Búsqueda secuencial de subconjuntos de atributos sobre un ranking

Roberto Ruiz, José C. Riquelme y Jesús S. Aguilar–Ruiz
Departamento de Lenguajes y Sistemas Informáticos
ETS Ingeniería Informática
Universidad de Sevilla
41012 Sevilla
{rruiz,riquelme,aguilar}@lsi.us.es

Resumen

La selección de atributos es una técnica de preprocesamiento que extrae atributos relevantes del conjunto total de atributos, englobando una búsqueda de subconjuntos que mejor se ajusten a una medida de evaluación. Estas técnicas son muy útiles en las tareas de clasificación, sin embargo, cuando el número de atributos se eleva, los métodos de búsqueda se hacen computacionalmente muy costosos. En este trabajo, combinamos la velocidad de los algoritmos de ranking con un método rápido de búsqueda sobre la lista de atributos. El método denominado IRU (Incremental Ranked Usefulness) se basa en la idea de relevancia y redundancia, en el sentido de que un atributo ordenado se escoge si añade información al incluirlo en el subconjunto. Una extensa comparativa con otros métodos de selección, utilizando bases de datos de mediana y alta dimensionalidad, demuestran la eficiencia y la eficacia de nuestra propuesta.

1. Introduction

En los últimos años se ha producido un importante crecimiento de las bases de datos en todas las áreas del conocimiento humano. Este incremento es debido principalmente al progreso en las tecnologías para la adquisición de los datos. Teóricamente, el tener más atributos daría más poder discriminatorio. Sin embargo, puede desencadenar algunos problemas: incremento del coste y de la complejidad computacional, aparición de muchos atributos redundantes y/o irrelevante, y la degradación en el error de clasificación.

La mayoría de los algoritmos de selección de atributos enfocan esta tarea como un problema de búsqueda, donde cada uno de los estados en la búsqueda se corresponde con un subconjunto distinto de atributos [2]. El proceso de búsqueda se combina con un criterio para evaluar el mérito de cada uno de los subconjuntos de atributos candidatos. Existen numerosas combinaciones posibles entre las diferentes técnicas de búsqueda y cada una de las medidas de atributos [14]. Sin embargo, los métodos de búsqueda pueden ser demasiado costosos en bases de datos con dimensión alta, especialmente si se aplica un algoritmo de aprendizaje como criterio de evaluación.

Los algoritmos de selección de atributos los podemos agrupar de dos maneras desde el punto de vista de la medida de evaluación escogida: dependiendo del modelo seguido (filtro o wrapper) o de la forma de evaluar los atributos (individual o subconjuntos). El modelo filtro evalúa los atributos de acuerdo con heurísticas basadas en características generales de los datos e independientes del método de clasificación a aplicar, mientras que el wrapper utiliza el comportamiento de un algoritmo de clasificación como criterio de evaluación de los atributos. El modelo wrapper escoge los atri-

Actas del III Taller Nacional de Minería de Datos y Aprendizaje, TAMIDA2005, pp.251-260
ISBN: 84-9732-449-8

butos que demuestran mejor clasificación, ayudando a mejorar el comportamiento del algoritmo de aprendizaje. En contra tiene un coste computacional mas elevado [12, 9] que el modelo filtro. Los métodos de ranking de atributos (FR), también denominado feature weighting [2, 5], asignan pesos a los atributos individualmente y los ordenan basándose en su relevancia con respecto al concepto destino o atributo clase, mientras que los algoritmos de selección de subconjunto de atributos (FSS) evalúan la bondad de cada uno de los subconjuntos candidatos.

En la categoría de algoritmos FR, los k primeros atributos formarán el subconjunto final. Ésta es una buena aproximación para bases de datos de dimensionalidad alta, dado su coste lineal con respecto al número de atributos. En algoritmos capaces de seleccionar subconjuntos de atributos, los subconjuntos candidatos se generan según alguna estrategia de búsqueda, existiendo diversas posibilidades. En [14] se hayan clasificados numerosos algoritmos de selección. Se pueden encontrar diferentes estrategias de búsquedas, exhaustivas, heurísticas y aleatorias, combinadas con distintos tipos de medidas para formar un gran número de algoritmos. La complejidad temporal es exponencial con respecto a la dimensionalidad de los datos en la búsqueda exhaustiva y cuadrática en la búsqueda heurística. En la búsqueda aleatoria, la complejidad puede ser lineal al número de iteraciones [15], pero la experiencia muestra que el número de iteraciones necesarias para encontrar un subconjunto óptimo es al menos cuadrático con respecto al número de atributos [4]. Los métodos de búsqueda más populares en aprendizaje supervisado no se pueden aplicar a este tipo de bases de datos debido al elevado número de atributos. Una de las pocas técnicas de búsqueda utilizadas en este dominio es la búsqueda secuencial hacia adelante [22, 7, 18] (también denominada hill-climbing o greedy).

