmo BT .

00001	*no es testor, siendo a_2 la fila de MB responsable , se aplica la proposición B
00010	
00011	
00100	
00101	
00110	
00111	
01000	*no es testor, siendo a_1 la fila de MB responsable, se aplica la proposición B
01001	
01010	
01011	
01100	*es testor, se almacena y se aplica la proposición A
01101	
01110	
01111	
10000	*no es testor, siendo a_2 la fila responsable, se aplica la proposición B
10001	
10010	
10011	
10100	
10101	

10110	
10111	
11000	*no es testor, siendo a_1 la fila de MB responsable, se aplica la proposición B
11001	
11010	
11011	
11100	*es testor (no típico, ya que se encuentra almacenada la lista 01100 luego no se almacena), se aplica la proposición A y FIN
11101	
11110	
11111	

2.8.6 Medida de la importancia informacional de los atributos (relevancia)

En la Teoría de Testores solo cobra importancia el concepto de relevancia, no existiendo la redundancia, Cuando se tienen varios testores típicos (subconjuntos óptimos de atributos) es común preguntarse: ¿Cuáles son los atributos más relevantes?, como se vio un testor típico es un subconjunto de atributos imprescindible para mantener las diferencias entre clases, es natural suponer que si un rasgo aparece en muchas combinaciones irreducibles, resulta más difícil prescindir del para describir de manera diferenciante las clases, sobre la base de esta idea se formula la definición de peso informacional de un atributo como la frecuencia relativa de aparición en la familia de todos los testores típicos.

Peso informacional frecuentista: Sea τ el número de testores típicos que tiene una cierta matriz $MA_{n,m,r}$ de un problema de clasificación supervisada y sea $\tau(i)$ el número de aquellos testores típicos en los que aparece la columna correspondiente al rasgo X_i , se dice que el peso informacional viene dado por la magnitud:

$$P(X_i) = \frac{\tau(i)}{\tau} \quad (Ec. 11)$$

para $i = 1, \dots, n$ $X_i \in R$.

Sin embargo la expresión anterior propuesta por Zhuravliov, si bien resulta intuitiva, al considerar más importante un atributo mientras más veces aparece en

la familia de todos los testores típicos, no considera las longitudes de los testores que contienen el atributo en cuestión, de modo que por ejemplo, si la familia de testores típicos de un problema fuera

$$\psi^* = \{\{X_1\}, \{X_2, X_5\}, \{X_2, X_3, X_4\}\} \quad (Ec. 12)$$

Se tiene que

$$P(X_1) = P(X_2) = P(X_4) = P(X_5) = 0.333 \text{ y } P(X_3) = 0.666$$

Resulta natural pensar que el rasgo X_1 es más relevante que X_3 , X_4 y X_5 aunque sólo aparece en un testor típico no requiere de ningún otro atributo para lograr diferenciar los objetos que están en clases diferentes, por lo que resulta natural considerar en la medida de que es menor la longitud de los testores típicos en los que aparece. Se asocia a cada rasgo una magnitud, que depende de estas longitudes y que se denota por

$$L(X) = \frac{\sum_{t \in \psi^*(X)} \frac{1}{|t|}}{|\psi^*(X)|} \quad (Ec. 13)$$

Siendo $\psi^*(X)$ la familia de todos los testores típicos que contienen al rasgo X , para el ejemplo anterior se tiene

$$L(X_1) = 1, L(X_2) = 0.418, L(X_3) = L(X_4) = 0.333, L(X_5) = 0.5 \quad (Ec. 14)$$

Estos valores no se corresponden con lo que se señaló como deseado con respecto a la relevancia pues aunque X_1 aparece como máximo valor X_5 tiene asociado un valor mayor que el de X_2 y debería esperarse lo contrario.

$P(X)$ toma en cuenta sólo la frecuencia de aparición del atributo en la familia de testores típicos; $L(X)$ por su parte considera únicamente la longitud de los testores típicos en los que aparece el rasgo, por lo que se puede establecer una medida de relevancia en función de estas dos magnitudes

$$\rho(X) = \theta(P(X), L(X)) \quad (Ec. 15)$$

Siendo θ una función de estas dos magnitudes como por ejemplo

$$\rho(X) = \alpha P(X) + \beta L(X) \text{ con } \alpha, \beta > 0 \text{ y } \alpha + \beta = 1 \quad (Ec. 16)$$

siendo α y β dos parámetros que ponderan la participación o influencia de $P(X)$ y de $L(X)$ respectivamente, si se concede a ambas la misma importancia se puede considerar $\alpha = \beta = 0.5$ entonces se tiene para el ejemplo

$$\rho(X_1) = 0.666, \rho(X_2) = 0.542, \rho(X_3) = \rho(X_4) = 0.333, \rho(X_5) = 0.416 \quad (\text{Ec. 17})$$

que son magnitudes acordes a lo intuitivo.

2.9 Branch & Bound (B&B)

Los algoritmos de ramificación y poda o *Branch & Bound*, se aplican en problemas de optimización. Estos algoritmos realizan una enumeración parcial del espacio de soluciones basándose en la generación de un árbol de expansión.

Se suele interpretar como un árbol de soluciones, donde cada rama lleva a una posible solución posterior a la actual. La característica de esta técnica y a la que debe su nombre es que el algoritmo se encarga de detectar en qué ramificación las soluciones dadas ya no están siendo óptimas, para “podar” esa rama del árbol y no continuar malgastando recursos y procesos en casos que se alejen de la solución óptima [Guerequeta, 2000].

El algoritmo *Branch & Bound (B&B)* de selección de atributos propuesto por [Narendra, 1977] tiene como característica principal evitar duplicidades en la creación de subconjuntos del árbol como se puede ver en la Figura 2.6, para un conjunto total de 5 elementos y la creación de subconjuntos con 2 elementos se tiene en las hojas todas las posibles combinaciones sin repetir ninguna de ellas y en la ramificación no se repite ningún subconjunto.

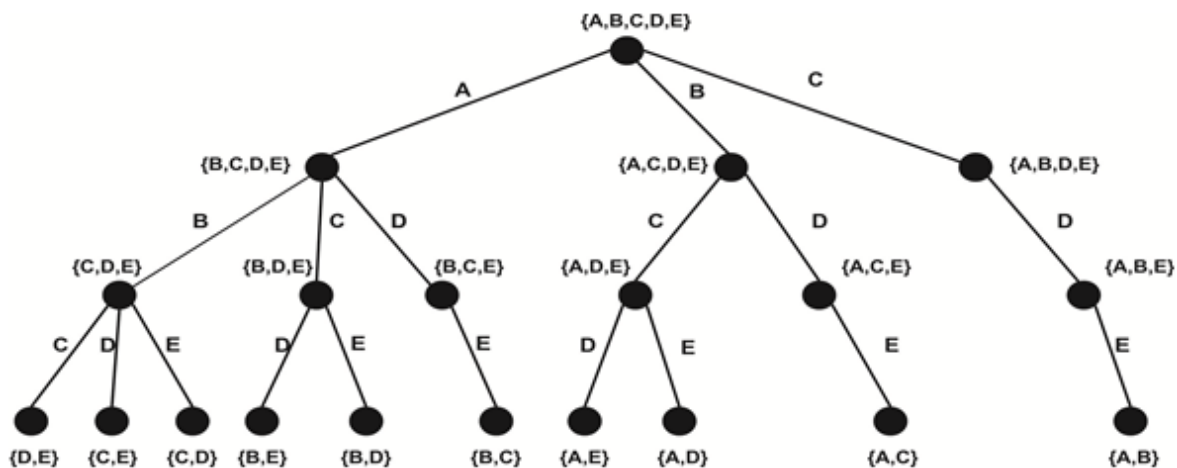


Figura 2.6. Árbol de enumeración [Narendra, 1978].

Pseudocódigo del árbol de enumeración propuesto por [Narendra, 1977]

n : Número de atributos del conjunto original sin incluir el atributo de clase.

m : Número de atributos del subconjunto deseado.

r : Número de atributos disponibles en ese subconjunto (nodo) para eliminar.

k : nivel de profundidad .

1. Asignar valores a n , m , r y k .
2. Asignar elementos al nodo raíz y ponerlo en la lista.
3. Repetir mientras haya elementos en la lista.
 - a. Tomar el último elemento de la lista.
 - b. Si la profundidad del nodo (k) es igual a $n-m$ borrar de la lista
 - c. Si no, generar q número de hijos que están dados por la siguiente fórmula $q=r-(n-m-k-1)$, con sus respectivos datos.
 - d. Agregar los hijos generados al final de la lista.
 - e. Remover el nodo padre de los hijos creados.
4. Fin

2.9.1 Heurística

El término Heurística ha adoptado diversas connotaciones a lo largo de la historia de IA. Actualmente el término heurística se utiliza más bien como un adjetivo que hace referencia a cualquier técnica que pueda mejorar el desempeño del caso promedio en una tarea de resolución de problemas, aunque no necesariamente permita mejorar el desempeño del peor de los casos. Específicamente en el área de los algoritmos de búsqueda, se refiere a una función mediante la cual se obtiene un estimado del costo de una solución [Russel, 1996].

Para n atributos hay 2^n posibles subconjuntos una búsqueda exhaustiva puede ser prohibitiva para encontrar el subconjunto óptimo, especialmente cuando n y el número de clases incrementa, por lo cual se usa comúnmente para la selección de atributos métodos heurísticos que exploran un espacio de búsqueda reducido. Estos métodos son típicamente voraces mientras buscan en su espacio de búsqueda, siguen una búsqueda consistente en elegir la opción óptima en cada paso local con la esperanza de llegar a una solución general óptima; tales métodos son eficaces en la práctica y pueden acercarse a la estimación de una solución óptima [Han, 2006].

2.9.2 Criterio de monotonía

Según [Larsson, 2006] una sucesión $\{a_n\}$ es monótona si sus términos son:

No decrecientes

$$a_1 \leq a_2 \leq a_3 \leq \dots \leq a_n \leq \dots \quad (\text{Ec. 18})$$

O, no crecientes

$$a_1 \geq a_2 \geq a_3 \geq \dots \geq a_n \geq \dots \quad (\text{Ec. 19})$$

2.9.3 Criterio de monotonía para el B&B

Si un subconjunto de atributos S_1 está incluido en un subconjunto S_2 , eso implica que la medida de evaluación del subconjunto S_1 no puede ser mayor que el resultado de evaluar S_2

$$S_1 \subset S_2 \Rightarrow J(S_1) \leq J(S_2) \quad (\text{Ec. 20})$$

La monotonía no es particularmente restrictiva, esto simplemente significa que un conjunto de atributos no debe ser mejor que cualquier otro conjunto mayor que contenga el subconjunto. Existen una gran variedad de criterios de selección de atributos que satisfacen la relación de monotonía, funciones discriminantes, mediciones de distancia, divergencia y dispersión son ejemplos [Fukunaga, 1990].

2.10 Discusión

En este capítulo se describieron algunos conceptos relacionados con la selección de atributos, inicialmente el término de selección de atributos fue riguroso en medida que hacer SA era encontrar el subconjunto con un rendimiento igual o mejor que el conjunto completo de atributos, después con [Kudo, 1997] el término de SA se hizo más flexible pudiéndose hacer selección de atributos respecto a una medida de evaluación, un tamaño establecido del subconjunto de atributos, o un contribución de ambas cosas; otros conceptos importantes que se han tergiversado son los de relevancia y optimalidad, en el subconjunto óptimo de atributos están los atributos relevantes, si hay atributos redundantes entonces existen diferentes subconjuntos de atributos que pueden ser igual de eficientes u óptimos, en algunos enfoques al buscar un solo subconjunto de atributos no contemplan la redundancia como óptima, en el enfoque lógico combinatorio no presentan estas ambigüedades, es por eso que se presentó un poco de cómo hacer la selección de atributos mediante la Teoría de Testores, esto debido a que no se encontraron artículos en donde hayan reportado su utilización que no sea en el CENIDET o su propia literatura, además de ser unos de los enfoques mejor fundamentados

Capítulo 3

Estado del arte y de la práctica

A pesar de que el tema de Selección de Atributos (SA) ha sido ampliamente estudiado, no se ha descubierto un algoritmo eficiente que funcione para las diferentes bases de datos de aprendizaje supervisado, la SA sigue mostrando interés en las diferentes áreas interdisciplinarias como son las dedicadas al aprendizaje automático, minería de datos, estadística o reconocimiento de patrones, por ser la SA parte fundamental de alguna etapa de desarrollo; algunos métodos de SA son adecuados bajo ciertas condiciones, otros son apropiados con otras condiciones de ahí que el tema siga estudiándose con empeño. En esta sección se presenta el estado del arte y de la práctica del tema de selección de atributos.

3.1 Estado del arte

3.1.1 Antecedentes

En esta sección se mencionan algunas Tesis desarrolladas en el CENIDET que tienen relación con el tema de selección de atributos que se está proponiendo.

Reconocimiento de texturas artificiales, aplicación a la inspección visual [Vergara, 2003]. Tiene como objetivo el desarrollo de una herramienta para definir patrones básicos de texturas, así como definir una textura y sus criterios de calidad asociados a esa textura; detectar en una textura arbitraria, textones y/o texturas previamente definidas en una base de conocimientos de referencia, con criterios de calidad asociados.

Para la etapa de extracción de características de texturas utilizó estadísticos de primer orden y de segundo orden, obteniendo un total de 42 variables; a estas variables les aplicó el algoritmo de *Bottom-Top* (BT) de la de Testores, con lo cual redujo las variables a sólo 10.

Desarrollo de una herramienta para el modelado automático de objetos alfaflexibles [Orbe, 2003]. La investigación de este documento es para contribuir con una metodología para modelar en forma automática los objetos alfaflexibles en una secuencia de imágenes, mediante la técnica *snakes* que actúa como una envolvente en el objeto por cada cuadro de video, esta técnica registra los cambios de los atributos del objeto que realizan el movimiento lo cual permite registrar la descripción de los objetos. También presenta la técnica de Teoría de Testores para determinar el orden de la importancia de los atributos.

Técnicas de Selección de Atributos para la Categorización Automática de Escenas Visuales [Ochoa, 2005]. Tiene como objetivo el estudio, selección e implementación de una técnica de reducción del espacio de representación y agrupamiento, para analizar datos multidimensionales obtenidos a partir de imágenes, para que posteriormente estos datos puedan ser utilizados en la clasificación de dichas imágenes.

Implementa un sistema informático, el cual recibe como entrada atributos de diferentes tipos de imágenes e información que permite realizar el agrupamiento y como salida entregará los grupos de imágenes que fueron creados y los criterios utilizados para su clasificación, primero clasifica las imágenes con el algoritmo de agrupamiento “*K-means*” y posteriormente efectúa la reducción de dimensiones a los objetos mediante la Teoría de Testores, utilizando el algoritmo *BT* para encontrar los Testores Típicos [Ruiz, 1999].

Desarrollo de una Metodología para la Selección de Atributos y Generación de Indicadores para la Aplicación de Minería de Datos a una Base de Datos real de Registros de Cáncer en Base Poblacional [Mexicano, 2007]. Este trabajo tiene como objetivo contribuir en la optimización del proceso de minería de datos en una Base de Datos (BD) de cáncer de base poblacional mediante el desarrollo de un método sistematizado para la selección de atributos y generación de indicadores.

En esta tesis se menciona proponer un nuevo enfoque de selección de atributos que consiste en la integración de esquemas de bases de datos para conformar un “metaesquema” al cual se le aplican técnicas de selección de atributos, valida los resultados de los atributos seleccionados utilizando el algoritmo de agrupación “*k-means*” y por cruzamiento con investigaciones clínicas que han reportado en la misma región altas tasas de mortalidad por cáncer gástrico.

Para la selección de atributos utilizó la herramienta de minería de datos Weka [Weka, 2016], probando todos los algoritmos que fueran factibles y eligiendo los atributos que mejor describieran la causa de mortalidad por tumores malignos, en la mayoría de los casos los mismos atributos fueron seleccionados independientemente del algoritmo seleccionado.

Indexado y Recuperación de Imágenes por Contenido [Troncoso, 2007]. Este trabajo tiene como objetivo el estudio de las técnicas de indexado y de recuperación automática de imágenes, seleccionar o diseñar una técnica e implementarla. Esta técnica deberá cumplir con ciertos criterios para la especificación del contenido y podrían ser, por una descripción textual o por una imagen o subimagen de referencia.

Desarrolló un sistema de recuperación de imágenes por contenido, que incluye: caracterización mediante descripciones parciales de los elementos de la imagen y recuperación automática de imágenes por contenido; para la selección de

variables del conjunto de descriptores utilizó la herramienta See5 [Rulequest, 2016], esta herramienta proporciona la selección de variables y la generación de reglas.

Caracterización Automática de Imágenes en un Dominio Transformado [Galeana, 2008]. Tiene por objetivo evaluar experimentalmente la hipótesis de que transformando el dominio de la imagen sería posible obtener una mejor caracterización de su contenido, para aquellos casos en que sus objetos no tienen una forma o color bien definidos. Implementa un conjunto de transformadas en los dominios: *wavelets* (*wavelet Haar*, *Daubechies 4*, *Daubechies 5* y *Daubechies 8*), de color (*RGB normalizado*, *HSI*, *YIQ*, *YQQ*, *YES*, *XYZ*, *CIE Luv*, *CIE Lab*, *CMY* y *YCbCr*) y paramétrico (*Hough* para líneas), con sus respectivos descriptores; 52 en el dominio *wavelet*, 39 en el dominio de color y 4 en el dominio paramétrico. Estos descriptores son estadísticos de primer orden, estadísticos de segundo orden, momentos y orientación de líneas.

Trabaja con un total de 95 descriptores por todos los dominios transformados y compara los resultados con las herramientas See5 [Rulequest, 2016], Weka [Weka, 2016] y RapidMiner [Rapidminer, 2016] en el dominio original de la imagen, obteniendo un buen resultado en el dominio transformado.

Caracterización de Texturas Naturales [Avellaneda, 2009]. Tiene por objetivo Estudiar las técnicas de segmentación de imágenes y las de caracterización de texturas, elegir una de estas técnicas y adaptarla o desarrollar una nueva para implementarla en la segmentación de imágenes con escenas naturales.

Da una metodología para realizar el proceso de segmentación y reconocimiento al mismo tiempo, es decir, segmentar una imagen y reconocer las texturas naturales presentes en ella.

En esta tesis implementan 70 descriptores para la caracterización de texturas y obtienen la selección de variables mediante la herramienta See5 [Rulequest, 2016]. De los 70 descriptores la herramienta See5 seleccionó la cantidad de 50.

COMENTARIOS

En los trabajos de [Avellaneda, 2009], [Troncoso, 2007] y [Galeana, 2008] se utilizó la herramienta See5 que es una técnica de selección de atributos embebida, ya que además de hacer la reducción de variables, proporciona un conjunto de reglas de clasificación, útiles para el proceso de reconocimiento.

En los trabajos [Vergara, 2003], [Orbe, 2003], [Ochoa, 2005], se utiliza como método para la reducción de variables el algoritmo BT de la Teoría de Testores. Gracias a estos trabajos se estudió más a detalle el enfoque lógico combinatorio que es poco mencionado en la literatura de SA y a diferencia de otros artículos los conceptos de SA, relevancia y optimalidad no se prestan a ambigüedades. Es

bueno mencionar que la Teoría de Testores, a pesar de ser una buena técnica en cuanto a la reducción de dimensiones, el algoritmo BT en comparación con otras técnicas para la reducción de dimensiones, es de las más costosas computacionalmente hablando, ya que tiene una complejidad computacional de tipo exponencial. En la Tabla 3.1, se muestra un resumen de los antecedentes del proyecto realizados en el CENIDET.

Tabla 3.1. Resumen de los antecedentes al proyecto de tesis.

Título	Objetivo General	Técnicas empleadas	Comentarios
Reconocimiento de texturas artificiales, aplicación a la inspección visual [Vergara, 2003]	Desarrollar una herramienta para definir patrones básicos de texturas, así como definir una textura y sus criterios de calidad asociados a esa textura; detectar en una textura arbitraria, textones y/o texturas previamente definidas en una base de conocimientos de referencia, con criterios de calidad asociados.	Para la etapa de extracción de características de texturas utilizó estadísticos de primer orden y de segundo orden, obteniendo un total de 42 variables. Para la etapa de reducción de variables aplicó el algoritmo de BT o de Testores Típicos.	A las 42 variables obtenidas en la etapa de extracción les aplicó el algoritmo de BT o de Testores Típicos, con lo cual redujo las variables a sólo 10.
Desarrollo de una herramienta para el modelado automático de objetos alfaflexibles [Orbe, 2003]	Desarrollar una herramienta para generar modelos primarios de los objetos, realizando la localización automática de sus parámetros, las variaciones en cada cuadro en la secuencia de imágenes y analizando la información de cada cuadro y parámetros, mediante Testores Típicos.	El algoritmo de selección de variables utilizado son los Testores Típicos, que permite obtener los rasgos o atributos más importantes del objeto que se está modelando.	Descripción detallada de cómo encontrar los Testores Típicos mediante el algoritmo BT y cómo medir la relevancia informacional de los rasgos en función del concepto de Testor Típico.

Tabla 3.1. Resumen de los antecedentes al proyecto de tesis (continuación).

Título	Objetivo General	Técnicas empleadas	Comentarios
Técnicas de Selección de Atributos para la Categorización Automática de Escenas Visuales [Ochoa, 2005]	Estudiar, seleccionar e implementar una técnica de reducción del espacio de representación y agrupamiento, para analizar datos multidimensionales obtenidos a partir de imágenes.	Para la etapa de clasificación utilizó el algoritmo <i>k-means</i> . Para la reducción de dimensiones utilizó los Testores Típicos.	Esta tesis es una herramienta computacional para la reducción de dimensiones con resultados óptimos pero computacionalmente es muy costosa.
Desarrollo de una Metodología para la Selección de Atributos y Generación de Indicadores para la Aplicación de Minería de Datos a una Base de Datos real de Registros de Cáncer en Base Poblacional. [Mexicano, 2007]	Contribuir en la optimización del proceso de minería de datos en una base de datos de cáncer de base poblacional mediante el desarrollo de un método sistematizado para la selección de atributos y generación de indicadores	Se utilizaron algoritmos de selección de atributos ya implementados en la herramienta libre Weka.	Menciona que en la mayoría de los casos los mismos atributos fueron seleccionados independientemente de los algoritmos seleccionados de la herramienta de minería Weka.
Indexado y Recuperación de Imágenes por Contenido [Troncoso, 2007]	Estudiar las técnicas de indexado y de recuperación automática de imágenes, seleccionar o diseñar una técnica e implementarla.	Los descriptores utilizados fueron reducidos por la herramienta de selección de atributos See5, así como las reglas de clasificación que fueron obtenidas por la misma herramienta.	Implementó el algoritmo ALVOT (Algoritmo de votación) para la clasificación de la escena presente en una imagen, a partir de las reglas dadas por la herramienta See5.
Caracterización Automática de Imágenes en un Dominio Transformado [Galeana, 2008]	Evaluar experimentalmente la hipótesis de que transformando el dominio de la imagen sería posible obtener una mejor caracterización de su contenido. Estudiar, seleccionar e implementar técnicas de transformación de dominio y funciones de caracterización.	Para la descripción en los dominios utilizó descriptores estadísticos de primer orden, descriptores estadísticos de segundo orden, momentos y orientación de líneas. Utiliza las herramientas de Weka, RapidMiner y See5 para analizar los descriptores en cada dominio.	Como trabajo futuro menciona como un punto de interés, desarrollar o implementar una herramienta de minería de datos, un algoritmo de clasificación o de selección de variables que sea invariante a la calidad de entrada de los datos.

Tabla 3.1. Resumen de los antecedentes al proyecto de tesis (continuación).

Título	Objetivo General	Técnicas empleadas	Comentarios
Caracterización de texturas naturales [Avellaneda, 2009]	Estudiar las técnicas de segmentación de imágenes y las de caracterización de texturas, elegir una de estas técnicas y adaptarla o desarrollar una nueva para implementarla en la segmentación de imágenes con escenas naturales.	Utiliza 70 descriptores para la caracterización de texturas, y hace la selección de variables a través de la herramienta See5.	Dentro de sus objetivos alcanzados menciona haber realizado la selección de variables a pesar de no estar contemplada en sus alcances, pero realizarla por ser una actividad importante.

3.1.2 Trabajos relacionados

Los siguientes trabajos muestran sólo un poco de la investigación que se ha hecho en el tema de SA ya que es un tema ampliamente estudiado y es difícil poner toda la investigación que se ha hecho al respecto.

A Branch and Bound Algorithm for Feature Subset Selection [Narendra, 1977]. En este artículo se presenta por primera vez el algoritmo *Branch & Bound* de SA, en el cual se afirma ser más eficiente y seleccionar el mejor subconjunto de atributos sin realizar una búsqueda exhaustiva, con las restricciones de que la medida usada para evaluar el subconjunto tiene que ser monótona y además para el funcionamiento del algoritmo se necesita conocer antes, el tamaño del subconjunto de atributos.

Wrapper for performance enhancement and oblivious decision graph. [Kohavi, 1995]. En esta tesis doctoral se investigan tres problemas básicos en aprendizaje automático que son la estimación de la precisión mediante *cross validation* y *.632 bootstrap*, además de la selección del subconjunto de atributos y sus parámetros de ajuste utilizando el enfoque *wrapper*. Presenta nuevas definiciones de relevancia para el conjunto óptimo de atributos.

Enfoque Lógico Combinatorio al Reconocimiento de Patrones [Ruiz, 1999]. Su objetivo central es brindar a los lectores de manera clara y directa la teoría y herramientas para resolver problemas de clasificación de variables y clasificación supervisada. Es un libro que está dedicado al enfoque lógico combinatorio dentro del reconocimiento de patrones, abarca los temas de selección de variables y clasificación supervisada, en el capítulo de selección de variables incluye la Teoría de Testores como una solución; trae una explicación detallada del algoritmo BT para el cálculo de todos los Testores Típicos como un método al problema de selección de variables.

Feature Selection Algorithms: A Survey and Experimental Evaluation [Molina, 2002]. Hace un estudio detallado de los algoritmos de selección de atributos, los caracteriza de acuerdo a su organización de búsqueda, medida de evaluación y generación de sucesores y ubica al *B&B* en organización de búsqueda como de tipo exponencial, su generación de sucesores hacia atrás “*backward*” y como medida de evaluación de tipo monotónica, hace énfasis en que es un algoritmo de búsqueda de selección de atributos óptimo con la desventaja de que el tamaño del subconjunto de búsqueda tiene que ser proporcionado por el usuario previamente.

Excluye de la evaluación experimental a los algoritmos *B&B*, *FOCUS*, *ABB* y *LVW* por el gran consumo de recursos computacionales, según estos autores.

An Introduction to Variable and Feature Selection [Guyon, 2003]. Este artículo menciona la importancia de la selección de atributos, como está aumentando la información de las bases de datos y van a seguir aumentando, las dificultades que se presentan como son la captura, almacenado, búsqueda, compartición, análisis y visualización, es necesaria la selección de atributos para la reducción de datos, las ventajas y desventajas de los diferentes tipos de técnicas que existen para SA y como poder escoger el algoritmo adecuado de acuerdo a las necesidades.

Fast Branch & Bound Algorithms for Optimal Feature Selection [Somol, 2004]. Propone una solución al problema del subconjunto óptimo de selección de atributos, mediante una mejora al algoritmo del *B&B* presentado por primera vez en 1977 por [Narendra, 1977]; como se sabe, la complejidad computacional del *B&B* es exponencial y gran parte de tiempo de cómputo se lleva en evaluar el criterio de selección, es por esto que propone un mecanismo para predecir el valor del subconjunto de atributos y así no evaluar todos los nodos que se van creando en la construcción del árbol de búsqueda.

A Review of Suboptimal Branch and Bound Algorithms [Nakariyakul, 2009]. El artículo trata acerca del algoritmo *B&B* que si bien es conocido por su eficiencia computacional cuando la dimensionalidad del espacio de características original es grande, el tiempo de ejecución requerido por el algoritmo *B&B* se vuelve excesiva, el tiempo de búsqueda se puede reducir en gran medida mediante el empleo de la mirada-hacia adelante “*look-ahead*” que es una buena solución, propuesta para acelerar la búsqueda en este artículo, este trabajo hace un estudio comparativo de cuatro variantes del *B&B* las cuales son: *BB Basic (Basic Branch & Bound)*, *BBO (Ordered B&B)*, *FBB (Fast Branch & Bound)*, y *ABB (Automatic Branch & Bound)*.

