

Un Sistema Inteligente para Selección de Características en Clasificación

TESIS DOCTORAL

**Universidad de Granada
E.T.S. Ingeniería Informática
Departamento de Ciencias de la
Computación e Inteligencia Artificial**

Antonio Araúzo Azofra

Granada, diciembre 2006



Universidad de Granada

Escuela Técnica Superior de Ingenierías Informática y
Telecomunicación
Departamento de Ciencias de la Computación
e Inteligencia Artificial

**Un sistema inteligente para selección de
características en clasificación**

MEMORIA DE TESIS PRESENTADA POR

Antonio Araúzo Azofra

COMO REQUISITO PARA
OPTAR AL GRADO DE DOCTOR
EN INFORMÁTICA



Universidad de Granada

Escuela Técnica Superior de Ingenierías Informática y
Telecomunicación
Departamento de Ciencias de la Computación
e Inteligencia Artificial

Un sistema inteligente para selección de características en clasificación

MEMORIA DE TESIS PRESENTADA POR

Antonio Araúzo Azofra

PARA OPTAR AL GRADO DE DOCTOR EN INFORMÁTICA

DIRECTOR:

José Manuel Benítez Sánchez

Fdo.: Antonio Araúzo Azofra

Fdo.: José Manuel Benítez Sánchez

Granada, Diciembre de 2006

Dedicado a mis padres.

Agradecimientos

Parece mentira que se vea luz al final del tunel. Después de tanto tiempo, pensaba que este momento nunca llegaría, que siempre me quedaría en la parada esperando el siguiente tranvía o que nunca conseguiría subirme al correcto. Pero al final le he dado la vuelta a la canción de Sabina, he conseguido coger el tranvía que espero me lleve al barrio de la “alegría”, y en parte, os lo debo a todos vosotros. Gracias de corazón.

En primer lugar, quiero darle las gracias a mi director de tesis por haberme acogido bajo su tutela cuando aún no era más que un proyecto de ingeniero y haber confiado en mí desde el primer momento. No siempre hemos estado de acuerdo; gracias por no haberme olvidado en un rincón oscuro cuando me he empeñado en hacer las cosas a mi manera o he retrasado el trabajo; por tu apoyo, tus desvelos y las horas que hemos pasado “*skypeando*”. Gracias José Manuel.

También quiero dar las gracias a Juan Luis Castro, por ayudarme a dar mis primeros pasos en “este mundillo”, a todos mis compañeros del grupo de investigación Sistemas Inteligentes y, en especial, a Manuel Parra y Nacho Robles por la ayuda que me han prestado a lo largo de este tiempo.

Tampoco puedo olvidarme de Paco Herrera. Sus charlas y consejos me han llevado a estar hoy donde estoy. Gracias por velar por nosotros.

A mis Padres, a quienes les debo todo, por sus enseñanzas, su paciencia infinita y todos estos años de vida tan felices. Sin su apoyo, desvelos y preocupaciones me habría desinflado hace mucho tiempo y esta memoria no sería hoy una realidad. Gracias por ser los mejores.

A mi hermana, que redactaría esto mucho mejor que yo. Ester, simplemente por ser tú, gracias de todo corazón.

A mi mujer. Gracias María, porque te has desvivido en ayudarme, en librarme de otras tareas mientras escribía, porque te has llegado a estresar más que yo para que acabara, por tu amor, porque eres lo más importante en mi vida y por soportar todo el tiempo que nos ha robado esta tesis de estar juntos.

Gracias también, a Paco, Sóle y Paco, por su cálida acogida, comprensión, ánimo, cariño y buenos momentos que me han hecho pasar.

A Carlos y Belén, a Belén y Carlos, esos amigos que siempre están ahí, en los buenos y malos momentos, en las ocasiones especiales y en el día a día. A ellos, que

saben lo que hay que sufrir para llegar a esto, gracias de todo corazón.

No puedo olvidarme de mis compañeros de faena, los que compartieron mis inicios en Granada, meceneros, exmeceneros y miembros de DECSAI; y los que me aguantan ahora en Córdoba: Antonio, Lorenzo y demás compañeros del área de proyectos, gracias por dejarme escabullirme en tantas ocasiones. A Sebas por sus consejos, amistad y acogida en Córdoba; y a los amigos de la extensión del C3, “Ceuta” y “Melilla”, por haberme dejado convertirme en “Perejil”. Gracias

Por último, y no por ello menos importante, a los amigos de siempre (Nacho, Maribel, Maribelilla, Jose, Javi, Celia, Inma ...), a los que han ido apareciendo en el camino durante estos años de trabajo (Carmen, Jesús, Aznarte, Puche, M^a Ángeles, Javi, ...) y, como no, a los GCubitas (lgs, erny, karmen, enlavin, dmolina, antonioz, nauj27 ...), en fin, gracias a todos por haberme permitido desaparecer del mapa durante tanto tiempo.

Y antes de que se me escape el tranvía, por los que me dejo en el tintero, GRACIAS A TODOS.

Índice general

1. Introducción, objetivos y desarrollo	1
1.1. Motivación	2
1.2. Objetivos	4
1.3. Estructura de la memoria	5
1.4. Notación	6
1.5. Terminología	7
2. Métodos de selección de características	9
2.1. Selección de características	9
2.2. Relevancia de las características	13
2.3. Clasificaciones de los métodos de selección de características	15
2.3.1. Según su modo de aplicación	15
2.3.2. Según una división del algoritmo en partes	18
2.3.3. Según otras propiedades de los métodos	19
2.4. Descomposición modular de los métodos de selección de características	20
2.5. Métodos de búsqueda	23
2.5.1. Espacio de búsqueda	23
2.5.2. Estrategias de búsqueda	24
2.5.2.1. Búsqueda completa	27
2.5.2.2. Búsqueda secuencial	28
2.5.2.3. Búsqueda probabilística	28
2.5.2.4. Búsqueda con meta-heurísticas	29
2.6. Medidas de la utilidad de las características	32
2.6.1. Medidas sobre características individuales	33
2.6.1.1. Ganancia de información o información mutua	33
2.6.1.2. Gain ratio	34
2.6.1.3. Índice de Gini	34
2.6.1.4. Relevancia en formación de reglas	34
2.6.1.5. ReliefF	34
2.6.2. Medidas sobre conjuntos de características	36
2.6.2.1. Medidas basadas en consistencia	37
Medida de consistencia básica.	38

Medida de consistencia de Liu	39
Medida de consistencia de la teoría de Rough sets.	41
2.6.2.2. Medidas basadas en la teoría de la información	42
Información mutua.	43
Incertidumbre simétrica.	43
MDL.	44
2.6.2.3. Medidas basadas en distancia	45
2.6.2.4. Medidas basadas en estimaciones sobre el rendimiento del algoritmo de aprendizaje	46
3. Mejoras de la selección de características	47
3.1. CFOCUS	48
3.1.1. El algoritmo FOCUS original	48
3.1.2. Extensión continua (CFOCUS)	52
3.1.3. Estudio empírico	56
3.1.3.1. Problema de las figuras geométricas	56
3.1.3.2. Forest Cover Type	61
3.1.3.3. Información contable	64
3.1.3.4. Conclusiones	67
3.2. FFOCUS	68
3.2.1. Motivación	68
3.2.2. Extensión lingüística (FFOCUS)	69
3.2.3. Estudio empírico	71
3.2.3.1. Pima Indians Diabetes Database	72
3.2.3.2. Iris Plant Database	72
3.2.3.3. Conclusiones	73
3.3. Medida de los pares de ejemplos inconsistentes	74
3.3.1. Antecedentes	74
3.3.2. Definición de la medida	75
3.3.3. Propiedades	77
3.3.4. Eficiencia e implementación	77
3.4. Nueva medida de conjuntos de características basada en Relief	78
3.4.1. RFSM (Relief Feature Set Measure)	79
3.4.2. Estudio empírico	82
3.4.2.1. Conjuntos de datos artificiales	83
3.4.2.2. Conjuntos de datos del mundo real	83
3.4.2.3. Conclusiones	84
4. Evaluación empírica de métodos de selección de características	85
4.1. Métodos basados en evaluación de conjuntos de características	85
4.1.1. Parámetros de los métodos	90