Las limitaciones de ambas aproximaciones, FR y FSS, sugieren claramente la necesidad de un modelo híbrido. Recientemente, se utiliza un nuevo marco de trabajo para la selección de atributos, donde se combinan varias de las técnicas antes mencionadas. Debido al gran número de atributos, el proceso de selección se descompone en dos fases: se empieza con una fase donde se evalúan los atributos individualmente, proporcionando un ranking ordenado según algún criterio filtro. En la segunda fase, se aplica un evaluador de subconjuntos de atributos (correlación, consistencia, divergencia, o un algoritmo de aprendizaje) a un número determinado de atributos del ranking anterior (los que superan un umbral, o los k primeros) siguiendo una estrategia de búsqueda. El método propuesto por Xing et al. [19], el propuesto por Yu y Liu [22], y otro por Guyon et al. [6] están entre los más referenciados que siguen este entorno de trabajo.

El modelo híbrido propuesto en este trabajo intenta aunar las ventajas de las diferentes aproximaciones en dos pasos: primero, siguiendo un modelo filtro o wrapper se genera una lista de atributos ordenada, y segundo, los atributos ordenados se van seleccionando utilizando un evaluador de subconjuntos (filtro o wrapper). Este método de búsqueda es válido para una lista de atributos ordenada por cualquier criterio, e igualmente, en la selección de los atributos puede intervenir cualquier medida capaz de evaluar subconjuntos. Además, esta manera de buscar subconjuntos ofrece la posibilidad de aplicar eficientemente un modelo wrapper en dominios de alta dimensión, obteniendo mejores resultados que con el modelo filtro.

Este trabajo tiene como objetivo el estudio y propuesta de un método de selección de atributos que se pueda aplicar a bases de datos de dimensión elevada en un marco de aprendizaje supervisado, en concreto para clasificación. Se utilizarán tres algoritmos de aprendizaje clasificadores para comparar los efectos de la selección de atributos, uno probabilístico (naive Bayes), otro basado en las técnicas de vecinos más cercanos (ib1) y un tercero basado en árboles de decisión (C4.5). Los algoritmos de aprendizaje empleados se han elegido por ser representativos de diferentes tipos de clasificadores, y se usan con frecuencia en los estudios comparativos y en bastantes aplicaciones de minería [16, 13].

El documento se organiza de la siguiente forma: en la siguiente sección se revisan los conceptos de relevancia y redundancia. En la sección 3 se presenta nuestra propuesta de relevancia y redundancia, describiendo el algoritmo IRU. En la sección 4 se muestran los resultados experimentales, y finalmente, en la sección 5 se recogen las conclusiones más interesantes.

2. Nociones de relevancia y redundancia de atributos

2.1. Relevancia

El propósito de los algoritmos de selección de atributos es el de identificar los relevantes de acuerdo a una definición de relevancia. Sin embargo, la noción de relevancia no ha sido aún rigurosamente justificada sobre un acuerdo de común entendimiento [1]. John et al. [9] establece tres categorías de relevancia, fuerte, débil e irrelevancia, permitiendo seleccionar los atributos en función del grado de relevancia, siendo los atributos fuertemente relevantes imprescindibles porque su eliminación añade ambigüedad a las muestras del conjunto de aprendizaje. Sin embargo los atributos débilmente relevantes pueden mantenerse o no, dependiendo de qué atributos contenga el conjunto de atributos seleccionados y del criterio que se desea optimizar. Los irrelevantes no son tenidos en cuenta. Wang [20] hace uso de conceptos de Teoría de la Información para definir la relevancia variable o entrópica de un atributo respecto a la clase. Blum y Langley [2] recogen una serie de definiciones de relevancia, entre ellas una definición de relevancia como medida de complejidad, es decir, encontrar el menor número de atributos necesarios para obtener un funcionamiento óptimo en el conjunto de aprendizaje con respecto al concepto que representa.