COMENTARIOS

Se ha publicado bastante acerca de selección de atributos que es complicado poner toda la investigación que existe, [Kohavi, 1995] presenta un marco de trabajo de selección de atributos mediante el enfoque *wrapper*, este enfoque utiliza un algoritmo inductor para evaluar la relevancia del subconjunto de atributos, la

mayoría de literatura de SA ha seguido esta marco de trabajo, existe también dentro del enfoque lógico combinatorio [Ruiz, 1999] la Teoría de Testores que encuentra los subconjunto óptimos de una base de datos de aprendizaje supervisado sin necesidad de algoritmo inductor.

A continuación en la Tabla 3.2 se muestra un resumen de los trabajos relacionados.

Tabla 3.2. Resumen de trabajos relacionados.

Nombre	Objetivo	Técnicas empleadas	Pruebas y resultados	Observaciones
<i>A Branch and Bound Algorithm for Feature Subset Selection</i> [Narendra, 1977]	Presentar el algoritmo que es muy eficiente y evita la búsqueda exhaustiva, garantizando la optimalidad.	Algoritmo <i>B&B</i> de SA con alguna medida de evaluación del subconjunto de atributos monótona.	Probado sobre una BD multiespectral de 24 atributos, 423 instancias y 2 clases, para encontrar el mejor subconjunto de 12 atributos utilizando como medida de evaluación, una función discriminante	El algoritmo <i>B&B</i> garantiza encontrar el subconjunto óptimo determinado por el usuario siempre y cuando la medida de evaluación del subconjunto de atributos sea monótona.
<i>Wrapper for performance enhancement and oblivious decision graph.</i> [Kohavi, 1995]	Investigar 3 problemas básicos en aprendizaje automático que son, estimación de la precisión, selección de atributos y su ajuste de parámetros.	Utilizó para encontrar el subconjunto de atributos el método <i>wrapper</i> .	Diferentes bases de datos reales y sintéticas con diferentes algoritmos inductores como el Naive Bayes, C4.5 e ID3	Propone nueva definiciones de relevancia y su relación con el subconjunto óptimo de atributos.
Enfoque Lógico Combinatorio al Reconocimiento de Patrones [Ruiz, 1999]	Brindar a los lectores de manera clara y directa la teoría y herramientas para resolver problemas de clasificación de variables y clasificación supervisada.	La Teoría de Testores como una solución al problema de reducción de variables, mediante el uso del algoritmo BT para el cálculo de los Testores Típicos.	Ejemplo del cálculo de los Testores Típicos mediante el algoritmo BT.	Trae una explicación detallada del algoritmo BT para el cálculo de todos los Testores Típicos como un método al problema de selección de variables.

Tabla 3.2. Resumen de trabajos relacionados (continuación).

Nombre	Objetivo	Técnicas Empleadas	Pruebas y Resultados	Observaciones
<i>Feature Selection Algorithms: A Survey and Experimental Evaluation.</i> [Molina, 2002]	Mostrar las características de diferentes algoritmos de selección de atributos, para saber decidir adecuadamente el algoritmo de selección de atributos dependiendo del contexto.	Se consideraron los aspectos de relevancia, irrelevancia, redundancia, así como el tamaño de muestra para establecer una calificación.	Se experimentó con 10 algoritmos y se muestran las gráficas de desempeño. Los resultados finales de los experimentos pueden ser vistos como un conocimiento útil para decidir cuál algoritmo utilizar en ciertas situaciones.	Según los autores de este artículo el B&B tiene un consumo elevado de recursos computacionales por lo que no fue evaluado, como se sabe si se escoge la heurística adecuada se puede obtener el subconjunto de atributos óptimo.
<i>An Introduction to Variable and Feature Selection.</i> [Guyon, 2003]	Dar a conocer la importancia y como mejorar la selección de atributos, así como los beneficios obtenidos de aplicar una selección de atributos	Describe las técnicas <i>filter</i> , <i>wrapper</i> y <i>embedded</i> y muestra sus ventajas y desventajas.	Presenta una serie de ejemplos y muestra la utilidad y limitaciones de técnicas ranking, señala que la dependencia de atributos no puede ser ignorada.	Resume los pasos que pueden ser tomados para enfrentar un problema de selección de atributos.
<i>Fast Branch and Bound Algorithms for optimal Feature Selection</i> [Somol, 2004]	Presentar un algoritmo para la selección de atributos más rápido que el B&B básico.	Se probaron los algoritmos B&B y FastB&B en datos de mamografías de 30 dimensiones.	El <i>Fast B&B</i> mostró eficiencia de 1.5 hasta 10 veces más rápido que el algoritmo B&B, aunque en casos extremos resulto peor que exhaustivamente.	Es recomendable usar el <i>Fast B&B</i> con criterios computacionalmente costosos, ya que reduce significativamente el costo computacional.
<i>A review of Suboptimal Branch & Bound Algorithms</i> [Nakariyakul, 2009]	Dar a conocer una solución "look-ahead" para acelerar la búsqueda del B&B	Se implementó esa heurística con el B&B, <i>ordered B&B</i> , <i>Fast B&B</i> y el <i>automatic B&B</i> .	La optimalidad con la estrategia "look-ahead" no está garantizada, se lograron soluciones optimas con los diferentes B&B, con la ventaja de reducir el costo computacional.	Este autor no recomienda el uso del B&B en BD's de dimensiones grandes, y para problema de seleccionar atributos en un rango mayor de 50 sugiere utilizar la estrategia "look-ahead".

3.1.3 Estado de la práctica

En esta sección se describen algunos de los sistemas de selección de atributos que se pueden encontrar, existe una amplia gama de herramientas de minería de datos o de reconocimiento de patrones (libres o privados) que incluyen métodos de selección de atributos, se tomarán los más utilizados o más ampliamente difundidos, aunque debido a que existe una gran cantidad de sistemas de SA la lista está incompleta.

PRTools es una *toolbox*¹ de Matlab para reconocimiento de patrones, el equipo de desarrollo de PRTools comenzó en 1993 en el grupo de reconocimiento de patrones (en 2004 renombrado *Quantitative Imaging*) de la *Delft University of Technology* ubicada en Países Bajos. Ofrece más de 300 rutinas de Matlab para la construcción de sistemas de reconocimiento de patrones. Dentro de las rutinas que ofrece están las de pre-procesamiento de imágenes, Matlab por sí mismo ofrece rutinas de pre-procesamiento de imágenes, PRTools contiene rutinas de propósito general que se pueden aplicar a un conjunto de archivos ubicados en una dirección de disco de manera automática, dentro de los algoritmos de SA que vienen implementados están de tipo de búsqueda hacia adelante, hacia atrás, flotante (adelante y atrás) y trae además el *Branch & Bound* el cual utiliza como criterio de evaluación la distancia euclidiana y Mahalanobis.

mRMR (*minimum Redundancy Maximum Relevance Feature Selection*) es una herramienta de selección de atributos de tipo *filter* desarrollada por el grupo de trabajo de Hanchuan Peng, el objetivo principal es seleccionar un subconjunto de atributos que mejor caracterice a la clase, utilizando para esto la información mutua en un esquema que minimice la redundancia entre atributos y maximice la relevancia con la clase. Esta herramienta se puede usar en tres formas, preparar los datos y subir el archivo al sitio web <http://research.janelia.org/peng/proj/mRMR> o descargar las versiones de Matlab o C/C++ y correrlos en la computadora deseada.

MLC++ es una librería de clases C++ para aprendizaje de máquina supervisado, MLC++ (hasta la versión 1.3.x) fue desarrollado en la Universidad de Stanford por Ron Kohavi, George John, Richard Long, David Manley y Karl Pfleger, era de dominio público; esa versión todavía se distribuye como tal por SGI Silicon Graphics International Corp. La versión más reciente SGI MLC++ (V2.0 y superior) incluye mejoras en MLC++, estas mejoras son de dominio de investigación y están disponibles tanto en código fuente y objeto a través de este sitio web <http://www.sgi.com/tech/mlc/index.html>, MLC++ proporciona algoritmos generales de aprendizaje automático que pueden ser utilizados por usuarios finales, analistas, profesionales e investigadores. El objetivo principal es proporcionar a los usuarios una amplia variedad de herramientas que pueden ayudar a extraer datos, acelerar el desarrollo de nuevos algoritmos de minería, aumentar la fiabilidad del

¹ Toolbox conjunto de instrucciones especializadas en un dominio en particular.

software, proporcionar herramientas de comparación y mostrar la información visual.

TOOLDIAG Es una colección de métodos estadísticos, para reconocimientos de patrones implementados en el lenguaje de programación C desarrollado por Thomas W. Rauber. El campo principal de aplicación es la clasificación, sólo acepta bases de datos con atributos continuos y no debe de haber valores faltantes, los atributos nominales no están permitidos, la interfaz de usuario es en línea de comandos, dentro de los métodos de selección de atributos se encuentran selección secuencial hacia adelante, selección secuencial hacia atrás, selección secuencial flotante hacia adelante, selección secuencial flotante hacia atrás, B&B y búsqueda exhaustiva, que utilizan como criterio de evaluación del subconjunto de atributos las medidas Minkoski, Euclídea, Chebichev, Chernoff, Bhattacharyya, Matusita, Divergencia, Mahalanobis, Patrick-Fisher entre otras, se encuentra disponible en la página <https://sites.google.com/site/tooldiag/>.

Orange Es una herramienta de código abierto para análisis y visualización de datos para usuarios principiantes y expertos, el software es mantenido y desarrollado por el laboratorio y la facultad de computación y ciencias de la información de la universidad de Ljubljana en Eslovenia, está desarrollado en Python y su uso es a través de una interfaz visual o scripts de Python, cuenta con librerías para aprendizaje de máquina, minería de textos y bioinformática, el método de selección de atributos que trae implementado es el Relief.

KEEL (*Knowledge Extraction based on Evolutionary Learning*) Esta herramienta permite a los usuarios analizar el comportamiento de algoritmos de aprendizaje evolutivo en diferentes tipos de problemas como: preprocesamiento de datos, regresión, clasificación, aprendizaje supervisado, aprendizaje no supervisado; entre otros.

Rapidminer Es una herramienta multiplataforma desarrollada en Java de código abierto, para análisis y minería de datos. La versión inicial fue desarrollada por el departamento de inteligencia artificial de la Universidad de Dortmund en 2001 por Ralf Klinkenberg, Ingo Mierswa, y Simon Fischer bajo el nombre de YALE (*Yet Another Learning Environment*). Cuenta con cientos de librerías para cargar datos, transformación de datos, modelado de datos, y visualización de datos de diferentes fuentes como Excel, Access, Oracle, IBM DB2, Microsoft SQL, Netezza, Teradata, MySQL, Postgres, SPSS y muchas más, se pueden integrar fácilmente los propios algoritmos o con Weka y R, gracias a su API de extensión. Cuenta con algoritmos de selección de atributos de tipo *wrapper* con búsqueda del subconjunto de atributos hacia adelante, hacia atrás, fuerza bruta y por algoritmos genéticos.

Elefant (*Efficient Learning, Large-scale Inference, and Optimisation Toolkit*) Es una biblioteca de código abierto para el aprendizaje automático licenciado bajo la *Mozilla Public License* (MPL). Elefant es desarrollado por el grupo de Análisis de

Datos Automatizado (ADA) del centro de Investigación Nacional de Tecnología de Información y Comunicación de Australia (NICTA). El sistema elefant ofrece las siguientes características, posee una interfaz gráfica de usuario para crear prototipos de aprendizaje automático, están algoritmos de aprendizaje automático que utilizan el poder de la tecnología multinúcleo y multihilo, tutoriales y documentación detallada de los algoritmos implementados. Dentro de los algoritmos de selección de atributos implementados en el sistema está el de la búsqueda del subconjunto de atributos usando el criterio independiente Hilbert-Smith con eliminación hacia atrás, disponible en <http://elefant.developer.nicta.com.au/Getting>.

Scikit-feature En la página web de la Universidad del Estado de Arizona [ASU, 2016] se puede encontrar un repositorio de una serie de algoritmos de selección de atributos los cuales están implementados en Matlab y en Python, contiene alrededor de unos 40 algoritmos de selección de atributos bajo licencia GPL.

R es un lenguaje de programación para análisis estadístico y gráfico de datos. Es un proyecto de software libre, como R forma parte de un proyecto colaborativo y abierto sus usuarios pueden publicar paquetes que extienden su configuración básica; el paquete relacionado con la selección de atributos se llama Fselector el cual contiene algoritmos de SA tipo filtro como cfs, chi-squared, ganancia de información, incertidumbre simétrica, correlación lineal, oneR, relief, consistencia entre otros; algoritmos de SA tipo *wrapper* con búsqueda de subconjunto de atributos primero el mejor, hacia atrás, hacia adelante, ascenso de colina, también incluye algoritmos que se basan en el peso de los atributos o en simples fórmulas.

Elvira El sistema o entorno Elvira (Elvira Consortium, 2002) es una plataforma para la construcción y uso de modelos gráficos probabilísticos, que fue construida con el apoyo del Ministerio de Ciencia y Tecnología, bajo dos proyectos de investigación: Elvira (TIC97-1135-C04, 1997-2000) y Elvira II (TIC2001-2973-C05, 2001-2004). El sistema Elvira incorpora métodos filter tipo ranking, en base a diferentes medidas extraídas del artículo [Ben-Bassat, 1982], estas medidas son las siguientes [Armañanzas,2004], Información mutua, Distancia Euclidiana, Distancia Matusita, Kullback-Leibler 1, Kullback-Leibler 2, Entropía de Shannon, Distancia Bhattacharyya.

Reduclas El sistema Reduclas es desarrollado en el CENIDET como parte de la tesis “Técnicas de selección de atributos para la categorización automática de escenas visuales” [Ochoa, 2005], este sistema utiliza la técnica de Teoría de Testores para la reducción de dimensiones, determina los Testores Típicos que son el conjunto mínimo de variables que describen de manera idéntica al total de objetos, tal como si se estuvieran utilizando todas las variables.

See5 El sistema See5 (Windows XP / Vista / 7 / 8) y su contraparte C5.0 (Unix) es una herramienta sofisticada de minería de datos para descubrir patrones, estos patrones pueden utilizarse para predecir los resultados en situaciones futuras y

ayudar en la toma de decisiones. Para maximizar la interpretabilidad los clasificadores See5 / C5.0 se expresan como árboles de decisión o conjuntos de reglas si-entonces, estas formas son generalmente fáciles de entender.

Weka El sistema Weka (*Waikato Environment for Knowledge Analysis*) entorno para análisis del conocimiento de la Universidad de Waikato, es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java, contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades.

A continuación en la Tabla 3.3 se muestra un resumen del estado de la práctica.

Tabla 3.3. Resumen del estado de la práctica.

Nombre	Centro de desarrollo	Algoritmos de SA	Tipo de licencia	Observaciones
PRTools	Delft University of Technology, Países Bajos.	búsqueda hacia adelante, hacia atrás, flotante (adelante y atrás) <i>Branch & Bound</i>	Para instituciones académicas o investigación sin ánimo de lucro, licencia pública. Para incluirla en uso comercial, se necesita pedir una licencia de distribución comercial.	Se encuentra desarrollado en Matlab por lo que necesita de este software para utilizar sus rutinas.
mRMR	Grupo de trabajo de Hanchuan Peng, Estados Unidos de América.	Método tipo <i>filter</i> que utiliza información mutua como medida de evaluación.	Para instituciones académicas o investigación sin ánimo de lucro, permiso para usar copiar y modificar. Para incluirla en uso comercial, se necesita pedir una licencia de distribución comercial.	Este software se encuentra disponible en versión para C/C++ y Matlab.
MLC++	Universidad de Stanford. Estados Unidos de América	Algoritmos de tipo <i>wrapper</i> , con técnicas de evaluación como validación cruzada y muestreo.	Hasta antes de la versión 2.0, licencia pública, a partir de la versión 2.0 solo se concede licencia para fines de investigación.	Son un conjunto de librerías de clases hechas en C++.

Tabla 3.3. Resumen del estado de la práctica (continuación).

Nombre	Centro de desarrollo	Algoritmos de SA	Tipo de licencia	Observaciones
TOOLDIAG	Thomas W. Rauber. Brazil	selección secuencial hacia adelante, secuencial hacia atrás, secuencial flotante hacia adelante y hacia atrás, <i>B&B</i> y búsqueda exhaustiva	Licencia pública.	Está desarrollado en C, no permite valores nominales ni perdidos (<i>missing</i>),
Orange	Universidad de Ljubljana Eslovenia	Metodos de SA tipo <i>wrapper</i> y ReliefF.	Licencia pública	Desarrollado en Phyton y cuenta con una interfaz gráfica para programación visual.
KEEL	Ministerio de educación y ciencia, España TIC2002-04036-C05, TIN2005-08386-C05 y TIN2008-06681-C06	Metodos de SA tipo <i>filter</i> , <i>wrapper</i> , heurísticos e híbridos.	Licencia pública general 3.	Herramienta de software hecha en Java.
Rapidminer antes (YALE)	Inicialmente desarrollado en Alemania por la Universidad de Dortmund, actualmente por la empresa Rapidminer	Métodos de SA tipo <i>wrapper</i> con búsqueda del subconjunto de atributos hacia adelante, hacia atrás, fuerza bruta y por algoritmos genéticos.	Licencia pública para uso académico o sin ánimo de lucro. Licencia comercial para otro uso.	desarrollado en Java, permite programación visual, se puede integrar con algoritmos de weka o R.
Elefant	Centro de investigación NICTA en Australia	Método <i>wrapper</i> con eliminación hacia atrás y medida de evaluación Hilbert-Smith.	Licencia pública Mozilla.	Desarrollado en Phyton, cuenta con interfaz gráfica para diseño de prototipos.
Scikit-feature	Universidad de Estado de Arizona en Estados Unidos de América	Métodos tipo <i>wrapper</i> , <i>filter</i> , métodos híbridos.	Licencia pública y para uso comercial licencia BSD.	Inicialmente desarrollado en Matlab, actualmente en Phyton.

Tabla 3.3. Resumen del estado de la práctica (continuación).

Nombre	Centro de desarrollo	Algoritmos de SA	Tipo de licencia	Observaciones
R	Inicialmente por la Universidad de Auckland, nueva Zelanda, actualmente es un proyecto colaborativo abierto a todos los usuarios.	Métodos tipo <i>filter</i> y <i>wrapper</i> además de algoritmos que se basan en simples fórmulas.	Licencia pública GNU.	R es un lenguaje de programación con un enfoque estadístico, puede integrarse con distintas bases de datos y cuenta con una interfaz para interactuar con Weka.
Elvira	Ministerio de educación y ciencia España. TIC97-1135-C04, TIC2001-2973-C05.	Información mutua, Distancia Euclídea, Distancia Matusita, Kullback-leibler 1 y 2 Entropía de Shannon, Distancia Bhattacharyya.	Licencia pública.	Software desarrollado en Java de métodos <i>filter</i> tipo ranking.
Reduclas	Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) en México.	Testores típicos.	Licencia pública.	Proporciona el código fuente en C. Este software solo permite procesar 30 atributos y 400 instancias.
See5	Centro de investigación Rulequest, Australia.	Árbol de decisión que utiliza la ganancia de información como medida de evaluación.	Licencia comercial.	Está desarrollado en C la versión de prueba permite procesar solo 400 instancias de entrenamiento o prueba.
Weka	Universidad de Waikato, Nueva Zelanda.	Métodos de SA tipo <i>filter</i> , <i>wrapper</i> , <i>embedded</i> , heurísticos, etc.	Licencia pública GNU.	Desarrollado en Java, cuenta con un extenso repertorio de algoritmos de SA.

3.2 Discusión

A pesar de que existen un gran número de herramientas de SA, todavía se siguen desarrollando nuevas herramientas de SA, existen diferentes herramientas que es un problema decidirse cuál usar. La mayoría de las herramientas sólo encuentran un solo subconjunto óptimo de atributos o muestran un ranking de atributos, el software Reduclas encuentra todos los subconjuntos óptimos que existen en la BD de aprendizaje supervisado, sacrificando optimalidad por eficiencia.

En México en el CENIDET en diferentes tesis aparecen mención al tema de SA, en un principio se utilizaba principalmente a la Teoría de Testores, existe también un *software*, Reduclas; después cambiaron de herramienta utilizando See5 y Weka para resolver el problema de selección de atributos, no se entiende ese cambio cuando en teoría y práctica para seleccionar atributos es mejor la Teoría de Testores, el motivo puede ser por las restricciones del software Reduclas que se limitan a 30 atributos y 400 instancias, pero al ser de uso libre y proporcionar el código se pueden hacer modificaciones para usarlo con más atributos o instancias y si el problema es el tiempo no utilizar todo el algoritmo, dejarlo sólo en un ranking de atributos.

Capítulo 4

Análisis, diseño e implementación del sistema

En este capítulo se presenta un análisis del algoritmo del *B&B* de selección de atributos, así como el diseño de la heurística que se aplicará para evaluar el subconjunto de atributos.

4.1. Análisis del *B&B*

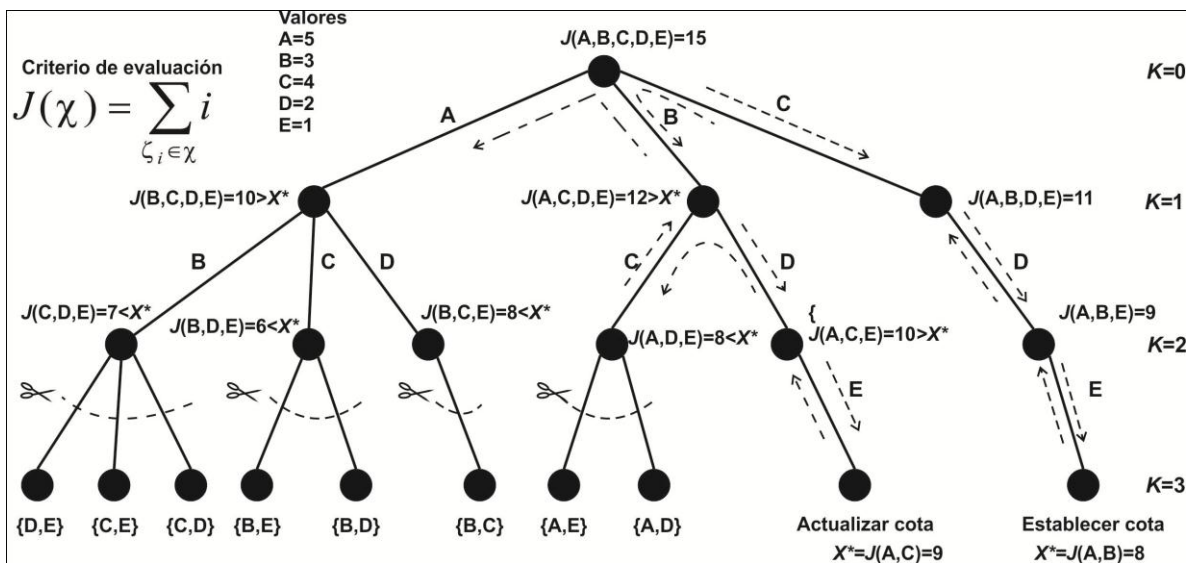


Figura 4.1. Árbol de búsqueda en donde $m=2$ atributos a seleccionar de un conjunto de $n=5$ atributos [Somol, 2004].

El algoritmo se desarrolla usando un árbol como se muestra en la Figura 4.1 siguiendo las flechas punteadas, consiste en ir eliminando atributos hasta llegar a un cierto número de atributos (predeterminados por el usuario). El subconjunto de atributos es evaluado usando una métrica monótonica y el valor obtenido es guardado como una cota (*bound*). A continuación se procede a quitar otros atributos del conjunto original, siguiendo un esquema ordenado de eliminación de atributos (esquema de enumeración); cada subconjunto así obtenido es evaluado. Si algún subconjunto obtiene una evaluación igual o peor que la cota, se detiene la

exploración de esa rama (es decir, se realiza una poda) puesto que continuar la exploración es inútil pues no conduce a una mejor solución que la que ya se tiene actualmente; lo que se quiere es realizar el mínimo de exploraciones (evaluaciones) de subconjuntos, pero obteniendo los atributos óptimos para esa medida de evaluación. Una descripción más detallada de la construcción del árbol de búsqueda se encontrará en [Narendra, 1977] y [Somol, 2004].

En un artículo publicado por [Wang, 2003] encontró datos importantes acerca del número de nodos que tiene el árbol de búsqueda del *B&B* de SA estos son:

El número de combinaciones de n elementos distintos, tomados m a la vez es:

$$C_n^m = \binom{n}{m} = \frac{n!}{m!(n-m)!} \quad (\text{Ec. 21})$$

Tomando en cuenta que n es el número o tamaño del conjunto completo de atributos y m es el número de atributos a seleccionar. El total de nodos en el árbol de búsqueda es:

$$C_{n+1}^{m+1} \quad (\text{Ec. 22})$$

El número de nodos hoja es:

$$C_n^m \quad (\text{Ec. 23})$$

El número de nodos de primer grado, es decir nodos que solo tienen un hijo en el árbol de búsqueda es:

$$C_{n-1}^{m+1} \quad (\text{Ec. 24})$$

En la Figura 4.2 se tiene un árbol de búsqueda con 5 atributos que son {A,B,C,D,E}, y se van a buscar los subconjuntos de 2 atributos que son {D,E}, {C,E}, {C,D}, {B,E}, {B,D}, {B,C}, {A,E}, {A,D}, {A,C}, {A,B}, utilizando las fórmulas de [Wang, 2003], para obtener la cuenta de los diferentes nodos se tiene:

Total de nodos en el árbol de búsqueda, Figura 4.2 es:

$$C_{n+1}^{m+1} = C_{5+1}^{2+1} = \binom{6}{3} = \frac{6!}{3!(6-3)!} = 20 \quad (\text{Ec. 25})$$

El número de nodos hoja, Figura 4.2 es:

$$C_n^m = C_5^2 = \binom{5}{2} = \frac{5!}{2!(5-2)!} = 10 \quad (\text{Ec. 26})$$

El número de nodos de primer grado, nodos huecos o sin rellenar mostrados en la Figura 4.2, es decir nodos que sólo tienen un hijo en el árbol de búsqueda, es:

$$C_{n-1}^{m+1} = C_{5-1}^{2+1} = \binom{4}{3} = \frac{4!}{3!(4-3)!} = 4 \quad (\text{Ec. 27})$$

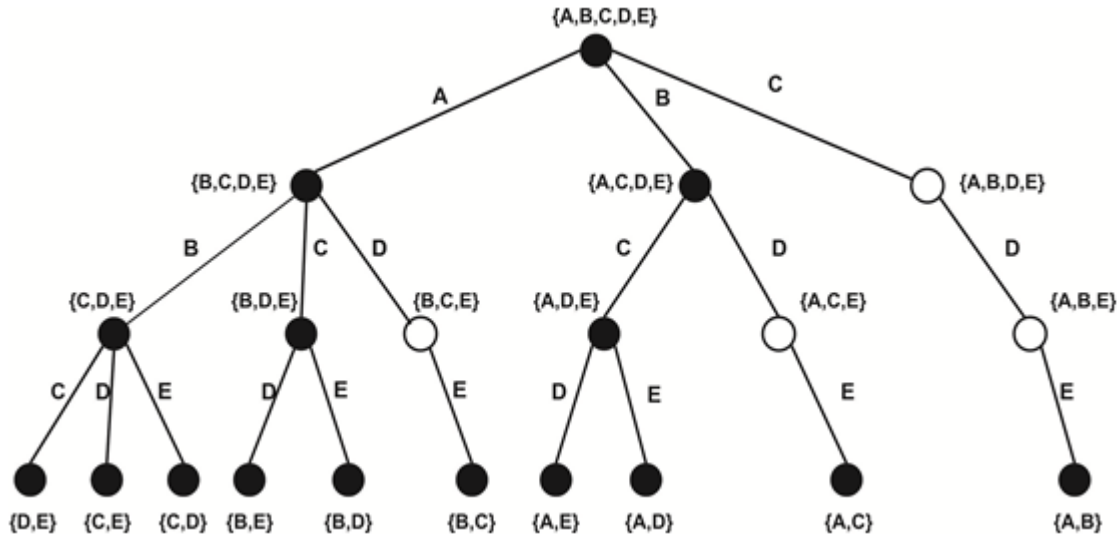


Figura 4.2. Árbol de búsqueda en donde $n=5$ atributos totales y $m=2$ atributos a seleccionar, nodos de primer grado mostrados sin rellenar [Yu, 1993].