4.1.1.1.	Parámetros de las medidas	90
4.1.1.2.	Parámetros de la búsqueda	90
4.2.	Métodos basados en la evaluación de características individuales . . .	92
4.2.1.	Criterios de corte	93
4.2.2.	Combinación con las medidas de características individuales . . .	97
4.3.	Metodología experimental	97
4.3.1.	Conjuntos de datos	97
4.3.2.	Clasificadores	103
4.3.2.1.	Naive-bayes	103
4.3.2.2.	Los k -vecinos más cercanos (k -NN)	104
4.3.2.3.	Árboles de clasificación	104
4.3.2.4.	Redes neuronales artificiales	104
4.3.2.5.	Parámetros elegidos en cada tipo de aprendizaje . . .	105
4.3.2.6.	Clasificadores sin selección de características	106
4.3.3.	Transformaciones de los datos	106
4.3.3.1.	<i>Discretización</i>	106
4.3.3.2.	<i>Continuización</i>	107
4.3.3.3.	Valores nulos o desconocidos	108
4.3.4.	Entorno de desarrollo y ejecución	108
4.3.4.1.	Entorno software	108
4.3.4.2.	Implementación de los métodos	109
4.3.4.3.	Servidores de cálculo	109
4.3.5.	Configuración de los experimentos	110
4.3.5.1.	Tests estadísticos	111
4.4.	Análisis de resultados	115
	Descripción de las gráficas y tablas de comparación in-	
	cludidas en este capítulo.	115
4.4.1.	Métodos que evalúan subconjuntos	117
4.4.1.1.	Resultados agrupados por medidas	117
	Ejemplos inconsistentes (Liu).	117
	Consistencia de Rough Sets (RSC).	117
	Pares de ejemplos inconsistentes (IEP).	117
	Información mutua (Inf).	121
	Incertidumbre simétrica (SU).	121
	Longitud mínima de la descripción (MDL).	121
	Medida de conjuntos basada en Relief (RFS).	124
	Envolvente (Wra).	124
4.4.1.2.	Resultados agrupados por métodos de búsqueda . . .	124
	Búsqueda exhaustiva (Ex).	128
	Ramificación y poda (BB).	128
	ABB.	128

Ascensión de colinas (SFS).	128
SBS	133
Filtro de Las Vegas (LVF).	133
Envolvente de Las Vegas (LVW).	133
Enfriamiento simulado (SA).	133
Algoritmos genéticos (GA).	138
Búsqueda tabú (TS).	138
4.4.1.3. Comparativa global de métodos de subconjuntos	138
4.4.2. Métodos que evalúan características individuales	147
4.4.2.1. Resultados agrupados por criterio de punto de corte	147
Las n mejores características (n17).	147
Una fracción de las características disponibles (p0.8).	147
Valoración por encima de un umbral (t0.1).	147
Valoración por encima de una fracción del ranking de las valoraciones (pm0.8).	147
Bajada de valoración absoluta (d0.2).	148
Bajada de valoración relativa (s1.5).	148
4.4.2.2. Resultados agrupados por medidas	148
Información mutua (info)	148
Gain ratio (gain)	148
Índice de Gini (gini)	148
Valoración de Relief-F (reli)	148
Potencial en la formación de reglas de decisión (rele)	148
4.4.2.3. Comparativa global de métodos individuales	160
4.4.3. Comparativa global de todos los métodos	160
4.4.4. Mejora del porcentaje de acierto al aplicar selección de carac- terísticas	163
4.4.5. Diferentes resultados según tipos de problemas	164
5. Sistema inteligente para selección de características	167
5.1. Requisitos del sistema	168
5.2. Diseño	170
5.2.1. Base de conocimiento	171
5.2.2. Motor de razonamiento	173
5.2.3. Módulo de justificación	173
5.3. Adquisición del conocimiento	174
5.3.1. Conocimiento de expertos	174
5.3.2. Aprendizaje Automático	175
5.4. Implementación y prueba	177
5.5. Validación del modelo	178
Clasificación de aceites de oliva	178

6. Conclusiones y líneas futuras de investigación	181
6.1. Conclusiones	181
6.2. Líneas de trabajo futuras	184
A. Tablas con los resultados experimentales	187
A.1. Muestra de los resultados experimentales	187
A.2. Otras tablas y figuras	194
B. Software desarrollado	199
B.1. Metodología de desarrollo	199
B.2. Dimensión	200
B.3. Herramientas	200
B.4. OrangeSNNS	201
Bibliografía	205

Índice de tablas

2.1. Medidas de la relevancia de conjuntos de características	37
3.1. Características seleccionadas en cada muestra de las figuras geométricas por CFOCUS con $U=0.025$	59
3.2. Características seleccionadas en cada muestra de las figuras geométricas por CFOCUS con $U=0.05$	59
3.3. Características seleccionadas en cada muestra de las figuras geométricas por CFOCUS con $U=0.1$	60
3.4. Características seleccionadas en cada muestra de las figuras geométricas por CFOCUS con $U=0.2$	60
3.5. Resultados de clasificación en Forest sin selección de características (% de acierto)	62
3.6. Resultados de clasificación en Forest usando CFOCUS (% de acierto)	63
3.7. Resultados de clasificación en Forest usando RELIEF-E (% de acierto)	63
3.8. Características seleccionadas para predicción contable por CFOCUS .	65
3.9. Resultados de clasificación obtenidos con CFOCUS y redes neuronales para el problema de predicción contable	66
3.10. Modelo obtenido con regresión logística para el problema de predicción contable	66
3.11. Resultados de clasificación obtenidos con regresión logística para el problema de predicción contable	66
3.12. Resultados de clasificación usando FFOCUS en Pima (% de acierto) .	72
3.13. Resultados de clasificación usando FFOCUS en Iris (% de acierto) . .	73
3.14. Rangos, cotas y valores interesantes de las medidas de inconsistencia	75
3.15. Comparación del número de características seleccionadas usando tres medidas diferentes en conjuntos de datos artificiales (\pm Err.std.) . . .	83
3.16. Comparación de RFSM con otros métodos en clasificación de problemas reales (% de acierto)	84
4.1. Métodos de selección de características ordenados por tipo de búsqueda y medida de relevancia	86
4.2. Principales características de los métodos de búsqueda (1/2)	88