Todas las definiciones anteriores son independientes del algoritmo de inducción utilizado, pero puede resultar que un atributo que es relevante según algunas de las definiciones anteriores, utilizado con un determinado algoritmo de inducción no intervenga en el proceso de inducción por lo que se considera irrelevante por dicho algoritmo. Una definición que si considera esta situación es la *utilidad incremental* [3] dada por Caruana y Freitag, siendo idónea para el propósito de obtener un subconjunto de atributos predictivo.

Definición 1. Dado un conjunto de aprendizaje D, un algoritmo de aprendizaje L, y un conjunto de atributos F, un atributo x_i es *incrementalmente útil* para L con respecto a F si la tasa de acierto de la hipótesis que L produce usando el conjunto de atributos $\{x_i\} \cup F$ es mejor que la tasa de acierto obtenida utilizando sólo el conjunto de atributos F.

2.2. Redundancia

El concepto de redundancia entre atributos se expresa normalmente en términos de correlación entre atributos. Normalmente se considera que dos atributos son redundantes si sus valores están completamente correlados. Se distingue normalmente dos tipos de medidas de la correlación entre dos atributos (X, Y): lineal y no lineal. En el primer caso, se utiliza el coeficiente de correlación de Pearson, y en el segundo, están las medidas basadas en los conceptos de entropía, o medida de la incertidumbre de una variable aleatoria. Se utiliza con frecuencia la incertidumbre simétrica, definida como

$$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right]$$

donde $H(X) = -\sum_i P(x_i)log_2(P(x_i))$ es la entropía de una variable X e $IG(X|Y) = H(X) - H(X|Y)$ es la ganancia de información de la variable X con respecto a Y.

Las definiciones de correlación anteriores se dan entre pares de variables, sin embargo no está tan claro como determinar cuando un atributo está correlado con un conjunto de atributos.

3. Búsqueda secuencial sobre un ranking

En esta sección daremos nuestra idea de relevancia y redundancia, teniendo en cuenta el objetivo fijado de permitir la utilización de cualquier tipo criterio tanto para elaborar el ranking como para generar el subconjunto de atributos. También se describe el método propuesto.

3.1. Propuesta para obtener relevancia y redundancia

Como se ha señalado antes, se utilizará un evaluador de subconjunto de atributos, que denominaremos *SubEvaluador*, para seleccionar un grupo reducido de ellos. Así, dado un SubEvaluador L, y dado un conjunto de atributos F, se busca en el espacio de F el subconjunto de atributos que mejor resultado de evaluación presente, utilizando el valor para comparar el comportamiento del SubEvaluador L sobre el subconjunto de prueba. El modelo wrapper es computacionalmente más costoso, sin embargo tiende a encontrar conjuntos de atributos mejor ajustados al algoritmo de aprendizaje destino, ofreciendo un comportamiento superior al modelo filtro. Entre las medidas más utilizadas en la evaluación de subconjuntos nos encontramos con la correlación (subconjuntos correlados con la clase y no correlados entre ellos), consistencia, divergencia y el resultado de aplicar un algoritmo de aprendizaje (wrapper)

La forma de realizar la búsqueda en el espacio de los atributos es un factor clave, sobretodo en los métodos wrappers. Entre otras, aparece junto a estrategias de búsquedas tales como búsqueda secuencial greedy, búsqueda best-first, y algoritmos genéticos [13], la mayoría de ellos tienen una complejidad temporal $O(N^2)$, no dando buenos resultados en bases de datos con decenas de miles de atributos. En algunas casos, no se puede preveer el número de veces que se va a ejecutar el clasificador, pudiendo llegar a ser el tiempo requerido del orden de cientos o miles de horas, suponiendo que el método no caiga primero en algún mínimo local y se detenga prematuramente. Por ejemplo, en una base de datos con veinte mil atributos, suponiendo que el conjunto final esté compuesto por cincuenta atributos (0.0025 % del total), una búsqueda greedy realizaría aproximadamente un millon de comprobaciones con el algoritmo de aprendizaje (N veces para elegir el mejor atributo, siendo N el número total de atributos, $N-1$ veces para encontrar el siguiente mejor atributo junto al primero, de esta manera tendríamos aproximadamente $20000 at. \times 50 seleccionados$), suponiendo un promedio de treinta segundos por cada una de ellas daría como resultado mas de ocho mil horas.