Existen algunas mejoras que se han hecho para reducir la complejidad computacional y agilizar la búsqueda del subconjunto de atributos, B. Yu and B. Yuan proponen una búsqueda más compacta del árbol de soluciones a través de un “árbol de solución mínima” [Yu, 1993] el cual simplifica la búsqueda acortando el recorrido a través de los nodos de primer grado, en el artículo explican que los nodos de primer grado pueden saltarse a la hora de evaluación para evaluar solamente las hojas. El recorrido más corto del árbol de búsqueda llamado árbol de solución mínima se muestra en la Figura 4.3.

Como se puede observar el número de operaciones para el peor de los casos en la construcción del árbol de búsqueda, utilizando el árbol de solución mínima es $C_{n+1}^{m+1} - C_{n-1}^{m+1}$, que es el total de nodos creados para el árbol de búsqueda menos los nodos de primer grado, un algoritmo con una complejidad computacional exponencial; la evaluación de eficiencia espacial o memoria utilizada por el *B&B* no es un problema, para la construcción del árbol de búsqueda se necesita una lista dinámica que cuando tenga más elementos, estos elementos no excederán en cantidad al total de atributos, debido a la liberación dinámica de memoria (nodos) que se va haciendo durante la construcción del árbol.

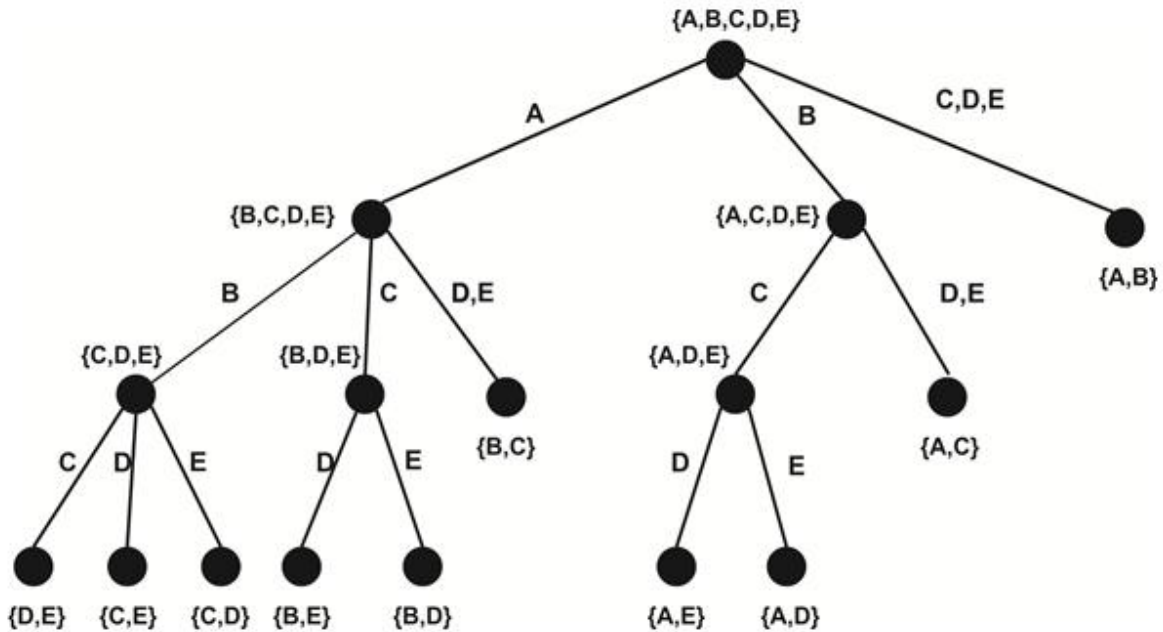


Figura 4.3. Árbol de solución mínima [Yu, 1993] [Wang, 2003].

Para una visualización gráfica de este análisis se muestra la Figura 4.4 obtenida de la Tabla 4.1, en su columna m indica el tamaño del subconjunto a encontrar esto es, cuántos elementos debe de contener cada hoja; la columna nodos hojas, muestra la cantidad de nodos hoja o combinaciones de m atributos que se generan a partir de un conjunto inicial de $n=30$ atributos; la columna total de nodos muestra la cantidad de nodos de todo el árbol, creados para obtener todos los subconjuntos de m atributos; la columna nodos árbol solución mínima, es el resultado de restar el total de nodos menos los nodos de primer grado; la columna nodos evaluados muestra el número de nodos creados y evaluados, la medida de evaluación monótona en este ejemplo consiste en maximizar la suma de los atributos, a cada atributo se le asignaron valores sucesivos como $A=1$, $B=2$, $C=3$, $D=4\dots$ y así hasta llegar al valor del atributo 30; la columna de tiempo de enumeración de todos los nodos, es el tiempo en segundos necesitado para la creación del árbol de búsqueda sin utilizar alguna medida de evaluación; la columna nodos evaluados muestra el tiempo de utilizar la medida de evaluación, como se puede observar es menor que la columna tiempo de enumeración, esto se debe a que en la medida de evaluación hay mucha diferencia de los valores de atributos y esto hace que puedan realizarse varias podas además de que la medida es muy sencilla, sólo asignarle un valor de 1-30 al atributo sin realizar alguna medida directa sobre cada atributo.

Tabla 4.1. Análisis del número de nodos en un árbol de búsqueda con $n=30$ atributos.

		Conjunto total de atributos $n=30$					
	m	Nodos hoja	Total de nodos	Nodos árbol solución mínima	$J(\chi) = \sum_{\xi_i \in X} i$ Nodos evaluados	Enumeración todos los nodos, t (s)	Nodos evaluados, tiempo (s)
		subconjuntos de m atributos	1	30	465	59	465
2	435		4495	841	4495	0.063	0.006
3	4060		31465	7714	30815	0.469	0.034
4	27405		169911	51156	157051	3.229	0.171
5	142506		736281	261261	621817	13.6	0.665
6	593775		2629575	1068795	1972600	47.331	2.157
7	2035800		7888725	3596580	5121165	146.333	5.669
8	5852925		20160075	10145070	11056671	375.248	12.74
9	14307150		44352165	24322155	20113434	834.647	24.477
10	30045015		84672315	50075025	31181163	1604.56	38.546
11	54627300		141120525	89224590	41625228	2719.46	52.278
12	86493225		206253075	138389160	48326987	4006.43	32.855
13	119759850		265182525	187623765	49276590	5243.72	30.577
14	145422675		300540195	222981435	44562170	6015.28	28.682
15	155117520		300540195	232676280	36093867	6089.57	23.794
16	145422675		265182525	213286590	26439200	5560.12	21.729
17	119759850		206253075	171655785	17677558	4304.78	14.243
18	86493225		141120525	121090515	10878824	2982.12	9.518
19	54627300		84672315	74657310	6205283	1786.18	6.617
20	30045015		44352165	40060020	3297561	966.713	4.046
21	14307150		20160075	18599295	1637168	344.564	2.23
22	5852925		7888725	7413705	759413	177.542	1.153
23	2035800		2629575	2510820	327802	55.383	0.527
24	593775		736281	712530	130310	17.016	0.23
25	142506		169911	166257	46535	3.766	0.083
26	27405		31465	31059	14154	0.704	0.026
27	4060		4495	4466	3273	0.095	0.007
28	435		465	464	465	0.524	0.001
29	30		31	31	31	0.079	0.001
30	1		1	1	1	0	0.001

Si se compara la columna de nodos evaluados con la columna de nodos enumerados, se puede observar que aun en el peor de los casos se realizan podas, es una cantidad menor de nodos creados que la columna nodos enumerados, esto no pasaría si tuvieran todos los atributos valores similares en donde si en el peor de los casos la cantidad de nodos creados sería una cantidad similar a la columna total de nodos.

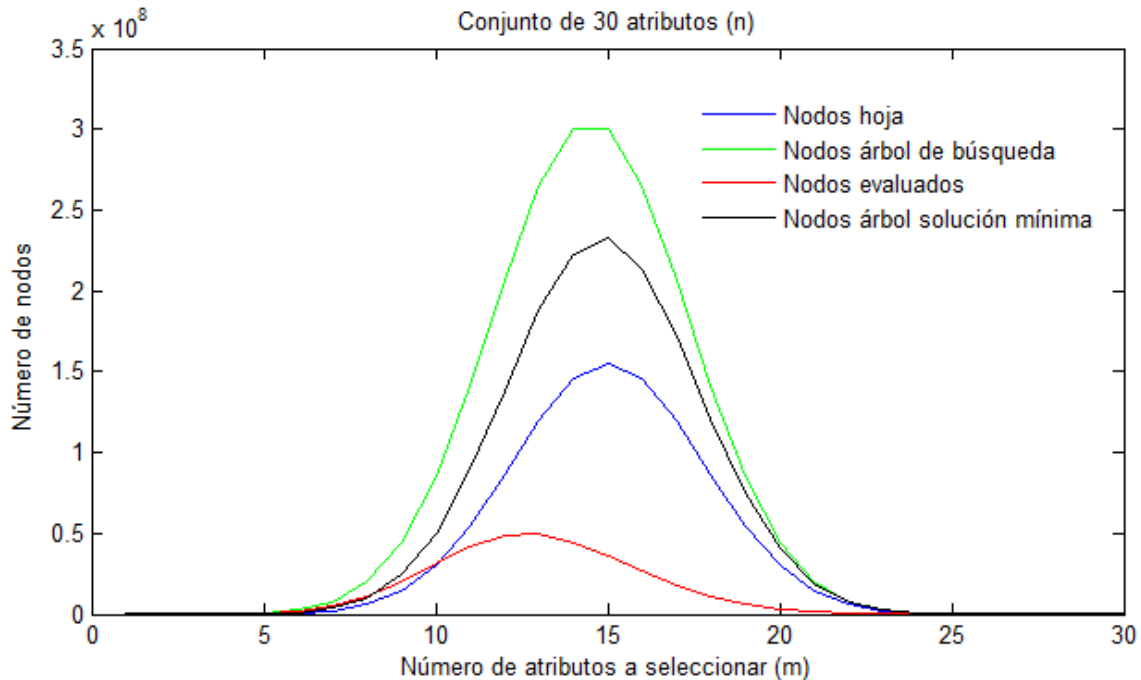


Figura 4.4. Análisis del número de nodos en un árbol de búsqueda con n=30 atributos.

La Figura 4.5 muestra los tiempos en segundos de pruebas de eficiencia temporal, estas pruebas fueron hechas con una computadora Lenovo G460 core i5, 8 gigas de RAM. La Figura 4.5 (a) es la gráfica de la columna enumeración todos los nodos de la Tabla 4.1, muestra el tiempo en segundos para enumerar todos los nodos del árbol y desplegarlos en pantalla, solo desplegar los subconjuntos creados sin ninguna medida de evaluación; la Figura 4.5 (b) es la gráfica de la columna nodos evaluados de la Tabla 4.1, muestra el tiempo en segundos para encontrar el subconjunto óptimo con la medida de evaluación de maximizar la suma de los atributos, como se puede ver es un tiempo menor que la columna enumeración todos los nodos debido a que existe diferencia en el valor de cada atributo, si los atributos tuvieran un valor similar en el peor de los casos sería un tiempo superior a la columna enumeración todos los nodos ya que aparte de ir creando todos los nodos del árbol se tiene que ir evaluando cada nodo, la evaluación del nodo es lo que en realidad consume tiempo de cálculo porque ahí es en donde se evalúa el subconjunto de atributos para poder realizar las podas, es por esto que derivado del tiempo consumido en la evaluación de cada nodo, diversos autores han propuesto mejoras para evitar evaluar todos los nodos, una mejora es la propuesta por [Somol, 2004].

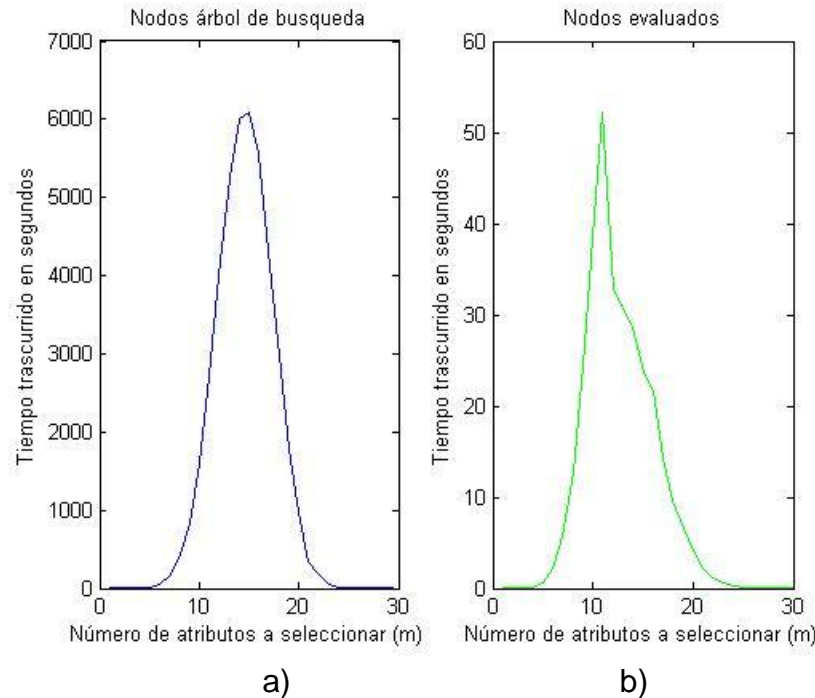


Figura 4.5. Análisis eficiencia temporal.

[Somol, 2004] propone una variante denominada FBB (*Fast Branch & Bound*) la cual tiene las siguientes ideas:

- Usar un mecanismo para predecir el valor del subconjunto de atributos, en vez de calcularlo usando el criterio real, el cual es más costoso de calcular.
- Llevar un “conteo promedio” de la importancia de cada atributo, es decir, al quitar un atributo examinar cuánto disminuye el valor del criterio de evaluación.
- Una vez se seleccionen los mejores atributos, calcular el valor con el criterio real (sólo a los mejores, según el mecanismo de predicción).

En la Figura 4.6 se presenta gráficamente la forma de operar del algoritmo. En el nodo raíz se evalúan todos los atributos usando el criterio real (lo cual es indicado por el nodo con una “C”). A continuación se elimina el atributo número 4 (siguiendo la rama de la derecha), se evalúa el nuevo subconjunto y se obtiene la diferencia con respecto al subconjunto anterior, que en este caso es $24.6 - 19.5 = 5.1$; el valor así obtenido es la primera estimación para cuando se elimina el atributo 4; en forma similar se obtienen los estimados para los atributos número 3, 2 y 1, y se obtiene la primera cota que es de 11.4. A continuación se regresa al nodo raíz en donde se elimina el atributo número 5, se evalúa el nuevo subconjunto y se compara contra la cota actual: al no haber poda, se procede a eliminar el atributo número 3; como de este atributo ya se tiene un estimado (que es 3.9) el nuevo subconjunto de atributos no se evalúa usando el criterio real, sino que se obtiene una predicción (marcada en el nodo por la letra “P”) la cual se obtiene de manera

económica restando, a la evaluación anterior, el valor estimado de quitar el atributo 3, esto es $18.5 - 3.9 = 14.6$. Como este valor no es mejor que la cota actual, no hay poda, siguiéndose el proceso normal. Cuando se eliminan los atributos 5, 4 y 2, se da el caso de que el valor predicho es peor que la cota actual (10.7 contra 11.4), por lo cual existe posibilidad de poda, la cual es comprobada calculando el valor real del subconjunto, que en este caso es de 10.9 y como sigue siendo peor que la cota actual, se realiza la poda, señalada por la tijera.

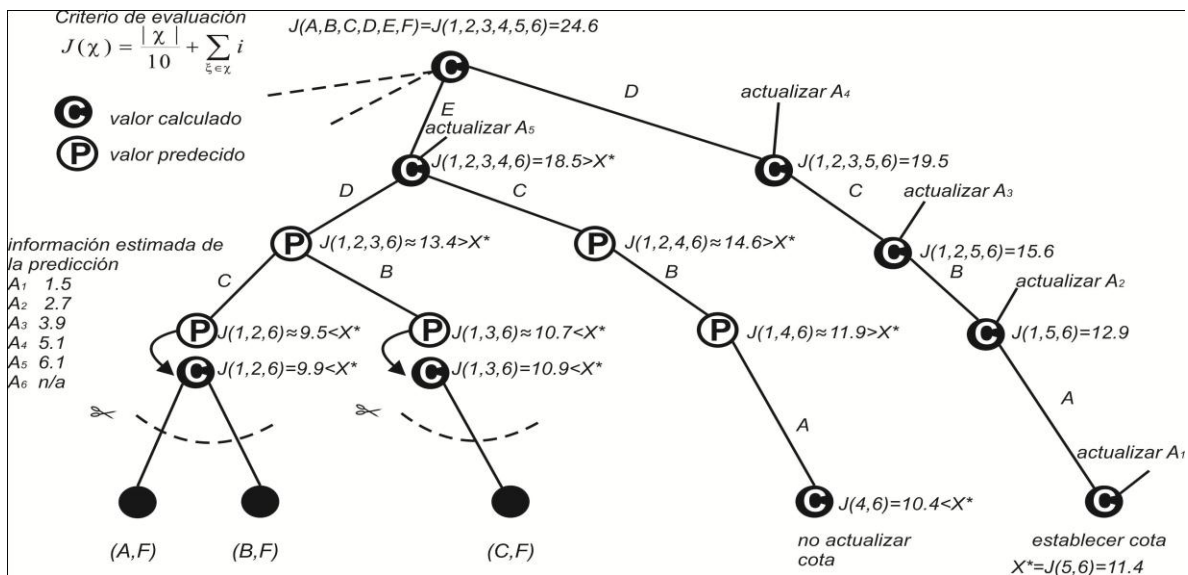


Figura 4.6. Ilustración del mecanismo de predicción del *Fast Branch & Bound* en un problema sintético donde $m=2$ y $n=6$ [Somol, 2004].

Los resultados que se obtienen con esta variante están estrechamente relacionados al mecanismo para predecir, que se obtiene con las estimaciones anteriormente obtenidas: si las predicciones son malas, el resultado final estará lejano al óptimo, por lo cual este sería el mayor inconveniente de lo propuesto por Somol. En los experimentos realizados por Somol existen casos extremos en que la variante propuesta resulta peor que realizar una búsqueda exhaustiva, aunque también se experimentó con casos donde lo propuesto resulta de 1.5 hasta 10 veces más rápido que el algoritmo de [Narendra, 1977].

4.2. Observaciones, detalles y desventajas del B&B

Comparado con la búsqueda exhaustiva para encontrar un subconjunto de tamaño m , el algoritmo del *B&B* requiere cálculos adicionales, no sólo evaluar los subconjuntos de tamaño m , sino que necesita evaluar todos los subconjuntos previos en las ramas del árbol construido.

El algoritmo del *B&B* no garantiza que siempre se van a podar ramificaciones para tener un número menor de evaluaciones que la búsqueda exhaustiva, las podas se van realizando conforme se van generando y evaluando los subconjuntos de

atributos, en el peor de los casos podría evaluarse cada nodo del árbol, con lo que resultaría peor que la búsqueda exhaustiva.

Puede tener un desempeño pobre debido a que empieza evaluando todos los atributos, cerca de la raíz la medida de evaluación consume más tiempo por ser subconjuntos grandes así como ser menos frecuentes las podas.

La medida de evaluación del subconjunto necesita ser monótona por lo cual no se puede utilizar la precisión como medida de evaluación.

Para su funcionamiento se tiene que conocer previamente la cantidad de los atributos relevantes, para proporcionarlos al *B&B* como dato de entrada.

El subconjunto óptimo proporcionado por el algoritmo del *B&B* es óptimo en relación a la medida de evaluación establecida, puede no ser el óptimo para representar a toda la base de datos o el subconjunto que más precisión tenga para un problema de clasificación.

El funcionamiento depende significativamente de la medida de evaluación del subconjunto y de la base de datos.

Por ser un método heurístico no se puede esperar trabajar bien en todos los problemas.

Los atributos deben de tener diferencias medibles, cuando los diferentes atributos contribuyen de manera similar a la medida de evaluación puede no haber podas y recorrerse todo el árbol.

En el peor de los casos su complejidad computacional es exponencial, es igual al total de nodos creados en el árbol de búsqueda que sería una cantidad de C_{n+1}^{m+1} y en el mejor de los casos es una complejidad computacional lineal equivalente a $n - m$ nodos evaluados, donde n es el conjunto total de atributos y m el conjunto de atributos a seleccionar.

4.3. Diseño del Sistema

Si la tarea de selección de atributos es para clasificación el objetivo de la selección de atributos será maximizar la exactitud de predicción. Es por lo tanto razonable que la precisión del clasificador sea la medida principal de evaluación [Liu, 1998].

La precisión no puede ser estimada hasta que el clasificador es construido con los atributos seleccionados y no es una medida monótonica por lo cual no se puede usar con el algoritmo del *B&B* por lo que se escogió como medida para encontrar los atributos relevantes, la correlación y se toma como hipótesis de que un buen subconjunto de atributos contiene atributos altamente correlacionados con la clase e independientes entre sí [Witten, 2011].

Existen en general dos tipos de medidas para la correlación entre dos variables aleatorias: lineal y no lineal [Yu, 2004]. De correlación lineal, la medida más conocida es el coeficiente de correlación lineal de Pearson, entre las medidas de correlación no lineal, muchas medidas se basan en el concepto de la entropía de la información, que es una medida de la incertidumbre de una variable aleatoria; también se usa como medida de evaluación para hallar atributos relevantes la distancia Bhattacharyya, que mide la similitud de dos distribuciones de probabilidad.

4.3.1. Medidas de evaluación escogidas

La medida escogida para evaluar los subconjuntos de atributos es la correlación, específicamente se usará el coeficiente simétrico de incertidumbre que es un tipo de información mutua normalizada *Symmetrical Uncertainty* (SU); haciendo uso de las medidas de entropía vistas en la sección 3.5 se llega a la medida propuesta por Witten el SU.

Coeficiente simétrico de incertidumbre [Witten, 2011].

$$SU(X, Y) = 2 \frac{IG(X|Y)}{H(X) + H(Y)} \quad (Ec. 28)$$

El SU, compensa la ganancia de información de atributos con muchos valores, y restringe su valor a un rango de [0,1]. Un valor de 1 indica que el conocimiento de cualquier atributo predice completamente el valor del otro. Un valor de 0 indica que X y Y son independientes.

Otra medida escogida para evaluar el subconjunto de atributos es la distancia Bhattacharyya.

Distancia Bhattacharyya [Armañanzas, 2004].

$$Bh(X, C) = \sum_{i=1}^{r_c} -\log \left[P(C_i) \sum_{j=1}^{r_x} \sqrt{P(x_j|c_i)P(x_j)} \right] \quad (Ec. 29)$$

Donde, X es la variable a evaluar, C es la variable clase, r_x es el número de estados que X puede tomar, y r_c es el número de estados que C puede tomar.

Esta distancia mide la dependencia que existe entre dos distribuciones de probabilidad. Las distribuciones que van a ser comparadas son las probabilidades *a priori* de una variable, contra la condicionada a la clase. Se intenta de esta forma ver qué grado de dependencia se encuentra entre ambas distribuciones; cuanto mayor sea ese grado, mayor será el peso de la variable analizada en el problema clasificatorio.

4.3.2. Diseño del criterio de evaluación para seleccionar atributos relevantes

Las medidas de evaluación escogidas funcionan sólo con valores nominales o discretos, por lo que se discretiza los valores de atributos que sean de tipo continuo, existen diferentes formas de discretización de valores continuos, la usada en este trabajo es una discretización no supervisada por división de intervalos con igual anchura siendo 10 el número de intervalos.

Con los datos discretos se saca el coeficiente simétrico de incertidumbre de cada par de atributos incluyendo la clase, se tiene una matriz ordenada donde las intersecciones de filas y columnas corresponde al SU de los diferentes atributos, un valor de 1 indica que el atributo X y Y están correlacionados completamente y un valor de 0 que son independientes, se tiene una matriz como la mostrada en la Tabla 4.2.

Tabla 4.2. Valores SU de cada par de atributos.

	Atr1	Atr2	Atr3	Atr4	Atr5	...Atrn	CLASE
Atr1	SU(Atr1,Atr1)	SU(Atr1,Atr2)	SU(Atr1,Atr3)	SU(Atr1,Atr4)	SU(Atr1,Atr5)	SU(Atr1,Atrn)	SU(Atr1,CLASE)
Atr2	SU(Atr2,Atr1)	SU(Atr2,Atr2)	SU(Atr2,Atr3)	SU(Atr2,Atr4)	SU(Atr2,Atr5)	SU(Atr2,Atrn)	SU(Atr2,CLASE)
Atr3	SU(Atr3,Atr1)	SU(Atr3,Atr2)	SU(Atr3,Atr3)	SU(Atr3,Atr4)	SU(Atr3,Atr5)	SU(Atr3,Atrn)	SU(Atr3,CLASE)
Atr4	SU(Atr4,Atr1)	SU(Atr4,Atr2)	SU(Atr4,Atr3)	SU(Atr4,Atr4)	SU(Atr4,Atr5)	SU(Atr4,Atrn)	SU(Atr4,CLASE)
Atr5	SU(Atr5,Atr1)	SU(Atr5,Atr2)	SU(Atr5,Atr3)	SU(Atr5,Atr4)	SU(Atr5,Atr5)	SU(Atr5,Atrn)	SU(Atr5,CLASE)
...Atrn	SU(Atrn,Atr1)	SU(Atrn,Atr2)	SU(Atrn,Atr3)	SU(Atrn,Atr4)	SU(Atrn,Atr5)	SU(Atrn,Atrn)	SU(Atrn,CLASE)

Según [Witten, 2011] una forma de eliminar atributos irrelevantes y redundantes es seleccionando un subconjunto de atributos que estén individualmente correlacionado con la clase y que tengan una pequeña intercorrelación entre ellos, por motivos de diseño de la heurística se modifica la Tabla 4.2, los valores del SU de los atributos con la clase permanecen igual (parte sombreada) y a los valores de cada par de atributos se le resta 1 luego se le saca valor absoluto, quedando como en la Tabla 4.3. Esto se hace porque lo que se necesita es que se sume la correlación con la clase y para que cuando se le sume la correlación entre atributos y estén completamente correlacionados se le sume una cantidad de cero o si hay independencia de atributos que tendría un valor cercano a cero se sume un valor cercano a uno.

Tabla 4.3. Valores SU modificados para la métrica monotónica.

