Universidad Autónoma de Baja California
Instituto de Ingeniería
Maestría y Doctorado en Ciencias e Ingeniería

Métodos Locales de Búsqueda no-deterministas en la Selección de Atributos

Tesis que para obtener el grado de:

MAESTRO EN CIENCIAS

Presenta

Marina Pamela Fernández Pérez

Director de Tesis:

Dr. Félix Fernando González Navarro

Mexicali, B. C. Noviembre 2014

Dedicatorias

A Dios por permitirme la vida
A mi familia por su apoyo
A mis maestros por sus consejos
A mis amigos por apoyarnos en los estudios

Reconocimientos

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por darme la oportunidad de seguir continuando en mis estudios, para mi superación persinal

A la Universidad Autónoma de Baja California (UABC) por el apoyo aportado a lo largo de este tiempo ya que es mi segunsa casa; y por el apoyo del proyecto y la realización de verlo terminado.

Al Instituto de Ingeniería por brindarme sus instalacines para la realización de la Tesis.

A mi asesor, Dr. Félix Fernando González Navarro, por el apoyo proporcionado, compartir sus conocimientos y por los consejos dados a lo largo de mi estancia y en la elaboración de la tesis.

A mi familia ya que sin su apoyo y paciencia no hubiera sido esto posble.

A mis compañeros y a todas las personas que conocí a lo largo de este trabajo, ya que me dieron un poco de su tiempo para discutir sobre mi tema y me ayudaron en mi estancia.

Resumen

La reducción de la dimensionalidad mediante la selección de atributos es uno de los pasos fundamentales del pre-procesado de datos, como fase previa al análisis de información. De entre los diversos algoritmos de reducción de dimensionalidad muchos se basan en el enfoque de envoltura (wrapper), que utilizan un clasificador (Vecinos cercanos, Bayesiano ingenuo, máquina de vector de soporte, análisis discriminante lineal y cuadrático) y un método de error de estimación (validación cruzada de 5X5) para la reducción de atributos. En este estudio se analizaron diferentes bases de datos, de muestras artificiales y reales, con datos continuos y discretos, con dimensiones diferentes cada una de ellas en donde el algoritmo de Aceptación por Umbral en las pruebas estadísticas de Friedman de obtuvo buenos resultados y un coste computacional menor contra los algoritmos de Recocido Simulado, Algoritmos Genéticos, y Búsqueda Tabu en la selección de atributos.

Contenido

Lista de Figuras

Lista de Tablas

Capítulo 1

Introducción

El problema de la Selección de Atributos (SA) es un campo dentro de la investigación de las ciencias de la computación muy activo en nuestros días. El propósito superior de éste es el de *reducir* el tamaño de los conjuntos de datos bajo análisis –i.e eliminar variables independientes o predictores–, sin degradar su capacidad de explicación de las variables dependientes o efectos. Como etapa previa al proceso de entrenamiento de clasificadores, la SA incide de manera significativa en varios aspectos de la clasificación o reconocimiento de objetos: mejora sustancialmente el desempeño de algoritmos de clasificación en términos de velocidad de aprendizaje, capacidad de generalización o su representación ya sea gráfica o interpretativa. Cuando existen atributos irrelevantes o redundantes en los datos de entrenamiento, los clasificadores están propensos realizar las tareas de reconocimiento de manera errónea. Así, la SA *genera* un subconjunto reducido de atributos a partir de los datos del problema, esto es llevando a cabo un proceso de búsqueda en todo el espacio de posibles soluciones. Es sabido que el número de subconjuntos posibles que se pueden generar a partir de un conjunto esta dado por 2^n posibilidades, donde n es el número de variables. Por ejemplo, para un conjunto $A = \{a, b, c\}$ el total de posibles subconjuntos a generar es de 8: $\{\phi, a, b, c, ab, ac, bc, abc\}$. En un problema de la vida real, el tamaños de los conjuntos de datos puede ser fácilmente de cientos a miles de variables. Un problema de $n = 500$, se tendrían alrededor de $3{,}27x10^150$ posibles subconjuntos, lo cual desde un punto de vista de ingeniería sería imposible de evaluar, ya sea en tiempo de procesamiento, capacidad de almacenamiento, entre otros aspectos. En este

sentido, la SA viene a *cooperar* en la búsqueda de una solución de menor tamaño, en tiempos razonables. Son dos grandes estrategias de búsqueda de subconjuntos las que se encuentran implementadas en los distintos algoritmos de SA: las búsquedas *deterministas*, es decir, aquellas que siempre inician y terminan en la misma solución y las *no-deterministas* las cuales pueden variar de una ejecución a otra pero con resultados aceptables. Este trabajo de investigación que culmina con la presente tesis, se centra en esta última modalidad. Son cuatro algoritmos de búsqueda los que son analizados en un contexto de SA, el Algoritmo de *Recocido Simulado (RS)*, el *Algoritmo Genético (AG)*, el Algoritmo de *Búsqueda Tabú (BT)*, y el algoritmo de *Aceptación por Umbral (AU)*, los cuales ofrecen diversas ventajas sobre los algoritmos de naturaleza determinista. Por otra parte, en esta tesis se plasma el uso del algoritmo de *Aceptación por Umbral* en la SA, ya que en la literatura científica se encuentran pocas o nulas referencias a éste, como una opción viable. De esta manera, éste es comparado con el resto de los algoritmos señalados en una serie de experimentos extensivos, afín de ofrecer un panorama de su potencialidad y como una herramienta fácil de implementar en tareas de minería de datos o reconocimiento de patrones.

Esta tesis está organizada de la siguiente manera en el capítulo uno realiza una introducción al tema bajo estudio, se hace el planteamiento del problema y se listan los objetivos planteados; en el capítulo dos se expone el marco teórico, en el cual se detalla cada uno de los algoritmos utilizados, así como las bases de datos utilizadas y la metodología que se siguió; en el capítulo tres se muestran los materiales y métodos utilizados para el desarrollo de los experimentos; en el capítulo cuatro se ofrecen los resultados experimentales; y por último se ofrece al lector un capítulo de conclusiones y algunas lineas de trabajo futuro.

1.1. Planteamiento del Problema

En el campo del descubrimiento de conocimiento conocido como KDP (Knowledge Discovery Process por sus siglas en inglés), o la minería de datos, existen diversas metodologías para el estudio de problemas. La metodología de Duda & Hart (2001), el KDP de Fayyad *et al.* (1996), el modelo CRISP-DM (Cross

Industry Standard Process for Data Mining) (Shearer, 2000), entre otros, comparten una etapa que actualmente es considerada esencial, que es la Selección de Atributos. Esta consiste en la reducción o eliminación de algunas variables que representan o explican el comportamiento de un sistema en particular. Esta reducción del problema en términos de variables, ayuda de manera significativa al proceso de reconocimiento de patrones, al facilitar la construcción de algoritmos inteligentes de menor tamaño, reducir la carga computacional o requerimientos de potencia de cálculos, y la oportunidad de ofrecer modelos finales más sencillos para explicar a un usuario final. Los métodos de selección de atributos basados en algoritmos estocásticos o no-deterministas, permiten obtener al menos un resultado sub-óptimo en un periodo de tiempo relativamente corto o problemas que resultarían imprácticos de resolver mediante la búsqueda de todas las posibles soluciones. Por otra parte, la calidad de los resultados –i.e en términos de tasas de reconocimiento o de error– usando los métodos de búsqueda bajo estudio son superiores a un simple método determinista.

De esta manera siendo varias las ventajas que ofrecen los métodos analizados a lo largo de esta tesis sobre otras opciones, es importante conocer sus características de desempeño, sus capacidades y sus debilidades. Así, este trabajo de investigación viene a llenar un hueco, desde un punto de vista experimental, del conocimiento comparativo entre ellos, además de contribuir en la propuesta de implementación en el campo de la SA.

1.2. Objetivo General

El objetivo de este trabajo de investigación es hacer un estudio comparativo de diversos métodos de Selección de Atributos basados en algoritmos no-deterministas que son la *Búsqueda Tabu*, *Algoritmos Genético*, *Recocido Simulado* y *Aceptación por Umbral*, en problemas de distinta naturaleza y complejidad -i.e. tipo de datos y tamaño del conjunto de datos.

1.2.1. Objetivos Específicos

1. Implementar varios algoritmos no-deterministas como motor de búsqueda de atributos en la SA.

2. Implementar la versión de *Aceptación por Umbral* para SA.
3. Realizar pruebas de desempeño de dichos algoritmos en datos de distinta naturaleza y dimensionalidad para tareas de clasificación.

1.2.2. Metas

1. Un estudio comparativo de algoritmos de SA.
2. Una implementación del Algoritmo de *Aceptación por Umbral* para SA.
3. Al menos dos publicaciones científicas.
4. La tesis de maestría

Capítulo 2

Marco Teórico

En este capítulo, se expondrán detalladamente los temas de relevancia para el desarrollo de esta investigación. Conceptos básicos, antecedentes de los algoritmos y la manera de implementarlos son aspectos explicados a lo largo de este capítulo. En los trabajos de comparación de desempeño, es de suma importancia determinar estadísticamente el potencial de un algoritmo sobre otro, por lo que se explicara a detalle la naturaleza del test estadístico utilizado para tal efecto.

2.1. Selección de Atributos

Como se explicó anteriormente, la SA es un campo dentro del pre-procesamiento de datos cuyo objetivo es reducir el número de variables que explican un problema –i.e. un conjunto de datos. Este proceso de reducción es guiado mediante alguna medida J que nos indica si esta reducción es positiva o negativa en términos de la capacidad de predictiva de estas variables con respecto a la variable dependiente.

En la literatura existen una gran cantidad de algoritmos de SA, los cuales pueden ser agrupados en dos grandes categorías:

1. El enfoque *wrapper*, el cual depende de un inductor (e.g. un algoritmo de clasificación) para determinar la habilidad discriminante de las variables independientes con respecto a la variable dependiente. Una de las desventajas que presenta este enfoque es su alto coste computacional, debido a que cada evaluación de variables requiere del entrenamiento de un algoritmo de aprendizaje.

2. El enfoque *filter*, donde el proceso está basado en los datos de manera independiente de un inductor (Liu, 1997). Este enfoque arroja como resultado una lista con valores numéricos asignado a cada variable, la cual indica el poder discriminativo de cada una de ellas, normalmente los valores mayores indican mejor relevancia de la variable.

Así, el proceso de selección de variables debe contener dos elementos principales:

1. La Función de Evaluación, que permite juzgar si un subconjunto es mejor (más relevante) que otro.

2. La Estrategia de Búsqueda, que decide como se siguen explorando nuevas soluciones.

En el espectro computacional de técnicas y algoritmos existen diversas funciones de evaluación y estrategias de búsqueda:

1. Funciones de evaluación: Descripción de conceptos mínimos (Almuallim & Dietterich, 1991), Información Mutua (Dan C. Marinescu, 2012), Conteo de Inconsistencias (Liu & Setiono, 1996), Separabilidad Interclase (Duda & Hart, 2001), entre otras.