En este trabajo se propone una búsqueda rápida sobre una parte mínima del espacio de atributos. Comenzando por el primer atributo de una lista ordenada por algún criterio de evaluación, se va comprobando la aportación de los atributos al resultado del SubEvaluador añadiéndolos uno a uno. En este caso, el algoritmo de evaluación se ejecuta siempre tantas veces como atributos tenga la base de datos, teniendo en cuenta que normalmente el SubEvaluador se construirá con muy pocos atributos. El algoritmo de ranking de atributos utiliza una función de puntuación obtenida con los valores de cada atributo y la clase. Por convención, se asume que una puntuación alta en un atributo es indicativa de una relevancia alta, y que los atributos se sitúan en orden decreciente de puntuación. Se considera definido el criterio de ranking para atributos individuales, indepedientemente del contexto de los demás.

Al realizar un ranking en bases de datos con muchos atributos, normalmente se tiene un elevado número de atributos con puntuaciones similares, y se critica la frecuente selección de atributos redundantes en el subconjunto final. Sin embargo, según [5], teniendo en cuenta que los atributos que son independientes e idénticamente distribuidos no son realmente redundantes, se puede reducir el ruido y por consiguiente obtener mejor separación entre las diferentes clases. Además, una correlación (en valor absoluto) muy alta entre variables no significa que no se complementen. En consecuencia, la idea de redundancia en este trabajo no se basa siempre en medidas de correlación, sino

en un criterio de evaluación de subconjuntos, pudiendo ser una aproximación filtro o wrapper, en el sentido de que un atributo es seleccionado si se obtiene información adicional cuando se añade al subconjunto de atributos elegidos previamente.

3.2. Utilidad incremental ordenada

En la selección de subconjuntos de atributos, es un hecho que se perciban como innecesarios dos tipos de atributos: los que son irrelevantes al concepto destino y los que son redundantes con otros atributos dados. Para ello, se define formalmente la utilidad incremental ordenada de manera que se identifique explícitamente los atributos relevantes y no se tenga en cuenta a los atributos redundantes, logrando así un aprendizaje más eficiente y efectivo.

Sea D un conjunto de datos etiquetados; y F un subconjunto de atributos de los datos D; se denomina *valor de medición* $\Gamma(D/F, L)$ al resultado de aplicar el evaluador de subconjuntos L, teniendo sólo en cuenta el subconjunto F.

Sea $R = \{x_i\}$, $i = 1 \ldots N$ un ranking de todos los atributos de D ordenados decrecientemente, y F el subconjunto de los i primeros de R, entonces $\Gamma(D/F \cup \{x_{i+1}\}, L) \not> \Gamma(D/F, L)$, siendo x_{i+1} condicionalmente independiente de la clase C dado el subconjunto de atributos F, y se puede omitir x_{i+1} sin comprometer el resultado obtenido por el evaluador de subconjuntos.

Definición 2. El atributo x_{i+1} en R es *incrementalmente útil* para L si no es condicionalmente independiente de la clase C dado F, de manera que el valor de medición que L produce usando el conjunto de atributos $\{x_{i+1}\} \cup F$ es mejor que el valor de medición obtenido utilizando sólo el conjunto de atributos F.

Partiendo de esta definición se implementa el algoritmo en el siguiente apartado.

3.3. Algoritmo

Existen dos fases bien diferenciadas en el algoritmo. En primer lugar, los atributos se ordenan según alguna medida de evaluación individual (línea 1–4). En segundo lugar, se tratará la lista de atributos una vez, recorriendo el ranking desde el principio hasta el último atributo ordenado (línea 5–12). Se obtiene el valor de medición del evaluador de subconjuntos con el primer atributo de la lista (línea 9) y se marca como seleccionado (línea 10–12). Se obtiene de nuevo el valor de medición del SubEvaluador, pero esta vez con el primer y segundo atributo. El segundo se marcará como seleccionado dependiendo de si el valor obtenido es mejor (línea 10). El siguiente paso es evaluar de nuevo con los atributos marcados y el siguiente de la lista, y así sucesivamente. Se repite el proceso hasta alcanzar el último atributo de la lista (línea 7). Finalmente, el algoritmo devuelve el mejor subconjunto encontrado, y se puede afirmar que no contendrá atributos irrelevantes ni redundantes.

```
Entrada: D-datos, U-criterio, L-eval.
Salida: MejorSub

lista R = {}
para cada atributo x_i ∈ D
    Puntuacion = calcula(x_i, U, D)
    añade x_i a R según Puntuacion
MejorEval = 0
MejorSub = ∅
para i = 1 hasta N
    TempSub = MejorSub ∪ {x_i} (x_i ∈ R)
    TempEval = SubEvaluador(TempSub, L)
    si (TempEval > MejorEval)
        MejorSub = TempSub
        MejorEval = TempEval
```