	X_1	X_2	X_3	X_4	X_5	$\dots X_n$	$CLASE(X_c)$
X_1	$SU^\circ(X_1, X_1)$	$SU^\circ(X_1, X_2)$	$SU^\circ(X_1, X_3)$	$SU^\circ(X_1, X_4)$	$SU^\circ(X_1, X_5)$	$SU^\circ(X_1, X_n)$	$SU^\circ(X_1, X_c)$
X_2	$SU^\circ(X_2, X_1)$	$SU^\circ(X_2, X_2)$	$SU^\circ(X_2, X_3)$	$SU^\circ(X_2, X_4)$	$SU^\circ(X_2, X_5)$	$SU^\circ(X_2, X_n)$	$SU^\circ(X_2, X_c)$
X_3	$SU^\circ(X_3, X_1)$	$SU^\circ(X_3, X_2)$	$SU^\circ(X_3, X_3)$	$SU^\circ(X_3, X_4)$	$SU^\circ(X_3, X_5)$	$SU^\circ(X_3, X_n)$	$SU^\circ(X_3, X_c)$
X_4	$SU^\circ(X_4, X_1)$	$SU^\circ(X_4, X_2)$	$SU^\circ(X_4, X_3)$	$SU^\circ(X_4, X_4)$	$SU^\circ(X_4, X_5)$	$SU^\circ(X_4, X_n)$	$SU^\circ(X_4, X_c)$
X_5	$SU^\circ(X_5, X_1)$	$SU^\circ(X_5, X_2)$	$SU^\circ(X_5, X_3)$	$SU^\circ(X_5, X_4)$	$SU^\circ(X_5, X_5)$	$SU^\circ(X_5, X_n)$	$SU^\circ(X_5, X_c)$
$\dots X_n$	$SU^\circ(X_n, X_1)$	$SU^\circ(X_n, X_2)$	$SU^\circ(X_n, X_3)$	$SU^\circ(X_n, X_4)$	$SU^\circ(X_n, X_5)$	$SU^\circ(X_n, X_n)$	$SU^\circ(X_n, X_c)$

Con la Tabla 4.3 se puede ir evaluando el subconjunto de atributos seleccionado en base a los criterios 1 y 2 propuestos a continuación.

Sea N el número de atributos de un subconjunto, y sea X_1, \dots, X_n los atributos de dicho subconjunto, X_c es el atributo de clase que es igual a X_{n+1} , además sea el coeficiente simétrico de incertidumbre modificado $SU^\circ(X_i, X_j)$ de cada par de los elementos del subconjunto, la evaluación de ese subconjunto de atributos es:

Criterio 1

$$\sum_{i=1}^N \sum_{j=1}^{N+1} SU^\circ(X_i, X_j) \quad \text{para } i \neq j \quad (\text{Ec. 30})$$

Criterio 2

$$\sum_{i=1}^N \left(\sum_{j=1}^N SU^\circ(X_i, X_j) \right) * \frac{SU^\circ(X_i, X_c)}{N} \quad \text{para } i \neq j \quad (\text{Ec. 31})$$

El criterio 1 sólo dice que son todas las combinaciones entre ellos incluida la clase, por ejemplo, para un subconjunto de 3 atributos (X_2, X_4, X_6) la evaluación del criterio 1 es igual la suma de lo siguiente: $SU^\circ(X_2, X_4) + SU^\circ(X_2, X_6) + SU^\circ(X_2, X_c) + SU^\circ(X_4, X_2) + SU^\circ(X_4, X_6) + SU^\circ(X_4, X_c) + SU^\circ(X_6, X_2) + SU^\circ(X_6, X_4) + SU^\circ(X_6, X_c)$.

Para el criterio 2 para un subconjunto de 3 atributos (X_2, X_4, X_6) la evaluación es: $[SU^\circ(X_2, X_4) + SU^\circ(X_2, X_6)] * \frac{SU^\circ(X_2, X_c)}{3} + [SU^\circ(X_4, X_2) + SU^\circ(X_4, X_6)] * \frac{SU^\circ(X_4, X_c)}{3} + [SU^\circ(X_6, X_2) + SU^\circ(X_6, X_4)] * \frac{SU^\circ(X_6, X_c)}{3}$.

Se sigue el mismo procedimiento para la medida de la distancia Bhattacharyya.

4.4. Implementación del sistema

Para la implementación del sistema se sigue el diagrama de la Figura 4.7 en el cual se observan 6 etapas.

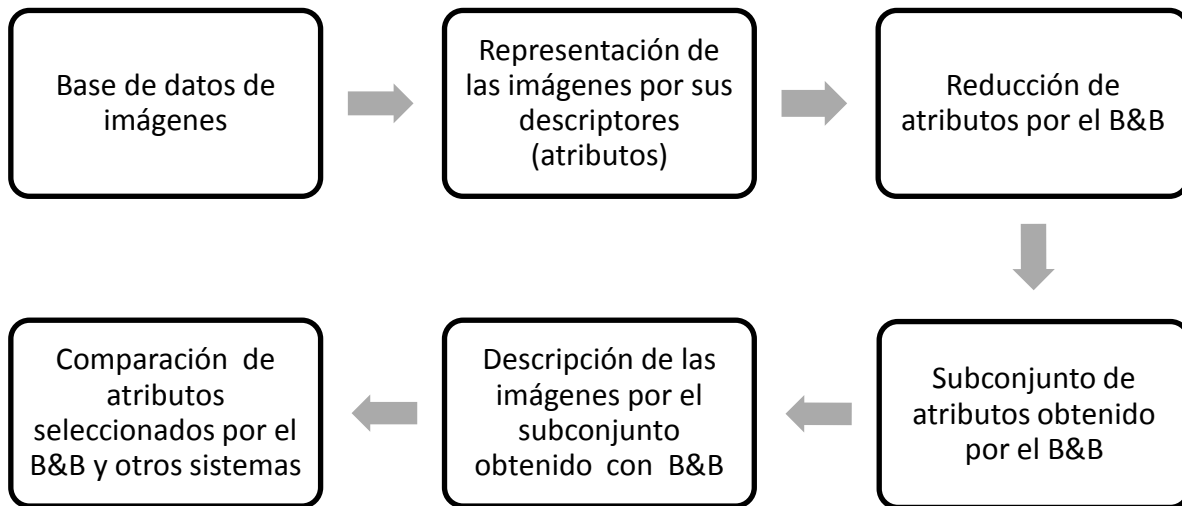


Figura 4.7. Diagrama de la metodología de la solución.

4.4.1. Base de datos de imágenes

Como primera etapa se tiene la adquisición de la base de datos de imágenes, para esta etapa se hace uso de las bases de datos usadas en las tesis de [Ochoa, 2005], particularmente se usarán 2 bases de datos de esa investigación que son: *Image segmentation dataset* [Segmentation, 2016] y *Image dataset* [Image, 2016], además de la Base de Datos *Brest Cancer Wisconsin (Diagnostic dataset)* [WDBC, 2016], que se encuentran explicadas en el anexo A.

4.4.2. Representación de imágenes por sus descriptores

Los descriptores escogidos se muestran en la Figura 4.8, los cuales fueron usados en tesis de maestría hechas en el CENIDET.

Todos los descriptores mostrados en la Figura 4.8 se implementaron para el dominio RGB y/o escala de grises, no se transformó el dominio como lo propuesto por [Galeana, 2008] y [Avellaneda, 2009].

Para la experimentación de los diferentes métodos de selección de atributos se utilizaron las bases de datos mencionadas en la sección 4.4.1, estas vienen con sus imágenes y con los valores de sus diferentes descriptores, los descriptores de estas bases de datos son diferentes a los implementados en la Figura 4.8, por lo

que para las pruebas se toman las bases de datos ya con los valores de sus diferentes descriptores, las bases de datos con valores continuos se discretizan en intervalos de 10 tomando el valor mínimo como 0 y el máximo como 9.

ESTADÍSTICOS DE PRIMER ORDEN	ESTADÍSTICOS DE SEGUNDO ORDEN	MOMENTOS
<ul style="list-style-type: none"> • Media • Varianza • Desviación estándar 	<ul style="list-style-type: none"> • Probabilidad máxima • Entropía • Contraste o inercia • Correlación • Energía • Homogeneidad local • Directividad • Uniformidad • Kurtosis 	<ul style="list-style-type: none"> • Normales • Centrales • De Hu • De Maïtra

Figura 4.8. Lista de descriptores implementados

4.4.3. Reducción de atributos por el B&B

Para la realización de esta etapa se utilizó el algoritmo del *B&B* de SA propuesto en [Narendra, 1977] y como medida de evaluación del subconjunto se utilizó la información mutua específicamente el coeficiente de incertidumbre simétrica que mide correlación no lineal, además de la distancia Bhattacharyya que mide la similitud de dos distribuciones de probabilidad

La forma de evaluar el subconjunto es en base a dos criterios de evaluación descritos en la sección 4.3.2.

4.4.4. Subconjunto de atributos obtenido por el B&B

En esta etapa sólo se va a registrar los resultados obtenidos por los dos criterios de evaluación propuestos para las medidas del coeficiente simétrico y la distancia Bhattacharyya.

4.4.5. Descripción de imágenes con el subconjunto obtenido por el B&B

En esta etapa para verificar la calidad y eficiencia del *B&B* se experimentará la precisión de instancias correctamente clasificadas con 2 clasificadores el J48 y el Naïve Bayes ambos de la herramienta Weka.

4.4.6. Comparación del subconjunto de atributos encontrado por el B&B y otros sistemas

En esta etapa se hará uso de las herramientas de SA Elvira, See5 y Reduclas , se comparará los subconjuntos obtenidos por estas herramientas y por el *B&B*; para

comparar en términos de la precisión del clasificador, los clasificadores a usar son el J48 y el Naïve Bayes de la herramienta Weka, en ambos clasificadores se entrenará y probará con todo el conjunto de instancias y con validación cruzada de 10 particiones.

4.5. Discusión

Los algoritmos eficientes son algoritmos que corren en tiempo polinomial, los algoritmos exponenciales comúnmente corren más lentamente que los polinomiales, lo que se quiere son algoritmos eficientes con resultados óptimos. En algunos casos no se conoce algoritmo de tiempo polinomial con resultados óptimos, la búsqueda del subconjunto óptimo de atributos de menor tamaño posible es un problema NP-completo como lo muestra [Davies, 1994], una manera de tratar problemas NP-completos es mediante métodos heurísticos, en muchos casos las búsquedas heurísticas dan mejoras significativas pero en otros no se consigue mejorar sustancialmente la eficiencia y la optimalidad por lo cual el problema persiste; el problema en general con las heurísticas son las suposiciones y restricciones que se tiene que hacer; para el uso del *B&B* hay que definir primeramente de qué tamaño es el subconjunto de atributos a encontrar, esta suposición equivale a resolver anticipadamente parte del problema de SA, pero aun conociendo el tamaño del subconjunto si utilizamos el *B&B* no implica una gran ventaja ya que como se vio en el análisis su complejidad computacional sigue siendo exponencial, por esta razón investigadores como [Pudil, 1994], [Somol, 2010], [Nakariyakul, 2009] que han propuesto mejoras al *B&B* no recomiendan su uso para problemas de SA de alta dimensionalidad.

Capítulo 5

Experimentación y resultados

En este capítulo se presentan las pruebas realizadas a los diferentes sistemas de selección de atributos para comparar su desempeño de calidad y eficiencia.

5.1. Detalles y objetivos de la experimentación

Para poder elegir un método de selección de atributos se pueden tomar en cuenta dos aspectos importantes que son la eficiencia del algoritmo y la calidad de la solución, el tiempo de procesamiento dice que tan eficiente es el método y la calidad de la solución da la optimalidad del subconjunto encontrado, por lo que se evaluará el desempeño del *B&B* propuesto contra algunos sistemas representativos de selección de atributos estas pruebas son:

- Pruebas de calidad
- Pruebas de eficiencia

5.1.1. Hardware

Los experimentos se llevaron a cabo en una computadora portátil Lenovo G460 con procesador Intel Core i5 a 2.67Ghz y con 8 GB de RAM, con sistema operativo Windows 7 Home Premium; para poder usar el sistema Reduclas se utilizó una máquina virtual, instalada en la misma computadora con un procesador de un núcleo a 2.67Ghz con 2 GB de RAM y sistema operativo XP.

5.1.2. Bases de datos

Se usarán 2 bases de datos usadas en [Ochoa, 2005] que son: *Image segmentation dataset* [Segmentation,2016], *Image dataset* [Image, 2016] y la Base de Datos (BD) *Brest cancer Wisconsin (Diagnostic dataset)* [WDBC,2016], que se encuentran explicadas en el anexo A.

5.1.3. Sistemas de selección de atributos

Los sistemas escogidos para comparar el desempeño de eficiencia y calidad contra el *B&B* son: Elvira [Elvira, 2016], See5 [Rulequest, 2016] y Reduclas [Ochoa, 2005] mismos que se encuentran descritos en el anexo B.

5.2. Pruebas de calidad y eficiencia

5.2.1. Pruebas de calidad

La calidad de un subconjunto de atributos puede ser medida de manera directa solo para bases de datos artificiales de las cuales se conoce previamente sus atributos relevantes, con sólo comparar el subconjunto encontrado por el método de SA contra el subconjunto conocido se puede establecer una similitud, para bases de datos reales no se conoce por anticipado el subconjunto óptimo de atributos, por lo cual vamos a usar una medida indirecta de calidad, esta medida corresponde a la precisión de un clasificador.

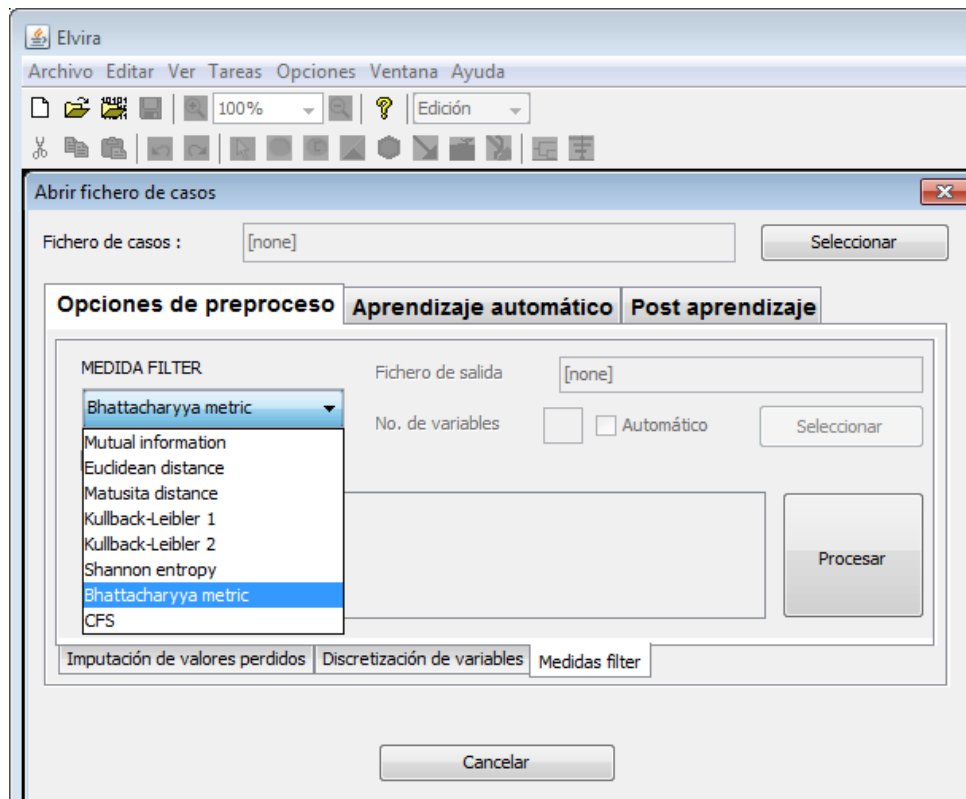


Figura 5.1. Herramienta Elvira.

La comparación se efectuó sobre las bases de datos *Image segmentation dataset* [Segmentation,2016], *Image dataset* [Image, 2016] y la *Brest Cancer Wisconsin (Diagnostic dataset)* [WDBC,2016], que se encuentran resumidas en el anexo A. Se seleccionaron 2 clasificadores de la herramienta Weka Figura 5.4 para evaluar la precisión del subconjunto seleccionado, son los clasificadores J48 y Naïve Bayes, para entrenar y evaluar el clasificador se usarán todas las instancias, también se evaluará el clasificador con validación cruzada en bloques de 10.

Para hacer la comparación se escogen 7 métodos de SA de la herramienta Elvira explicada en el anexo B la cual se muestra en la Figura 5.1 estos métodos *filter*

son: Información Mutua, Distancia Euclidiana, Distancia Matusita, Kullback-Leibler 1, Kullback-Leibler 2, Entropía de Shannon y Distancia Bhattacharyya, además de la herramienta Reduclas Figura 5.2 que utiliza la Teoría de Testores y la herramienta See5, explicadas en el anexo B.



Figura 5.2. Herramienta Reduclas.

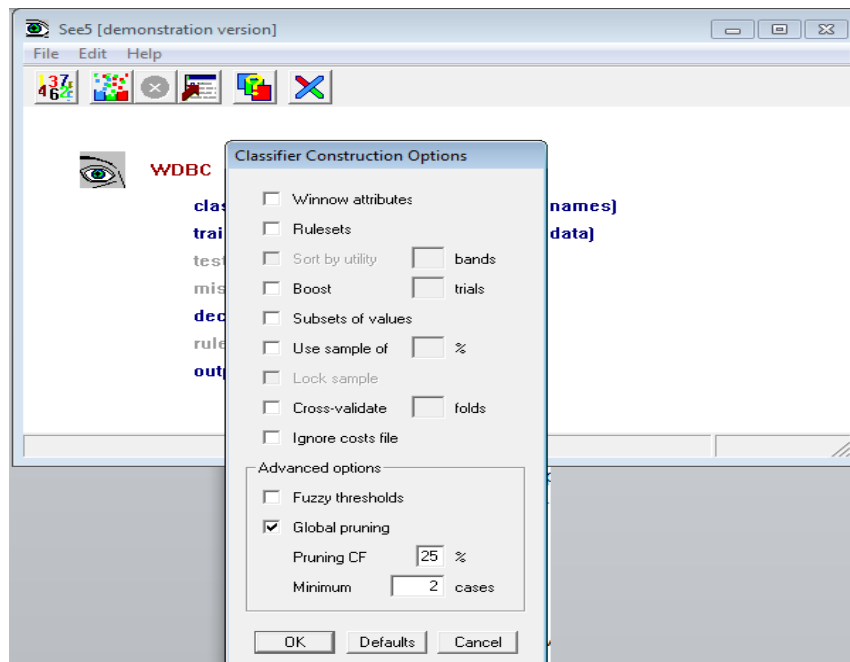


Figura 5.3. Herramienta See5.

Los métodos de la herramienta Elvira generan un ranking de atributos es decir ordenan los atributos de mayor a menor dependiendo de la medida usada, los rankings no detectan redundancia, como la salida es un ranking es difícil saber cuántos y cuáles son los atributos necesarios para lograr una buena clasificación. Para comparar subconjuntos de atributos del ranking vamos a usar la forma de [Ruiz, 2006] la cual consiste en empezar con el primer atributo e irle añadiendo uno por uno hasta usar todos los atributos disponibles, aunque no es una comparación justa ya que en este caso se va a evaluar una lista ordenada de atributos contra el método de *B&B* que encuentra un subconjunto en particular de n atributos, se puede ver cuál subconjunto de atributos permite un mejor desempeño del clasificador.



Figura 5.4. Herramienta Weka.

La Tabla 5.1 muestra los rankings obtenidos por la herramienta Elvira al evaluar la BD *Image* con las medidas de: Información Mutua, Distancia Euclidiana, Distancia Matusita, Kullback-Leibler 1, Kullback-Leibler 2, Entropía de Shannon y Distancia Bhattacharyya. Para cada método del ranking, la fila R_k presenta el ranking de atributos generado con uno de los métodos, bajo esta fila, el resultado de clasificación con J48 y Naïve Bayes utilizando el número de atributos del ranking indicado en la primer fila de la Tabla 5.1, bajo el título Subconjunto; al lado de cada clasificador se tiene las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador, CV-10: validación cruzada en bloques de 10.

En la Tabla 5.1 no aparece el tiempo para obtener los rankings el cual es de 0.4 segundos para cualquier método.

Tabla 5.1. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD *Image*, de los rankings obtenidos por la herramienta Elvira. Rk: ranking de atributos; TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

Subconjunto de		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
Rk Mutual informatio		1,	3,	13,	15,	4,	2,	12,	8,	17,	6,	5,	21,	11,	20,	19,	16,	18,	14,	7,	10,	9	
J48	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
	CV-10	73.17	75.61	93.50	97.56	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
Naïve Bayes	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100
	CV-10	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100	100
Rk Euclidian distance		1,	3,	13,	4,	15,	2,	12,	8,	17,	5,	6,	21,	11,	20,	19,	16,	18,	7,	14,	10,	9	
J48	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
	CV-10	73.17	75.61	93.50	97.56	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
Naïve Bayes	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100
	CV-10	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100	100
Rk Matusita distance		1,	3,	13,	15,	4,	2,	12,	17,	8,	6,	5,	21,	20,	11,	16,	19,	18,	14,	7,	10,	9	
J48	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
	CV-10	73.17	75.61	93.50	97.56	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
Naïve Bayes	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100
	CV-10	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100	100
Rk Kullback-Leibler 1		1,	3,	13,	15,	4,	2,	12,	8,	17,	6,	5,	21,	11,	20,	19,	16,	18,	7,	14,	10,	9	
J48	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
	CV-10	73.17	75.61	93.50	97.56	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
Naïve Bayes	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100
	CV-10	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100	100
Rk Kullback-Leibler 2		1,	3,	13,	15,	4,	2,	12,	8,	17,	6,	5,	21,	11,	20,	19,	16,	18,	14,	7,	10,	9	
J48	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
	CV-10	73.17	75.61	93.50	97.56	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
Naïve Bayes	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100
	CV-10	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100	100
Rk Shannon entropy		1,	3,	13,	15,	4,	2,	12,	8,	17,	6,	5,	21,	11,	20,	19,	16,	18,	14,	7,	10,	9	
J48	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
	CV-10	73.17	75.61	93.50	97.56	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
Naïve Bayes	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100
	CV-10	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100	100
Rk Bhatacharyya		1,	3,	4,	13,	15,	2,	12,	8,	17,	5,	6,	21,	20,	11,	19,	16,	18,	7,	14,	10,	9	
J48	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
	CV-10	73.17	75.61	93.50	97.56	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.19	
Naïve Bayes	TS	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	99.18	99.19	100	100	100	100	100	100	100	100	100
	CV-10	73.17	75.61	93.50	97.56	99.19	99.19	99.19	99.19	99.19	99.19	99.19	100	100	100	100	100	100	100	100	100	100	100

La Tabla 5.2 muestra los resultados del *B&B* utilizando información mutua y los criterios explicados en la sección 4.3.2 para la BD *Image*; bajo la columna #AT está el tamaño del subconjunto, a un lado del tamaño del subconjunto se encuentra los atributos que componen ese subconjunto y abajo del subconjunto de atributos la precisión de los clasificadores J48 y Naïve Bayes, utilizando las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

Tabla 5.2. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD *Image*, de los subconjuntos del *B&B* obtenidos con la medida de información mutua. TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10; t(s): tiempo en segundos.

#AT	Criterio 1				t(s)	Criterio 2				t(s)
	J48		Naïve Bayes			J48		Naïve Bayes		
	TS	CV-10	TS	CV-10		TS	CV-10	TS	CV-10	
1	{1}				0.01	{1}				0.01
	73.17	73.17	73.17	73.17		73.17	73.17	73.17	73.17	
2	{3,8}				0.01	{1,13}				0.01
	90.24	90.24	90.24	90.24		95.12	95.12	95.12	95.12	
3	{1,4,5}				0.02	{1,3,13}				0.02
	95.12	95.12	95.12	95.12		97.56	97.56	97.56	97.56	
4	{1,4,5,20}				0.05	{1,3,13,15}				0.02
	95.12	95.12	95.93	95.12		99.18	97.56	99.18	99.18	
5	{1,5,7,10,20}				0.12	{1,3,13,15,17}				0.05
	92.68	92.68	92.68	92.68		99.18	97.56	99.18	99.18	
6	{7,9,10,12,17,20}				0.28	{1,2,3,13,15,17}				0.09
	86.17	85.36	86.99	85.36		99.18	97.56	99.18	99.18	
7	{7,9,10,11,12,17,20}				0.58	{1,2,3,4,13,15,17}				0.14
	91.86	91.86	89.43	88.61		99.18	97.56	99.18	99.18	
8	{2,4,5,7,9,11,14,20}				0.94	{1,2,3,4,8,13,15,17}				0.16
	90.24	84.55	91.05	91.05		99.18	97.56	99.18	99.18	
9	{3,4,5,7,9,10,16,20,21}				1.44	{1,2,3,4,8,12,13,15,17}				0.16
	94.3	94.3	94.3	94.3		99.18	99.18	99.18	99.18	
10	{2,4,5,6,7,9,10,16,20,21}				1.75	{1,2,3,4,6,8,12,13,15,17}				0.14
	91.05	91.05	92.68	91.86		99.18	99.18	99.18	99.18	
15	{2,4,5,6,7,9,10,11,12,14,16,17,18,20, 21}				0.3	{1,2,3,4,5,6,8,11,12,13,15,16,17,20,21}				0.02
	97.56	91.05	97.56	97.56		99.18	99.18	100	100	
20	{1,2,3,4,5,6,7,8,9,10,11,12,14,15,16, 17,18,19,20,21}				0.01	{1,2,3,4,5,6,7,8,10,11,12,13,14,15,16,17,18,19, 20,21}				0.01
	99.18	99.18	100	100		99.18	99.18	100	100	

La Tabla 5.3 muestra los resultados del *B&B* utilizando la distancia Bhattacharyya y los criterios explicados en la sección 4.3.2 para la BD *Image*, bajo la columna #AT está el tamaño del subconjunto a un lado del tamaño del subconjunto se encuentra los atributos que componen ese subconjunto y abajo del subconjunto de atributos la precisión de los clasificadores J48 y Naïve Bayes, utilizando las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada con 10 particiones.

Tabla 5.3. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD *Image*, de los subconjuntos del *B&B* obtenidos con la distancia Bhattacharyya. TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada con 10 particiones; t(s): tiempo en segundos.

#AT	Criterio 1				t(s)	Criterio 2				t(s)
	J48		Naïve Bayes			J48		Naïve Bayes		
	TS	CV-10	TS	CV-10		TS	CV-10	TS	CV-10	
1	{1}				0.01	{1}				0.01
	73.17	73.17	73.17	73.17		73.17	73.17	73.17	73.17	
2	{1,4}				0.01	{1,4}				0.01
	91.05	91.05	91.05	91.05		91.05	91.05	91.05	91.05	
3	{1,3,4}				0.01	{1,3,4}				0.02
	93.49	93.49	93.49	93.49		93.49	93.49	93.49	93.49	
4	{1,3,4,5}				0.03	{1,3,4,5}				0.02
	97.56	97.56	97.56	97.56		97.56	97.56	97.56	97.56	
5	1,2,3,4,5}				0.1	{1,2,3,4,5}				0.05
	97.56	97.56	97.56	96.74		97.56	97.56	97.56	96.74	
6	{1,2,3,4,5,12}				0.23	{1,2,3,4,5,13}				0.09
	97.56	97.56	97.56	96.74		97.56	97.56	97.56	95.93	
7	{1,2,3,4,5,8,12}				0.44	{1,2,3,4,5,12,13}				0.16
	99.18	96.74	99.18	98.37		100	100	100	99.18	
8	{1,2,3,4,5,8,12,13}				0.69	{1,2,3,4,5,8,12,13}				0.14
	100	100	100	99.18		100	100	100	99.18	
9	{1,2,3,4,5,8,12,13,17}				0.86	{1,2,3,4,5,8,12,13,15}				0.16
	100	100	100	100		99.18	99.18	99.18	98.37	
10	{1,2,3,4,5,8,12,13,15,17}				0.89	{1,2,3,4,5,8,12,13,15,17}				0.14
	99.18	99.18	99.18	99.18		99.18	99.18	99.18	99.18	
15	{1,2,3,4,5,6,7,8,9,11,12,13,15,17,21}				0.11	{1,2,3,4,5,6,7,8,9,11,12,13,15,17,21}				0.11
	99.18	99.18	100	100		99.18	99.18	100	100	
20	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20,21}				0.01	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19,20,21}				0.01
	99.18	99.18	100	100		99.18	99.18	100	100	

La Tabla 5.4 muestra el ranking de la BD *Image Segmentation* de los métodos Información Mutua, Distancia Euclidiana, Distancia Matusita, Kullback-Leibler 1, Kullback-Leibler 2, Entropía de Shannon y Distancia Bhattacharyya. Para cada método de ranking, la fila Rk presenta el ranking de atributos generado con uno de los métodos, y bajo esta fila, el resultado de clasificación con J48 y Naïve Bayes utilizando el número de atributos del ranking indicado en la primera fila, bajo el título Subconjunto; al lado de cada clasificador se tienen las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

En la Tabla 5.4 no aparece el tiempo para obtener los rankings el cual es de 0.4 segundos para cualquier método.