2. Estrategias de búsqueda: Método de Branch y Bound (Fukunaga, 1990), Búsqueda Secuencial hacia Delante y hacia Atrás (Kittler, 1986), Búsqueda Flotante (Pudil *et al.*, 1994), etc.

La SA puede ser vista como un problema de búsqueda, donde cada estado en el espacio de búsqueda corresponde a un subcojunto de atributos. Si el problema de la SA es visto como una búsqueda abierta en el espacio de hipótesis –i.e. el conjunto de soluciones posibles– entonces el número de los subconjuntos potenciales a evaluar es exponencial, lo cual lo hace un problema prácticamente intratable (Oommen *et al.*, 2008).

De esta manera, se pueden distinguir dos tipos de estrategias de búsqueda en la SA.

1. Las deterministas, son aquellas que son completamente predictivas, de tal manera que siempre se producirá la misma salida, como por ejemplo el algoritmo de Búsqueda Secuencial Flotante hacia Adelante (SFFS por sus siglas en inglés) propuesta por Pudil *et al.* (1994); la clásica búsqueda hacia delante, búsqueda hacia atrás o el método de *Branch & bound*, etc. Este métods requieren la monoticidad de la evaluación inducida. Esto implica que cuando un atributo es agregado/removido al/del subconjunto actual, el valor de la función de evaluación no disminuye. En la mayoría de las aplicaciones prácticas este enfoque es computacionalmente impráctico, y la corriente principal de investigación en la SA ha sido dirigida a los métodos de búsqueda secuencial sub-óptimos. En particular, un algoritmo secuencial de SA posee un tiempo polinomial el cual es motivado por la definición de relevancia.

2. Las búsquedas no deterministas o estocásticas, basadas en algoritmos que poseen características estocásticas o probabilísticas. En éstos las configuraciones en cada etapa de la búsqueda de atributos que se encuentran en el subconjunto, son adheridos o removidos con algún criterio de tipo probabilístico.

El trabajo de investigación que se presenta en este documento se centra en los cuatro grandes algoritmos estocásticos de la literatura: Los *Algoritmos Genéticos (AG)*, la *Búsqueda Tabú (BT)*, el *Recocido Simulado (RS)* y la *Aceptación por Umbral (AU)*, que a continuación explicamos a detalle.

2.1.1. Algoritmos Genéticos

Los *Algoritmos Genéticos* fueron propuestos en 1975 por John Holland de la universidad de Michigan. Son métodos adaptativos que se utilizan para resolver problemas de búsqueda y optimización, en donde tratan de encontrar la mejor solución entre un conjunto de soluciones posibles y están basados en los procesos de evolución biológica (Stender, 2007).

Los *AG's* trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema determinado. A cada individuo

se le asigna un valor, relacionado con la efectividad de dicha solución. Cuanto mayor sea el valor de un individuo mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos descendientes de los anteriores, los cuales comparten algunas características de sus padres.

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y contiene una mayor proporción de *buenas* características. Así, a lo largo de las generaciones las buenas características se propagan a través de la población, favoreciendo el cruce de los individuos mejor adaptados. De esta manera van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el *Algoritmo Genético* ha sido bien diseñado, la población convergerá hacia una solución óptima del problema.

Los elemento básicos de un AG se listan a continuación:

Población inicial: Los algoritmos genéticos comienzan con un conjunto de k estados generados aleatoriamente, llamados "población". Cada estado o ïndividuo.[es]tá representado con una cadena sobre un alfabeto finito, siendo el más común una cadena de 1's y 0's.

Evaluación: Una función de idoneidad $f(x)$ que devuelve valores altos para estados mejores. En el contexto de SA, esta función es modelada con el desempeño de algún algoritmo de clasificación.

Selección: Cuando se hace la evaluación de la población inicial, se evalúan y se toman los mejores dos individuos con el objetivo de que se reproduzcan.

Criterio de paro: Si el cromosoma seleccionado es más alto que la eficiencia actual se toma, de lo contrario se continua cruzando y mutando hasta el criterio de paro, el cual si la eficiencia no es mejorada, se para y termina el algoritmo.

Operador de cruce: El método de çruce de un punto.[el] cual es el cruce más sencillo, consiste en seleccionar una posición aleatoria en las cadenas progenitoras y se intercambian los genes a la izquierda de esta posición.

Operador de mutación: La mutación se considera un operador básico, que proporciona un pequeño elemento de aleatoriedad en la vecindad de los individuos de la población. El objetivo del operador de mutación es producir nuevas soluciones a partir de la modificación de un cierto número de genes de una solución

existente, con la intención de fomentar la variabilidad dentro de la población. Este operador es definido por la siguiente fórmula (P.VenKataraman, 2002):

$$C = \begin{cases} x_i, i \neq k \\ x_k, i = k \end{cases} \tag{2.1}$$

Nueva Generación: Generar nueva población con los nuevos ïndividuos"que permiten realizar una exploración de toda la información almacenada hasta el momento en la población y combinarla para crear mejores individuos. Esta nueva generación esta dada por la siguiente expresión:

$$u_1 = \begin{cases} x_1, if, i = r \\ y_1, contrario \end{cases} \quad u_1 = \begin{cases} y_1, if, i = r \\ x_1, contrario \end{cases} \tag{2.2}$$

En la figura 2.1.1 se localiza el diagrama de flujo que se siguió para la elaboración del algoritmo.

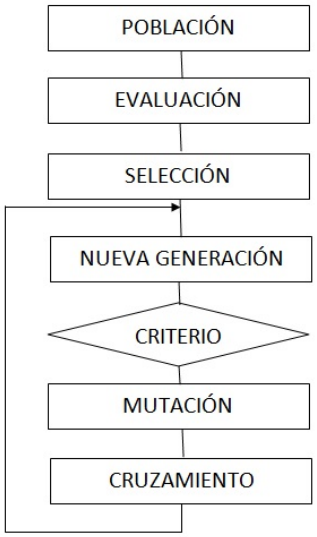

Figura 2.1: Diagrama de Flujo, Algoritmo Genético

Algoritmo 1 Pseudocódigo de un *Algoritmo Genético*

```
Generar una Población inicial
Evaluar
Selección
while Criterio de paro do
    Nueva Generación
    Criterio de paro
    Mutación
    Cruce
end while
Volver al paso 3 hasta el criterio de parada
```

2.1.2. Búsqueda Tabú

Los orígenes de la *Búsqueda Tabú* se remonta a finales de los años 70. Oficialmente el nombre y la metodología fueron introducidos por Fred Glover. Este algoritmo dota de inteligencia a los algoritmos de búsqueda local para que exploren el espacio de soluciones más allá del óptimo local. Toma de la Inteligencia Artificial el concepto de memoria con el objetivo de almacenar información de lo sucedido y así dirigir mejor la búsqueda. En este sentido puede decirse que hay cierto aprendizaje y que la búsqueda es inteligente. Este método permite moverse a una solución aunque no sea tan buena como la actual, de modo que se puede escapar de óptimos locales y continuar estratégicamente la búsqueda de soluciones aún mejores.

En general la BT tiene los siguientes elementos (Díaz, 2006):

1. Solución inicial: La búsqueda debe comenzar desde una solución que satisfaga las restricciones del problema, puede ser generada aleatoriamente o con funciones.

2. Movimiento: Es un procedimiento aleatorio o determinístico con el que se genera una solución aceptable a partir de la solución inicial.

3. Vecindad: Dada una solución S, la vecindad $N(S)$ es el conjunto de todas la soluciones aceptables que pueden ser generadas por la ejecución de un movimiento sobre la solución actual S.

4. Lista tabú: Es un mecanismo de memoria que trata de evitar que la búsqueda entre en un ciclo o quede atrapada en un óptimo local. Cuando un movimiento que genera una nueva solución es aceptado, se añade a la lista tabú para que no vuelva a ser elegido.

5. Criterio de parada: La búsqueda termina cuando se alcanza un número dado de iteraciones sin mejorar la solución o después de un tiempo predefinido.

El pseucódigo realizado es el siguiente (Zhang & Sun, 2002):

Algoritmo 2 Pseudocódigo de la *Búsqueda Tabú*

```
1: Elegir un candidato aleatorio del vecindario N(x).
2: Lista Tabu = Vacia
3: Mejor solución = 0
4: while No se cumpla el criterio de parada do
5:    Seleccionar la mejor solución no tabu del vecindario y almacenar en
      solución-actual
6:    if ((solución actual) es mejor que (mejor solución)) then
7:       mejor solución=solución actual
8:       Lista Tabu = Actualizar lista tabu
9:       if La Longitud excede el tamaño predefinido then
10:         Remover el primer vecino de la Lista Tabu
11:      end if
12:   end if
13: end while
14: Volver al paso 4 hasta encontrar el criterio de terminación
```

2.1.3. Recocido Simulado

El *Recocido Simulado (RS)* es una técnica estocástica inspirada en la mecánica estadística para encontrar la solución óptima global en grandes problemas de optimización combinatoria. El *RS* es un método que casi no necesita información acerca de la estructura del espacio de búsqueda. El algoritmo funciona asumiendo que alguna parte de la solución actual proviene de una potencialmente mejor y esa parte debería ser retenida mediante la exploración de los vecinos de la solución actual. Asumiendo que la función objetivo es minimizada. Entonces el *RS* puede saltar de colina en colina y por lo tanto escapar o simplemente evitar soluciones sub-óptimas.

Cuando un sistema S (considerado como un conjunto de estados posibles) está en equilibrio térmico a una temperatura dada T, la probabilidad de que se encuentra en un cierto estado S, llamado $P_T(s)$ depende de T y de la energía $E(s)$ del estado s. Esta probabilidad sigue una distribución de Boltzmann:

$$P_T(s) = \frac{exp_(- \frac{E(s)}{kT})}{Z}, with, Z = \sum_{s \epsilon S} exp_(- \frac{E(s)}{kT}) \tag{2.3}$$

donde k es la constante de Boltzmann y Z actua como un factor de normalización. Metrópolis y sus colaboradores desarrollaron un método de relajación estocástica que trabaja para simular el comportamiento de un sistema en una determinada temperatura T (Metropolis *et al.*, 1953). Siendo s el estado actual y s' un estado vecino, la probabilidad de hacer una transición de s a s' es la razón $P_T(s \leftarrow s')$ de la probabilidad de estar en s y la probabilidad de estar en s':

$$P_T(s \leftarrow s') = \frac{P_{T(s')}}{P_T(s)} = exp(-\frac{\Delta E}{kT}) \tag{2.4}$$

donde tenemos definido $\delta E = E(s') - E(s)$. Por lo tanto, la aceptación o rechazo de s' como el nuevo estado depende de la diferencia de las energías de ambos estados en T. Si $P_T(s') < P_T(s)$ entonces el movimiento es siempre aceptado. Si $P_T(s') < P_T(s)$ entonces se acepta con probabilidad $P_T(s, s') < 1$ (esta situación corresponde a una transición a un estado de mayor energía).