Algoritmo 1: Algoritmo IRU

La eficiencia de la primera parte del algoritmo anterior es obvia, dado que sólo requiere el cálculo de N puntuaciones y su ordenación, mientras que en la segunda parte, la complejidad temporal depende del algoritmo de evaluación de subconjunto escogido, siendo, obviamente, más costoso en el caso de una aproximación wrapper. Se debe tener en cuenta que el SubEvaluador se ejecuta N (número total de atributos) veces con un número muy reducido

de atributos, sólo los seleccionados. Por tanto, se proporciona un buen punto de partida. De hecho, los resultados obtenidos a partir de un orden aleatorio de atributos (sin un ranking previo) presentaron tres inconvenientes: 1) inestabilidad, dado que la solución es no determinista; 2) mayor número de atributos seleccionados; y 3) mayor tiempo de computación al trabajar el algoritmo de evaluación de subconjuntos con un mayor número de atributos desde las primeras iteraciones.

Como se puede observar en el algoritmo, el primer atributo es siempre seleccionado. Esto no debe representar un gran inconveniente en bases de datos de elevada dimensionalidad, basándonos en la probable existencia de diferentes conjuntos de atributos que contengan parecida información. Por consiguiente, IRU se basa en la idea de que el mejor atributo de un ranking lleva a soluciones cercana a la óptima. El principal inconveniente de hacer un recorrido siguiendo esta dirección (*hacia adelante*), es la posibilidad de no detectar interacciones básicas entre atributos que sean muy interesantes para la clasificación. Es decir, puede ocurrir que atributos que por separado son irrelevantes, el estar juntos en un determinado subconjunto les haga ser muy importantes. En el caso *hacia atrás*, se remedia en parte el inconveniente de la generación *hacia adelante*, aunque permanezcan interacciones ocultas, sin embargo necesita más tiempo de computación. Debemos tener en cuenta que debido al gran número de atributos de estas bases de datos, el coste computacional del backward es mucho mayor, y si se utiliza una aproximación wrapper, aún mayor. Esto es así por empezar con el conjunto completo de atributos e ir eliminándolos uno a uno.

4. Experimentos

En esta sección se pretende evaluar nuestra propuesta en términos de exactitud de clasificación, grado de dimensionalidad y velocidad sobre los atributos seleccionados, para mostrar como IRU se comporta en situaciones donde existen un número medio y alto de atributos. La comparación se efectuó sobre dos grupos diferentes de bases de datos, según sea su dimensionalidad menor o mayor que cien atributos, del repositorio UCI[1].

Tabla 1: Datos.

Datos	Ejemplos	At´s	Clases
letter	16	20000	26
segment	19	2310	7
german-credit	20	1000	2
horse-colic	22	368	2
mushroom	22	8124	2
autos	25	205	7
horse-co.OR	27	368	2
hypothyroid	29	3772	4
sick	29	3772	2
ionosphere	34	351	2
soybean	35	683	19
kr-vs-kp	36	3196	2
anneal	38	898	6
anneal.OR	38	898	6
waveform	40	5000	3
sonar	60	208	2
splice-2	60	3190	3
audiology	69	226	24
Arrhythmia	279	452	16
Isolet	617	1560	26
ADS	1558	3279	2

Las propiedades de estas bases de datos se encuentran resumidas en la Tabla 1. La principal característica del segundo grupo de bases de datos es su gran número de atributos. Se seleccionaron tres algoritmos de aprendizaje muy diferentes, C4.5, ib1 y Naive Bayes, para evaluar la exactitud sobre los atributos seleccionados por cada uno de los algoritmos de selección de atributos.

Como se comentó en la sección anterior, en la primera fase del algoritmo se elabora una lista con todos los atributos ordenados según su relevancia de acuerdo con el criterio elegido. Para demostrar la validez de nuestra propuesta se han escogido tres criterios diferentes de generar una lista ordenada de atributos:

- Ganancia de información (IG), basada en el concepto de entropía de la teoría de la información, es una medida de la incertidumbre de una variable aleatoria.