La Tabla 5.5 muestra los resultados del *B&B* utilizando Información Mutua y los criterios explicados en la sección 4.3.2 para la BD *Image Segmentation*, bajo la columna #AT está el tamaño del subconjunto a un lado del tamaño del subconjunto se encuentra los atributos que componen ese subconjunto y abajo del subconjunto de atributos la precisión de los clasificadores J48 y Naïve Bayes utilizando las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

La Tabla 5.6 muestra los resultados del *B&B* utilizando la distancia Bhattacharyya y los criterios explicados en la sección 4.3.2 para la BD *Image Segmentation*, bajo la columna #AT está el tamaño del subconjunto a un lado del tamaño del subconjunto se encuentra los atributos que componen ese subconjunto y abajo del subconjunto de atributos la precisión de los clasificadores J48 y Naïve Bayes utilizando las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

La Tabla 5.7 y 5.8 muestra el ranking de la BD WDBC de los métodos Información Mutua, Distancia Euclidiana, Distancia Matusita, Kullback-Leibler 1, Kullback-Leibler 2, Entropía de Shannon y Bhattacharyya. Para cada método de ranking, la fila Rk presenta el ranking de atributos generado con uno de los métodos, y bajo esta fila, el resultado de clasificación con J48 y Naïve Bayes utilizando el número de atributos del ranking indicado en la primera fila, bajo el título Subconjunto; al lado de cada clasificador se tienen las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

En la Tabla 5.7 y 5.8 no aparece el tiempo para obtener los rankings el cual es de 0.5 segundos para cualquier método.

Tabla 5.4. Resultados de precisión de los clasificadores de Weka J48 y NaïveBayes para la BD *Image Segmentation*, de los rankings obtenidos por la herramienta Elvira. Rk: ranking de atributos; TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

Subconjunto de		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
Rk Mutual informatio		17,	11,	19,	16,	15,	12,	10,	13,	18,	1,	14,	2,	8,	6,	9,	7,	5,	4,	3	
J48	TS	100	100,	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	99.75	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Naïve Bayes	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Rk Euclidian distance		19,	16,	15,	18,	12,	11,	10,	17,	7,	13,	14,	9,	8,	6,	5,	4,	2,	1,	3	
J48	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Naïve Bayes	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Rk Matusita distance		10,	11,	12,	15,	16,	17,	19,	13,	18,	1,	2,	14,	8,	6,	9,	7,	5,	4,	3	
J48	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	99.75	100	99.75	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Naïve Bayes	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Rk Kullback-Leibler 1		17,	16,	19,	15,	12,	11,	10,	13,	18,	1,	14,	2,	8,	6,	9,	7,	5,	4,	3	
J48	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Naïve Bayes	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Rk Kullback-Leibler 2		19,	16,	15,	12,	11,	10,	17,	13,	18,	14,	2,	8,	6,	9,	1,	7,	5,	4,	3	
J48	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Naïve Bayes	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Rk Shannon entropy		14,	2,	1,	8,	6,	18,	9,	7,	4,	5,	13,	19,	17,	16,	15,	12,	11,	10,	3	
J48	TS	84.5	93.25	94	94	94	97	97	97	97	97	99.5	100	100	100	100	100	100	100	100	
	CV-10	84.5	92.25	91	91.75	91.75	96.25	96.25	96.25	96.25	96.25	99.5	100	100	100	100	100	100	100	100	
Naïve Bayes	TS	84.5	92.75	92.25	94.5	94.75	98.5	98.25	98	98.5	98	99.75	100	100	100	100	100	100	100	100	
	CV-10	84.5	92.5	90.75	93	93.25	98.25	97.75	97	97	96.5	99.25	100	100	100	100	100	100	100	100	
Rk Bhattacharyya		19,	12,	17,	16,	15,	11,	10,	13,	18,	1,	2,	14,	8,	6,	9,	7,	5,	4,	3	
J48	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
Naïve Bayes	TS	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	
	CV-10	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	

Tabla 5.5. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD *Image Segmentation*, de los subconjuntos del *B&B* obtenidos con la medida de información mutua.

#AT	Criterio 1				t(s)	Criterio 2				t(s)
	J48		Naïve Bayes			J48		Naïve Bayes		
	TS	CV-10	TS	CV-10		TS	CV-10	TS	CV-10	
1	{17}				0.01	{1}				0.01
	100	100	100	100		60	57.75	60	57.75	
2	{1,17}				0.01	{1,17}				0.01
	100	100	100	100		100	100	100	100	
3	{1,9,17}				0.02	{1,15,17}				0.02
	100	100	100	100		100	99.75	100	100	
4	{1,9,15,17}				0.02	{1,15,16,17}				0.02
	100	99.75	100	100		100	100	100	100	
5	{2,5,7,17,18}				0.06	{2,7,15,16,17}				0.06
	100	100	100	99.5		100	100	100	100	
6	{2,5,7,15,17,18}				0.13	{2,7,13,15,17,18}				0.16
	100	99.75	100	100		100	99.75	100	100	
7	{2,5,7,15,17,18,19}				0.22	{2,7,13,15,16,17,18}				0.16
	100	100	100	100		100	100	100	100	
8	{2,5,7,9,15,17,18,19}				0.31	{2,7,13,14,15,16,17,18}				0.22
	100	100	100	100		100	100	100	100	
9	{2,7,9,12,13,14,15,17,18}				0.38	{2,7,9,13,14,15,16,17,18}				0.27
	100	99.75	100	100		100	100	100	100	
10	{2,7,9,12,13,14,15,16,17,18}				0.41	{1,2,7,9,13,14,15,16,17,18}				0.25
	100	100	100	100		100	100	100	100	
11	{2,6,7,9,12,13,14,15,16,17,18}				0.34	{1,2,3,7,9,13,14,15,16,17,18}				0.2
	100	100	100	100		100	100	100	100	
12	{2,3,6,7,9,12,13,14,15,16,17,18}				0.23	{1,2,3,6,7,9,13,14,15,16,17,18}				0.14
	100	100	100	100		100	100	100	100	
13	{2,3,6,7,8,9,12,13,14,15,16,17,18}				0.14	{1,2,3,6,7,9,13,14,15,16,17,18,19}				0.08
	100	100	100	100		100	100	100	100	
14	{2,3,6,7,8,9,12,13,14,15,16,17,18,19}				0.06	{1,2,3,6,7,8,9,13,14,15,16,17,18,19}				0.05
	100	100	100	100		100	100	100	100	
15	{1,2,3,6,7,8,9,12,13,14,15,16,17,18,19}				0.03	{1,2,3,6,7,8,9,10,13,14,15,16,17,18,19}				0.02
	100	100	100	100		100	100	100	100	
16	{1,2,3,5,6,7,8,9,12,13,14,15,16,17,18,19}				0.02	{1,2,3,5,6,7,8,9,12,13,14,15,16,17,18,19}				0.02
	100	100	100	100		100	100	100	100	
17	{1,2,3,5,6,7,8,9,11,12,13,14,15,16,17,18,19}				0.01	{1,2,3,5,6,7,8,9,10,12,13,14,15,16,17,18,19}				0.02
	100	100	100	100		100	100	100	100	
18	{1,2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19}				0.01	{1,2,3,4,5,6,7,8,9,10,12,13,14,15,16,17,18,19}				0.01
	100	100	100	100		100	100	100	100	
19	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19				0.01	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19				0.01
	100	100	100	100		100	100	100	100	

Tabla 5.6. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD *Image Segmentation*, de los subconjuntos del *B&B* obtenidos con la distancia Bhattacharyya.

#AT	Criterio 1				t(s)	Criterio 2				t(s)
	J48		Naïve Bayes			J48		Naïve Bayes		
	TS	CV-10	TS	CV-10		TS	CV-10	TS	CV-10	
1	{10}				0	{10}				0.01
	100	100	100	100		100	100	100	100	
2	{2,3}				0.02	{2,3}				0.01
	79.75	78.75	79.75	78.75		79.75	78.75	79.75	78.75	
3	{2,3,4}				0.02	{2,3,4}				0.02
	80.5	79.75	80	78.75		80.5	79.75	80	78.75	
4	{2,3,4,5}				0.02	{2,3,4,5}				0.02
	81.25	80.25	81.5	80.25		81.25	80.25	81.5	80.25	
5	{1,2,3,4,5}				0.05	{1,2,3,4,5}				0.02
	86.75	84.5	82.5	80		86.75	84.5	82.5	80	
6	{1,2,3,4,5,16}				0.06	{1,2,3,4,5,16}				0.02
	100	100	100	100		100	100	100	100	
7	{1,2,3,4,5,16,19}				0.09	{1,2,3,4,5,15,16}				0.03
	100	100	100	100		100	100	100	100	
8	{1,2,3,4,5,15,16,19}				0.12	{1,2,3,4,5,10,15,16}				0.02
	100	100	100	100		100	100	100	100	
9	{1,2,3,4,5,10,15,16,19}				0.11	{1,2,3,4,5,10,15,16,19}				0.02
	100	100	100	100		100	100	100	100	
10	{1,2,3,4,5,10,15,16,17,19}				0.11	{1,2,3,4,5,10,11,15,16,19}				0.02
	100	100	100	100		100	100	100	100	
11	{1,2,3,4,5,10,13,15,16,17,19}				0.08	{1,2,3,4,5,6,10,11,15,16,19}				0.02
	100	100	100	100		100	100	100	100	
12	{1,2,3,4,5,7,10,13,15,16,17,19}				0.06	{1,2,3,4,5,6,10,13,15,16,17,19}				0.02
	100	100	100	100		100	100	100	100	
13	{1,2,3,4,5,7,10,11,13,15,16,17,19}				0.03	{1,2,3,4,5,7,8,10,13,15,16,17,19}				0.02
	100	100	100	100		100	100	100	100	
14	{1,2,3,4,5,7,8,10,11,13,15,16,17,19}				0.02	{1,2,3,4,5,6,7,10,11,12,13,15,16,19}				0.02
	100	100	100	100		100	100	100	100	
15	{1,2,3,4,5,7,8,10,11,12,13,15,16,17,19}				0.02	{1,2,3,4,5,7,8,10,11,12,13,15,16,17,19}				0.01
	100	100	100	100		100	100	100	100	
16	{1,2,3,4,5,7,8,9,10,11,12,13,15,16,17,19}				0.01	{1,2,3,4,5,6,7,8,10,11,12,13,15,16,17,19}				0.01
	100	100	100	100		100	100	100	100	
17	{1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,19}				0.01	{1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,19}				0.01
	100	100	100	100		100	100	100	100	
18	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19}				0.01	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,19}				0.01
	100	100	100	100		100	100	100	100	
19	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19				0.01	1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19				0.01
	100	100	100	100		100	100	100	100	

Tabla 5.7. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD WDBC, de los rankings obtenidos por la herramienta Elvira. Rk: ranking de atributos; TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

Subconjunto de	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Rk Mutual informatio	28	23	8	21	24	3	1	7	4	27	11	6	13	26	14
J48	TS	91.74	93.5	94.2	94.2	94.2	94.2	94.2	94.55	95.25	95.25	95.25	95.25	95.25	95.25
	CV-10	91.74	91.04	93.85	93.5	93.5	93.5	93.15	92.09	91.92	92.44	93.5	93.5	93.5	93.5
Naïve Bayes	TS	91.74	94.02	94.38	94.55	93.85	93.67	92.97	94.55	93.5	93.85	94.2	94.55	93.67	94.2
	CV-10	91.74	93.15	94.2	94.02	93.5	93.32	92.97	94.38	93.5	93.32	93.85	94.02	93.5	93.67
Rk Euclidian distance	13	11	14	24	7	4	23	8	21	27	3	18	1	28	26
J48	TS	81.9	82.43	83.3	90.86	93.67	93.85	93.85	94.38	94.55	94.55	94.55	94.55	94.73	95.25
	CV-10	81.55	80.84	81.55	90.51	92.79	92.27	91.56	92.09	93.67	93.85	94.02	94.02	93.85	93.67
Naïve Bayes	TS	81.9	82.43	82.6	88.22	89.81	91.04	92.27	92.27	92.62	93.67	93.85	93.15	92.97	93.67
	CV-10	81.55	82.43	82.6	87.87	89.81	90.69	91.39	91.92	92.62	93.15	93.32	92.97	92.44	93.32
Rk Matusita distance	23	28	8	21	24	3	1	4	7	27	11	6	14	13	26
J48	TS	91.74	93.5	94.2	94.2	94.2	94.2	94.2	94.55	94.55	95.25	95.25	95.25	95.25	95.25
	CV-10	91.56	91.04	93.85	93.5	93.5	93.5	93.15	92.62	91.92	92.44	93.5	93.5	93.5	93.5
Naïve Bayes	TS	91.74	94.02	94.38	94.55	93.85	93.67	92.97	92.09	93.5	93.85	94.2	94.55	94.02	94.2
	CV-10	91.56	93.15	94.2	94.02	93.5	93.32	92.97	91.74	93.5	93.32	93.85	94.02	94.02	93.5
Rk Kullback-Leibler 1	28	23	8	21	24	3	1	7	4	27	6	11	13	26	14
J48	TS	91.74	93.5	94.2	94.2	94.2	94.2	94.2	94.55	95.25	95.25	95.25	95.25	95.25	95.25
	CV-10	91.74	91.04	93.85	93.5	93.5	93.5	93.15	92.09	91.92	92.44	93.15	93.5	93.5	93.5
Naïve Bayes	TS	91.74	94.02	94.38	94.55	93.85	93.67	92.97	94.55	93.5	93.85	94.02	94.55	93.67	94.2
	CV-10	91.74	93.15	94.2	94.02	93.5	93.32	92.97	94.38	93.5	93.32	94.02	94.02	93.5	93.67
Rk Kullback-Leibler 2	23	8	28	21	24	3	1	7	4	27	14	11	6	13	26
J48	TS	91.74	91.74	94.2	94.2	94.2	94.2	94.2	94.55	95.25	95.25	95.25	95.25	95.25	95.25
	CV-10	91.56	91.04	93.85	93.5	93.5	93.5	93.15	92.09	91.92	92.44	93.15	93.5	93.5	93.5
Naïve Bayes	TS	91.74	92.62	94.38	94.55	93.85	93.67	92.97	94.55	93.5	93.85	93.85	94.02	94.02	94.2
	CV-10	91.56	92.62	94.2	94.02	93.5	93.32	92.97	94.38	93.5	93.32	93.67	94.02	94.02	93.5
Rk Shannon entropy	7	27	28	6	21	26	8	3	24	2	23	1	25	22	9
J48	TS	87.7	87.7	91.74	91.74	95.25	95.25	94.73	94.73	95.25	95.25	95.25	95.25	95.25	95.25
	CV-10	87.7	86.99	90.51	89.63	93.32	93.15	93.85	93.67	93.5	94.02	94.02	94.02	94.02	94.02
Naïve Bayes	TS	87.7	88.22	91.56	90.16	92.79	91.39	91.74	92.44	93.67	94.2	95.25	95.25	95.78	95.78
	CV-10	87.52	88.22	91.39	89.63	92.44	91.21	91.39	92.44	93.5	93.85	94.73	94.73	95.43	95.61
Rk Bhattacharyya	23	8	28	21	24	3	1	4	7	27	6	11	26	14	13
J48	TS	91.74	91.74	94.2	94.2	94.2	94.2	94.2	94.55	94.55	95.25	95.25	95.25	95.25	95.25
	CV-10	91.56	91.04	93.85	93.5	93.5	93.5	93.15	92.62	91.92	92.44	93.15	93.5	93.5	93.5
Naïve Bayes	TS	91.74	92.62	94.38	94.55	93.85	93.67	92.97	92.09	93.5	93.85	94.02	94.55	94.2	94.38
	CV-10	91.56	92.62	94.2	94.02	93.5	93.32	92.97	91.74	93.5	93.32	94.02	94.02	94.02	93.67

Tabla 5.8. Resultados de precisión de los clasificadores de Weka J48 y NaïveBayes para la BD WDBC, de los ranking obtenidos por la herramienta Elvira. Rk: Ranking de atributos; TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en boques de 10 (continuación).

Subconjunto de		16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Rk Mutual informator		22	2	18	25	29	5	16	9	30	17	19	20	10	15	12
J48	TS	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.61	95.61	95.61	95.61	96.49	95.43
	CV-10	94.2	94.02	94.2	93.67	93.67	93.67	93.5	93.5	93.15	93.15	93.15	93.15	93.15	93.15	93.67
Naïve Bayes	TS	94.55	94.73	94.73	94.9	94.9	95.08	95.08	94.9	94.55	94.55	94.55	94.73	94.73	94.9	94.55
	CV-10	94.2	94.38	93.85	94.55	94.2	94.73	94.38	94.73	93.85	94.02	94.02	94.02	94.02	94.02	94.2
Rk Euclidian distance		6	20	2	16	22	30	5	29	25	17	9	12	19	10	15
J48	TS	95.25	95.25	95.25	95.08	95.08	95.25	95.25	95.25	95.25	95.61	95.61	95.61	95.08	95.43	95.43
	CV-10	93.67	93.85	94.2	93.32	93.5	93.15	93.15	92.97	93.15	93.15	93.15	93.15	92.79	92.62	93.67
Naïve Bayes	TS	93.85	94.02	94.55	93.85	94.2	94.2	94.02	94.55	95.08	94.38	94.55	94.38	94.38	94.55	94.55
	CV-10	93.32	93.67	94.2	93.85	94.02	94.02	93.67	94.02	94.2	93.85	94.02	93.85	93.85	94.02	94.2
Rk Matusita distance		22	2	18	25	29	5	16	9	30	17	19	20	10	15	12
J48	TS	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.61	95.61	95.61	95.61	96.49	95.43
	CV-10	94.2	94.02	94.2	93.67	93.67	93.67	93.5	93.5	93.15	93.15	93.15	93.15	93.15	93.15	93.67
Naïve Bayes	TS	94.55	94.73	94.73	94.9	94.9	95.08	95.08	94.9	94.55	94.55	94.55	94.73	94.73	94.9	94.55
	CV-10	94.2	94.38	93.85	94.55	94.2	94.73	94.38	94.73	93.85	94.02	94.02	94.02	94.02	94.02	94.2
Rk Kullback-Leibler 1		22	2	18	25	29	5	16	9	30	17	19	20	10	15	12
J48	TS	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.61	95.61	95.61	95.61	96.49	95.43
	CV-10	94.2	94.02	94.2	93.67	93.67	93.67	93.5	93.5	93.15	93.15	93.15	93.15	93.15	93.15	93.67
Naïve Bayes	TS	94.55	94.73	94.73	94.9	94.9	95.08	95.08	94.9	94.55	94.55	94.55	94.73	94.73	94.9	94.55
	CV-10	94.2	94.38	93.85	94.55	94.2	94.73	94.38	94.73	93.85	94.02	94.02	94.02	94.02	94.02	94.2
Rk Kullback-Leibler 2		2	22	18	25	29	5	16	9	30	17	19	20	10	12	15
J48	TS	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.61	95.61	95.61	95.61	95.43	95.43
	CV-10	94.02	94.02	94.2	93.67	93.67	93.67	93.5	93.5	93.15	93.15	93.15	93.15	93.15	92.62	93.67
Naïve Bayes	TS	94.02	94.73	94.73	94.9	94.9	95.08	95.08	94.9	94.55	94.55	94.55	94.73	94.73	94.55	94.55
	CV-10	93.67	94.38	93.85	94.55	94.2	94.73	94.38	94.73	93.85	94.02	94.02	94.02	94.02	94.02	94.2
Rk Shannon entropy		10	29	4	18	16	11	5	30	13	14	19	12	15	20	17
J48	TS	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	96.31	95.43
	CV-10	93.5	93.5	93.5	93.5	93.5	94.02	94.02	93.67	93.85	93.15	93.15	93.15	93.67	93.67	93.67
Naïve Bayes	TS	95.43	95.96	95.61	95.25	95.25	94.9	94.9	94.73	94.9	94.73	94.73	95.08	95.08	95.08	94.55
	CV-10	95.25	95.61	95.08	95.08	94.55	94.55	94.73	94.38	94.2	94.38	94.38	94.38	94.2	94.02	94.2
Rk Bhattacharyya		22	18	2	25	29	5	16	9	30	17	19	20	10	15	12
J48	TS	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.25	95.61	95.61	95.61	95.61	96.49	95.43
	CV-10	94.2	94.38	94.2	93.67	93.67	93.67	93.5	93.5	93.15	93.15	93.15	93.15	93.15	93.15	93.67
Naïve Bayes	TS	94.55	94.02	94.73	94.9	94.9	95.08	95.08	94.9	94.55	94.55	94.55	94.73	94.73	94.9	94.55
	CV-10	94.2	93.85	93.85	94.55	94.2	94.73	94.38	94.73	93.85	94.02	94.02	94.02	94.02	94.02	94.2

Tabla 5.9. Resultados de precisión de los clasificadores de Weka J48 y NaïveBayes para la BD WDBC, de los subconjuntos del B&B obtenidos con la medida de información mutua.

#AT	Criterio 1				t(s)	Criterio 2				t(s)
	J48		Naïve Bayes			J48		Naïve Bayes		
	TS	CV-10	TS	CV-10		TS	CV-10	TS	CV-10	
1	{24}				0.02	{24}				0.02
	88.57	88.4	88.57	88.4		88.57	88.4	88.57	88.4	
2	{17,24}				0.11	{8,24}				0.06
	90.33	90.15	90.15	89.63		92.44	91.91	92.61	99.44	
3	{17,22,24}				0.83	{14,24,28}				0.52
	91.03	88.57	90.86	89.8		94.37	94.02	94.55	94.2	
4	{5,17,22,24}				3.81	{8,14,24,28}				2.02
	92.61	91.21	94.02	93.84		94.37	93.67	94.37	93.84	
5	{5,17,22,24,29}				15.3	{8,14,24,27,28}				4.95
	94.9	89.63	95.07	94.2		94.55	92.44	94.02	94.02	
6	{5,15,17,22,24,29}				54.1	{8,14,23,24,27,28}				11.6
	93.49	90.33	95.07	94.02		94.55	91.91	94.72	94.72	
7	{5,14,15,17,19,22,29}				15.5	{8,13,14,23,24,27,28}				23.7
	88.92	87.17	89.63	87.87		94.9	91.56	94.37	94.2	
8	{5,12,14,15,17,22,29,30}				44.6	{7,8,13,14,23,24,27,28}				52.3
	88.4	84.88	87.52	85.58		94.9	92.09	94.55	94.2	
9	{2,5,12,14,15,17,19,29,30}				113	{3,7,8,13,14,23,24,27,28}				29.3
	91.73	88.22	89.27	86.81		94.9	92.09	94.2	94.2	
10	{1,2,5,12,14,15,17,19,29,30}				223	{3,7,8,11,13,14,23,24,27,28}				46.1
	90.15	88.22	94.9	94.37		95.07	93.32	94.02	93.67	
15	{1,2,5,9,12,14,15,17,19,20,22,25,28,29,30}				1285	{1,3,4,6,7,8,11,13,14,21,23,24,26,27,28}				42.6
	93.84	91.91	94.72	93.67		95.25	93.49	94.02	93.67	
20	{2,5,9,10,11,12,14,15,17,18,19,20,22,24,25,26,28,29,30}				287	{1,2,3,4,6,7,8,11,13,14,18,21,22,23,24,25,26,27,28,29}				2.22
	94.37	93.14	95.25	94.37		95.25	93.67	94.9	94.2	
25	{1 2 4 5 8 9 10 11 12 13 14 15 16 17 18 19 20 22 24 25 26 27 28 29 30}				1.59	{1,2,3,4,5,6,7,8,9,11,13,14,16,17,18,21, 22,23,24,25,26,27,28,29,30}				0.03
	95.95	93.49	94.72	93.84		95.6	93.14	94.55	94.02	
26	{1 2 4 5 6 8 9 10 11 12 13 14 15 16 17 18 19 20 22 24 25 26 27 28 29 30}				0.31	{1,2,3,4,5,6,7,8,9,11,13,14,16,17,18,19, 21,22,23,24,25,26,27,28,29,30}				0.02
	95.95	93.49	94.37	93.14		95.6	93.14	94.55	94.02	
27	{1 2 4 5 6 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 24 25 26 27 28 29 30}				0.05	{1,2,3,4,5,6,7,8,9,11,13,14,16,17,18,19, 20,21,22,23,24,25,26,27,28,29,30}				0.02
	96.13	92.26	94.72	93.84		95.6	93.14	94.72	94.02	
28	{1,2,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 18,19,20,21,22,24,25,26,27,28,29,30}				0.01	{1,2,3,4,5,6,7,8,9,11,13,14,15,16,17,18, 19,20,21,22,23,24,25,26,27,28,29,30}				0.01
	95.43	94.2	94.2	93.67		96.48	93.32	94.55	94.02	
29	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 18,19,20,21,22,24,25,26,27,28,29,30}				0.01	{1,2,3,4,5,6,7,8,9,10,11,13,14,15,16,17, 18,19,20,21,22,23,24,25,26,27,28,29,30}				0.01
	95.43	94.2	94.9	94.55		96.48	93.14	94.9	94.02	
30	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 18,19,20,21,22,23,24,25,26,27,28,29,30}				0.01	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17, 18,19,20,21,22,23,24,25,26,27,28,29,30}				0.01
	95.43	93.67	94.55	94.2		95.43	93.67	94.55	94.2	

Tabla 5.10. Resultados de precisión de los clasificadores de Weka J48 y Naïve Bayes para la BD WDBC, de los subconjuntos del B&B obtenidos con la distancia Bhattacharyya.

#AT	Criterio 1				t(s)	Criterio 2				t(s)
	J48		Naïve Bayes			J48		Naïve Bayes		
	TS	CV-10	TS	CV-10		TS	CV-10	TS	CV-10	
1	{23}				0.02	{23}				0.02
	91.73	91.56	91.73	91.56		91.73	91.56	91.73	91.56	
2	{23,30}				0.02	{1,28}				0.02
	93.49	92.09	92.44	92.26		94.2	91.91	93.32	93.32	
3	{4,17,28}				0.06	{1,14,28}				0.06
	91.73	91.56	94.9	92.97		94.2	91.91	92.79	92.44	
4	{3,14,17,28}				0.3	{1,3,17,28}				0.2
	91.73	91.03	93.84	92.97		92.79	91.03	93.49	93.49	
5	{4,9,14,17,28}				1.2	{1,3,14,17,28}				0.66
	91.73	91.56	94.2	93.67		92.79	91.03	92.61	92.09	
6	{1,3,9,14,17,28}				4.06	{1,2,3,14,17,28}				1.81
	91.73	91.03	93.14	92.26		92.79	90.86	93.84	93.32	
7	{1,3,9,13,14,17,28}				11.4	{1,2,3,8,14,17,28}				4.18
	92.79	90.86	92.26	92.26		95.07	88.22	94.72	94.02	
8	{1,2,3,9,13,14,17,28}				27.2	{1,3,4,9,13,14,17,28}				7.82
	92.79	90.5	94.2	94.02		95.6	91.73	91.38	91.38	
9	{1,2,3,6,9,13,14,17,28}				57.1	{1,2,3,4,6,9,14,17,28}				13.1
	92.79	90.5	94.2	93.49		93.84	91.56	92.97	92.44	
10	{1,2,3,6,9,13,14,16,17,28}				108	{1,2,3,4,9,13,14,16,17,28}				16.8
	92.79	92.05	93.84	92.97		93.84	91.56	93.14	93.14	
15	{1,3,4,5,6,8,9,10,12,13,14,16,17,22,28}				292	{1,2,3,4,5,6,7,8,9,12,13,14,16,17,28}				17.6
	92.79	90.68	94.2	93.84		95.09	88.92	93.84	93.32	
20	{1,2,3,4,5,6,8,9,10,12,13,14,16,17,19,21,22,24,25,28}				37.1	{1,2,3,4,5,6,7,8,9,10,12,13,14,16,17,19,21,22,24,28}				0.78
	95.78	90.68	94.9	94.72		95.6	93.14	94.55	94.72	
25	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,28}				0.24	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,28}				0.02
	96.66	92.61	94.9	94.37		95.07	92.44	94.9	94.37	
26	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,26,28}				0.06	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,26,28}				0.01
	95.07	92.44	94.55	94.2		95.07	92.26	94.55	94.2	
27	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,26,28}				0.02	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,16,17,18,19,20,21,22,23,24,25,26,27,28}				0.01
	95.07	92.26	94.55	94.2		95.07	92.79	94.72	94.55	
28	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28}				0.01	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28}				0.01
	95.43	93.49	94.72	94.37		96.48	93.14	94.72	94.37	
29	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30}				0.01	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,30}				0.01
	96.48	93.14	94.37	93.67		95.43	93.67	94.37	93.67	
30	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30}				0.01	{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30}				0.01
	95.43	93.67	94.55	94.2		95.43	93.67	94.55	94.2	

La Tabla 5.9 muestra los resultados del *B&B* utilizando información mutua y los criterios explicados en la sección 4.3 para la BD WDBC, bajo la columna #AT están el tamaño del subconjunto a un lado del tamaño del subconjunto se encuentra los atributos que componen ese subconjunto y abajo del subconjunto de atributos la precisión de los clasificadores J48 y Naïve Bayes utilizando las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

La Tabla 5.10 muestra los resultados del *B&B* utilizando la distancia Bhattacharyya y los criterios explicados en la sección 4.3.2 para la BD WDBC, bajo la columna #AT están el tamaño del subconjunto a un lado del tamaño del subconjunto se encuentra los atributos que componen ese subconjunto y abajo del subconjunto de atributos la precisión de los clasificadores J48 y Naïve Bayes utilizando las letras TS y CV-10 que representan: TS: uso de todos los datos para entrenar y evaluar el clasificador; CV-10: validación cruzada en bloques de 10.