4.3. Principales características de los métodos de búsqueda (2/2)	89
4.4. Conjuntos de datos (1/4)	99
4.5. Conjuntos de datos (2/4)	100
4.6. Conjuntos de datos (3/4)	101
4.7. Conjuntos de datos (4/4)	102
4.8. Rendimiento medio de los métodos de clasificación sin selección de características	106
4.9. Ejemplo de aplicación de los tests estadísticos para comparar 4 me- didas en reducción de características	113
4.10. Comparación de métodos de búsqueda para la medida Liu	118
4.11. Comparación de métodos de búsqueda para la medida RSC	119
4.12. Comparación de métodos de búsqueda para la medida IEP	120
4.13. Comparación de métodos de búsqueda para la medida Inf	122
4.14. Comparación de métodos de búsqueda para la medida SU	123
4.15. Comparación de métodos de búsqueda para la medida MDL	125
4.16. Comparación de métodos de búsqueda para la medida RFS	126
4.17. Comparación de métodos de búsqueda para la medida Wra	127
4.18. Comparación de medidas para el método de búsqueda Ex	129
4.19. Comparación de medidas para el método de búsqueda BB	130
4.20. Comparación de medidas para el método de búsqueda ABB	131
4.21. Comparación de medidas para el método de búsqueda SFS	132
4.22. Comparación de medidas para el método de búsqueda SBS	134
4.23. Comparación de medidas para el método de búsqueda LVF	135
4.24. Comparación de medidas para el método de búsqueda LVW	136
4.25. Comparación de medidas para el método de búsqueda SA	137
4.26. Comparación de medidas para el método de búsqueda GA	139
4.27. Comparación de medidas para el método de búsqueda TS	140
4.28. Comparación de métodos basados en evaluación de conjuntos de ca- racterísticas sobre 35 conjuntos de datos	145
4.29. Comparación de todos los métodos basados en evaluación de conjun- tos de características sobre 20 conjuntos de datos	147
4.30. Comparación de medidas para el método de corte n17	149
4.31. Comparación de medidas para el método de corte p0.8	150
4.32. Comparación de medidas para el método de corte t0.1	151
4.33. Comparación de medidas para el método de corte pm0.8	152
4.34. Comparación de medidas para el método de corte d0.2	153
4.35. Comparación de medidas para el método de corte s1.5	154
4.36. Comparación de métodos de búsqueda para la medida info	155
4.37. Comparación de métodos de búsqueda para la medida gain	156
4.38. Comparación de métodos de búsqueda para la medida gini	157
4.39. Comparación de métodos de búsqueda para la medida reli	158

4.40. Comparación de métodos de búsqueda para la medida rele	159
4.41. Comparación de los métodos de selección de características que usan valoración individual de las características para cada algoritmos de aprendizaje	161
4.42. Comparación conjunta de los mejores métodos de selección de características que usan valoración individual o de subconjuntos	162
5.1. Resultados obtenidos con el sistema experto en un problema de clasificación de aceites	180
A.1. Muestra de los resultados detallados de la batería de experimentos descrita en el capítulo 4	194
A.2. Rendimiento de los métodos de clasificación sin selección de características en cada conjunto de datos	195
A.3. Comparación de los métodos de selección de características que usan valoración individual de las características	196
A.4. Comparación de porcentaje de acierto del mejor método en global y no realizar la selección de características	197

Índice de figuras

2.1. Proceso de aprendizaje	10
2.2. Proyección de los datos sobre el conjunto S de características	11
2.3. Aplicación de la selección de características	11
2.4. Selección <i>vs.</i> Extracción de características	12
2.5. Relevancia de las características	15
2.6. Selección de características de tipo filtro	16
2.7. Selección de características envolvente del algoritmo de aprendizaje	17
2.8. Selección de características inmersa en el algoritmo de aprendizaje	17
2.9. Taxonomía de Jain y Zongker para los métodos de selección de características	19
2.10. Descomposición modular del proceso de selección de características	21
2.11. Diagrama Hasse del retículo formado por $\mathcal{P}(\{a, b, c, d, e\})$	24
2.12. Número de conjuntos de tamaño x en el espacio de búsqueda de N características: a) $N = 5$, b) $N = 10$, c) $N = 20$, y d) $N = 50$	25
2.13. Comparación sobre escala logarítmica del número de conjuntos de tamaño x en los espacios de búsqueda de 5, 10, 20 y 50 características	26
2.14. Algoritmo de enfriamiento simulado	30
2.15. Algoritmo Genético	31
2.16. Algoritmo general de todas las versiones de Relief	35
3.1. Ejemplos de un conjunto de datos y sus respectivos conflictos	49
3.2. Algoritmo FOCUS	50
3.3. Algoritmo FOCUS-2	51
3.4. Figuras geométricas y los valores de sus características	57
3.5. Lista de características del problema de predicción contable	64
3.6. Aplicación final de CFOCUS en el proceso de aprendizaje del problema de predicción contable	65
3.7. Metodología de filtro compartiendo información lingüística	69
3.8. Elección de etiquetas: a) a ciegas <i>vs.</i> b) realizada por un experto.	70
3.9. Medida de separación de Castro [Sán00, CSCZ04]	71
3.10. Algoritmo para calcular la medida de pares de ejemplos inconsistentes	78

3.11. Ejemplo de cálculo rápido de las medidas de pares de ejemplos inconsistentes y de Liu	79
4.1. Valoración de características con una medida individual	93
4.2. Criterio de corte: las m características más relevantes	95
4.3. Criterio de corte: un porcentaje de las características disponibles . . .	95
4.4. Criterio de corte: características cuya valoración sea superior a un umbral	95
4.5. Criterio de corte: umbral como porcentaje de la variación máxima . .	96
4.6. Criterio de corte: hasta una diferencia superior a un umbral dado . .	96
4.7. Criterio de corte: hasta una pendiente superior a un umbral dado . .	96
4.8. Envolverte convexa según métodos de búsqueda para C45	142
4.9. Envolverte convexa según tipo de medida para C45	143
4.10. Envolverte convexa de todos los métodos para cada clasificador . . .	162
4.11. Envolverte convexa según tipo de método para C45	163
4.12. Comparación de métodos de búsqueda para la medida Liu en problemas con menos de 8 características	164
4.13. Comparación de métodos de búsqueda para la medida Liu en problemas con entre 9 y 20 características	165
4.14. Comparación de métodos de búsqueda para la medida Liu en problemas con más de 20 características	166
5.1. Esquema del sistema experto desarrollado	170
5.2. Etiquetas lingüísticas de la variable que indica el número de características del problema	171
5.3. Etiquetas lingüísticas de la variable que indica el número de ejemplos del problema	172
5.4. Etiqueta lingüística de las variables que indican la importancia de cada uno de los objetivos de la selección de características	172
5.5. Regiones para la generación de reglas basadas en los mejores resultados obtenidos	176
5.6. Regiones para la generación de reglas basadas en el menor tiempo de ejecución	177
5.7. Reglas disparadas en el sistema experto para el problema de clasificación de aceites	179
5.8. Métodos recomendados por el sistema experto para el problema de clasificación de aceites	180
B.1. Página principal de presentación de OrangeSNNS (1/2)	202
B.2. Página principal de presentación de OrangeSNNS (2/2)	203
B.3. Diagrama describiendo el funcionamiento de OrangeSNNS	204

Capítulo 1

Introducción, objetivos y desarrollo

La meta de todo problema de clasificación es encontrar un modelo que use como entradas distintas variables sobre el problema y dé como respuesta la clase de entre un número finito de ellas a la que se asigna el dato cuyos valores se usan como entradas. En todas las áreas del conocimiento aparecen problemas de clasificación que han de ser resueltos. Encontrar buenos clasificadores en una cuestión de importancia capital. Sin embargo, no se conoce ninguna solución de naturaleza puramente algorítmica para este problema. Casi siempre hay que recurrir a técnicas de Inteligencia Artificial.