[1] http://www.ics.uci.edu/ mlearn/MLRepository.html

- Relief (RL) basándose en la técnica del vecino más cercano asigna un peso a cada atributo. Sus creadores fueron Kira y Rendell [8] y posteriormente fue modificado por Kononenko [11]. El peso de cada atributo se va modificando en función de la habilidad para distinguir entre los valores de la variable clase.

- SOAP [17] (Selection of Attributes by Projections) (SP) Este criterio se basa en un único valor denominado NCE (Número de Cambios de Etiqueta), que relaciona cada atributo con la etiqueta que sirve de clasificación. Este valor se calcula proyectando los ejemplos de la base de datos sobre el eje correspondiente a ese atributo (los ordenamos por él), para a continuación recorrer el eje desde el menor hasta el mayor valor del atributo contabilizando el número de cambios de etiqueta que se producen.

Una vez obtenido el ranking, para evaluar los subconjuntos se utiliza el resultado del SubEvaluador. Los criterios de evaluación de subconjuntos empleados se han elegido por ser representativos de diferentes tipos de aproximaciones, y se usan con frecuencia en los estudios comparativos:

- Medida de correlación (Corr) evalúa la bondad de un subconjunto de atributos basándose en la hipótesis de que un buen subconjunto contiene atributos muy correlacionados con la clase, pero poco correlados entre ellos.

- Medida de consistencia (Cons) intenta encontrar un número mínimo de atributos que sea capaz de separar las clases igual que el conjunto completo de atributos. Una inconsistencia se define como una instancia que teniendo los mismos valores para los atributos, difiere en la etiqueta de la clase.

- Resultado de clasificación (Wra) o aproximación Wrapper, utiliza el algoritmo de aprendizaje objetivo para estimar la bondad del subconjunto de atributos.

Los experimentos se llevaron a cabo utilizando implementaciones de los algoritmos existente en el entorno WEKA [21], así como nuestra propuesta también fue implementada en dicho entorno. Para cada base de datos del primer grupo (inferiores a cien atributos), se ejecutó nuestra propuesta elaborando el ranking con las tres medidas individuales antes mencionadas (SO, RL e IG), y obteniendo el subconjunto final de atributos con los tres criterios referenciados (Corr, Cons y Wra). Dada la elevada dimensionalidad de los datos, se limita la comparación a técnicas secuenciales hacia adelante (SF) (ver Sección **??**), que para su comparación se combina con el criterio de subconjuntos correspondiente. Se almacenó el tiempo de ejecución y el número de atributos seleccionados por cada algoritmo. Entonces, se aplicaron los tres clasificadores (c4, ib1 y nb) sobre las bases de datos originales, así como sobre las recién obtenidas, conteniendo sólo los atributos seleccionados por cada uno de los algoritmos de selección. En cada caso se guardó la exactitud obtenida mediante validación cruzada diez.

Tabla 2: Resultados obtenidos con C4.5 y los datos del 1^{er} grupo Tabla 1; 1–Promedio de exactitud; 2–Porcentaje medio de atributos retenidos con respecto al original; 3–Tiempo total de ejecución (en horas).

		C4.5 (88,02)			
		SP	RL	IG	SF
Corr	1	86,54	86,24	85,91	85,88
	2	0,25	0,23	0,21	0,20
	3	0,003	0,807	0,003	0,003
Cons	1	86,87	87,25	87,58	86,71
	2	0,47	0,44	0,42	0,32
	3	0,01	0,81	0,01	0,03
Wra	1	88,05	88,86	88,06	88,02
	2	0,31	0,33	0,29	0,22
	3	1,43	2,28	1,18	5,23
Prom.1		87,15	87,45	87,18	86,87
Prom.2		0,34	0,33	0,31	0,25
Total	3	1,44	3,89	1,19	5,26

Las Tablas 2, 3 y 4 proporcionan para c4, ib1

Tabla 3: Resultados obtenidos con IB1.