Para poder tener un análisis comparativo de las diferentes tablas se grafican y visualmente se observa su comportamiento.

En la Figura 5.5 se observa la gráfica generada para la BD *Image* con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador J48 con todos los datos para entrenar y evaluar, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.6 se observa la gráfica generada para la BD *Image* con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador J48 con validación cruzada en bloques de 10, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.7 se observa la gráfica generada para la BD *Image* con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador Naïve Bayes con todos los datos para entrenar y evaluar, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.8 se observa la gráfica generada para la BD *Image* con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador J48 con validación cruzada en bloques de 10, en el eje de las abscisas se tiene el número de

atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.9 se observa la gráfica generada para la BD *Segmentation* con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador J48 con todos los datos para entrenar y evaluar, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.10 se observa la gráfica generada para la BD *Segmentation* con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador J48 con validación cruzada en bloques de 10, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.11 se observa la gráfica generada para la BD *Segmentation* con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador Naïve Bayes con todos los datos para entrenar y evaluar, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.12 se observa la gráfica generada para la BD *Segmentation* con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador Naïve Bayes con validación cruzada en bloques de 10, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.13 se observa la gráfica generada para la BD WDBC con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador J48 con todos los datos para entrenar y evaluar, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.14 se observa la gráfica generada para la BD WDBC con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador J48 con validación cruzada en bloques de 10, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.15 se observa la gráfica generada para la BD WDBC con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador Naïve Bayes con todos los datos para entrenar y evaluar, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

En la Figura 5.16 se observa la gráfica generada para la BD WDBC con los rankings obtenidos con la herramienta Elvira y los subconjuntos obtenidos por el *B&B* con sus diferentes medidas y criterios, utilizando el clasificador Naïve Bayes con validación cruzada en bloques de 10, en el eje de las abscisas se tiene el número de atributos utilizados en la clasificación y en el eje de las ordenadas se tiene la precisión de clasificación.

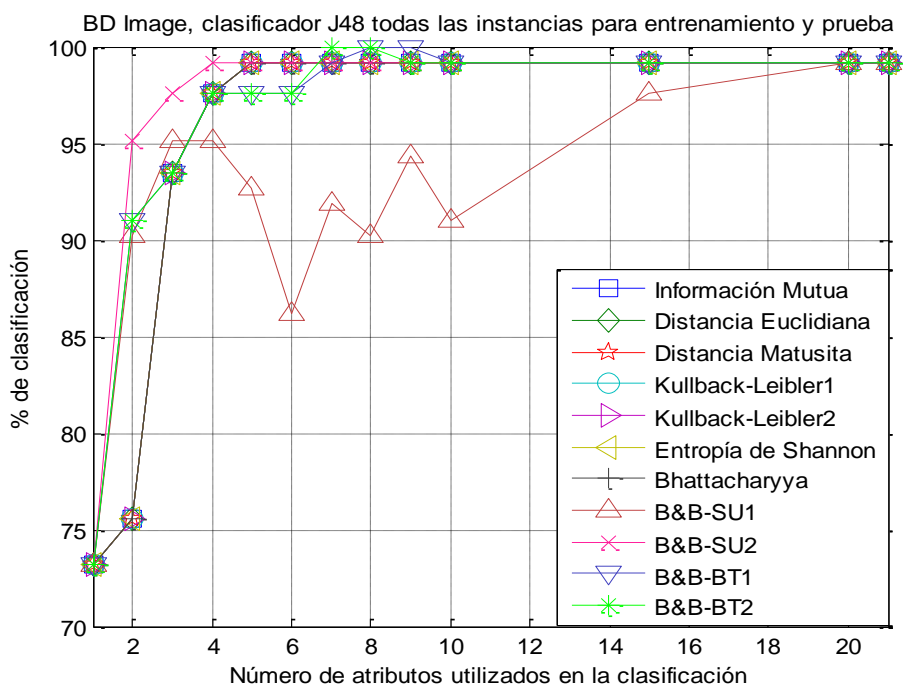


Figura 5.5. Clasificador J48 aplicado a la BD *Image*.

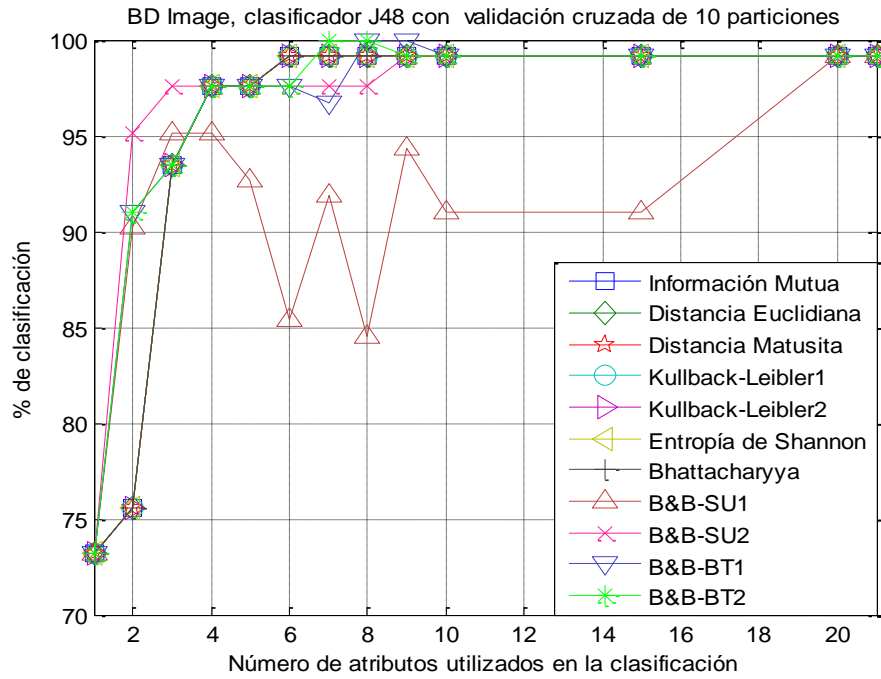


Figura 5.6. Clasificador J48 con validación cruzada aplicado a la BD Image.

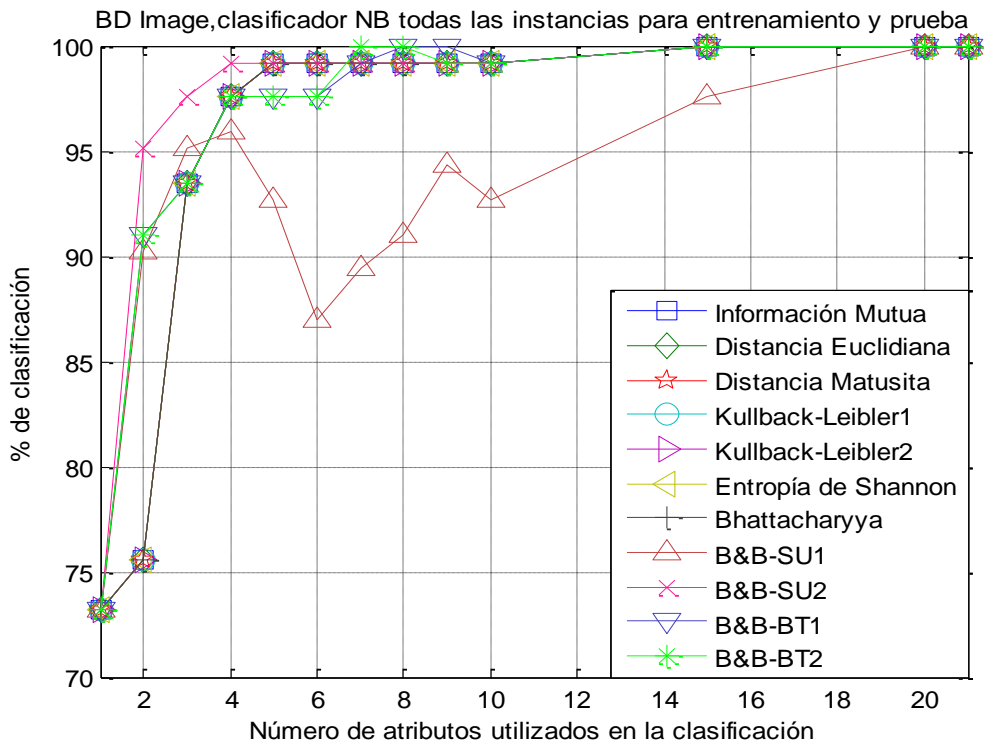


Figura 5.7. Clasificador NB aplicado a la BD Image.

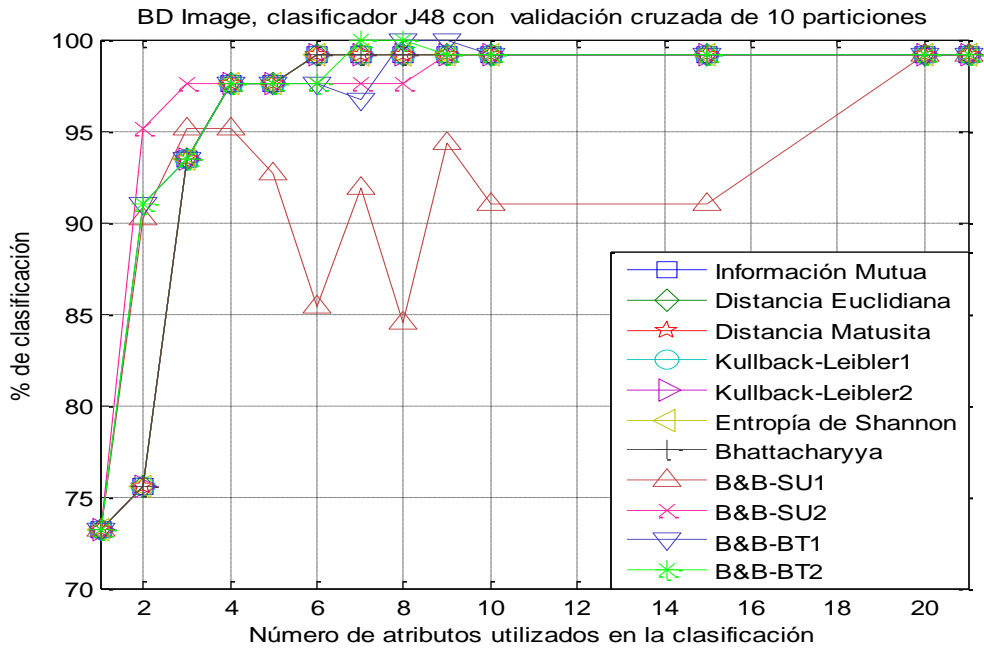


Figura 5.8. Clasificador NB con validación cruzada aplicado a la BD *Image*.

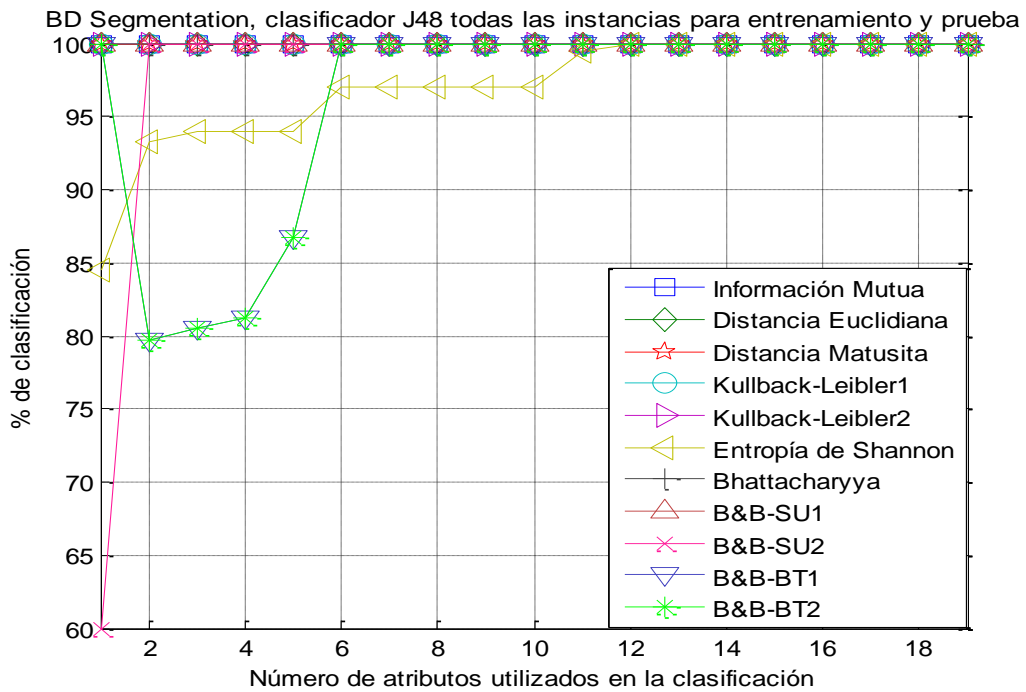


Figura 5.9. Clasificador J48 aplicado a la BD *Segmentation*.

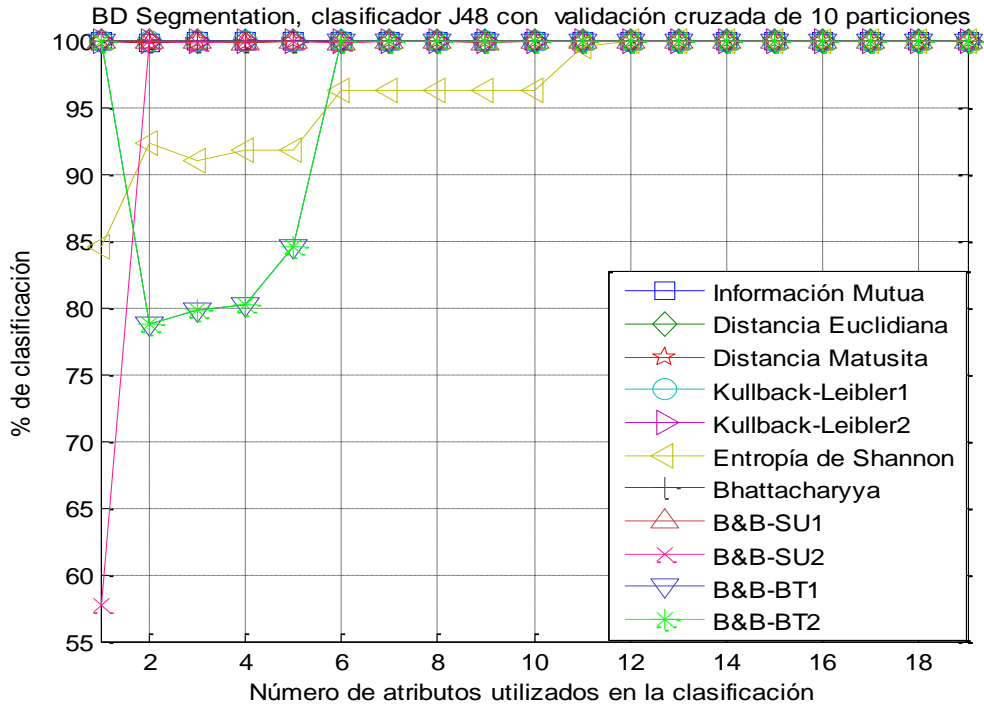


Figura 5.10. Clasificador J48 con validación cruzada aplicado a la BD Segmentation

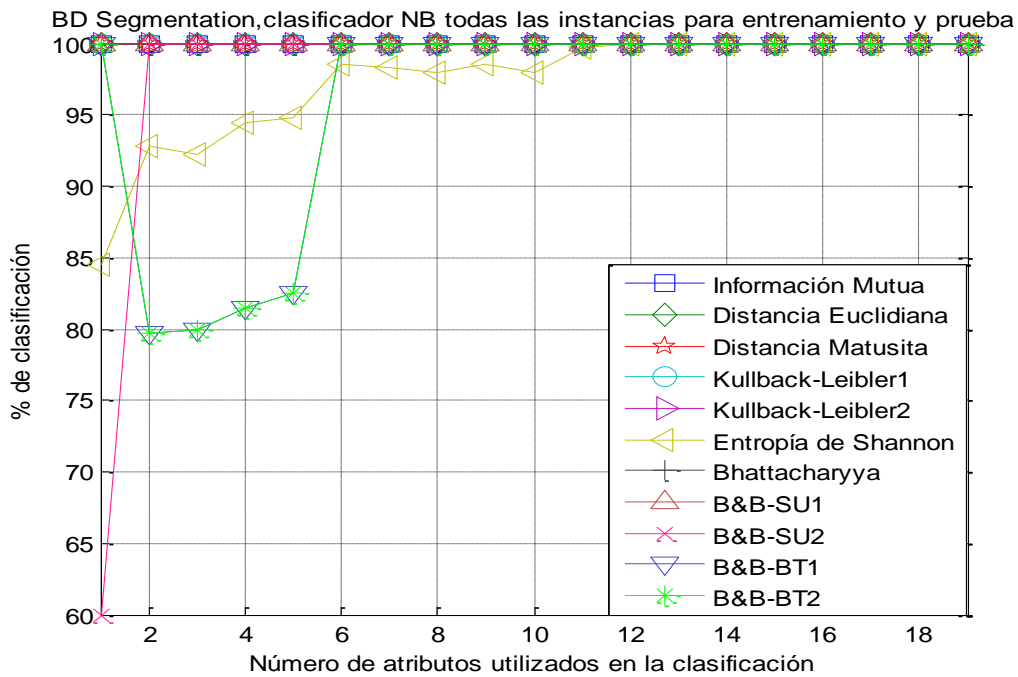


Figura 5.11. Clasificador NB con validación cruzada aplicado a la BD Segmentation.

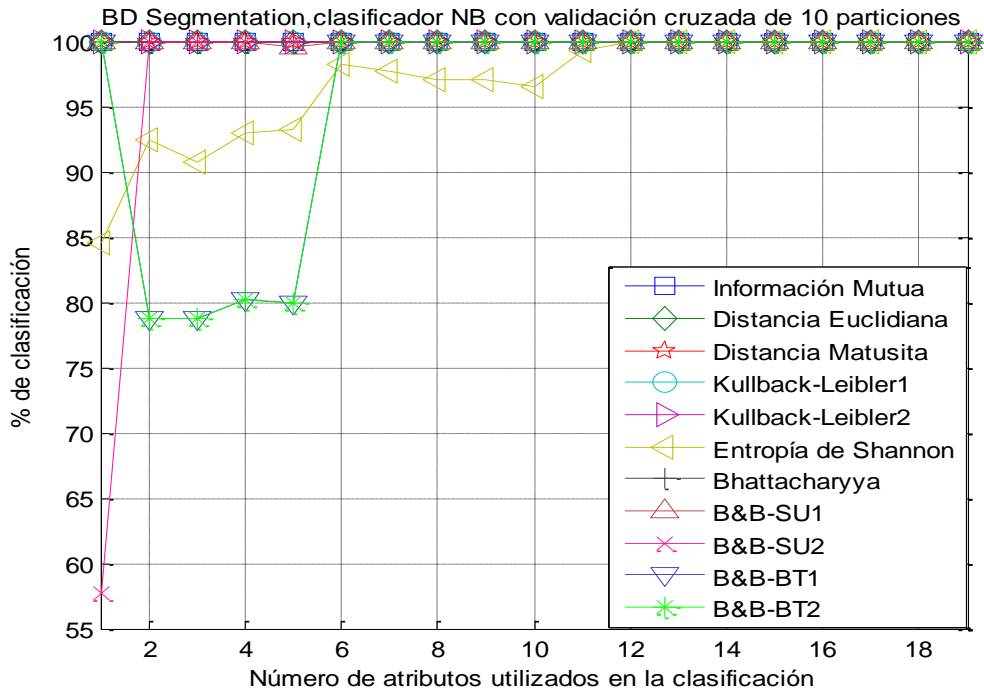


Figura 5.12. Clasificador NB con validación cruzada aplicado a la BD Segmentation.

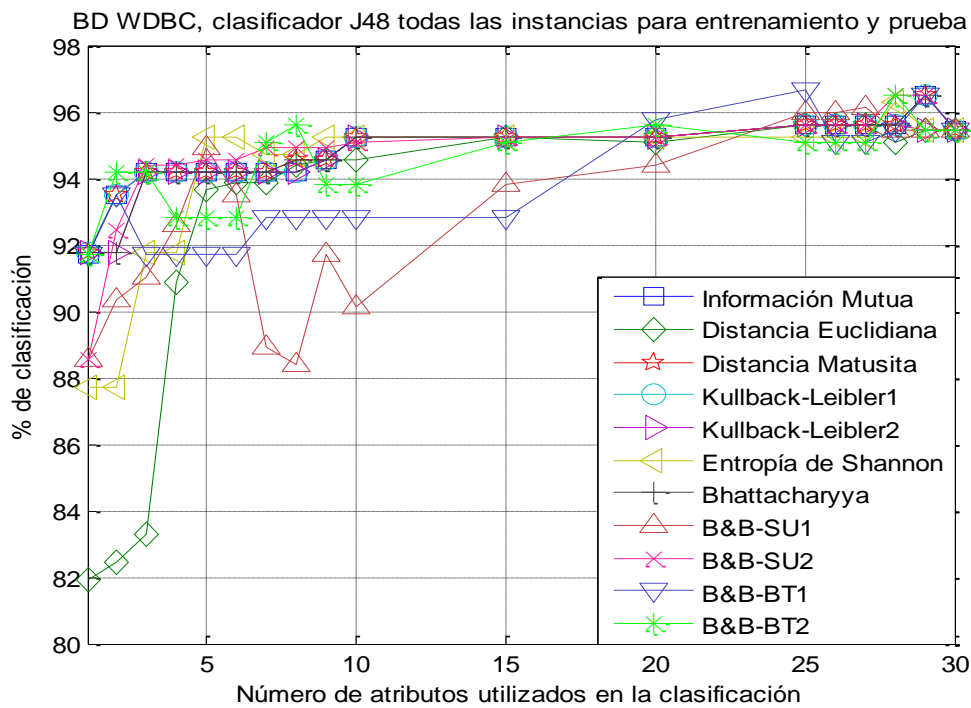


Figura 5.13. Clasificador J48 aplicado a la BD WDBC.

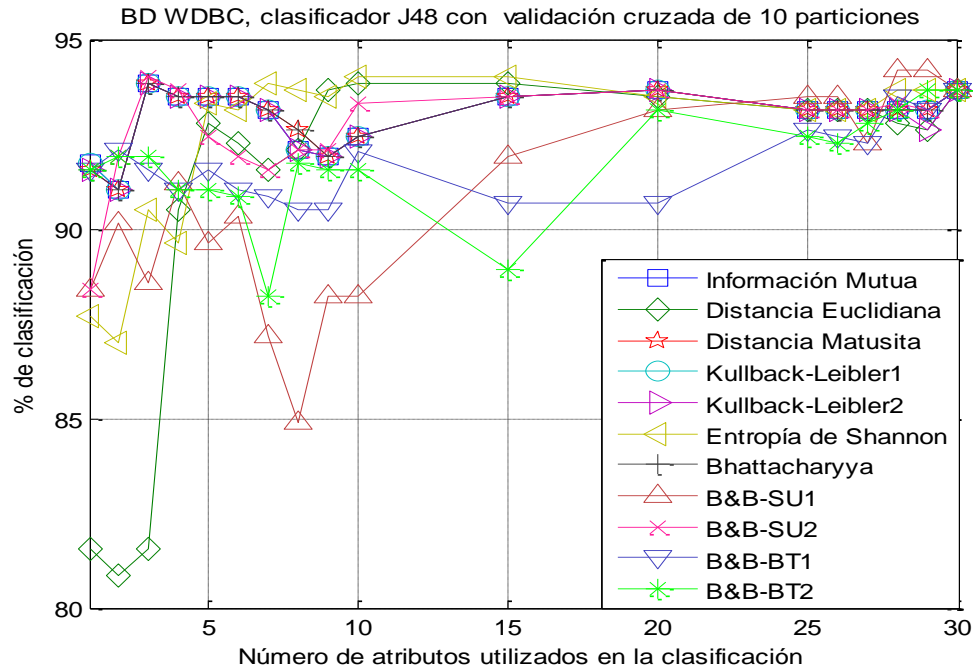


Figura 5.14. Clasificador J48 con validación cruzada aplicado a la BD WDBC.

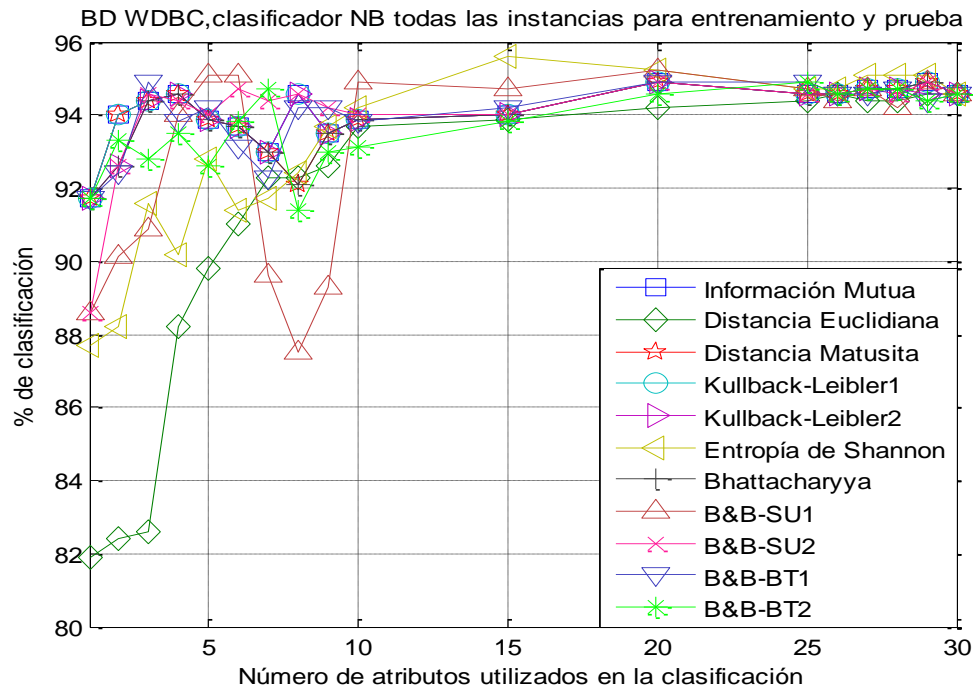


Figura 5.15. Clasificador NB aplicado a la BD WDBC.