La situación más habitual es que la principal fuente de información para poder construir el clasificador sea un conjunto de datos ya clasificados. En ese caso, se suele recurrir a técnicas de Aprendizaje Automático. Cada ejemplo usado para el aprendizaje viene descrito por un conjunto de características. Con el avance de la tecnología, cada vez se abordan problemas más complejos, y se dispone de mucha más información para abordarlos. Por tanto, aparecen frecuentemente problemas en los que se parte de ejemplos con un elevado número de características y, puesto que se pretende aprender de los ejemplos, normalmente se desconoce *a priori* cuáles de estas características pueden ser de utilidad.

En principio, lo ideal para la resolución de un problema de clasificación es disponer de la máxima información posible. Así, puede parecer que cuanto mayor sea el número de características empleadas mejor. En cambio, en la práctica, esto no es siempre así [Koh94]. El rendimiento de los algoritmos de aprendizaje reales se puede deteriorar ante la abundancia de información. Muchas características pueden ser

completamente irrelevantes para el problema. Además, varias características redundantes pueden estar proporcionando la misma información. Por tanto, podríamos prescindir perfectamente de algunas características sin perjudicar nada la resolución del problema, e incluso mejorar los resultados en diversos aspectos y reducir varios tipos de costes; ventajas que detallamos más adelante. En definitiva, interesa realizar una *selección de características*.

La utilización de la selección de características se puede ver como una aplicación de la estrategia de Divide y Vencerás. Ante un problema, esta estrategia propone su división en partes para resolver cada una de ellas individualmente. Normalmente, es más sencillo atacar la resolución de las divisiones que el problema completo. La selección de características sería una de estas partes del problema de construir un clasificador.

Además de en la construcción de clasificadores, la selección de características es empleada en todas las aplicaciones del Aprendizaje Automático y en muchas otras áreas, como por ejemplo: Estadística, Reconocimiento de Patrones, Minería de Datos, Predicción y Modelado de Sistemas. Tiene aplicaciones en campos muy diversos [LY05] como en: categorización de textos, recuperación de imágenes, gestión de las relaciones con los clientes [MZ06], detección de intrusos, análisis genético, clasificación de vídeo [MC04], detección de fallos en maquinaria [SABAA03], y otros muchos.

1.1. Motivación

La utilización de métodos de selección de características en el desarrollo de clasificadores, en particular, y en un proceso de aprendizaje, en general, puede aportar algunas de las siguientes ventajas:

- La eficiencia (en tiempo y/o en espacio) de la mayoría de los algoritmos de aprendizaje depende del número de características empleado. Por tanto, seleccionando un conjunto de características más pequeño el algoritmo funcionará más rápido y/o con menor consumo de memoria u otros recursos.
- Mejora en los resultados obtenidos: Algunos de los algoritmos de aprendizaje, que trabajan muy bien con pocas características relevantes, ante la abundancia de información pueden tratar de usar características irrelevantes y ser confundidos por las mismas, ofreciendo resultados peores. Así que la selección

de características puede ayudar a obtener mejores resultados a un algoritmo indicándole en qué características centrar la atención.

- Reducción de los costes de adquisición de datos: En ciertas aplicaciones, la medición de las características supone un coste no despreciable, por ejemplo, si la medición exige la realización de análisis químicos. La exclusión de características no necesarias puede suponer un ahorro importante.
- Aplicabilidad de ciertas técnicas: Ciertos algoritmos de aprendizaje, como por ejemplo los basados en reglas difusas, sufren de la maldición de la dimensión creciente (*curse-of-dimensionality*). Su eficiencia viene dada por alguna función del número de características con crecimiento muy rápido (por ejemplo, de orden exponencial). Esto hace que no sean aplicables a problemas con un gran número de características. En consecuencia, la selección de características amplía el campo de aplicación de técnicas que de otro modo no serían aplicables a gran cantidad de problemas.
- Mejora de la interpretabilidad: Algunas técnicas de aprendizaje proporcionan como salida información comprensible para el ser humano. Esta información suele ser mucho más fácil de interpretar y comprender si está basada en un número reducido de características.
- Reducción de los recursos necesarios para el almacenamiento y transmisión de la información de las características no seleccionadas.
- Evitar repetir partes del proceso de aprendizaje: Hay ocasiones en las que el problema requiere la repetición del proceso aprendizaje numerosas veces, pero puede no ser necesario repetir la selección de características. Dividiendo el problema en selección de características y resto del aprendizaje, se evita repetir una de las partes del proceso.

La selección de características es un problema que continúa plenamente vigente porque, siendo una etapa fundamental en cualquier proceso de modelado, aún no se ha encontrado una solución definitiva. Debido a esto, siguen siendo interesantes las aportaciones, y continuamente están apareciendo trabajos con nuevas propuestas o explicando la aplicación de la selección de características en nuevos entornos.

De las propuestas existentes, algunas representan soluciones interesantes para una amplia variedad de problemas, y cualquier mejora en su rendimiento resulta beneficiosa. Otras propuestas son de aplicación limitada por las restricciones que plantean; extenderlas simplemente para que puedan ser utilizadas en otros tipos de problemas también supone una aportación útil.

Ningún método de los propuestos hasta ahora resuelve completamente el problema, es decir, no se ha encontrado un método de selección de características definitivo y universalmente válido. Una de las razones para ello es la complejidad que implica este problema. Esta complejidad hace sospechar que no existe esa solución universal. Podría pensarse que la estrategia envolvente [KJ97] es esa solución general, sin embargo, no es así. Primero porque no es aplicable en muchos problemas, y en segundo lugar, porque no siempre es la estrategia óptima para guiar la búsqueda, como mostramos en este trabajo.

Para poder encontrar vías alternativas de solución es requisito imprescindible conocer el problema a fondo. La mayoría de los trabajos publicados ofrecen una perspectiva parcial y sesgada, en función del campo de trabajo y los intereses de los autores. Las escasas monografías y trabajos de revisión tampoco han ofrecido una perspectiva completa del problema de selección de características dejándose sin explorar ni evaluar algunas de las posibilidades disponibles.

La complejidad del problema de la selección de características hace difícil abordar su solución desde una perspectiva puramente teórica. Hemos de recurrir entonces a la vía experimental para conocer el comportamiento de los métodos sobre datos de problemas reales. Estamos convencidos de que una solución universal entendida como un algoritmo tradicional no existe para el problema de la selección de características. Las soluciones hay que buscarlas a través de otros enfoques. La Inteligencia Computacional* ofrece una alternativa que se ha mostrado eficaz en la resolución de problemas complejos.

1.2. Objetivos

Los objetivos que nos planteamos alcanzar tras el desarrollo de esta tesis doctoral son:

- **Estudiar a fondo el problema de la selección de características.** Si bien, no parece posible encontrar una solución definitiva, aportar una perspectiva global y ordenada permitirá encontrar nuevas vías de solución. Pretendemos profundizar en la naturaleza del problema y exponer todas sus características.