		IB1 (85,9)			
		SP	RL	IG	SF
CORR	1	82,68	83,70	82,82	82,82
	2	0,25	0,23	0,21	0,20
	3	0,003	0,807	0,003	0,003
CONS	1	86,59	86,77	85,89	84,21
	2	0,47	0,44	0,42	0,32
	3	0,01	0,81	0,01	0,03
WRA	1	87,31	88,29	88,01	87,32
	2	0,40	0,35	0,34	0,26
	3	16,23	13,04	12,46	75,14
PROM.1		85,53	86,25	85,58	84,78
PROM.2		0,37	0,34	0,32	0,26
TOTAL 3		16,24	14,65	12,47	75,17

Tabla 4: Resultados obtenidos con NB.

		NB (80,34)			
		SP	RL	IG	SF
CORR	1	81,33	82,26	80,16	80,63
	2	0,25	0,23	0,21	0,20
	3	0,003	0,807	0,003	0,003
CONS	1	81,05	81,44	81,04	80,02
	2	0,47	0,44	0,42	0,32
	3	0,01	0,81	0,01	0,03
WRA	1	84,88	85,56	85,76	85,58
	2	0,34	0,33	0,32	0,23
	3	0,18	1,19	0,18	1,01
PROM.1		82,42	83,09	82,32	82,08
PROM.2		0,35	0,33	0,32	0,25
TOTAL 3		0,19	2,81	0,19	1,04

y nb respectivamente, los resultados obtenidos con las dieciocho bases de datos que conforman el primer grupo de la Tabla 1. Se muestra: 1–el promedio de las exactitudes; 2–el porcentaje medio de atributos retenidos con respecto al original; y 3–el tiempo total de ejecución en horas. Se diferencian por el método utilizado (IRU con sp, rl e ig; y SF) y por la medida de subconjuntos (Corr, Cons y Wra). El promedio del clasificador con las bases de datos originales se encuentra en la primera línea de cada tabla, así como en las tres últimas se observa el promedios de exactitudes (PROM.1) y de retención de atributos (PROM.2), y el total de tiempo para cada método (TOTAL 3).

Los resultados observados en las tres Tablas son muy similares. Los valores de las exactitudes para las tres aproximaciones IRU, con los tres clasificadores y con los tres evaluadores de subconjuntos, son ligeramente superiores a los valores aportados por SF, destacando las tasas de aciertos obtenidas cuando el algoritmo IRU parte del ranking elaborado con RL. En cuanto al número de atributos seleccionados, el promedio es favorable a las aproximaciones de SF (por eso pierde en las exactitudes). Además, es importante resaltar, que el tiempo necesitado por SF para la selección de atributos es muy superior al necesario para la aplicación de IRU. Esta diferencia se acentúa con la aproximación wrapper, y en especial con el clasificador IB1. Una excepción se produce para el clasificador NB, donde RL tarda más tiempo que SF, pero frente a SP e IG, SF pierde en casi todos los casos.

En la Tabla 5 se muestran los resultados obtenidos para la aproximación wrapper mediante el clasificador NB con el método IRU y con SF. Se observa la tasa de aciertos, el porcentaje de atributos seleccionados frente al total y el tiempo en segundos necesarios para cada algoritmo de selección. De esta Tabla vamos a resaltar tres aspectos interesantes:

1. No existen diferencias significativas entre los resultados de exactitud y porcentaje de atributos retenidos obtenidos con IRU y SF.

2. El tiempo necesario para obtener los resultados del punto anterior en SF es aproximadamente diez veces superior al empleado por IRU.

3. Las exactitudes de clasificación obtenidas con el conjunto total de atributos en los dos primeros casos, son significativamente menores que las obtenidas con los dos algoritmos de selección.

Tabla 5: Resultados para la aproximación wrapper NB con el método IRU y con SF. AC–Aciertos(%), AC–Atributos, %RET–Atributos seleccionados frente al total y T–Tiempo en segundos.

Datos	IRU				SF				Original
	Ac	At	%Ret	t	Ac	At	%Ret	t	
Arrhythmia	73,01	7	0,025	251	74,35	15	0,054	4089	61,74
Isolet	87,17	34	0,055	7533	89,23	29	0,047	74644	83,77
ADS	96,12	15	0,010	657	96,77	23	0,015	6060	96,58