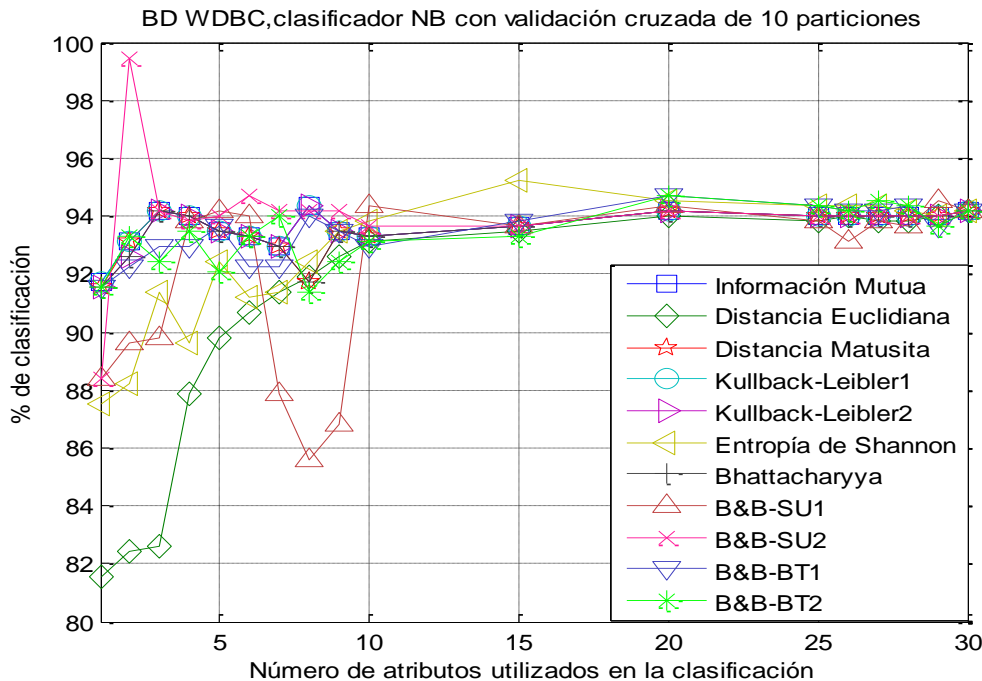


Figura 5.16. Clasificador NB con validación cruzada aplicado a la BD WDBC.

COMENTARIOS: No existen tendencias claras de que algún método SA obtenga mejores resultados de clasificación, por lo cual no se puede decidir que ranking o medida de evaluación del *B&B* es mejor para clasificar, el comportamiento de la precisión del clasificador es diferente para cada BD, para la BD *Segmentation* el método de SA basado en la entropía de Shannon es de los que tiene más baja precisión del clasificador no así con la BD *WDBC* en la cual es uno de los mejores, se nota más aleatoriedad de precisión del clasificador con el *B&B* específicamente con la medida de evaluación de información mutua y el criterio de evaluación 1.

5.3. Pruebas de eficiencia

Las pruebas de eficiencia no se pueden hacer bajo las mismas condiciones puesto que no se comparó con algún método de selección de atributos que al igual que el *Branch & Bound* tenga como requisito una cardinalidad fija para la salida de los atributos seleccionados, por lo que la comparación de eficiencia temporal no es imparcial.

En la sección anterior los métodos *filter* de la herramienta *Elvira* son muy rápidos en comparación con el *B&B* y obtienen resultados mejores o similares en calidad de la solución.

En la Tabla 5.10 se tiene una comparación de eficiencia y calidad de la herramienta *See5* vs *B&B*; *B&B_SU1* corresponde a la medida de información mutua criterio de evaluación 1, *B&B_SU2* corresponde a la información mutua

criterio de evaluación 2, *B&B_BT1* corresponde a la distancia Bhattacharyya criterio de evaluación 1, *B&B_BT2* corresponde a la distancia Bhattacharyya criterio de evaluación 2; bajo la columna Herramienta está el nombre de la herramienta utilizada, después se tiene el tiempo en segundos de procesamiento para encontrar el subconjunto de atributos, bajo subconjunto seleccionado está el subconjunto de atributos seleccionado por la herramienta, utilizando las letras J48 TS: uso del clasificador J48 con todos los datos para entrenar y evaluar el clasificador; J48 CV-10: uso del clasificador J48 con validación cruzada en bloques de 10 ; NB TS: uso del clasificador Naïve Bayes con todos los datos para entrenar y evaluar el clasificador; NB CV-10 uso del clasificador Naïve Bayes con validación cruzada en bloques de 10

Tabla 5.10. Resultados eficiencia y calidad See5 vs B&B

BD Image 21 atributos See5 vs B&B						
Herramienta	t(s)	subconjunto seleccionado	J48 TS	J48 CV-10	NB TS	NB CV-10
See5	0	{1,13,2,15}	99.18	97.56	99.18	99.18
B&B_SU1	0.048	{1,4,5,20}	95.12	95.12	95.93	95.12
B&B_SU2	0.016	{1,3,13,15}	99.18	97.56	99.18	99.18
B&B_BT1	0.031	{1,3,4,5}	97.56	97.56	97.56	97.56
B&B_BT2	0.016	{1,3,4,5}	97.56	97.56	97.56	97.56
BD Segmentation 19 atributos See5 vs B&B						
Herramienta	t(s)	subconjunto seleccionado	J48 TS	J48 CV-10	NB TS	NB CV-10
See5	0	{19}	100	100	100	100
B&B_SU1	0.01	{17}	100	100	100	100
B&B_SU2	0.01	{1}	60	57.75	60	57.75
B&B_BT1	0.01	{10}	100	100	100	100
B&B_BT2	0.01	{10}	100	100	100	100
BD WDBC 30 atributos See5 vs B&B						
Herramienta	t(s)	subconjunto seleccionado	J48 TS	J48 CV-10	NB TS	NB CV-10
See5	0.1	{23,28,27,2,22,24,3}	94.37	93.32	95.95	95.6
B&B_SU1	15.537	{5,14,15,17,19,22,29}	88.92	87.17	89.63	87.87
B&B_SU2	23.727	{8,13,14,23,24,27,28}	94.9	91.56	94.37	94.2
B&B_BT1	11.406	{1,3,9,13,14,17,28}	92.79	90.86	92.26	92.26
B&B_BT2	4.181	{1,2,3,8,14,17,28}	95.07	88.22	94.72	94.02

En la Tabla 5.11 se tiene una comparación de eficiencia y calidad de la herramienta Reduclas vs *B&B*, *B&B_SU1* corresponde a la medida de información mutua criterio de evaluación 1, *B&B_SU2* corresponde a la medida de información mutua criterio de evaluación 2, *B&B_BT1* corresponde a la medida de la distancia Bhattacharyya criterio de evaluación 1, *B&B_BT2* corresponde a la medida de la

distancia Bhattacharyya criterio de evaluación 2; bajo la columna Herramienta está el nombre de la herramienta utilizada, después se tiene el tiempo de procesamiento para encontrar el subconjunto, bajo subconjunto seleccionado está el subconjunto de atributos seleccionado por la herramienta, utilizando las letras J48 TS: Uso del clasificador J48 con todos los datos para entrenar y evaluar el clasificador; J48 CV-10: Uso del clasificador J48 con validación cruzada en bloques de 10; ; NB TS: uso del clasificador Naïve Bayes con todos los datos para entrenar y evaluar el clasificador; NB CV-10 uso del clasificador Naïve Bayes con validación cruzada en bloques de 10

En la Tabla 5.11 se observa que la herramienta Reduclas obtiene mejores resultados de calidad del clasificador y encuentra todos los subconjuntos de atributos con los cuales se puede hacer una clasificación sin que las instancias se confundan de clase, la realización de la Tabla 5.11, no fue imparcial, ya que para ejecutar la herramienta Reduclas se ejecutó el programa en una máquina virtual con un núcleo de Intel core i5 a 2.67Ghz y 2G de RAM

Tabla 5.11. Resultados eficiencia y calidad Reduclas vs B&B

BD Image 21 atributos Reduclas vs B&B						
Herramienta	t(s)	subconjunto seleccionado	J48 TS	J48 CV-10	NB TS	NB CV-10
Reduclas	0.6	{1,3,4,12,13}	100	100	100	100
Reduclas	0.6	{1,2,4,12,13}	100	100	100	100
B&B_SU1	0.124	{1,5,7,10,20}	92.68	92.68	92.68	92.68
B&B_SU2	0.047	{1,3,13,15,17}	99.18	97.56	99.18	99.18
B&B_BT1	0.095	{1,2,3,4,5}	97.56	97.56	97.56	96.74
B&B_BT2	0.095	{1,2,3,4,5}	97.56	97.56	97.56	96.74

BD Segmentation 19 atributos Reduclas vs B&B						
Herramienta	t(s)	subconjunto seleccionado	J48 TS	J48 CV-10	NB TS	NB CV-10
Reduclas	0.5	{10}	100	100	100	100
Reduclas	0.5	{11}	100	99.75	100	99.75
Reduclas	0.5	{12}	100	100	100	100
Reduclas	0.5	{15}	100	99.75	100	99.75
Reduclas	0.5	{16}	100	100	100	100
Reduclas	0.5	{17}	100	100	100	100
Reduclas	0.5	{19}	100	100	100	100
B&B_SU1	0.01	{17}	100	100	100	100
B&B_SU2	0.01	{1}	60	57.75	60	57.75
B&B_BT1	0.01	{10}	100	100	100	100
B&B_BT2	0.01	{10}	100	100	100	100

Los métodos de SA de las herramientas Elvira, *B&B* y See5, producen diferentes subconjuntos de atributos que producen diferentes comportamientos con los clasificadores J48 y Naïve Bayes, solo la herramienta Reduclas obtuvo un buen comportamiento para las diferentes bases de datos, las demás herramientas analizadas no existen tendencias claras para las bases de datos, algunos son buenos con cierta base de datos pero con otra producen diferentes resultados.

El algoritmo *B&B* obtuvo resultados similares en las pruebas de calidad en comparación con los métodos *filter*, en eficiencia son mejor los métodos *filter*; los métodos *filter* tienen mejores resultados en eficiencia que el Sistema Reduclas (Teoría de Testores) con la disminución de calidad

La Teoría de Testores, a pesar de ser una buena técnica de selección de atributos, en comparación con otras técnicas para selección de atributos, es de las más costosas computacionalmente hablando ya que tiene una complejidad computacional de tipo exponencial [Ochoa, 2005].

La herramienta See5 es eficiente y tiene buena calidad en las pruebas hechas escogió subconjuntos muy parecidos a la herramienta Reduclas que fue la más alta en calidad pero hizo el cálculo del subconjunto más rápido.

5.4. Discusión

La evaluación de métodos de SA debería de hacerse en igualdad de condiciones, es muy diferente si se comparan propiedades de un método en concreto, en el caso de comparar la optimalidad del subconjunto de atributos, la optimalidad es definida de diferentes maneras, una es por la medida de evaluación usada y el subconjunto óptimo de una medida puede no ser el óptimo con una medida de evaluación diferente; otra forma de evaluar la calidad de un algoritmo de SA es generando BD sintéticas con funciones conocidas, al conocer la función que predice la clase se puede comparar su calidad de solución y una medida indirecta para evaluar la optimalidad es usar el rendimiento de un clasificador, incluso el clasificador juega un papel muy importante en su elección ya que los clasificadores escogidos no toman en cuenta interdependencia, esto hace que si se quiere evaluar una base de datos que contenga como solución óptima un conjunto de atributos con dependencia como por ejemplo una BD de tipo XOR el algoritmo inductor será incapaz de construir un modelo clasificador y dará resultados erróneos, los clasificadores probabilísticos por lo general asumen independencia de variables, por lo que se puede tener conclusiones erróneas sobre el método de SA evaluado, con esto se expone que la evaluación de calidad de un método de SA depende de diferentes factores no solo el rendimiento del clasificador. Si se quiere hacer SA para representación óptima de datos no es necesario utilizar un algoritmo inductor para evaluar el subconjunto de datos, existen métodos como la Teoría de Testores que encuentran los subconjuntos óptimos que representan a toda la base de datos sin necesidad de un algoritmo inductor.

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones del trabajo de tesis, los objetivos y alcances conseguidos, las aportaciones y finalmente el trabajo futuro que podría continuar de esta investigación.

6.1. Objetivos logrados

En la Tabla 6.1 se presentan los objetivos propuestos y su comentario de cómo se llevaron a cabo en este trabajo de investigación.

Tabla 6.1. Objetivos en la investigación.

Objetivos propuestos	Comentario
Estudiar e Implementar el algoritmo <i>B&B</i> propuesto por [Narendra, 1977].	Se implementó el algoritmo de <i>B&B</i> en lenguaje de programación C.
Definir una heurística para seleccionar atributos relevantes de bases de datos de aprendizaje supervisado.	Se definieron 2 criterios de evaluación para las medidas de información mutua y distancia Bhattacharyya.
Implementar el algoritmo <i>B&B</i> con la heurística definida.	Se implementaron 2 heurísticas para el <i>B&B</i> .
Comparar el subconjunto de atributos obtenido por el algoritmo <i>B&B</i> con la heurística definida con diferentes herramientas de selección de atributos.	Se comparó eficiencia y calidad de solución contra los sistemas See5, Elvira, Reduclas.

En la Tabla 6.2 se presentan los alcances y limitaciones planteados, en este trabajo de investigación.

Tabla 6.2. Alcances y limitaciones.

Alcances y limitaciones	Comentario
Estudiar el área de selección de atributos y el <i>B&B</i> original.	Se realizó un estudio del estado del arte en el cual se revisaron más de 50 artículos del área de selección de atributos y del algoritmo del <i>B&B</i> .
Revisar los problemas típicos de Visión Artificial relacionados con selección de atributos, se van a revisar las tesis del CENIDET para ver los diferentes descriptores usados en ellas.	Se encontraron 7 tesis en el CENIDET que tratan el tema de SA, se programaron en lenguaje C más de 30 descriptores usados en esas tesis, el programa se incluye en el anexo C.
Las bases de datos podrán tener atributos discretos con clase discreta, y atributos continuos con clase discreta.	Para bases de datos con atributos continuos se realizó un programa de discretización en lenguaje C, incluido en el anexo C.
Implementar el <i>B&B</i> propuesto por [Narendra, 1977].	Se implementó en lenguaje C el algoritmo del <i>B&B</i> propuesto por [Narendra, 1977] el cual se incluye en el anexo C.
Implementar la adaptación del <i>B&B</i> con la heurística monótona para Visión Artificial y experimentar con bases de datos de imágenes.	Se implementaron 2 heurísticas para el <i>B&B</i> y se experimentó con bases de datos de la tesis de [Ochoa, 2005]
Realizar un estudio comparativo.	Se realizó un estudio comparativo de eficiencia y calidad contra las herramientas de SA Elvira, See5 y Reduclas.
Utilización de un conjunto total de atributos similar a la que usa [Somol, 2004] el cual utiliza un conjunto de 30 atributos.	Se experimentó con base de datos de 30 atributos y menores, pero el sistema puede servir para cualquier cantidad de atributos.
Se requiere definir previamente el tamaño del subconjunto deseado de atributos.	Es un parámetro inicial importante para el funcionamiento del algoritmo del <i>B&B</i> .
No se puede saber si el tamaño escogido por el usuario es el correcto ya que el algoritmo <i>B&B</i> original, sólo encuentra el óptimo del tamaño seleccionado, (siempre y cuando se escoja la heurística correcta de tipo métrica monótonica).	El tamaño escogido por el usuario no implica que ese número de atributos sea el óptimo para un base de datos en particular, el subconjunto obtenido por el <i>B&B</i> es óptimo con respecto a la medida de evaluación establecida, puede no ser el óptimo para clasificación o representación de la BD.
En esta tesis no se modificará la restricción de <i>B&B</i> en el sentido de que en la cardinalidad del subconjunto de Atributos debe ser predefinido "a priori" por el usuario.	El tamaño del subconjunto a encontrar es un parámetro de entrada para el algoritmo del <i>B&B</i> el cual no puede ser modificado.

6.2. Productos

Para la realización de esta investigación se implementaron diferentes programas los cuales se incluyen en el CD anexo a este trabajo y se explican en el anexo C. Estos programas implementados en lenguaje C son:

- Algoritmo de *B&B* propuesto por [Narendra, 1977]
- Medida de información mutua.
- Medida de la distancia Bhattacharyya
- Discretización de atributos continuos.
- *B&B* con los criterios de evaluación propuestos.

Se realizó la publicación de un artículo en el XI Congreso Internacional sobre Innovación y Desarrollo Tecnológico, celebrado del 2 al 4 de abril de 2014, en Cuernavaca Morelos, México; el cual se muestra en la Figura 6.1.

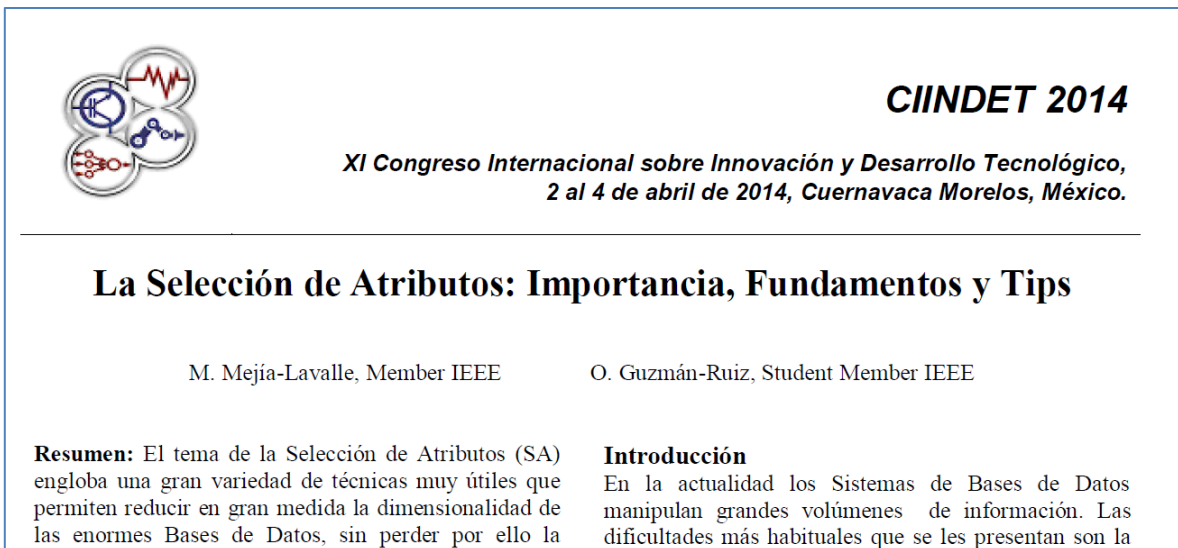


Figura 6.1. Artículo publicado en el CIINDET 2014.

El principal producto de esta investigación es el diseño de 2 criterios de evaluación del subconjunto de atributos (sección 4.3.2), para el algoritmo del *B&B* de SA.

6.3. Conclusiones

Del estudio realizado en el tema de selección de atributos, el análisis efectuado al algoritmo del *B&B* de selección de atributos, así como la experimentación con diferentes sistemas de selección de atributos se puede exponer lo siguiente:

- Existe abundante literatura del tema de selección de atributos que en algunos casos no se entiende que realmente mejoran o es trascendente de su método propuesto, en algunos algoritmos existe bastante similitud, [Ruiz, 2006] cataloga a más de 60 algoritmos de SA, se siguen publicando bastantes artículos, que pocas contribuciones importantes aparecen actualmente, los investigadores proponen un método y dan características particulares en donde su algoritmo puede funcionar y terminan por declarar que el algoritmo funciona diferente para cada base de datos y no pueden dar las características o decir bajo qué condiciones su método funcionará como en [Somol 2004] o como en [Molina, 2002] que piensa que es peligroso confiar en un solo algoritmo de SA. Es interesante que a pesar de toda la investigación existente se haga poca mención a la Teoría de Testores que resultó ser el mejor método de SA para aprendizaje supervisado, porque a diferencia de los otros algoritmos de selección de atributos, trabaja para cualquier vector de atributos esto es para cualquier tipo de dato ya sea continuo o discreto y es capaz de encontrar todos los subconjuntos óptimos del vector de atributos proporcionado.
- El problema en general de las heurísticas son las suposiciones y restricciones que se tienen que hacer, esto puede resultar más problemático ya que pueden implicar supuestos no comprendidos, en el caso del algoritmo del *B&B* una suposición muy importante que se tiene que hacer es proporcionar el tamaño del subconjunto de atributos, cuando se supone que el problema en general de la SA es encontrar el tamaño del subconjunto de atributos óptimo.
- La eficiencia temporal del algoritmo del *B&B* evaluado desde su complejidad computacional según [Somol, 2010] es exponencial $O(2^n)$ por lo que su eficiencia si se basa en esta complejidad computacional puede ser igual o peor aún que los métodos exhaustivos.
- En la mayoría de los artículos de mejoras al *B&B* solo estudian la complejidad computacional, dejando de lado la calidad de la solución (optimalidad), siendo que los dos términos son igual de importantes, de nada sirve un algoritmo eficiente si no entrega resultados favorables en cuanto a calidad o si por el contrario se tiene una calidad excelente pero no se puede obtener una respuesta en un tiempo razonable, no se podría utilizar el método con resultados óptimos.
- La eficiencia algorítmica se debe de tomar en cuenta a la hora de evaluar algoritmos de SA, a veces se le pone poco interés por el aumento constante de

velocidad de cómputo, al trabajar con pocos atributos pareciera que son eficientes luego con un tamaño mayor de atributos al ser algoritmos exponenciales resultan inviables; el algoritmo del *B&B* tiene una complejidad computacional exponencial por lo que resultará inviable para un vector de atributos de alta dimensionalidad, el tiempo computacional del *B&B* es empleado por la construcción del árbol de búsqueda y principalmente el criterio de evaluación; para los métodos *wrapper* su criterio de evaluación es costoso porque involucra la construcción de un clasificador.

- El objetivo de encontrar el subconjunto óptimo de atributos en algún problema de visión artificial, estrictamente no puede ser resuelto con las medidas y criterios propuestos en esta investigación para el algoritmo del *B&B*, por lo que habrá que seguir en la búsqueda de una buena heurística que lleve a la solución de manera lineal; el término subconjunto óptimo en la literatura de SA es visto de diferentes formas, uno se basa en que la optimalidad del subconjunto está dada por aquel subconjunto de atributos que brinde la mejor medida de evaluación del subconjunto escogido de alguna medida de evaluación vistas en la sección 3.5, la otra en que la optimalidad está dada por todos aquellos subconjuntos minimales que conservan la distribución de la clase del vector de atributos proporcionado. En este sentido si el problema en cuestión es de clasificación el *B&B* no es lo más adecuado para resolverlo, ya que como principales restricciones tiene que la medida de evaluación debe de ser monótona y saber de antemano el tamaño (cardinalidad) del subconjunto óptimo

6.4. Trabajo futuro

El algoritmo del *B&B* para SA tiene algunas restricciones y limitantes, por lo que es necesario aliviar esos inconvenientes con algún trabajo futuro como:

- Diseñar un algoritmo que utilice los principios del *B&B* pero sin la restricción de la cardinalidad previa.
- Buscar medidas monótonas o aproximadas que estén acopladas con el rendimiento del clasificador.
- Encontrar heurísticas para el *B&B* que permitan realizar podas y lleven a la solución en tiempo polinómico.
- *B&B* para bases de datos con cualquier tipo de datos, estos datos pueden ser: continuos, discretos, ausencia de información y mixtos.

Referencias

Referencias bibliográficas

- [Almuallim, 1991] Almuallim, H., Dietterich, T. 1991. Learning with many irrelevant features. In Proc. Of the Ninth Nat. Conf. on AI, MIT Press, pp. 547-552.
- [Armañanzas,2004] Rubén Armañanzas Arnedillo, Medidas de Filtrado de Selección de Variables mediante la plataforma Elvira; Facultad de informática, Universidad del país Vasco; Agosto 2004
- [Avellaneda, 2009] Diana Avellaneda Avellaneda, (2009) Caracterización de Texturas Naturales. Tesis de maestría CENIDET, Cuernavaca, Morelos.
- [Barbara, 1997] Barbara, D., DuMouchel, W., Faloutsos, C., Haas, P., V Hellerstein, J., Ioannidis, Y., Jagadish, H., Johnson, T., Ng, R., Poosala, V., Ross, K., and Sevcik, K.(1997). The New Jersey Data Reduction Report. Bulletin of the Technical Committee on Data Engineering, 20(4).
- [Ben-Bassat, 1982] Moshe BenBassat. Use of Distance Measures, Information Measures and ErrorBound in Feature Evaluation. P. R. Krishnaiah and L. N. Kanal, eds., Handbook of Statistics, Vol. North-Holland Publishing Company (1982) 773- 791
- [Davies, 1994] Scott Davies, Stuart Russel, NP-completeness of searches for smallest posible feature sets. Association for the Advancement of Artificial Intelligence (AAAI) Technical Report FS-94-02. Compilation copyright, 1994
- [Domingos, 2001] Domingos Pedro, When and how to subsample. Report on the KDD-2001 Panel, SIGKDD Explorations, Volumen 3 Issue 2, 74-75.
- [Fukunaga, 1990] Fukunaga K. Introduction to Statical Patter Recognition, Second ed. Academic Press, Inc., 1990.
- [Galeana, 2008] Deysy Galeana Pérez (2008). Caracterización Automática de Imágenes en un Dominio Transformado tesis de maestría CENIDET, Cuernavaca, Morelos.

- [Guerequeta, 2000] Rosa Guerequeta, Antonio Vallecillo. Técnicas de Diseño de Algoritmos. Servicio de publicaciones de la Universidad de Málaga. Segunda Edición, Mayo 2000 .
- [Guyon, 2003] Guyon Isabelle, Andree Elisseef An Introduction to Variable and Feature Selection; Journal of Machine Learning Research 3 (2003) 1157-1182.
- [Han, 2006] J. Han, M. Kamber, J. Pei. Data Mining: Concepts and Techniques. Second Edition. University of Illinois at Urbana-Champaign. Micheline Kamber. Estados Unidos, 2006.
- [John, 1994] George H John, Ron Kohavi, Karl Pfleger. Irrelevant Features and the Subset Selection Problem. Machine Learning Proceedings of the Eleventh International Conference. Morgan Kaufmann Publishers San Francisco CA. 1994.
- [Kira, 1992] Kira, K., Rendell, L. 1992. The feature selection problem: traditional methods and a new algorithm. Tenth Nat. Conf. on AI, MIT Press, pp. 129-134.
- [Kohavi, 1995] Ron Kohavi, Tesis doctoral. Wrappers for performance enhancement and oblivious decision graphs, Stanford University Septiembre 1995.
- [Kohavi, 1997] Kohavi, R., John, G. Wrappers for feature subset selection. Artificial Intelligence, 97(1-2):273-324, December 1997.
- [Kudo,1997] M. Kudo and J. Sklansky. A Comparative Evaluation of medium and large-scale Feature Selectors for Pattern Classifiers. In Proc. of the 1st Int. Workshop on Statistical Techniques in Pattern Recognition, pages 91–96, Prague, Czech Republic, 1997.
- [Larsson, 2006] Roland E. Larsson, Bruce H. Edwards. Introducción al Algebra Lineal. Editorial Limusa, Versión autorizada en Español de la obra publicada en Inglés: Elementary Linear Algebra ,2004
- [Liu, 1998] Huan Liu, Hiroshi Motoda. Feature selection for knowledge discovery and data mining. Kluwer Academic Publishers, New York in 1998
- [Liu, 2001] Huan Liu and Hiroshi Motoda. Instance Selection and Construction for Data Mining. Kluwer International. Series in Engineering and Computer Science.