* *Computational Intelligence*, en inglés

- **Mejorar algunos de los métodos propuestos.** Para algunos métodos limitados a problemas muy restringidos, nos proponemos ampliar su campo de aplicación a problemas con características más complejas. También fijamos como objetivo mejorar el rendimiento de otros métodos (permitiendo que detecten características redundantes) y mejorar la eficiencia computacional de la implementación de algunas técnicas.
- **Definir un esquema modular en el que tengan cabida todos los métodos de selección de características,** que nos permita la implementación de métodos existentes, la propuesta de nuevos métodos por combinación de sus módulos, y la aplicación de técnicas de otras áreas a la selección de características.
- **Evaluar experimentalmente diversos métodos de selección de características.** Pretendemos determinar cuáles funcionan mejor sobre un conjunto de problemas reales.
 - **Diversidad:** Utilizaremos el esquema modular definido para evaluar métodos que representen a todas las tipologías.
 - **Rigurosidad:** Además, pretendemos que el rigor sea una propiedad inherente a este estudio insistiendo en la aplicación de las últimas propuestas de metodologías empíricas que respalden con las máximas garantías estadísticas la calidad de los resultados alcanzados.
- **Uso de la inteligencia computacional para la elección de métodos de selección de características.** Aunando el conocimiento experto adquirido con el desarrollo de los objetivos anteriores, junto con técnicas de modelado basadas en aprendizaje, nos proponemos diseñar un sistema inteligente que ayude en la elección de un método de selección de características adecuado para un problema concreto.

1.3. Estructura de la memoria

Esta memoria está compuesta por la presente introducción, cinco capítulos más, en los que se desarrolla la investigación realizada, y un capítulo de comentarios finales con las conclusiones generales y los trabajos futuros. Se adjuntan dos apéndices, uno con tablas adicionales de resultados y otro con detalles sobre el software desarrollado.

En el capítulo 2, se hace un estudio a fondo de la selección de características. En primer lugar, introducimos el problema de la selección de características en sis-

temas de clasificación, y revisamos el concepto de relevancia de las características. A continuación, se explican las clasificaciones publicadas de los métodos de selección de características, para seguidamente proponer una descomposición modular del proceso de selección. Finalmente, se revisan algunos de los métodos de búsqueda y medidas de relevancia que pueden ser usadas en selección de características.

En el capítulo 3, proponemos cuatro mejoras para selección de características. Las dos primeras son extensiones para ampliar el campo de aplicación del algoritmo FOCUS [AD94], y las dos siguientes son medidas de la relevancia de un conjunto de características.

En el capítulo 4, realizamos un estudio empírico que abarca la práctica totalidad del espectro de métodos de selección de características. Se consideran dos grandes grupos de métodos, los que realizan una evaluación individual de las características y los que trabajan con conjuntos. Dentro de ellos, hay métodos ya descritos previamente en la bibliografía y otros que son nuevas propuestas creadas por combinación de módulos. Se describe la metodología seguida y los resultados obtenidos. Y finalmente, se realiza un detenido y riguroso análisis de los mismos.

En el capítulo 5, se describe el uso de los resultados analizados en el capítulo 4 para diseñar un sistema inteligente que ayude en la elección y aplicación de un método de selección de características.

En el último capítulo, se resumen los resultados obtenidos en esta memoria, comentando los mismos y planteando algunos de los trabajos futuros que se pueden abordar en el área.

1.4. Notación

Sin perjuicio de que en cada ilustración, tabla o fórmula se usen y expliquen los símbolos específicos necesarios, se pretende dar homogeneidad a la memoria. Para ello, se han usado letras minúsculas para denotar elementos de un conjunto y letras mayúsculas para los conjuntos. Además, se mantienen constantes los siguientes símbolos:

- $s \in S$ una característica perteneciente al conjunto de características seleccionadas (S).

- $f \in F$ una característica perteneciente al conjunto de todas las características (F).
- $e \in E$ un ejemplo perteneciente a los datos que conocemos del problema de clasificación (E).
- T conjunto de ejemplos de entrenamiento.
- $n = |F|$ el número de características total.
- $m = |E|$ número de ejemplos en los datos.

1.5. Terminología

Aunque no aparecen en el diccionario de la RAE, se han utilizado los verbos *Discretizar* y *Continuizar* junto con sus derivados, para significar convertir los datos en discretos y continuos respectivamente. Lo hemos hecho a semejanza de los términos ingleses *Discretize* y *Continuize* que indican de forma concreta y clara un proceso habitual y, pese a no aparecer en un diccionario general como Merriam-Webster, son términos aceptados en el área.

En un problema de clasificación, entre las diversas formas de denominar a cada tupla de una tabla de datos usada para el aprendizaje hemos elegido “ejemplo”. Frente a otras denominaciones posibles como “caso” o “instancia”, nuestra elección incluye en su significado la connotación de servir de ilustración o explicar el problema del que queremos aprender. Descartamos usar la palabra “muestra”, porque ésta, aun en singular, puede significar varias tuplas.

Capítulo 2

Métodos de selección de características

Todas las ventajas mencionadas en el capítulo anterior han atraído la atención de la comunidad científica del ámbito del Aprendizaje Automático, lo que ha motivado que se hayan desarrollado multitud de métodos de selección de características. Para caracterizarlos, se han propuesto algunas clasificaciones [DL97, JZ97, Lan94].

En este capítulo, centramos y definimos concretamente el problema de selección de características para clasificación que abordamos en esta tesis. A continuación, describimos las clasificaciones mencionadas, para proponer posteriormente una descomposición modular del proceso de selección de características. Finalmente, usando esta descomposición, describimos los métodos de selección de características más conocidos, dejando abierta la posibilidad de crear otros métodos mediante la combinación de sus módulos.

2.1. Selección de características

Normalmente, para aplicar Aprendizaje Automático en un problema, seguimos el proceso reflejado en la figura 2.1. En un sistema de aprendizaje, pueden utilizarse diversas herramientas en combinación con el propio algoritmo de aprendizaje, como por ejemplo: la *discretización* o *continuización* de características, la sustitución de valores nulos, la selección de ejemplos [BL97, LA04], o la extracción de características derivadas. Todas estas herramientas ayudan al proceso de aprendizaje, ya

sea permitiendo aplicar técnicas diferentes, mejorando los resultados, o reduciendo costes. La selección de características forma parte de este grupo de herramientas.