Es importante tener en cuenta que esta probabilidad depende de como aumenta y disminuye con la temperatura T. Al final, habrá una T suficiente baja (el punto de congelación), en el que estas transiciones serán muy improbables, y el sistema se considerará congelado. En fin de maximizar la probabilidad de encontrar estados de mínima energía en cada T, el equilibrio térmico debe ser alcanzado. Para ello según Metropolis *et al.* (1953), hay un programa de recocido, diseñado para evitar que el proceso quedara atrapado en un mínimo local.

El algoritmo propuesto por Kirkpatrick (1984) consiste en usar la idea de Metropolis en cada T para una cantidad finita de tiempo. En este algoritmo la T se fija en un valor inicialmente alto, pasando el tiempo suficiente para llegar al equilibrio térmico. Entonces un pequeña reducción en la T se lleva a cabo y el proceso se repite hasta que el sistema es considerado congelado.

Si el enfriamiento está bien diseñado, el estado final alcanzado puede ser considerado una solución casi óptima. Sin embargo, el proceso entero es inherentemente lento. Principalmente por el requisito del equilibrio térmico en cada T.

Hay cuatro aspectos asociados con la temperatura:

1. La temperatura inicial. Si esta es suficientemente alta, entonces casi cualquier parte del espacio de búsqueda puede ser visitado desde el principio. Debe ser suficientemente largo para hacer movimientos cuesta arriba y cuesta abajo cerca del mismo.

2. El programa de recocido (o velocidad de enfriamiento). La temperatura debe disminuir tan cerca al cero al final del tiempo asignado.

3. El tiempo dedicado en cada temperatura, es decir, el número de aceptaciones permitidos en la temperatura dada antes de un nuevo enfriamiento se lleva acabo. Si es demasiado alta, entonces la convergencia es muy lenta, y si es demasiado baja, entonces el espacio de la solución es mal examinada y el proceso de la relajación global puede fallar.

4. La temperatura final.

El pseudocódigo utilizado para RS es el siguiente:

Algoritmo 3 Pseudocódigo de *Recocido Simulado*

1: Seleccionar una solución inicial.
2: Elegir una Temperatura inicial, $T > 0$.
3: **while** No se cumpla el criterio de parada **do**
4: Buscar una solución del entorno y evaluar $\triangle E$(incremento de la función objetivo).
5: **if** $f(j) \leq f(i)$ entonces $i := j$ **then**
6: Aceptar el movimiento.
7: **end if**
8: **if** $exp(f(i) - f(j))/T >$ número aleatorio en $[0, 1)$ entonces $i := j$ **then**
9: Aceptar el movimiento
10: Después de no conseguir ninguna mejora durante un tiempo, o tras un número de iteraciones.
11: **else**
12: **if** No hay mejora o k iteraciones **then**
13: Reducir T
14: **end if**
15: **end if**
16: **end while**
17: Volver al paso 3 hasta encontrar el criterio de terminación

2.1.4. Aceptación por Umbral

El algoritmo de *Aceptación por Umbral* fue propuesto por Deuck y Scheuer en 1990, e independientemente por Moscato & Fontanari (1990). Este algoritmo es una variante del algoritmo de *Recocido Simulado* cuyo principio de operación se puede enunciar como sigue (Duarte, 1993):

Dada una solución inicial, todos los movimientos que produzcan una mejora en la función objetivo son aceptados. En cambio, los movimientos que empeoran la función objetivo son aceptados, si esta pérdida de calidad es menor que un umbral dado. Con esto se logra salir de óptimos locales y encontrar soluciones de mayor calidad.

Una de las diferencias respecto al *Recocido Simulado* es que la función de aceptación de movimientos es determinista y no aleatoria. *AU* es un método iterativo en el que se fija un umbral determinado en cada iteración. Generalmente, el umbral empieza en valores elevados para, posteriormente decrecer de forma monótona hasta que se hace cero. El umbral determina en cada iteración el nivel

máximo de pérdida de calidad que se le permite a la solución. Para cada iteración se hace una búsqueda local exhaustiva de la vecindad. El pseudocódigo es muy parecido al de *RS*.

El esquema básico que gobierna al algoritmo de *Aceptación por Umbral* es el siguiente (R. & Yepes, -):

Algoritmo 4 Pseudocódigo de *Aceptación por Umbral*

1: Elegir un umbral inicial, $U > 0$
2: Buscar una solución del entorno y evaluar ΔE(incremento de la función objetivo)
3: **if** $\Delta E > -U$ **then**
4: Aceptar el movimiento evaluando $P_k(\Delta E)$:
5: **end if**
6: **if** No hay mejora o k iteraciones **then**
7: Reducir U
8: **end if**
9: Volver al paso 3 hasta encontrar el criterio de terminación

En el contexto de la metaheurística la función de aceptación del movimiento está dada por la siguiente ecuación:

$$P_k(\Delta E) = \begin{cases} 1 & si \quad \Delta E \leq Th_k \\ 0 & en \quad caso \quad contrario \end{cases}$$

donde ΔE es el cambio de la función objetivo y Th_k la función que determina el umbral en la iteración k.

Por otra parte es de fácil implementación y su rápida desempeño reduce considerablemente el esfuerzo computacional. Algunos de los trabajos más conocidos sobre AU son Lin *et al.* (1995), Scheermesse & Bryngdahl (1995) y Nissen & Paul (1995).

Al momento de escribir este documento de tesis, no se ha encontrado en la literatura científica una implementación propia de la AU como estrategia de búsqueda en la SA.

2.2. Método de Validación

En los procesos de generación de modelos, ya sea entrenamiento de algoritmos de clasificación o selección de subconjuntos de atributos, es importante proveer los resultados de una forma objetiva y libre de sesgos. Es sabido que una leve variación en un conjunto de datos –e.g. eliminar datos o variables– arrojará, en términos de clasificación o generación de subconjuntos de variables, diferentes resultados. Es por ello que se deberán aplicar métodos para eliminar estas variaciones y ofrecer lecturas en lo que pudiera ser un *promedio* del resultado con variaciones aceptables. Existen en la literatura diversas formas de proceder en este sentido. Los métodos de remuestreo como Bootstrap (Tibshirani *et al.*, 2002), los Métodos de Montecarlo (Daniel, 2001) y la Validación Cruzada (Trevor Hastie & Fridman, 2008)son los más conocidos. Este último fue el seleccionado para validar los resultados experimentales y que se explica en la siguiente sección.

2.2.1. Validación Cruzada

La validación cruzada o cross-validation (como se conoce en inglés) es una técnica utilizada para evaluar el desempeño de un determinado parámetro objetivo y garantizar su independiencia de la partición entre datos de entrenamiento y prueba. Este método consiste en repetir y calcular la medida de desempeño promedio, que en nuestro caso es la media aritmética de la tasa de reconocimientos de un clasificador, obtenida de las medidas de evaluación sobre diferentes particiones, repitiendo este proceso un número de veces especificado. En la figura 2.2 se esquematiza este proceso.

El trabajo de investigación fue inicialmente desarrollado usando la versión de validación 10x10, es decir, 10 veces 10 particiones de datos. Sin embargo su coste computacional resultó demasiado elevado, así que se redujo a 5X5.

2.3. Algoritmos de Clasificación

En esta sección se explicarán los algoritmos de clasificación utilizados en la SA en la modalidad *Wrapper*.

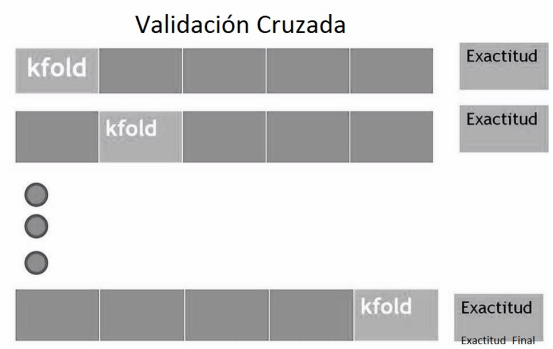

Figura 2.2: Esquema k-fold cross validation, con n particiones y un solo clasificador, repitiendo este mismo proceso n veces y tomando su media final

2.3.1. Vecinos Cercanos

Vecinos cercanos o *KNN* (por sus siglas en inglés) es un método de clasificación supervisada el cual está basado en casos o instancias. Este método supone que los vecinos más cercanos comparten una *buena* similitud entre ellos y el caso de prueba. Consiste en comparar la nueva instancia a clasificar con los k casos más cercanos dada una medida de distancia. El nuevo caso se ubicará en la clase mayoritaria de estos k vecinos.

La métrica o medida de distancia más utilizada es la distancia euclidiana (Pang-Ning Tan, 2006). Sea $D(x, y)$ la distancia entre dos puntos, la distancia euclidiana se define como:

$$D(x, y) = \sqrt{\sum_{k=1}^{n}(x_k - y_k)^2} \tag{2.5}$$

donde n es el número de dimensiones y x_k y y_k son respectivamente los atributos.

2.3.2. Naïve Bayes

El clasificador de *Naïve Bayes (NB)* es un algoritmo que basa su funcionamiento en el ampliamente conocido *Teorema de Bayes* (Pang-Ning Tan, 2006). Éste nos permite expresar la probabilidad *a posteriori* en términos de la probabilidad *a priori*, la probabilidad condicionada por la clase$P(X|Y)$ y la evidencia

$P(X)$:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{2.6}$$

NB asume que la presencia o ausencia de una característica particular no está relacionada con la presencia o ausencia de cualquier otra característica, teniendo en cuenta la variable de clase; esto significa que se da por hecho que existe una independencia entre todas las variables que componen o definen los casos en particular. Una ventaja del *Naïve Bayes* es que sólo se requiere una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios para la clasificación.

De esta manera, el cálculo de las probabilidades condicionadas por la clase pueden expresarse de la siguiente manera para datos de tipo discretos:

$$P(X|Y = y) = \prod_{i=1}^{d} P(X_i|Y = y) \tag{2.7}$$

donde cada caso esta formado d-atributos o variables $X = x_1, x_2...x_n$.

2.3.3. Análisis discriminante Lineal y Cuadrático

El *Análisis discriminante lineal y cuádratico* o *LDA* y *QDA* por sus siglas en inglés son algorítmos que constituyen reglas discriminantes desde la base del enfoque bayesiano o regla de bayes. En el caso de *LDA* donde un conjunto de patrones en un espacio de características de n-dimensión, pertenecientes a dos clases, es linealmente separable si y solo si el espacio se puede dividir mediante un hiperplano en dos regiones, de forma que cada región contenga patrones de una única clase. Estas funciones discriminantes están dadas por la siguiente expresión:

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T + \log \pi_k \tag{2.8}$$

donde σ es la covarianza de los datos, μ_k es el vector de medias y π_k son las probabilidades de clase.