- [Maravall, 1994] Dario Maravall. Reconocimiento de Formas y Visión Artificial. Editorial RA-MA 1994.
- [Mejía, 2007] Mejía Lavalle Manuel (2007) Selección de Atributos considerando inter-dependencias. Tesis Doctoral ITESM, Cuernavaca Morelos.
- [Mejía, 2014] Mejía-Lavalle M. Guzmán-Ruiz O. La Selección de Atributos: Importancia fundamentos y tips. XI Congreso Internacional sobre Innovación y Desarrollo Tecnológico, CIINDET 2014, Cuernavaca Morelos, México.
- [Mexicano, 2007] Adriana Mexicano Santoyo (2007), Desarrollo de una Metodología para la Selección de Atributos y Generación de Indicadores para la Aplicación de Minería de Datos a una Base de Datos Real de Registros de Cáncer en Base Poblacional, tesis de maestría CENIDET, Cuernavaca, Morelos.
- [Molina, 2002] Molina, L., Belanche, L., Nebot, A. (2002). Feature selection algorithms, a survey and experimental evaluation. IEEE Int. Conf. on Data Mining, Maebashi City Japan, pp. 306-313.
- [Nakariyakul, 2009] Songyot Nakariyakul, A Review of Suboptimal Branch and Bound Algorithms, 2009 International Conference on Computer Engineering and Applications IPCSIT vol.2 (2011) IACSIT Press, Singapore.
- [Narendra, 1977] Narendra, M., Fukunaga, K. (1977) A Branch and Bound algorithm for feature subset selection. IEEE Transactions on Computers C-26(9): 917-922.
- [Ochoa, 2005] Ochoa Somuano Jorge (2005) Técnicas de Selección de Atributos para la Categorización Automática de Escenas Visuales. Tesis CENIDET, Cuernavaca Morelos.
- [Orbe, 2003] Orbe Trujillo Eduardo (2003) Desarrollo de una Herramienta para el Modelado Automático de Objetos Alfalexibles. Tesis CENIDET, Cuernavaca Morelos
- [Pajares, 2008] Pajares Gonzalo, De la cruz, Jesús; Visión por computador-Imágenes digitales y aplicaciones, Segunda edición, Alfaomega, Ra-ma 2008.
- [Pudil, 1994] P. Pudil, J. Novovicova, J. Kittler. Floating search methods in feature selection. Elsevier Pattern Recognition Letters 15

- (1994) 1119-1125
- [Quinlan, 1993] Quinlan, J. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers. San Mateo California
- [Shanon, 1948] C. E. SHANNON A Mathematical Theory of Communication. Reprinted with corrections from The Bell System Technical Journal Vol. 27, pp. 379–423, 623–656, July, October, 1948
- [Somol, 2004] Somol Petr, Pudil Pavel (2004) Fast Branch & Bound Algorithms for Optimal Feature Selection; IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 26, No. 7
- [Somol, 2008] Somol, P.; Novovicová, J.; Grim, J., & Pudil, P. 2008. Dynamic oscillating search algorithms for feature selection. In ICPR 2008 Los Alamitos, CA, USA: IEEE Computer Society
- [Somol, 2010] Petr Somol, Jana Novovicová and Pavel Pudil. Efficient Feature Subset Selection and Subset Size Optimization. Pattern Recognition, Recent Advances, 75-98. In-Teh, Vukovar, Croatia 2010.
- [Somol, 2011] Petr Somol, Jana Novovicova, Pavel Pudil. Are Better Feature Selection Methods Actually Better?. Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies, 2011.
- [Ruiz, 1999] Ruiz Shulcloper José, Adolfo Guzmán, J.F. Martínez, (1999) Enfoque Lógico Combinatorio al Reconocimiento de Patrones, Avance en Reconocimiento de Patrones, Centro de Investigación en Computación, Instituto Politécnico Nacional, México.
- [Ruiz, 2006] Roberto Ruiz Sánchez. Heurísticas de selección de atributos para datos de gran dimensionalidad. Tesis doctoral, Universidad de Sevilla, Junio de 2006.
- [Russel, 1996] S. Russell, P. Norvig. Inteligencia Artificial: un enfoque moderno. Prentice Hall Hispanoamericana. México, 1996. ISBN: 968-880-682-X
- [Troncoso, 2007] Perla Aurora Troncoso Rey (2007) Indexado y Recuperación de Imágenes por Contenido. Tesis de maestría CENIDET, Cuernavaca, Morelos.
- [Vergara, 2003] Osslan Osiris Vergara Villegas, Reconocimiento de texturas artificiales, aplicación a la inspección visual. Tesis de

- maestría CENIDET, Cuernavaca, Morelos.
- [Wang, 2003] Zhenxiao Wang. An Improved Branch & Bound Algorithm in Feature Selection. Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing 9th International Conference, 2003.
- [Walpole, 2012] Ronald E. Walpole, Raymond H. Myers, Sharon I. Myers y Keying Ye. Probabilidad y Estadística para Ingeniería y Ciencias 9ª edición 2012. Pearson Education, Inc.
- [Witten,2011] Witten Ian H, Eibe Frank, Mark A. Hall; Data Mining, Practical machine learning tools and techniques; third edition, Elsevier,2011
- [Yu, 1993] B. Yu, B. Yuan. A More Efficient Branch and Bound Algorithm for Feature Selection. Pattern Recognition, Vol. 26, pp. 883-889, 1993.
- [Yu, 2004] Lei Yu, Huan Liu; Efficient Feature Selection via Analysis of Relevance and Redundancy; Journal of Machine Learning Research 5 (2004) 1205–1224

Referencias en línea

- [ASU, 2016] <http://featureselection.asu.edu/> fecha de consulta noviembre de 2016
- [Elvira, 2016] <http://www.ia.uned.es/investig/proyectos/elvira/#proyecto> fecha de consulta noviembre de 2016
- [Image, 2016] <http://www.cs.washington.edu/research/imagedatabase/> fecha de consulta noviembre de 2016
- [Rapidminer, 2016] <http://rapidminer.com/products/studio/> fecha de consulta noviembre de 2016
- [Rulequest, 2016] <http://www.rulequest.com/see5-info.html> fecha de consulta noviembre de 2016
- [Segmentation, 2016] <http://archive.ics.uci.edu/ml/datasets/Image+Segmentation> fecha de consulta noviembre de 2016

[WDBC, 2016] <http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wdbc.data> fecha de consulta noviembre de 2016

[Weka, 2016] <http://www.cs.waikato.ac.nz/ml/weka/> fecha de consulta noviembre de 2016

Anexos

Anexo A

Base de datos

Descripción de las bases de datos usadas

IMAGE SEGMENTATION DATA SET [Segmentation,2016]

Esta base de datos es usada en la tesis de [Ochoa, 2005], La base de imágenes original tiene 7 clases. Sin embargo, para el caso de prueba presentado en este documento se utilizan sólo dos clases (Pasto y Camino) de los datos de entrenamiento de la base de imágenes original. Por otro lado, de los 300 objetos por clase de la base de imágenes original se tomaron para los casos de prueba de esta tesis sólo 200 por clase, los otros cien objetos de cada clase se dejaron para la fase de prueba.

Cuenta con las siguientes características:

Título: Datos de imágenes segmentadas (*Image Segmentation*)

Creador: Grupo de Visión, Universidad de Massachussets

Donador: Grupo de Visión (Carla Brodley, brodley@cs.umass.edu)

Fecha: Noviembre de 1990

URL: <http://archive.ics.uci.edu/ml/datasets/Image+Segmentation>

Información relevante: Los objetos se tomaron al azar de un conjunto de 7 bases de datos, con imágenes tomadas al aire libre. Las imágenes fueron segmentadas de manera que se creara una clasificación para cada pixel. Cada caso es una región 3x3.

Numero de objetos: datos de entrenamiento: 210, datos para prueba: 2100

Número de variables: 19 variables continuas

Información de las variables:

- 1) **Region-centroid-col:** columna del pixel central de la región.
- 2) **Region-centroid-row:** fila del pixel central de la región.
- 3) **Region-pixel-count:** número de pixeles en una región = 9.

- 4) **Short-line-density-5:** el resultado de un algoritmo de extracción de línea que cuenta cuántas líneas de longitud 5 (cualquier orientación) con contraste bajo, menor o igual a cinco, pasan por una región.
- 5) **Short-line-density-2:** el resultado de un algoritmo de extracción de línea que cuenta cuántas líneas de longitud 5 (cualquier orientación) con contraste alto, mayor a cinco, pasan por una región.
- 6) **Vedge-mean:** mide el contraste de pixeles adyacentes horizontales en la región si hay 6 se obtiene la media. Esta variable es usada para detectar el borde vertical.
- 7) **Vegde-sd:** mide el contraste de pixeles adyacentes horizontales en la región si hay 6 se obtiene la desviación estándar. Esta variable es usada para detectar el borde vertical.
- 8) **Hedge-mean:** mide el contraste de pixeles adyacentes verticales para obtener la media. Se usa para la detección de una línea horizontal.
- 9) **Hedge-sd:** mide el contraste de pixeles adyacentes verticales para obtener la desviación estándar. Se usa para la detección de una línea horizontal.
- 10) **Intensity-mean:** media de la región en $(R + G + B) / 3$.
- 11) **Rawred-mean:** el promedio de la región en el valor R.
- 12) **Rawblue-mean:** el promedio de la región en el valor B.
- 13) **Rawgreen-mean:** el promedio de la región en el valor G.
- 14) **Exred-mean:** mide el exceso en el color rojo $(2R - (G + B))$.
- 15) **Exblue-mean:** mide el exceso en el color azul $(2B - (G + R))$.
- 16) **Exgreen-mean:** mide el exceso en el color verde $(2G - (R + B))$.
- 17) **Value-mean:** transformación no lineal 3-d en RGB.
- 18) **Saturation-mean:** saturación no lineal 3-d en RGB
- 19) **Hue-mean:** color no lineal 3-d en RGB

Valores de variables faltantes: Ninguno

Distribución de clases: Clases: 1.Fachadas de ladrillo, 2.Cielo, 3.Follaje, 4.Cemento, 5.Ventana, 6.Camino, 7.Pasto

Objetos por clase: 30 para cada clase en los objetos de entrenamiento 300 para cada clase en los objetos de prueba

IMAGE DATASET [Image, 2016]

Esta base de datos es usada en la tesis de [Ochoa, 2005], la base de imágenes original tiene 21 clases, para el caso de prueba presentado en la tesis se utilizan sólo tres clases (SpringFlowers, SwissMountains, Yellowstone)

Título: Imágenes de exteriores representadas por sus atributos en forma binaria

Información de origen:

Creador: Departamento de Ciencias Computacionales e Ingeniería.

Fecha: Febrero de 2014 (fecha de consulta)

URL: <http://www.cs.washington.edu/research/imagedatabase/>

Representación en binario: Jorge Ochoa Somuano

Fecha: Mayo de 2004

Información relevante: Los objetos se obtuvieron como imágenes y un archivo descriptivo que contenía las variables contenidas por cada imagen, se eliminaron las redundantes y las demás se colocaron en forma binaria en un nuevo archivo de tipo texto, de acuerdo al siguiente criterio: si existía dicha variable en la imagen se colocaba un 1; de lo contrario se colocaba un 0.

Numero de objetos: 123

Número de variables: 21

Información de las variables: Cada variable está representada por una columna.

- 1) Flowers
- 2) Building
- 3) Bushes
- 4) Trees
- 5) Ground
- 6) Sidewalk
- 7) Grass
- 8) Clear sky
- 9) Water
- 10) People
- 11) Tree trunk
- 12) Mountains
- 13) Rocks
- 14) Cloudy sky
- 15) Snow
- 16) Greenery
- 17) Partial cloudy sky
- 18) Greener
- 19) Thermal pool
- 20) rocks

21)Hills

Valores de variables faltantes: Ninguno

Distribución de clases:

Clases: 1.SpringFlowers, 2.SwissMountains, 3.Yellowstone

Objetos por clase: 1.- 48, 2.- 30, 3.- 45. Cada objeto está representado por una línea.

BREAST CANCER WISCONSIN (DIAGNOSTIC DATA SET) [WDBC, 2014].

Es una base de datos que contiene 569 instancias y 32 atributos, para la realización de pruebas en este trabajo se utilizaron sólo 400 instancias y 31 atributos quitando el atributo de número de identificación.

Título: Diagnostic Breast Cancer Wisconsin por sus atributos en forma binaria
Información de origen:

Creador: University of Wisconsin, Clinical Sciences Center

Fecha: Noviembre 1995

URL:<http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin>

Información relevante: Los atributos se calculan a partir de una imagen digital de una biopsia por aspiración con aguja fina (*fine needle aspiration biopsy*, FNA), y se describen los núcleos de las células presentes en la imagen.

Numero de objetos: 569

Número de variables: 32

Información de las variables: Cada variable está representada por una columna.

Atr1.-ID Number

Atr2-30.-

- a) Radius (mean of distances from center to points on the perimeter)
- b) Texture (standard deviation of gray-scale values)
- c) Perimeter
- d) Area
- e) Smoothness (local variation in radius lengths)
- f) Compactness ($\text{perimeter}^2 / \text{area} - 1.0$)
- g) Concavity (severity of concave portions of the contour)

h) Concave points (number of concave portions of the contour)

i) Symmetry

j) Fractal dimension ("coastline approximation" - 1)

Atr31.- Diagnosis (M = *malignant*, B = *benign*)

Valores de variables faltantes: Ninguno

Distribución de clases: B=Benigno. M=maligno

Objetos por clase: 357 casos benignos, 212 casos malignos

Anexo B

Herramientas de SA

SISTEMA ELVIRA

El sistema o entorno Elvira (Elvira Consortium, 2002) es una plataforma para la construcción y uso de modelos gráficos probabilísticos, que fue construida con el apoyo del Ministerio de Ciencia y Tecnología, bajo dos proyectos de investigación: Elvira (TIC97-1135-C04, 1997-2000) y Elvira II (TIC2001-2973-C05, 2001-2004).

El sistema Elvira Figura B.1, incorpora métodos *filter* tipo ranking, en base a diferentes medidas extraídas del artículo [Ben-Bassat, 1982], estas medidas son las siguientes[Armañanzas,2004]:

- Información mutua
- Distancia Euclídea
- Distancia Matusita
- Kullback-Leibler 1
- Kullback-Leibler 2
- Entropía de Shannon
- Distancia Bhattacharyya

En la Figura B.1 se puede apreciar la pestaña superior activada de opciones de preprocesado, en la pestaña inferior medidas *filter* se muestran las diferentes medidas, por las que se puede obtener un ranking de los atributos.

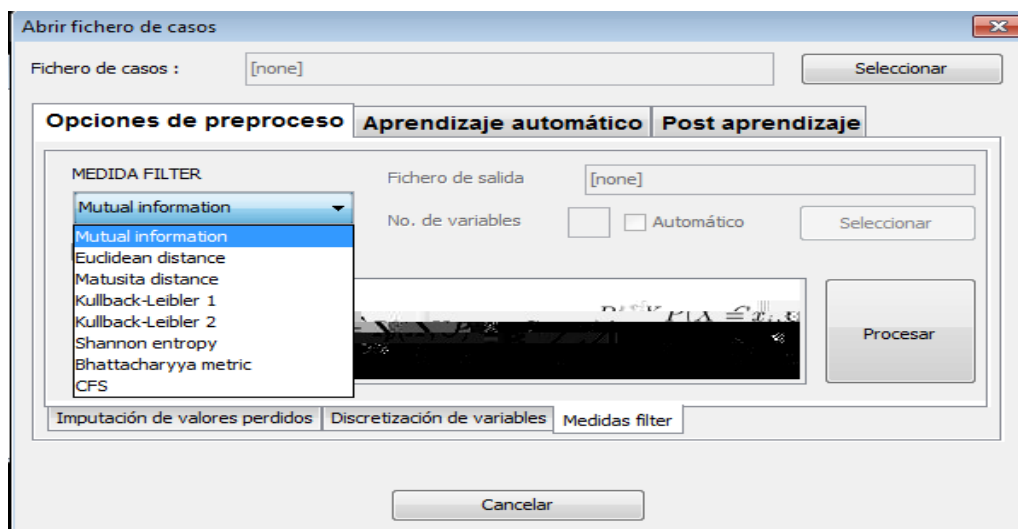


Figura B.1. Sistema Elvira.

INFORMACIÓN MUTUA: En base a la teoría de la información, calcula la relación que existe entre una variable y otra dada. La semántica de esta medida se centra en la cantidad de incertidumbre que el conocimiento de una variable es capaz de despejar con respecto al estado en el que se encuentre la segunda. Esta medida se encuentra acotada en el intervalo [0,1]; valores cercanos a 1 marcan una alta correlación, mientras que, valores cercanos a 0 indican independencia entre las variables analizadas.

Su expresión matemática se presenta a continuación, donde: X es la variable a evaluar, C es la variable clase, r_x es el número de estados que X puede tomar, y r_c es el número de estados que C puede tomar.

$$I_p(X, C) = \sum_{i=1}^{r_x} \sum_{j=1}^{r_c} P(x_i, c_j) \log \frac{P(x_i, c_j)}{P(x_i)P(c_j)} \quad (\text{Ec. B1})$$

DISTANCIA EUCLÍDEA: Su expresión matemática, en base a los mismos parámetros que en el caso anterior, se formula como sigue.

$$D_c(X, C) = \sqrt{\sum_{i=1}^{r_x} \sum_{j=1}^{r_c} \sum_{k=1}^{k < j} P(c_k)P(c_j) |P(x_i|c_k)^2 - P(x_i|c_j)^2|^2} \quad (\text{Ec. B2})$$

DISTANCIA MATUSITA: La formulación original de esta métrica mide la distancia entre dos distribuciones de probabilidad. Al generalizarse, se intenta medir la distancia media entre las diferentes distribuciones marginales de cada uno de los valores del atributo con los valores de la clase. Su expresión matemática se formula como:

$$D_m(X, C) = \sum_{i=1}^{r_c} \sum_{j=1}^{j < i} P(c_i)P(c_j) \left[\sum_{k=1}^{r_x} \sqrt{P(x_k|c_i)P(x_k|c_j)} \right] \quad (\text{Ec. B3})$$

DIVERGENCIA DE KULLBACK-LEIBLER: La divergencia de Kullback-Leibler es el método más conocido para la medición de distancias entre dos distribuciones de probabilidad. Su formulación genérica es:

$$D_{kl}(P(X), Q(X)) = \sum P(x_i) \log \frac{P(x_i)}{Q(x_i)} \quad (\text{Ec. B4})$$

Primero se pondera cada comparación con las probabilidades marginales de cada clase:

$$KL(X, C) = \sum_{i=1}^{r_c} \sum_{j=1}^{j < i} P(c_i)P(c_j) KL_{ij}(X, C)_1 \quad (\text{Ec. B5})$$

KULLBACK-LEIBLER 1: Con las probabilidades marginales *a priori*.

$$KL_{ij}(X, C)_1 = D_{kl}(P(X|c_i), P(X)) + D_{kl}(P(X|c_j), P(X)) \quad (\text{Ec. B6})$$

KULLBACK-LEIBLER 2: las probabilidades condicionales *a priori*.

$$KL_{ij}(X, C)_2 = D_{kl}(P(X|c_i), P(X|c_j)) + D_{kl}(P(X|c_j), P(X|c_i)) \quad (\text{Ec. B6})$$

ENTROPIA DE SHANNON: Su formulación, con los mismos parámetros que en los demás casos, es:

$$Sh(X, C) = \sum_{i=1}^{r_c} \sum_{j=1}^{j < i} P(c_i)P(c_j)H_{ij}(X) \quad (\text{Ec. B7})$$

$$H_{ih}(X) = - \sum_{k=1}^{r_x} P(x_i|c_i) \log_2 P(x_i|c_j) + P(x_i|c_j) \log_2 P(x_i|c_i) \quad (\text{Ec. B8})$$

DISTANCIA BHATTACHARYYA: Esta distancia mide la dependencia que existe entre dos distribuciones de probabilidad. Las distribuciones que van a ser comparadas son las probabilidades *a priori* de una variable, contra la condicionada a la clase. Se intenta de esta forma ver qué grado de dependencia se encuentra entre ambas distribuciones; cuanto mayor sea ese grado, mayor será el peso de la variable analizada en el problema clasificatorio.

$$Bh(X, C) = \sum_{i=1}^{r_c} -\log \left[P(C_i) \sum_{j=1}^{r_x} \sqrt{P(x_j|c_i)P(x_j)} \right] \quad (\text{Ec. B9})$$

UMBRAL DE CORTE

El mayor problema que se presenta una vez realizado un ranking de atributos es decidir con cuántas de esas variables más importantes se toman, empezando por la parte superior del ranking, y cuáles son las que rechazo por suponer que no aportan información alguna.

Existen diferentes alternativas para abordar el problema, pero todas ellas están basadas en el hecho de que, en algún momento, la ganancia en información que se produce por la adición de otra variable no es significativa.

De las técnicas que intentan aproximar un resultado óptimo se ha seleccionado el método presentado en [Molina, 2002], en el cual se refiere como método del codo, o punto de corte óptimo.

Este método va a intentar determinar automáticamente ese punto de corte fijando la posición de corte cuando exista una diferencia significativa entre la reducción de incertidumbre despejada por un atributo y el siguiente en la lista.

El algoritmo propuesto para ello se enuncia de la siguiente forma:

1. Eliminar los atributos con coeficientes más lejanos por abajo que dos veces la varianza total, desde el valor medio del estadístico que se esté computando.

2. Definir $s_i = w_i + w_{i-1}$ donde w_i es el valor del estadístico para el atributo i .
3. A partir de estos valores definir

$$\sigma_j = \sum_{i=2}^j s_i \quad (\text{Ec. B10})$$

4. El objetivo es encontrar el atributo x_j tal que

$$K_{x_j} = 1 - \frac{\sigma_j}{\sigma_n} \frac{n-j}{n} \quad (\text{Ec. B11})$$

Sea máximo, n =número de atributos

5. El punto de corte estará entre el atributo x_j y x_{j+1} .

SISTEMA SEE5/C5.0

El sistema See5 (Windows XP / Vista / 7 / 8) y su contraparte C5.0 (Unix) son herramientas sofisticadas de minería de datos para descubrir patrones, estos patrones se pueden utilizar para predecir los resultados en situaciones futuras y ayudar en la toma de decisiones, Figura B.2.

Algunas características importantes:

- Ha sido diseñado para analizar bases de datos sustanciales que contienen miles de millones de registros, también se aprovecha de los ordenadores con hasta ocho núcleos en una o varias CPU's (incluyendo Intel Hyper-Threading) para acelerar el análisis.
- Los clasificadores See5 / C5.0 se expresan como árboles de decisión o conjuntos de reglas si-entonces, estas formas son generalmente más fáciles de entender que las redes neuronales.
- See5 / C5.0 está disponible para Windows XP / Vista / 7 / 8 y Linux.
- See5 / C5.0 es fácil de usar y no se necesita ningún conocimiento especial de estadística o de aprendizaje automático.
- RuleQuest proporciona el código fuente C para que los clasificadores construidos por See5 / C5.0 se pueden utilizar en sistemas propios de la organización.

Es un software privativo la versión de demostración no permite procesar más de 400 casos de entrenamiento o prueba.

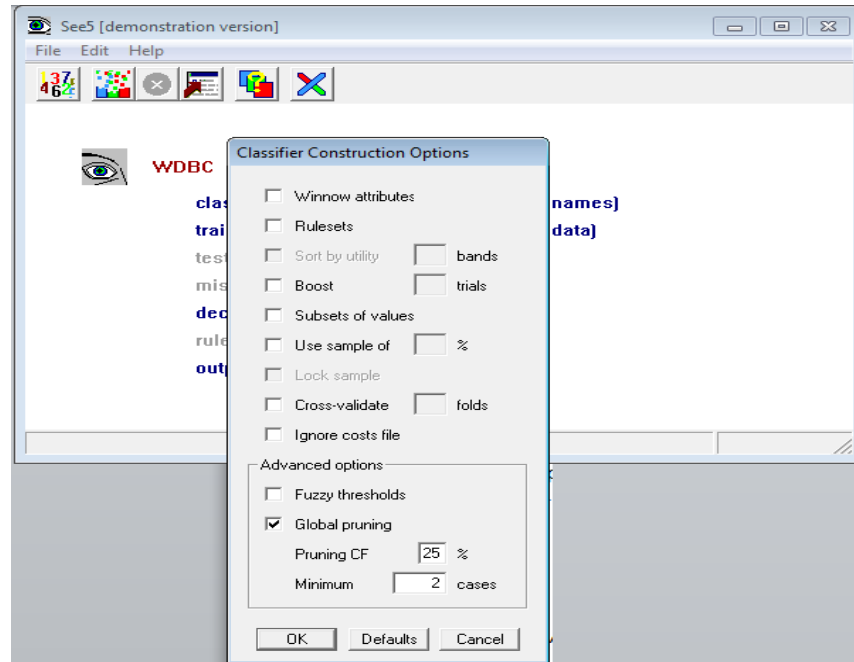


Figura B.2. Sistema See5.

SISTEMA WEKA

El sistema Weka (*Waikato Environment for Knowledge Analysis*) entorno para análisis del conocimiento de la Universidad de Waikato, es una plataforma de software para el aprendizaje automático y la minería de datos escrito en Java y desarrollado en la Universidad de Waikato, Figura B.3.

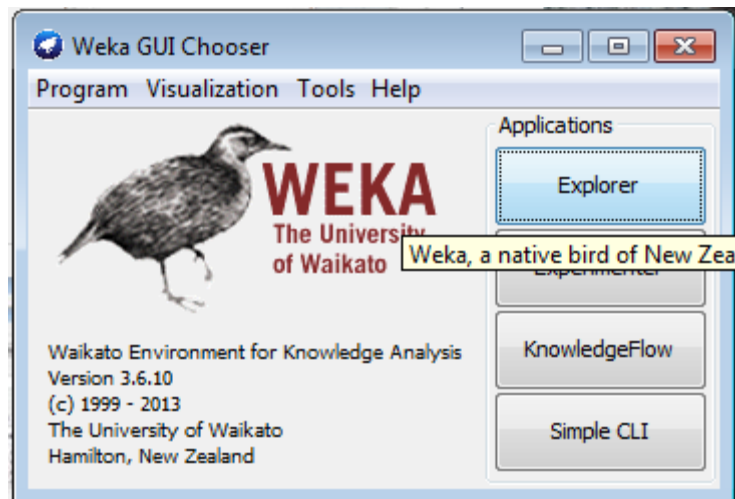


Figura B.3. Sistema Weka.

SISTEMA REDUCLAS

El sistema Reduclas, Figura B.4, es desarrollado en el CENIDET como parte de la tesis “Técnicas de selección de atributos para la categorización automática de escenas visuales” [Ochoa, 2005], este sistema utiliza la técnica de Teoría de Testores para la reducción de dimensiones, determina los testores típicos que son el conjunto mínimo de variables que describen de manera idéntica al total de objetos, tal como si se estuvieran utilizando todas las variables.

Esta versión sólo permite el uso de 400 muestras y 30 atributos pero al disponer el código fuente puede ser modificado para su uso con más muestras o atributos.



Figura B.4. Sistema Reduclas.

Anexo C

Contenido del CD

Programas

Código fuente del *B&B* propuesto por [Narendra, 1997]

D:\Contenido_CD\BB_Narendra\BB.c

Código fuente para la medida de información mutua.

D:\Contenido_CD\BB_InfMutua\infmtut.c

Código fuente para la medida de distancia Bhattacharyya.

D:\Contenido_CD\BB_InfMutua\Metric_Bat.cpp

Código fuente del *B&B* con los criterios de evaluación.

D:\Contenido_CD\Criterios1_2\BB_criterios.cpp

Código fuente para discretizar atributos continuos.

D:\Contenido_CD\Discretiza\ DISCRETIZA.cpp

Bases de datos

Base de datos *Image segmentation dataset* [Segmentation, 2016].

D:\Contenido_CD\Base_de_datos\ GRASS_DISC_20_400.txt

Base de datos *Image dataset* [Image, 2016]

D:\Contenido_CD\Base_de_datos\ YELLOWS_DISC_22_12.txt

Base de datos *Brest cancer Wisconsin (Diagnostic dataset)* [WDBC,2016]

D:\Contenido_CD\Base_de_datos\ WDBC_DISC_31_569.txt