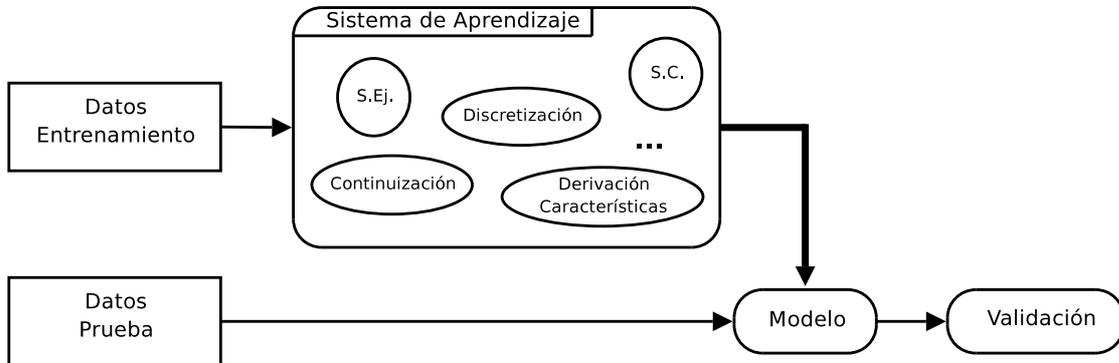


Figura 2.1: Proceso de aprendizaje

En la figura 2.1, hemos denominado “modelo” a la salida del sistema de aprendizaje para recoger todas las posibles formas de conocimiento aprendido (clasificadores, regresores, deductores, ...). También se muestra el proceso de validación sobre datos independientes, que debe ser aplicado sobre todo sistema de aprendizaje serio. Esta validación la suponemos externa al proceso de aprendizaje y, por tanto, no se incluye en los diagramas posteriores para permitir mayor claridad y evitar confusiones.

En esta tesis, trabajaremos con modelos basados en aprendizaje supervisado (aquél en el que se tienen ejemplos que incluyen como muestra el valor a aprender) y, de forma más específica, nos centraremos en los problemas de clasificación (valor a aprender categórico). Aunque hay muchas similitudes, y técnicas que también son aplicables en los otros tipos de aprendizaje o de problemas, en adelante, nos referiremos únicamente a los que son objeto de esta tesis.

Los problemas de aprendizaje y los tipos de ejemplos disponibles pueden ser muy diversos (series temporales, documentos estructurados, imágenes, ...), pero es muy frecuente que se trabaje con tuplas de datos, bien porque el problema venga especificado de esta forma, o bien, porque se realice alguna transformación de los datos más complejos para trabajar con tuplas. Una tupla es un vector con los valores categóricos o numéricos para cada característica del ejemplo. Finalmente, el conjunto de datos con el que se trabaja (ver ejemplo en figura 2.2) es similar a una tabla del modelo relacional [SK93].

El objetivo de la selección de características es reducir la dimensión de los datos. Esto se consigue eligiendo características que sean útiles para resolver el problema de aprendizaje y descartando las demás. Usando el modelo relacional, una

$F =$	$\{f_1$	f_2	f_3	\dots	$f_n\}$	Cl
e_1	8.1	Sí	ESO	...	Casad.	A
e_2	3.4	No	Diplom.	...	Divor.	B
e_3	-2.5	No	Doctor	...	Solt.	A
			\vdots			
e_n	-1.0	No	Ingeni.	...	Viud.	C

 $\xrightarrow{\Pi_S}$

$S =$	$\{f_1$	$f_3\}$	Cl
e_1	8.1	ESO	A
e_2	3.4	Diplom.	B
e_3	-2.5	Doctor	A
		\vdots	
e_n	-1.0	Ingeni.	C

Figura 2.2: Proyección de los datos sobre el conjunto S de características

vez elegido un conjunto de características (S), la operación de proyección reduce el conjunto de datos, como se puede ver en la figura 2.2. Aunque, teóricamente, si la distribución estadística completa se conociese, usar más características sólo podría mejorar los resultados, en los escenarios prácticos de aprendizaje puede ser mejor usar un conjunto reducido de características [KJ97]. Además, la selección de características nos puede aportar las ventajas ya mencionadas en el apartado 1.1.

La selección de características se denomina, a veces, “selección de subconjuntos de características” para resaltar el hecho de que se elige un subconjunto relacionado frente a la elección de varias características de forma individual.

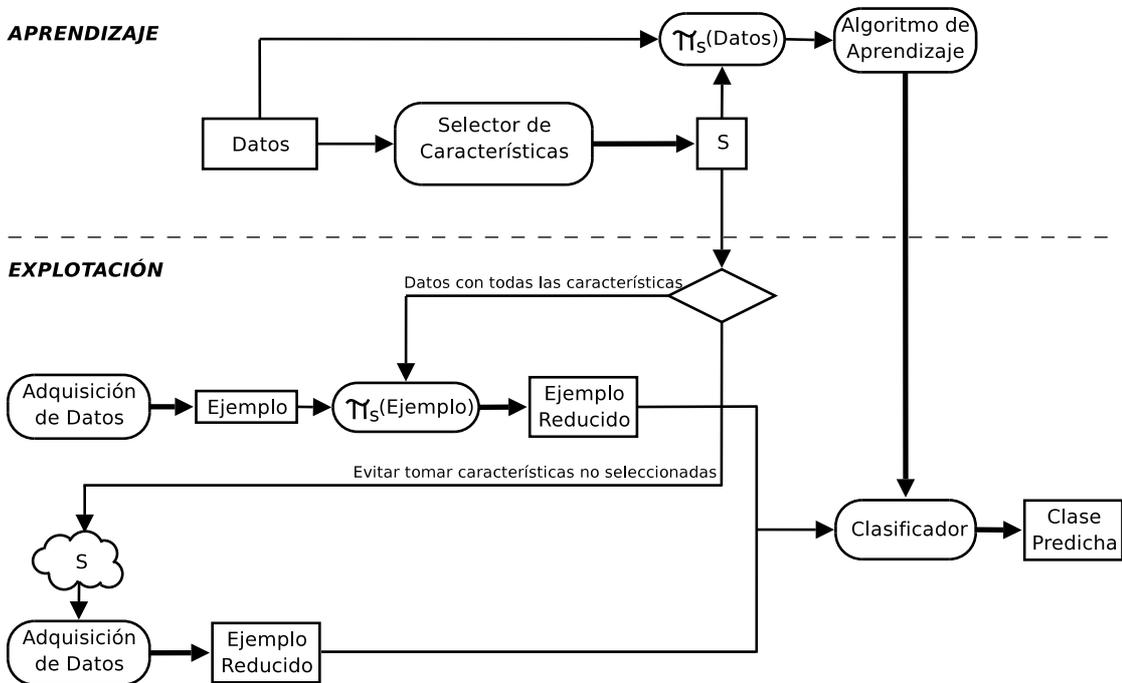


Figura 2.3: Aplicación de la selección de características

En la figura 2.3 se ilustra la aplicación de los métodos de selección de características en problemas de clasificación. En la fase de aprendizaje, pretendemos

obtener un modelo que realizará la clasificación (clasificador). El método de selección de características elige el subconjunto de características S ; con él se realizará la proyección de los datos que se pasan al algoritmo de aprendizaje. Después de la aplicación de éste, como resultado del aprendizaje, tenemos un clasificador y un conjunto de características seleccionadas. En la fase de explotación, pretendemos averiguar la clase a la que pertenece un ejemplo usando el clasificador aprendido. Hay dos posibilidades. La primera es que los datos siempre tengan todas las características. En este caso, aplicaremos la proyección del conjunto S sobre el ejemplo. La segunda posibilidad es que podamos beneficiarnos de la selección de características para reducir la adquisición de datos. En este caso, sólo se tomarán los datos de las características seleccionadas del ejemplo a clasificar.

Hay otras técnicas relacionadas con la selección de características pero distintas. Con el mismo objetivo, reducir la dimensionalidad, tenemos la extracción de características [PN98]. Ésta, a partir de las características originales, crea unas nuevas características derivadas, que pretenden resumir el conocimiento aportado por las anteriores (ver figura 2.4). Se emplean, por ejemplo, el análisis de componentes principales (*PCA*), el análisis de factores y, más enfocado a clasificación, el análisis discriminante.