En la figura 2.3 se representa un hiperplano generado por un *LDA*:

El *Discriminante Cuadrático (QDA)* es similar al *Discriminante Lineal (LDA)*, con la diferencia de que el hiperplano generado es una superficie de naturaleza

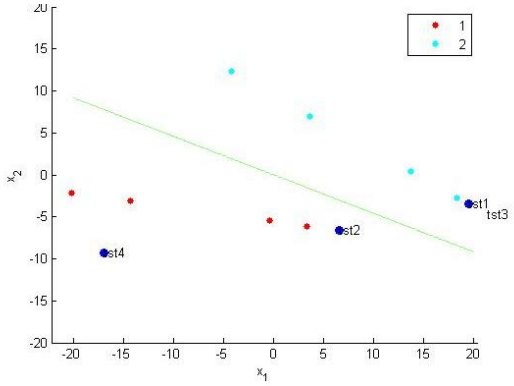

Figura 2.3: Gráfica de un *Discriminante lineal*

cuadrática(Clarke *et al.*, 1979). La función discriminante generada esta dada por la siguiente expresión:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) + \log\pi_k \tag{2.9}$$

En la figura 2.4 se ejemplifica el hiperplano generado por un *QDA*:

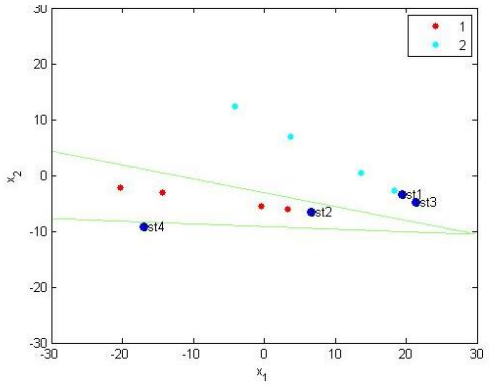

Figura 2.4: Gráfica de un *Discriminante cuadrática*

2.3.4. Máquina de Vectores de Soporte

La *Máquina de Vectores de Soporte* o *SVM* por sus siglas en inglés, es una técnica de clasificación que ha tenido considerable atención en los últimos años. Tiene como origen el aprendizaje estadístico y demuestra un enorme potencial en

muchas aplicaciones de distintas áreas de la ingeniería (Pang-Ning Tan, 2006). Este método es capaz de trabajar con problemas de clasificación y regresión.

La **SVM** construye un hiperplano de *Máximo Margen*, es decir, encuentra un hiplerplano en donde existe una máxima distancia entre determinados puntos de entrenamiento y éste. Estos puntos son los llamados *vectores de soporte*. En la figura 2.5 se ejemplifica en un modelo de dos dimensiones las clases separadas por este hiperplano.

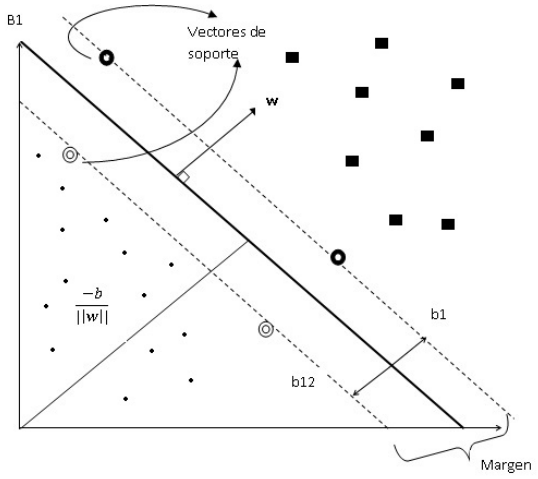

Figura 2.5: Margen *SVM* (hiperplano)

Para encontrar este tipo de regiones, la *SVM* utiliza un operador matemático llamado *Kernel*, el cual tiene como función principal *trasladar* los datos a una dimensión mucho mayor que la actual, a fin de encontrar estos hiperplanos que separen de mejor manera las clases. Los tipos de funciones *Kernel* que se utilizaron los siguientes:

$$Kernel \;\; Lineal = X_i - X_j^T \tag{2.10}$$

$$Kernel \;\; de \;\; Base \;\; Radial \;\; RBF = exp - \frac{\|x-y\|^2}{2\sigma^2} \tag{2.11}$$

Capítulo 3

Materiales y Métodos

En este capítulo se explicará la metodología seguida para desarrollar esta investigación. Se detallarán los datos utilizados en los experimentos, además del hardware y software utilizados como herramientas para la implementación de algoritmos y pruebas.

3.1. Metodología

La metodología utilizada es el *Ciclo del Sistema del Reconocimiento de Patrones* propuesta por Duda & Hart (2001). Mediante algoritmos matemáticos se pretende que las máquinas sean capaces de emular las tareas de reconocimiento que rutinariamente hacen los seres humanos. La metodología consta de cinco grandes etapas –ver Figura 3.1:

Figura 3.1: Metodolgía (Duda & Hart, 2001)

1. Colección de los datos: En la sección de descripción de datos se habla específicamente de ellos –ver sección 3.3.

2. Selección de atributos: Una de las etapas de mayor importancia es la selección de atributos. Esta consiste en la reducción o eliminación de las variables que representan el sistema bajo estudio. Esta reducción del problema, en términos de variables, ayuda de manera significativa al proceso de reconocimiento de patrones, al facilitar la construcción de algoritmos inteligentes de menor tamaño, reducir la carga computacional o requerimientos de potencia de cálculos, y la oportunidad de ofrecer modelos finales más sencillos para explicar a un usuario final. La forma de seleccionar/eliminar variables puede hacerse mediante algoritmos especializados y cuyo espectro de posibilidades y opciones es basto. Actualmente, aquellos basados en métodos de optimización ofrecen una potencia de búsqueda muy superior a los basados en métodos tradicionales. Así, esta etapa representa el punto central de esta tesis. Este trabajo de investigación emplea algoritmos estocásticos, ya descritos, para *buscar* los mejores candidatos –ver sección 2.1.

3. Elección del modelo: Es importante seleccionar aquellos algoritmos que mejor se adapten a la situación experimental. De esta manera, se seleccionaron algoritmos que posean el mínimo número de parámetros de ajuste, en en el mejor de los casos, sin parámetros. Los algoritmos seleccionados fueron explicados en la sección 2.3.

4. Entrenamiento de Clasificadores: Esta etapa consta del entrenamiento de clasificadores. Para este propósito, se utilizó el método de validación cruzada en su modalidad de 5x5 –ver sección 2.2.1.

5. Evaluación de clasificadores: Con el propósito de dar certeza estadística a la capacidad de generalización del modelo seleccionado es importante evaluar su rendimiento. Para ello utilizó la prueba de Friedman seguido de un *post-hoc test*. El test de Friedman Demsar (2006) consiste en analizar k tratamientos en el que la hipótesis nula que se contrasta es que las respuestas asociadas a cada uno de estos tratamientos tienen la misma distribución o distribuciones con la misma media. Se utilizó como umbral de aceptación/rechazo de la H_O igual a $p \leq 0{,}05$.

3.2. Hardware y Software utilizados

Para los experimentos preliminares se utilizó el asistente matemático Matlab de la compañía Mathworks en su versión R2012a, corriendo sobre Microsoft Windows 7 Professional. Posteriormente, una vez verificada la correcta operacionalidad de los algoritmos implementados, se procedió a realizar los experimentos finales bajo Matlab en su versión 2012a en ambiente linux. Esta última etapa fue llevada a cabo en un clúster de computación con 24 procesadores Intel (R) CPU x565 @2.679Gz y un sistema de memoria de 94 Gb. Los algoritmos fueron ejecutados de forma paralela sobre 12 procesadores.

3.3. Descripción de los datos

Fue objetivo de esta investigación el tratar de cubrir todos los escenarios posibles para la prueba de algoritmos y su posterior comparación. Esto es, se integró una batería de datos con diversas características, por ejemplo, de gran dimensionalidad y gran cantidad de casos, poca dimensionalidad y pocos datos, gran desbalance entre dimensión vs. casos, datos de naturaleza real y artificial, etc.

- Del sitio web *Feature Selection Challenge* (NIPS), la cual contiene una de las bases de datos más usadas en el campo de la computación, se tomó *Madelon* que contiene 2000 muestras y 500 atributos;
- Del campo de expresión genética se utilizaron los conjuntos de datos públicos de *Cáncer de colon* que consta de 62 muestras y 2000 atributos, el *Breast Cancer* que tiene 78 muestras y 24481 variables, y el de *Cáncer de prostata* con 136 muestras y 12,600 atributos.
- De la sitio WEB de la UCI of California at Irvine se uso la base de datos de Breast Cancer (Wisconsin) que contiene 569 muestras y 30 atributos.
- Se crearon bases de datos artificiales usando el generador web encontrado en Torres *et al.* (2014). Cinco conjuntos de datos de 200 muestras por 50 atributos fueron generados con las siguientes características: *Data1* con los

Bases de Datos

Name	Muestras	Atributos	Origen	Tipo
Breast Cancer W.	569	30	Real	Continuos
Madelon	2000	500	Real	Denso
Cáncer de Colon	62	2,000	Microarreglos	Continuos
Leukemia	38	7,129	Microarreglos	Continuos
Breast Cancer	78	100	Microarreglos	Continuos
Prostata	136	12,600	Microarreglos	Discretos
Data1	200	50	Artificial	Continuos
Data2	200	50	Artificial	Continuos
Data3	200	50	Artificial	Continuos
Data4	200	50	Artificial	Continuos
Data5	200	50	Artificial	Continuos

Tabla 3.1: Tabla de bases de datos

primeros cinco atributos relevantes y los 45 son atributos irrelevantes; *Data2* tiene atributos 1,10,30,40 como relevantes y los demás son irrelevantes; *Data3* los atributos relevantes están acomodados en los atributos 46-50 y los irrelevantes de 1-45; *Data4* y *Data5* contienen 25 atributos relevantes y 25 irelevantes, cambiando entre ellas las relevantes en posiciones pares y en otra base de datos las relevantes en posiciones impares. A las bases de datos de *Mandelon*, *Leukemia*, *Cáncer de Prostata* y *Breast Cancer* se les hizo una preselección utilizando el filtro de Score de Fisher, quedando cada uno de ellos con 100 atributos. Es importante mencionar que todas las bases de datos contienen dos clases. Un resumen de la geometría de las bases de datos se muestra en la tabla 3.1.

Capítulo 4

Resultados y Discusión

4.1. Desempeño de los algoritmos de SA

Análisis por algoritmo de clasificación

En la tabla 4.1 se muestra el desempeño –i.e. tasa de reconocimientos– de los clasificadores con una validación cruzada de 5X5. En ésta se puede observar que el algoritmo de *AU* tiene los mejores promedios en siete de los nueve algoritmos de clasificación implementados, *1NN*, *3NN*, *5NN*, *7NN*, *NB*, *SVM-R* y *SVM-L*.