5. Conclusiones

El éxito de muchos esquemas de aprendizaje, en sus intentos para construir modelos de datos, pasa por la habilidad para identificar un subconjunto pequeño de atributos altamente predictivos. La inclusión de atributos irrelevantes, redundantes o con ruido en la fase del proceso de construcción del modelo puede provocar un comportamiento predictivo pobre y un incremento computacional. La aplicación de los métodos de búsqueda más populares en aprendizaje automático a bases de datos con un gran número de atributos puede ser prohibitivo. Sin embargo, en este trabajo, se propone una nueva técnica de selección de atributos que permite utilizar una aproximación wrapper para encontrar un buen conjunto de atributos para clasificación. Se utiliza la definición de Utilidad Incremental Ordenada para decidir, al mismo tiempo, si un atributo es relevante y no redundante, o no (no relevante o redundante). La técnica extrae los mejores atributos no consecutivos del ranking, intentando evitar estadísticamente la influencia de los atributos innecesarios en la clasificación posterior. Esta nueva heurística, denominada IRU, muestra un excelente comportamiento comparado con la técnica tradicional de búsqueda secuencial hacia adelante, no sólo considerando la exactitud de la clasificación en relación con la aproximación filtro, sino también en relación con el coste computacional de la aproximación wrapper.

En nuestros próximos trabajos profundizaremos en las distintas opciones de búsquedas de subconjuntos de atributos aplicables sobre los rankings, evitando su dependencia con el primero.

6. Agradecimientos

Este trabajo está financiado, dentro del Plan nacional de investigación, por los proyectos TIN2004-00159 y TIN2004–06689–C03–03.

Referencias

[1] Bell, D., Wang, H.: A formalism for relevance and its application in feature subset selection. Machine Learning **41** (2000) 175–195

[2] Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. In Greiner, R., Subramanian, D., eds.: Artificial Intelligence on Relevance. Volume 97. (1997) 245–271

[3] Caruana, R., Freitag, D.: How useful is relevance? In: Working notes of the AAAI fall symp. on relevance, N. Orleans, LA, AAAI Press (1994) 25–29

[4] Dash, M., Liu, H., Motoda, H.: Consistency based feature selection. In: Pacific-Asia Conf. on Knowledge Discovery and Data Mining. (2000) 98–109

[5] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research **3** (2003) 1157–1182

[6] Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machine. Machine Learning **46** (2002) 389–422

[7] Inza, I., Larrañaga, P.L., Blanco, R., Cerrolaza, A.: Filter versus wrapper gene selection approaches in dna microarray domains. Artificial Intelligence in Medicine **31** (2004) 91–103

[8] Kira, K., Rendell, L.: A practical approach to feature selection. In: 9th Int. Conf. on Machine Learning, Aberdeen, Scotland, Morgan Kaufmann (1992) 249–256

[9] Kohavi, R., John, G.: Wrappers for feature subset selection. Artificial Intelligence **1-2** (1997) 273–324

[10] Koller, D., Sahami, M.: Toward optimal feature selection. In: 13th Int. Conf. on Machine Learning, Bari, IT, Morgan Kaufmann (1996) 284–292

[11] Kononenko, I.: Estimating attributes: Analysis and estensions of relief. In: European Conf. on Machine Learning, Vienna, Springer Verlag (1994) 171–182

[12] Langley, P.: Selection of relevant features in machine learning. In: Procs. Of the AAAI Fall Symposium on Relevance. (1994) 140–144

[13] Liu, H., Motoda, H.: Feature Selection for Knowlegde Discovery and Data Mining. Kluwer Academic Publishers, London, UK (1998)

[14] Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Trans. on Knowledge and Data Eng. **17** (2005) 1–12

[15] Liu, H., Setiono, R.: A probabilistic approach to feature selection: a filter solution. In: 13th Int. Conf. on Machine Learning, Morgan Kaufmann (1996) 319–327

[16] Mitchell, T.: Machine Learning. McGraw Hill (1997)

[17] Ruiz, R., Riquelme, J., Aguilar-Ruiz, J.: Projection-based measure for efficient feature selection. Journal of Intelligent and Fuzzy System **12** (2002) 175–183

[18] Xiong, M., Fang, X., Zhao, J.: Biomarker identification by feature wrappers. Genome Res **11** (2001) 1878–87

[19] Xing, E., Jordan, M., Karp, R.: Feature selection for high-dimensional genomic microarray data. In: Proc. 18th Int. Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA (2001) 601–608

[20] Wang, H., Bell, D., Murtagh, F.: Relevance approach to feature subset selection. In: Feature extraction, construction and selection. Kluwer Academic Publishers (1998) 85–97

[21] Witten, I., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)

[22] Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of machine learning research **5** (2004) 1205–24