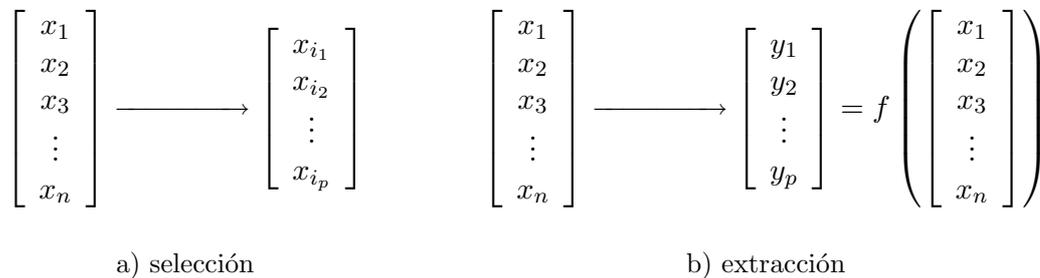


Figura 2.4: Selección *vs.* Extracción de características

Por otra parte, otra técnica que también evalúa las características es la ponderación de características [WAM97]. En este caso, lo que se busca es asignar un peso adecuado a cada característica que ayude a ciertos algoritmos de aprendizaje (por ejemplo kNN) a trabajar con las características.

En ambas técnicas, la selección de características puede ser vista como un caso específico de la misma. Respecto a la ponderación, la selección de características es una ponderación con pesos binarios, indicando cada uno si la característica está seleccionada o no. Respecto a la extracción, en las transformaciones de la extracción de características, la selección de características se puede alcanzar con una matriz de ceros con unos cuantos unos indicando qué características se incluyen. Sin embargo, esto no significa que debamos enfocar la solución de la selección de características

mirando los problemas más generales. En la práctica, estos problemas se resuelven de formas muy distintas. Al ser más específico, el punto de vista de la selección de características nos permite centrarnos más en nuestro objetivo. Hay una gran cantidad de métodos específicos para selección, que habrían sido muy difíciles de crear pensando en los problemas generales.

Otra técnica que tiene el potencial de realizar selección de características, aun no siendo ese su objetivo, es la *discretización* [SL95]. Al ir dividiendo en intervalos las características numéricas, los métodos relativamente avanzados pueden decidir no dividir algunas características. Estas características con un solo intervalo, sólo tendrán un valor para el algoritmo de aprendizaje, tornándose inútiles. Así, lo más indicado es eliminarlas y, por tanto, estaremos realizando selección de características junto con la *discretización*.

La extracción de características es más potente que la selección de características a la hora de concentrar el conocimiento en unas pocas características. No obstante, hay varias razones que pueden aconsejar el uso de la selección. Ésta permite descartar completamente características, y así, por ejemplo, reducir el coste de adquisición de datos. Además, es posible mantener la interpretabilidad de los datos, cosa que difícilmente ocurre con las características transformadas. Otra ventaja de la selección es que trabaja sin problemas con características no numéricas. Finalmente, la decisión de elegir entre una y otra es también una cuestión de qué parte del trabajo interesa hacer en el preprocesado, y qué parte le dejamos al algoritmo de aprendizaje, por lo que, debemos tener en cuenta las propiedades del algoritmo de aprendizaje para evitar hacer un trabajo que puede que haga él mejor. Esta decisión es algo que debe estudiarse en cada problema concreto, pues todas las técnicas mencionadas tienen su ámbito idóneo de aplicación.

2.2. Relevancia de las características

La selección de características normalmente persigue elegir aquellas características que son relevantes para el problema. Por ello, es interesante prestar algo de atención al concepto de relevancia, ya que nos puede ayudar a la hora de diseñar y comprender mejor los métodos de selección de características. Hay diversas definiciones de característica relevante. La más directa, por venir de la definición de dependencia estadística, es la siguiente.

Definición 1

F_i es relevante sii $\exists v, y$ con $p(F_i = v) > 0$ tales que

$$p(Y = y|F_i = v) \neq p(Y = y)$$

donde F_i e Y son variables aleatorias que representan la característica f_i y la clase respectivamente.

El inconveniente de esta definición es que sólo tiene en cuenta la característica de forma individual, cuando en realidad, varias características influyen simultáneamente en el valor de la clase. Así, por ejemplo, si dos características binarias F_1 y F_2 determinan el resultado Y mediante la función XOR, suponiendo valores equiprobables, ambas características serían irrelevantes según la definición 1. Necesitamos, por tanto, una definición de relevancia de características en el problema como un todo.

Hay varios trabajos que revisan y estudian las definiciones de relevancia [JKP94, Lan94, BL97, KJ97, LA04]. Finalmente en ellos se propone la definición de dos tipos de relevancia: fuerte y débil. Definiciones claras y comúnmente aceptadas, que exponemos a continuación.

Entendemos que una característica es fuertemente relevante cuando ésta es imprescindible, entendiendo esto como que ninguna otra característica puede reemplazar su aportación y, por tanto, su eliminación supone pérdida de capacidad de predicción.

Sea R_i el conjunto de las variables aleatorias de todas las características excepto f_i , esto es $R = F - F_i$. Denotaremos r a una asignación de valores a todas las características de R_i .

Definición 2 (Relevancia fuerte)

F_i es fuertemente relevante sii $\exists v, y, r$ con $p(F_i = v, R_i = r) > 0$ tales que

$$p(Y = y|F_i = v, R_i = r) \neq p(Y = y|R_i = r)$$

Esta definición deja fuera a características potencialmente relevantes, que podrían llegar a ser imprescindibles, si no se incluyen otras que aporten su misma información; son las que llamamos débilmente relevantes. Por ejemplo, en un grupo de características redundantes, ninguna de ellas será fuertemente relevante, pero puede ser necesario que alguna de ellas esté presente para no perder capacidad de predicción.

Definición 3 (Relevancia débil)

F_i es débilmente relevante sii no es fuertemente relevante y $\exists R'_i \subset R_i$ para el que

$\exists v, y, r'$ con $p(F_i = v, R'_i = r') > 0$ tales que

$$p(Y = y|F_i = v, R'_i = r') \neq p(Y = y|R'_i = r')$$

Se consideran irrelevantes aquellas características que no son relevantes, ni fuertemente ni débilmente. En la figura 2.5, se muestra el espacio de características de un problema según su relevancia. Se puede ver como la selección de características óptima incluye todas las características fuertemente relevantes y algunas débilmente relevantes, considerando óptima aquella selección que aporta máxima capacidad de predicción con el menor número de características.