En un segundo lugar se ubica consistentemente el algoritmo de *BT*, también en siete de los nueve algoritmos de clasificación. Los dos restantes, *LDA* y *QDA* tienen mejor tasa de reconocimientos usando atributos encontrados por *BT*.

Examinando la contraparte de los resultados, es decir, aquellos algoritmos de SA que ofrecen un pobre desempeño se encuentra el algoritmo *AG*. En 10 de los 11 clasificadores, los subconjuntos de atributos seleccionados por *AG* arrojan las más bajas tasas de reconocimientos promedio, con excepción de *LDA* usando los subconjuntos generados por *RS*. Esto puede ser ilustrado con el clasificador *SVM-L* y el conjunto de datos *Madelon*. Mientras que *AU* alcanza un 91.04 %, *AG* ofrece un reducido 53.14 % de tasa de reconocimiento.

Análisis por conjunto de datos

Analizando los resultados con respecto a que algoritmo de SA muestra un mejor papel en los conjuntos de datos, el algoritmo *AU* arroja el mejor resultado en nueve de los 11 conjuntos con tasas de reconocimiento por arriba del 90 %, a

excepción de *BCW* y *D5*. De igual manera, *BT* también se posiciona en segundo lugar de desempeño en nueve de los 11 conjuntos. Es meritorio resaltar que en el conjunto de *Leukemia*, *AU* ofrece un 100 % de exactitud.

De la misma manera, los peores resultados son generados por *AG* en *Colon*, *Madelon*, *BC*, *Próstata* y *Leukemia*. En los conjuntos artificiales, en todos ellos *AG* es ubicado en el último lugar.

En resumen, las mejores combinaciones (indicadas con la abreviación del algoritmo de SA más el algoritmo de clasificación) para cada una de las bases de datos analizadas son indicadas en la tabla 4.1.

	1NN				3NN				5NN			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	91.05	98.19	96.10	95.43	94.95	96.76	95.43	94.10	91.71	96.38	98.19	92.48
BCW	94.34	93.18	94.52	91.74	94.87	93.22	93.07	92.80	95.04	93.36	93.53	93.46
Madelon	70.20	88.54	88.12	74.72	74.86	90.42	89.52	76.84	79.28	90.56	90.10	79.74
BC	72.87	**79.37**	72.93	74.88	68.65	76.07	74.32	68.77	67.17	77.42	75.48	67.23
Prostata	82.66	91.92	90.33	87.22	83.63	**91.93**	91.05	90.30	85.59	89.84	90.15	82.39
Leukemia	96.36	**100.00**	99.43	97.36	93.71	97.93	97.43	91.64	89.64	95.71	95.00	96.29
Data1	77.80	89.80	84.00	87.50	78.80	**92.00**	87.50	89.90	75.60	91.60	86.30	87.10
Data2	74.00	91.00	87.80	90.80	77.60	92.40	89.00	88.80	79.00	93.60	88.00	92.60
Data3	73.20	89.50	87.80	88.00	79.30	91.10	85.60	91.30	75.90	91.20	88.00	89.40
Data4	83.00	83.00	82.00	77.80	82.00	84.50	86.60	75.90	83.40	86.60	88.10	75.10
Data5	78.90	84.90	81.00	77.00	77.80	88.10	86.80	74.00	81.70	86.40	87.00	73.10
Promedio	81.31	**89.94**	87.64	85.68	82.38	**90.40**	88.76	84.94	82.18	**90.24**	89.08	84.44

	7NN				LDA				QDA			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	93.62	93.33	98.19	93.62	92.48	97.43	93.52	91.33	62.68	63.84	63.46	59.34
BCW	95.01	93.99	93.81	91.14	97.08	96.87	97.51	97.01	**98.03**	97.50	97.96	94.97
Madelon	78.60	90.40	89.44	67.20	79.28	90.56	90.10	79.74	58.14	73.38	73.02	59.46
BC	62.47	75.77	72.25	72.93	64.20	59.80	67.35	57.03	53.12	62.57	60.28	56.33
Prostata	84.46	91.33	89.70	84.56	83.41	85.56	87.23	82.84	58.85	69.99	69.99	68.51
Leukemia	91.64	92.29	94.29	89.21	**100.00**	**100.00**	97.50	93.36	97.43	**100.00**	97.50	87.43
Data1	79.00	91.80	84.70	**92.00**	80.20	81.50	82.50	78.10	76.80	88.80	88.00	87.30
Data2	79.40	92.40	85.60	81.20	81.80	84.80	83.60	81.80	83.20	90.80	90.80	87.80
Data3	76.50	**91.60**	84.80	92.50	80.80	81.70	81.70	78.00	80.60	88.70	88.50	86.60
Data4	83.70	87.70	87.00	73.70	91.00	86.90	**91.30**	85.70	87.90	89.60	89.50	78.10
Data5	83.50	88.60	87.80	78.80	89.70	88.30	90.50	85.50	83.60	90.50	88.80	77.70
Promedio	82.54	**89.93**	87.96	83.35	85.45	86.67	**87.53**	82.76	79.33	87.90	**89.60**	83.86

	NB				SVM R				SVM L			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	66.33	96.15	94.59	87.74	62.68	63.84	63.46	59.34	62.86	**98.76**	96.86	93.33
BCW	96.80	97.12	96.66	94.34	**98.03**	97.50	97.96	94.97	97.22	97.22	97.01	96.03
Madelon	62.68	63.84	63.46	59.34	58.14	73.38	73.02	59.46	53.14	**91.04**	90.92	72.58
BC	50.00	60.32	57.98	59.75	53.12	62.57	60.28	56.33	63.53	65.42	68.77	60.78
Prostata	59.13	69.88	68.68	67.79	58.85	69.99	69.99	68.51	81.92	88.10	90.74	78.69
Leukemia	95.00	**100.00**	97.50	97.00	97.43	**100.00**	97.50	87.43	71.07	**100.00**	**100.00**	98.36
Data1	86.20	89.20	88.50	88.10	76.80	88.80	88.00	87.30	71.10	91.60	91.20	88.20
Data2	87.60	89.20	87.60	83.80	83.20	90.80	90.80	87.80	68.00	**93.80**	92.40	88.00
Data3	87.40	88.80	86.60	88.40	80.60	88.70	88.50	86.60	69.00	91.50	89.10	87.70
Data4	90.40	90.10	91.10	84.20	87.90	89.60	89.50	78.10	62.00	89.70	86.10	83.10
Data5	**91.40**	90.60	89.90	85.30	83.60	90.50	88.80	77.70	51.50	89.70	84.00	81.40
Promedio	79.36	**85.02**	83.87	81.43	76.40	**83.24**	82.53	76.69	68.30	**90.62**	89.74	84.38

Tabla 4.1: Desempeño de los Clasificadores 5X5 CV

Los algoritmos de SA con mayor estabilidad en su desempeño –i.e. menor desviación estándar en las lecturas–, se encuentran en *AU* y *BT* con $s < 1$. En el

Conjunto	Combinación	Tasa de reconocimiento
Colon	AU+SVM-L	98.76 %
BCW	AG+(QDA,SVM-R)	98.03 %
Madelon	AU+SVM-L	91.04 %
BC	AU+1NN	79.37 %
Próstata	AU+3NN	91.93 %
Leukemia	AU+(1NN,LDA,QDA,NB,SVM-L,SVM-R) AG+LDA, BT+SVM-L	100 %
Data1	AU+3NN, RS+7NN	92 %
Data2	AU+SVM-L	93.80 %
Data3	AU+7NN	91.60 %
Data4	BT+LDA	91.30 %
Data5	AG+NB	91.40 %

Tabla 4.2: Mejores combinaciones *Algoritmo+Clasificador*

caso de *RS*, en solo un clasificador ha presentado valores similares, al igual que *AG*. En el caso contrario, los datos que presentan mayor variabilidad son *BC* con el algoritmo de *AG*

4.2. Análisis estadístico del desempeño

Los resultados de la prueba de Friedman –ver Figura 4.3– indican que el algoritmo de *AU* supera por mucho al resto de algoritmos, a pesar de que estos son de mayor complejidad en su accionar, y sin importar el tamaño de las bases de datos e incluso su naturaleza. En siete de los nueve clasificadores, el algoritmo de *AU* ha obtenido el rango más pequeño, por lo que se le considera el algoritmo con mejor desempeño, seguido por *BT*. El análisis de *Post Hoc* en la tabla de la derecha muestra que no hay diferencias significativas entre AU y BT para $\alpha < 0{,}05$; sin embargo hay que tener en cuenta que la complejidad del primero resulta ser significativamente menor que el segundo.

4.3. Tiempos de Procesamiento

En la tabla 4.3 se muestra el tiempo de procesamiento por cada experimento. El algoritmo de *AU* ha mostrado rendimientos sólidos, es decir, si bien no es el más rápido, la demanda computacional es relativamente aceptable.

En *AU*, el tiempo promedio de procesamiento para la mayoría de los clasificadores es de 0.43 hrs. A diferencia de *RS* con un coste computacional es un poco más alto.

Algoritmos	Posición
AG	3.5
AU	**1.23**
BT	2.36
RS	2.91

1NN

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
AG	4.129	0.00004	0.017	0.051
RS	3.055	0.002	0.025	0.051
BT	2.064	0.039	0.05	0.05

Algoritmos	Posición
AG	3.3636
AU	**1.2727**
BT	2.1818
AG	3.1818

3NN

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
AG	3.798	0.0002	0.017	0.047
RS	3.468	0.001	0.025	0.047
BT	1.651	0.099	0.05	0.05

Algoritmos	Posición
AG	3.46
AU	**1.73**
BT	2
RS	2.82

5NN

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
AG	3.138	0.002	0.017	0.02
RS	1.981	0.048	0.025	0.02
BT	0.495	0.620	0.05	0.05

Algoritmos	Posición
AG	3.23
AU	**1.64**
BT	2.18
RS	2.95

7NN

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
AG	2.890	0.004	0.017	0.036
RS	2.395	0.017	0.025	0.036
BT	0.990	0.322	0.05	0.05

Algoritmos	Posición
AG	2.64
AU	2.09
BT	**1.5**
RS	3.77

LDA

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
RS	4.128	0.00004	0.016	0.038
AG	2.064	0.039	0.025	0.038
AU	1.073	0.283	0.05	0.05

Algoritmos	Posición
AG	3.36
AU	**1.27**
BT	1.90
RS	3.46

QDA

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
RS	3.963	0.0001	0.017	0.04
AG	3.798	0.0002	0.025	0.04
BT	1.156	0.248	0.05	0.05

Algoritmos	Posición
AG	3.05
AU	**1.27**
BT	2.41
RS	3.27

NB

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
RS	3.633	0.0003	0.017	0.051
AG	3.220	0.001	0.025	0.051
BT	2.064	0.039	0.05	0.05

Algoritmos	Posición
AG	2.64
AU	2.09
BT	**1.5**
RS	3.77

SVM Lineal

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
RS	4.129	0.00004	0.017	0.038
AG	2.064	0.039	0.025	0.0378
AU	1.073	0.283	0.05	0.05

Algoritmos	Posición
AG	3.59
AU	**1.2727**
BT	1.86
RS	3.27

SVM RBF

algoritmos	$z = \frac{(R_0 - R_i)}{SE}$	p	Holm	Li
AG	4.211	0.00003	0.017	0.038
RS	3.633	0.0003	0.025	0.038
BT	1.073	0.283	0.05	0.05

Tabla 4.3: Ranking promedio de los algoritmos y la comparación *Post Hoc* con $\alpha = 0{,}05$ para cada clasificador contra el mejor (la señalada en negrita en la tabla de la izquierda).

Los tiempos computacionales más altos se han observado usando el clasificador de *SVM-L*, mientras que los tiempos más cortos han sido con *NB*. Para todos los

clasificadores se ha visto este patrón de comportamiento. Por otra parte se puede apreciar que el algoritmo de *BT* ofrece los tiempos más cortos.

Para la base de datos *Madelon*, y a pesar de que se utilizó el Score de Fisher para reducir la dimensionalidad previa, los algoritmos asociados al *SVM* ofrecen tiempos de procesamiento de alrededor de 30 hrs. en *AU*.

	1NN				3NN				5NN			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	0,29	2,63	0,19	1,39	0,24	2,58	0,28	2	0,45	2,64	0,17	2,67
BCW	0,01	0,05	0,01	0,06	0,01	0,05	0,01	0,1	0,01	0,05	0,004	0,2
Madelon	2,05	1	0,13	0,6	3,24	1,05	0,12	0,5	2,89	1,09	0,15	0,76
BC	0,06	0,13	0,01	0,07	0,05	0,13	0,02	0,07	0,03	0,13	0,01	0,07
Prostata	0,02	0,14	0,01	0,07	0,02	0,14	0,01	0,07	0,03	0,14	0,01	0,1
Leukemia	0,03	0,38	0,01	0,13	0,08	0,38	0,02	0,22	0,04	0,38	0,02	0,14
Data1	0,02	0,07	0,003	0,04	0,01	0,07	0,004	0,05	0,01	0,07	0,003	0,05
Data2	0,01	0,07	0,003	0,04	0,01	0,07	0,25	0,04	0,01	0,07	0,003	0,04
Data3	0,01	0,07	0,004	0,04	0,02	0,07	0,003	0,1	0,03	0,07	0,003	2,23
Data4	0,02	0,07	0,01	0,04	0,02	0,07	0,01	0,04	0,02	0,07	0,01	0,04
Data5	0,02	0,07	0,01	0,03	0,01	0,07	0,01	0,04	0,02	0,07	0,01	0,04
Promedio	0.24	0.43	**0.05**	0.24	0.34	0.43	**0.08**	0.29	0.32	0.43	**0.05**	0.58

	7NN				LDA				QDA			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	0,53	2,56	0,26	2,03	3,56	4	0,49	2,99	1,06	4,95	0,54	3,16
BCW	0,01	0,05	0,003	0,08	0,02	0,07	0,01	0,61	0,02	0,09	0,36	0,74
Madelon	1,24	1,1	0,01	3,43	2,36	1,07	0,08	3,67	1,99	1,36	0,12	2,98
BC	0,03	0,13	0,01	0,07	0,03	0,2	0,02	0,15	0,03	0,26	0,01	0,16
Prostata	0,02	0,14	0,01	0,12	0,03	0,21	0,03	0,11	0,04	0,25	0,01	0,14
Leukemia	0,05	0,37	0,02	0,21	0,06	0,59	0,03	0,40	0,14	0,74	0,05	0,51
Data1	0,02	0,07	0,003	0,04	0,02	0,11	0,01	0,12	0,02	0,13	0,01	0,12
Data2	0,02	4,32	0,21	0,04	0,01	0,1	0,36	0,07	0,04	0,13	0,01	0,07
Data3	0,02	0,07	0,003	0,04	0,01	0,1	0,01	0,07	0,02	0,13	0,01	0,21
Data4	0,01	0,07	0,01	0,04	0,03	0,1	0,01	0,12	0,07	0,13	0,01	0,12
Data5	0,02	0,08	0,01	0,05	0,03	0,11	0,01	0,07	0,04	0,13	0,01	0,0
Promedio	0.18	0.81	**0.07**	0.56	0.56	0.61	**0.10**	0.76	0.32	0.75	**0.10**	0.75

	NB				SVM R				SVM L			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	0,64	0,72	0,1	0,45	0,35	2,23	0,15	1,31	0,72	5,26	0,58	3,6
BCW	0,003	0,01	0,001	0,05	0,06	0,14	0,01	0,7	0,07	0,2	0,02	0,68
Madelon	0,25	0,22	0,02	0,6	2,99	30,46	4,84	19,52	2,99	30,46	4,84	19,52
BC	0,01	0,04	0,004	0,02	0,02	0,11	0,01	0,07	0,03	0,41	0,04	0,5
Prostata	0,01	0,04	0,003	0,02	0,05	0,19	0,02	0,25	0,02	0,14	0,02	0,2
Leukemia	0,01	0,06	0,01	0,03	0,04	0,23	0,02	0,38	0,02	0,14	0,02	0,20
Data1	0,01	0,02	0,001	0,03	0,02	0,13	0,01	0,1	0,67	0,34	0,05	0,92
Data2	0,004	0,03	0,002	0,02	0,03	0,16	0,01	0,08	0,03	0,32	0,01	0,29
Data3	0,01	0,02	0,001	0,04	0,02	0,13	0,01	0,15	0,02	0,38	0,01	0,15
Data4	0,01	0,02	0,002	0,01	0,06	0,17	0,01	0,08	0,66	0,64	0,11	1,01
Data5	0,01	0,02	0,001	0,01	0,02	0,16	0,01	0,08	0,28	0,42	0,09	1,08
Promedio	0.24	0.43	**0.05**	0.24	0.34	0.43	**0.08**	0.29	0.32	0.43	**0.05**	0.58

Tabla 4.4: Tiempo de Procesamiento del CPU por clasificadores y bases de datos. Los resultados son reportados en horas.

4.4. Selección de atributos en datos artificiales

Dada la *verdad conocida* acerca de los datos artificiales –i.e. cuales atributos son relevantes e irrelevantes– las tablas 4.5, 4.6 Y 4.7 muestran los atributos que fueron seleccionados por cada algoritmo de SA y de clasificación. Dos aspectos

son discutidos principalmente, el tamaño del subconjunto generado y la calidad de estos. Este último es entendido como la capacidad de seleccionar lo que se supone que debe seleccionar.

El algoritmo de SA *GA* genera por mucho los subconjuntos de atributos más grandes. Observando la tabla 4.5, un pobre desempeño se identifica debido a que gran cantidad de variables irrelevantes son incluidas. *BT* puede ubicarse en segundo lugar al agregar un poco de irrelevancia, pero carece de algunas variables relevantes. *AU* y *BT* generan los subconjuntos más pequeños, sin embargo es interesante notar que la calidad de los subconjuntos generados por *AU* es altamente satisfactoria.

En la tabla 4.6 los resultados muestran la misma tendencia para *AG*, esto respecto a la relevancia e irrelevancia de los atributos seleccionados además del tamaño. *BT* sigue el mismo patrón respecto a su tamaño y calidad. *RS* selecciona subconjuntos pequeños, pero agregando algo de irrelevancia. Para el caso de *AU*, éste sólo incluye atributos relevantes, excepto dos atributos irrelevantes con el clasificador de *SVM*.

En la tabla 4.7 se ven los mismos resultados decepcionantes para *AG*. *BT* mejora su calidad, agregando menos irrelevancia que su contraparte en la tabla 4.6. *RS* muestra subconjuntos pequeños, pero añadiendo los dos tipos de atributos casi en la misma proporción. Y por último *AU* agrega sólo seis atributos irrelevantes, teniendo casi el mismo patrón visto en la tabla 4.6.

	Atributos de la Base de datos artificial, Data 1 Algorithms			
	GA	TA	TS	SA
1NN	1-5,7,12,16,17,21-23,27-30 34,36,38,40,43,45,47-50	1,5,9	1,5,39	1,3,4
3NN	1,5,7,9,15,21,23,27,29 33,37,38,40,44,45,48	1,2,5	1,2,5,7	1,2,5
5NN	1,2,4,5,7,9,11,12,14 16,18,19,21,25,26,27 33,39,40,42,44,47,49	1,2,5	1,4,5,13,40	1,3,5,37
7NN	1-7,9,13,15-17,20,26 27,29,37,39,40,42	1,5	1,5,37	1,4
LDA	1,2,4,5,7,11,14,16 18,19,21,24-26,34-36 41,42,45,49,50	1,12	1,3,5,13,20,21,29 36,37,39,49	1-3,7,9,10,13,14 21,23,24,27,28,39
QDA	1,2,4,5,11,12,15,17,18,20-22 26,29,33,35,37-39,43,45	1,5,40	1,2,5,12,16,20,21,40	1-4
NB	1-5,7,8,10,16-18,23,28,29 33,34,37,39,40,44,47,50	1,2,4,5,13,27	1,2,4,5,36,38	1-4
LINEAL SVM	1,2,5,6,10,11,13,15,16,19,26 28,31,34,39,45,48-50	1,5,31,39	1,5,18,30,,32,39,47	1-4
RBF SVM	1,2,4,5,11,24,26,37,39,49	1,5	1,5,13	1-4

Tabla 4.5: Los atributos seleccionados de la base de datos artificial *Data1*. Los primeros cinco atibutos son relevantes y los 45 restantes son irrelevantes.

	Atributos de la Base de datos artificial, Data 4 Algorithms			
	GA	TA	TS	SA
1NN	2,4-6,8,10,12,13,16,18 20,21,24,26,28,29 30,31,33,36,40,4-48,50	16,20,40,48,50	6,8,12,18,20,22,45,50	20,40,45-48
3NN	1,3-5,8,12-14,16,18 20,25,28,30,32,33,37 38,40,44,45,46-48,50	6,12,20,32,40,50	6,8,12,16,20,21,26,28 31,32,34,40,44,46,50	2,12,40,45-47 42-44
5NN	6,7,8,10-12,16-18 20,21,25,26,28,31 33,35-37,40,47,48 50	6,8,12,22,20,40 50	2,6,8,12,16,17,20,21 25,26,39,40, 46,48 50	6,11
7NN	4-8,12,14,16,21 22,25,31-34,37 38-40,43,45,47 48,50	4,6,8,12,20,22,40 46,50	1,8,10,12,14,20,26 30,32,34,40,44,46 48,50	6,11
LDA	3,4,6-8,10,12,14,19,20 22,23,32,33,37,39,40 41,43,44,46,48,50	12,28,32,40,46,48 50	1,4,6,11,12,14,20,22 26,32,40,44,45,48,50	6,8,12-16,18,31,37 39,41,42,44-47,50
QDA	6-8,12,14,17,19,20,21 25,32,34,36,37,39,40 42,44,47,48,50	6,8,12,14,20,40,46 50	4,6,12,14,20,40,44 48,50	6,7,11,12,20,21 26-30,33-38,50
NB	1,2,4,5,6,7,8,10,11,14 16,20-22,24-26,28-30,32 35-37,40,41,44-48,50	4,6,12,20,40,42,48 50	4,6,8,12,14,16,20 22,27,32,40,44,47 48,50	6,8,12,16,18,23-26 28-31,35-39,46,47 50
Linear SVM	4,6-8,10,11,14,15,17 19-22,25,26,29-31 35-38,42-45,48-50	6,7,10,12,18,20,26 31,32,36,40,50	3,5,7,11,12,18,22 28,29,31,32,36,40 48,50	6,10-16,19,34-37,39 40,50
RBF SVM	1,5,10,17,27,31,32,40 42,43,46,48,49	6,8,12,14,20,40 48-50	6,9,12,14,20,32 40,50	6,9,10,11,40,50

Tabla 4.6: Atributos seleccionados de la base de datos artificial *Data4*. Los atributos pares son relevantes y los impares son irrelevantes.

	Atributos de la Base de datos artificial, *Data5* Algorithms			
	GA	TA	TS	SA
1NN	1,5,7,9,11,12,18,19,22,25,26 32-35,37,38,40,47-49	5,7,11,13,15,19 39,43,49	7,9,11,21,25 29,36,47,49	11,19,20,39
3NN	1,5-7,10,11-13,15,17-19,22-25 27-29,37-40,42,43,45,46	5,7,11,19,21,22,31 35,39,45,47,49	4,5,11,13,19,21 39,45,47,49	5,10,37
5NN	1,5-7,10-13,15 17-19,22-25,27-29 37-40,42,43,45 46,48,49	5,7,11,19,21,22 31,35,39,45,47 49	4,5,11,13,19 21,39,45,47 49	5,10,37
7NN	1,3,5,7-11,13,15 18,20,21,23,25 29,31,34,35,39 43,47,48,49	5,7,11,19,22,30 31,39,45,47,49	2,3,5,7,8,11 15,19,21,22 39,41,46,47 49	5,6,11,18,19 27,29,30,34 35,37,38,49
LDA	1-3,5,9,11-17,19,21,22,24,25,27 31,32,37,39,41,43,44,45,47,49	11,27,28,31,39 43,44,47,49	5,9,11,13,19,25,39 43,47,49,50	5,7,10,11,18-31 33-35,37,41-46,49
QDA	1,2,5,7,9,13,14,17,19,22,26,30 31,33,35,38,39,42,45-47,49	5,7,11,13,19,39,43,49	5,7,11,13,17 19,39,43,49	3-5,7,10-13,15,17,20,22 23,25,26,30-32,37,42,49
NB	3,5,7,8,11-13,15,17,19,22,23,25 27,29,30-32,36,39,41,43,44,45,47	2,3,5,7,11,19,31 39,47,49,50	3,5,11,19,39 42,47,48,49	5,7-9,10-17,20 37,38,46,49
Linear SVM	1,5,6,7,8,10,18-21,25,32,33,35 38,41,42,46-49	5,11,13,19,39 47,49	5,11,13,19,22,34,39 40,43,46-49	5-18,39,49
RBF SVM	1,4,6,7,9,15,16,20,22,23,24,25,28 29,32,33,36,37,40,42,43,45,48-50	5,7,11,21,22 39,47,49	4,11,13,15,17 21,47,49	5,11,19,36 44,45

Tabla 4.7: Atributos seleccionados de la base de datos artificial *Data5*. Los atributos impares son relevantes y los pares son irrelevantes.

Como experimento adicional y para verificar la consistencia en el desempeño en términos de clasificación y calidad de subconjuntos de atributos del algoritmo de AU, se optó por la combinación de AU+NB. Ésta se replicó 40 veces a fin de encontrar un comportamiento promedio, es decir, los atributos que más se selec-

cionan sabiendo cual deben ser seleccionados, usando el conjunto de *Data1*. *NB* fue preferido ya que su cálculo es ligero y no requiere parámetros de ajuste. En la figura 4.1 se muestra la distribución de frecuencias de los atributos que han sido seleccionados. Cabe recalcar que en el conjunto de datos *Data1*, los primeros cinco atributos son relevantes y los 45 restantes son irrelevantes. Se puede observar que AU selecciona en su mayoría los atributos que debe –i.e. los primeros cinco relevantes– y descarta en su gran medida a los atributos irrelevante. Este resultado revela un desempeño interesante para el algoritmo de AU en términos de desempeño y calidad de las soluciones.

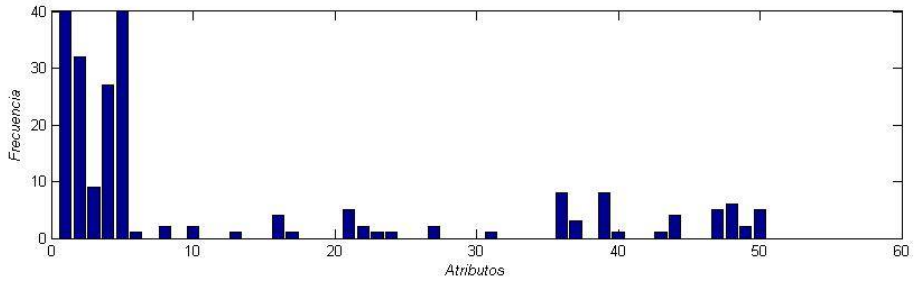

Figura 4.1: Distribuión de los atributos seleccionados por la combinación de AU+NB en la base de datos *Data1*

Capítulo 5

Conclusiones

5.1. Conclusiones

Dados los resultados experimentales expuestos en el capítulo anterior, algunas ideas generales son listadas a continuación:

1. El algoritmo de *Aceptación por Umbral* ofrece un desempeño de clasificación satisfactorio y en algunos casos superior, en comparación al resto de los los algoritmos de SA.

2. En términos esfuerzo computacional, la *Búsqueda Tabú* es el algoritmo que mejor se comporta por mucho. En algunos casos supera a otros algoritmos con un factor de 10x el tiempo de procesamiento.

3. El tamaño de los subconjuntos generados por *Aceptación por Umbral*, *Búsqueda tabú* y *Recocido Simulado* fue aceptablemente bajo. Al contrario del *Algoritmo Genético* en donde, mucha irrelevancia fue aceptada.

4. La calidad de los subconjuntos de atributos generados por *Aceptación por Umbral* es por mucho la mejor. En este tenor, el *Algoritmo Genético* mostró débil y decepcionante desempeño.

De esta manera se puede concluir que el algoritmo de SA de *Aceptación por umbral* se posiciona dentro de los algoritmos no deterministas como el mejor, con marcada diferencia contra el resto de las estrategias de búsqueda de algoritmos

analizadas. Esta conclusión recibe valor añadido por su relativa sencillez de implementación. Esta conclusión resulta interesante debido ya que a la fecha hay pocos o nulos trabajos en los que se emplee este motor de búsqueda de atributos en la reducción de la dimensionalidad. Siendo este aspecto unos de los puntos principales del trabajo de investigación expuesto en esta tesis.

5.2. Trabajo Futuro

Como trabajo futuro se plantean varias linea de investigación:

1. Ampliar el estudio a otros algoritmos o estrategias de búsqueda como *Optimización por Enjambre de Partículas*, *Optimización de Colonia de Hormigas* y/o *Sistemas Inmunes Artificiales*.

2. Generar las versiones de estos algoritmos en su forma paralela, con el propósito de incrementar la velocidad análisis.

3. Diseñar e implementar los algoritmos analizados para que funcionen en la búsqueda de atributos para clasificación en dominios de naturaleza más especializada o mayormente complejos e.g. Cloud Computing, secuencias de video e imágenes y series de temporales.

5.3. Productos derivados de la tesis

1. M. Fernández, & González F. (2014). Métodos de búsqueda no-deterministas en la selección de atributos. En Editor (Ed.). *Avances de la Computación en México.* En edición.

2. Marina P. Fernández-Pérez and Félix F. González-Navarro (2014). Non-deterministic local search methods for feature selection. An experimental study. *MICAI 2014* On edition. $http://www.micai.org/2014/pre-print/papers/paper_029.pdf$.

Anexos A

Atributos seleccionados en las bases de datos artificiales dos y tres

	Algoritmos			
	AG	AU	BT	RS
1NN	2, 4, 11, 12, 14-16, 18, 25, 28, 30-33, 38,41, 43, 45, 47, 48, 50	46, 50, 15	31, 50, 46, 47	46, 49, 48
3NN	2, 3, 5, 7, 8, 11, 14, 15,17, 19, 20, 23, 29, 33, 35, 37, 38, 40, 42, 44, 46, 47, 50	50, 46, 46	13, 46, 50	50, 46
LDA	1, 6, 7, 10, 14, 15, 20, 22, 24, 27-31,34-36, 38, 44, 45-50	46,40	22,46,49,28,7	46, 32, 47, 2, 11, 15, 21, 20, 9, 8, 7, 16, 38, 37, 36, 35, 34, 33
QDA	5, 10, 11, 15, 16, 18, 19, 20, 24, 27, 35, 37, 46, 47, 49, 50	46, 50, 35, 45	35, 46, 50, 45	46,49,48,47
NB	1, 8, 13, 16, 20-22, 27, 28, 34, 37, 39, 40, 41, 46-50	46, 50, 49, 47	2, 46, 50, 40, 27, 16, 37	46, 49, 48, 47
SVM Lineal	1, 3, 6, 8-13,15-22 24, 26, 29, 32, 33, 35, 36, 37, 38, 40, 42, 44, 46, 47, 48	46, 50, 34, 42, 45, 4	15, 46, 50	46, 49, 48, 47
SVM RBF	1, 3, 6, 8-22,24, 26, 29, 32, 33, 35, 36, 37,38, 40, 42, 44, 46, 47, 48	46, 50	15, 46, 50	46, 49, 48, 47

Tabla A.1: Atributos seleccionados para la base de datos artificial *Data 2*. Del atributo 46 al 50 son relevantes y el resto es irrelevante

	Algoritmos			
	AG	AU	BT	RS
1NN	1-4, 7, 8, 11, 12, 14, 16- 18, 20, 23, 26, 27, 28, 30, 31, 33, 38, 40, 44, 50	40, 1, 5	30, 1, 40	40, 1, 4
3NN	1, 3, 5-7, 13, 15- 18, 20, 25, 27, 29, 32, 35, 39, 40, 42, 44, 50	40, 1, 21	45, 1, 40, 29	40, 1, 6, 16
LDA	1, 3, 9, 11, 15-21, 24, 29, 30, 32, 37, 40, 42, 43, 47, 49	1, 47	49, 1, 40, 12	1, 32, 28, 40, 4, 8, 5, 33, 13, 22, 21
QDA	1, 3, 4, 8, 13, 16-22, 25, 31, 33, 34, 36, 37, 39, 40, 50	1, 40, 39, 43	31, 40, 1, 30	1, 39, 38, 37, 36, 35
NB	1- 4, 7, 8, 14, 27, 29, 38, 40, 43, 45, 46, 50	1, 40, 7, 29, 25, 43	12, 1, 40, 13, 44, 29, 26	1,39,38,37,36,35,34,33,32,31 ,30,29,28,27,26,25,24,23,22,21
SVM Lineal	2, 4, 6, 10, 13-16, 18, 19, 21, 24-26, 28, 33, 35, 37, 39, 40, 41, 45, 47-49	1, 40, 26	41, 1, 40	40, 1, 45, 37, 36
SVM RBF	2, 4, 6, 10, 13-19, 21, 24-26, 28, 33, 35, 37, 39, 40, 41, 45, 47, 48, 49	1, 40, 37	41, 1, 40	1, 39, 38, 37, 36

Tabla A.2: Atributos seleccionados para la base de datos artificial *Data 3*. Los atributos 1, 20, 30, 40 y 50 son relevantes y el resto son irrelevantes

Anexos B

Desviaciones estándar asociadas a la tabla 4.1

	Clasificadores											
	KNN1				KNN3				KNN5			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	1.70	1.66	2.83	3.64	5.72	4.09	3.26	4.60	3.43	3.31	1.66	5.01
BCW	0.34	0.49	0.55	0.48	0.49	0.38	0.58	0.58	0.15	0.38	0.48	0.32
Madelon	0.91	0.86	1.03	1.32	0.78	0.42	0.64	0.87	0.98	1.10	0.73	1.22
BC	2.08	2.83	2.15	3.37	4.12	1.66	1.99	4.87	3.02	1.94	3.55	2.54
Prostata	1.50	1.28	1.06	1.30	1.28	0.75	1.83	1.33	0.85	1.67	0.67	1.89
Leukemia	1.41	0.00	1.27	0.19	1.44	1.16	0.15	1.35	0.25	2.21	0.00	2.12
Data1	0.97	1.04	2.03	2.32	1.92	0.94	1.12	0.42	2.33	1.24	1.10	0.42
Data2	2.24	2.55	1.10	2.17	2.51	2.30	2.00	0.84	4.00	1.14	2.74	0.89
Data3	1.82	0.79	1.72	0.50	1.44	0.89	1.29	0.97	1.19	0.67	1.15	0.65
Data4	1.00	1.17	2.18	0.76	1.17	1.62	1.08	2.27	2.53	2.07	1.24	1.39
Data5	1.29	1.24	1.27	1.12	1.60	1.78	1.15	0.94	1.25	1.85	1.84	1.02

	Clasificadores											
	KNN7				LDA				QDA			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	3.22	2.71	1.66	2.04	3.60	2.77	3.31	1.70	4.48	6.00	1.35	3.83
BCW	0.27	0.36	0.40	0.52	0.10	0.19	0.19	0.25	0.23	0.29	0.16	0.16
Madelon	0.71	0.68	0.59	1.30	1.78	0.67	0.45	0.85	1.58	0.81	0.61	0.74
BC	1.71	2.65	2.30	1.90	7.62	1.18	2.16	4.39	4.77	0.65	1.30	2.51
Prostata	1.57	1.31	1.04	2.07	1.66	2.28	1.13	1.51	0.68	1.00	1.10	1.60
Leukemia	1.18	0.31	1.04	0.99	0.00	0.00	0.00	2.76	0.15	0.00	0.00	2.49
Data1	1.32	1.04	1.25	0.61	1.52	0.71	0.35	1.29	1.60	0.57	2.26	1.04
Data2	1.52	3.13	2.07	0.84	4.09	2.17	2.70	3.35	3.27	1.30	0.55	2.39
Data3	3.16	0.65	2.02	0.61	1.35	0.45	0.91	0.79	1.60	0.76	1.06	1.39
Data4	1.52	1.72	1.06	1.57	1.37	0.55	0.76	0.91	1.47	0.74	1.06	1.19
Data5	1.00	0.82	1.04	2.17	0.45	1.04	0.94	2.15	0.42	1.12	1.79	2.25

	Clasificadores											
	NB				SVM R				SVM L			
	AG	AU	BT	RS	AG	AU	BT	RS	AG	AU	BT	RS
Colon	1.11	0.94	1.13	2.44	0	1.70	2.20	4.80	1.51	0.07	0.06	2.33
BCW	0.15	0.30	1.13	0.34	0.32	0.23	0.41	0.24	0.26	0.00	0.10	0.12
Madelon	1.02	1.48	1.13	0.59	1.83	0.58	0.81	1.05	1.83	0.58	0.81	1.05
BC	2.46	0.09	1.13	0.65	1.28	0.06	1.20	0.62	2.98	0.09	0.09	3.78
Prostata	0.41	2.27	1.13	1.19	0.64	0.98	0.66	1.19	0.97	0.65	0.35	2.00
Leukemia	0.00	0.00	0.00	1.11	0.00	0.00	0.00	2.52	0.00	0.00	0.00	1.12
Data1	1.48	0.97	1.13	1.08	0.74	0.42	0.67	1.35	1.68	1.64	0.96	1.27
Data2	1.56	0.97	1.13	1.44	0	2.17	1.14	3.39	0.00	2.41	1.14	2.19
Data3	0.96	1.10	1.13	1.43	0	0.50	0.22	1.52	0.00	0.45	0.22	1.52
Data4	0.89	0.42	1.13	1.15	1.84	0.91	0.22	1.60	1.64	0.91	1.34	1.39
Data5	1.34	0.42	1.13	0.57	0	1.04	1.77	0.55	2.08	1.44	0.89	0.94

Referencias

Almuallim, H. & Dietterich, T. (1991). Learning with many irrelevant features. In *Proc. of the 9th National Conf. on AI (AAAI-91)*, vol. 2, 547–552, AAAI Press, Anaheim, California. 6

Clarke, W.R., Lachenbruch, P.A. & Broffitt, B. (1979). How non-normality affects the quadratic discriminant function. *Communications in Statistics-Theory and Methods*, **8**, 1285–1301. 19

Dan C. Marinescu, G.M.M. (2012). Classical and quantum information. Academic Press. 6

Daniel, P.S.d.R. (2001). DeducciÛn de distribuciones: el mÈtodo de monte carlo. En Fundamentos de EstadÌstica. 16

Demsar, J. (2006). Statical comparation of classifiers over multiple data sets. *Machine Learning Research*, 1–30. 22

Díaz, A.H. (2006). Algoritmo tabú para un problema de distribución de espacios. http://www.upo.es/RevMetCuant/pdf/vol1/art2.pdf. 10

Duarte, A. (1993). Metaheurísticas dykinson. 14

Duda, R. & Hart, P. (2001). *Pattern Recognition and Scene Analysis*. John Wiley and Sons. vi, 2, 6, 21

Fayyad, U.M., Piatetsky-Shapiro, G. & Smyth, P. (1996). Advances in knowledge discovery and data mining. chap. From Data Mining to Knowledge

Discovery: An Overview, 1–34, American Association for Artificial Intelligence, Menlo Park, CA, USA. 2

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press. 6

Kirkpatrick, S. (1984). Optimization by simulated annealing: Quantitative studies. *Journal of Statistical Physics*, **34**. 12

Kittler, J. (1986). *Handbook of Pattern Recognition and Image Processing*, chap. Feature selection and extraction, 59–83. Academic Press. 6

Lin, C., Haley, K. & Sparks, C. (1995). A comparative study of both standar and adaptative version of threshold acceptiong and simulated annealing algorithms in three scheduling problems. *European Journal of Operational Research*, **83**. 15

Liu, H. & Setiono, R. (1996). A probabilistic approach to feature selection - a filter solution. In *Int. Conf. on Machine Learning*, 319–327. 6

Liu, M.D..H. (1997). Feature selection for classification. *Intelligent Data Analysis*, **1**, 131–156. 6

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. & Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**. 12

Moscato, P. & Fontanari (1990). Convergence and finite-time behavior of simulated annealing. *Advances in Applied Probality*, **18**, 747–771. 14

NIPS (????). Feature selection challenge,feature extraction workshop. Http://www.nipsfsc.ecs.soton.ac.uk/datasets/. 23

Nissen, V. & Paul, H. (1995). A modification of threshold accepting and its application to the quadratic assigment problem. *OR Spektrum*, **17**, 205–210. 15

of California at Irvine, U. (????). Machine learning respository. Https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic). 23

Oommen, T., Misra, D. & Twarakavi, e.a. (2008). An objective analysis of support vector machine based classification for remote sensing. mathematical geosciences. Mathematical Geosciences. 6

Pang-Ning Tan, V.K., Mitchael Stein Bach (2006). *Introduction to Statistical Pattern Recognition*. Introduction to Data Mining. 17, 20

Pudil, P., Ferri, F., Novovicova, J. & Kittler, J. (1994). Floating search methods for feature selection. *Pattern recognition letters*, **15**, 1119–1125. 6, 7

P.VenKataraman (2002). *Applied Optimization with matlab programming*. Wiley-Interscience. 9

R., J. & Yepes, V. (-). OptimizaciÛn de rutas mediantela b?squeda en entornos variables y aceptaciÛn por umbrales estoc·sticos. http://personales.upv.es/vyepesp/CIT2004STA.pdf. 15

Scheermesse, T. & Bryngdahl, O. (1995). Threshold accepting for constrained half toning. *Optics Communications*, **115**, 13–18. 15

Shearer, C. (2000). The crisp-dm model: The new blueprint for data mining. *Journal of Data Warehousing*, **5**. 3

Stender, J. (2007). Parallel genetic algorithms: Theory and applications, ios press. Segunda edición. 7

Tibshirani, R., Hastie, T., Narasimhan, B. & Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, **99**, 6567–6572. 16

Torres, J., Spola, N., Alvarez, E. & Monard, M. (2014). A framework to generate synthetic multi-label datasets. *ELSEVIER*, 155–176. 23

Trevor Hastie, R.T. & Fridman, J. (2008). *The elements of Statiscal Learning*. Springer. 16

Zhang, H. & Sun, G. (2002). Feature selection using tabu search method. *Pattern Recognition*, **35**, 701 – 711, image/Video Communication. 11