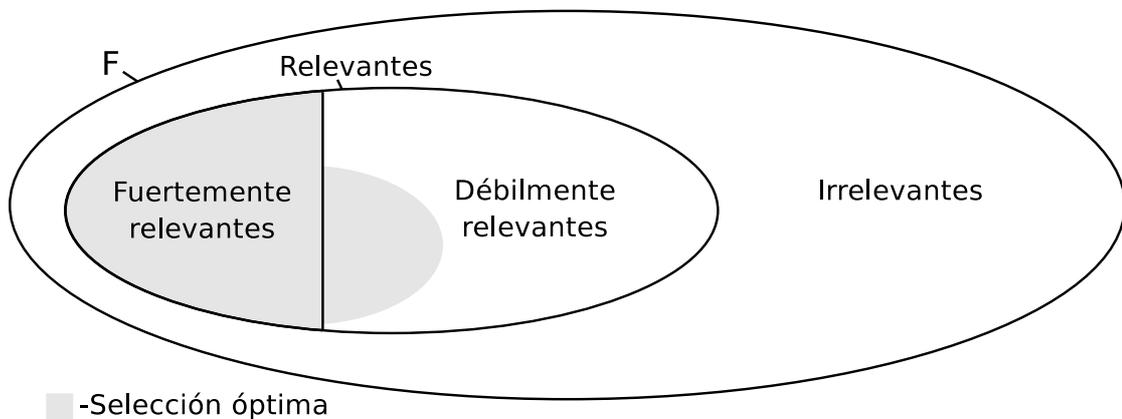


Figura 2.5: Relevancia de las características

2.3. Clasificaciones de los métodos de selección de características

Claramente, la selección de características es un problema muy importante en diversas parcelas de la Inteligencia Artificial. Por eso, muchos investigadores han realizado numerosos esfuerzos para proponer métodos que lo resuelvan. Es interesante ver formas de clasificar los diversos métodos disponibles, para conocerlos mejor y tener una visión ordenada de todas las posibilidades.

2.3.1. Según su modo de aplicación

Una forma de clasificarlos, ampliamente aceptada, es según su modo de aplicación [BL97]. Podemos distinguir las siguientes categorías:

- La metodología de **filtro** (*filter*) es probablemente la primera y más conocida. En ella, se aplica primero el algoritmo de selección de características y, posteriormente, el de aprendizaje empleando sólo las características seleccionadas. La única información que intercambian es el conjunto de características, lo que aporta la principal ventaja de este tipo de métodos: que son independientes del algoritmo de aprendizaje. Por ello, pueden ser utilizados con cualquier algoritmo de aprendizaje, independientemente de su eficiencia u otras propiedades, que sí afectan a otros modos de aplicación.

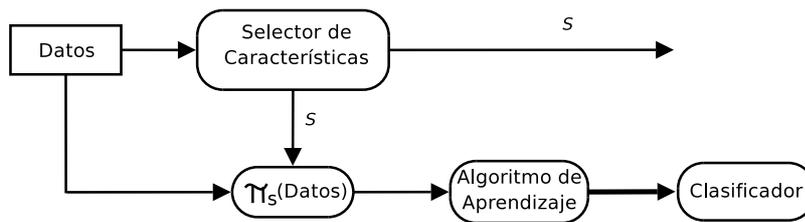


Figura 2.6: Selección de características de tipo filtro

- En la estrategia **envolvente** (*wrapper*), el método de selección de características usa el algoritmo de aprendizaje para evaluar la calidad de los conjuntos de características, utilizando alguna medida de la calidad de las soluciones que obtiene éste con cada uno de los conjuntos de características candidatos. En este proceso, hay un flujo de información en ambos sentidos. En uno, el método de selección de características indica un conjunto de características a usar y, en el otro sentido, se devuelve una evaluación de lo útiles que son esas características. Esto se repite hasta que finalmente se selecciona un conjunto definitivo.

La principal ventaja de esta estrategia es que la selección de características tiene una evaluación de las características en el entorno real en que serán aplicadas y, por tanto, tiene en cuenta las posibles particularidades del algoritmo de aprendizaje que se va a usar. Sin embargo, se genera una relación de dependencia entre ambos algoritmos que impone ciertos requisitos sobre el algoritmo de aprendizaje, ya que éste deberá ser capaz de trabajar con los conjuntos de características que determine probar el método envolvente.

- Inmersa** (*embedded*): En algunos algoritmos de aprendizaje, la selección de características está incluida en el mismo como una parte no separable. En este caso, la ventaja es que la selección de características está diseñada de forma específica para ese aprendizaje, con lo que se espera que su rendimiento sea mejor.

No deben confundirse éstos con los algoritmos de aprendizaje que toleran características irrelevantes. Los algoritmos de selección de características inmer-

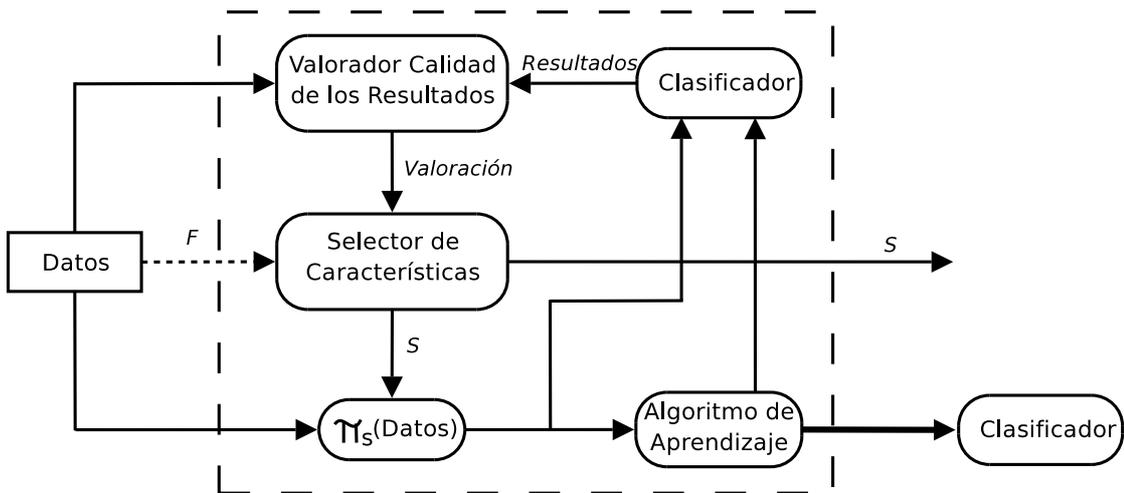


Figura 2.7: Selección de características envolvente del algoritmo de aprendizaje

Los indicadores indican qué características han seleccionado, y los clasificadores que generan sólo necesitarán esas. Sin embargo, la tolerancia a características irrelevantes no reduce las características usadas. Al indicar las características seleccionadas, éstas pueden ser usadas por otro algoritmo de aprendizaje y, así, usar el método inmerso como filtro en otro sistema de aprendizaje. Por ejemplo, C4.5 lleva inmerso un método de selección, siendo posible saber que características usa un árbol generado, y de esta manera usarlas como selección para otro método [LKM01].

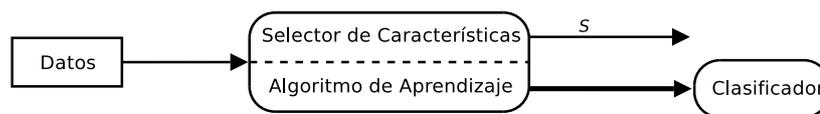


Figura 2.8: Selección de características inmersa en el algoritmo de aprendizaje

- Además, incluimos una categoría para considerar los métodos **híbridos** (*hybrid*), que combinan algunas de las estrategias anteriores. Una posibilidad interesante es realizar una búsqueda rápida tipo filtro y, posteriormente, usar la estrategia envolvente para afinar la selección (ejemplos: [Boz02], [ÖUT06]). También incluiríamos en esta categoría el uso que apuntábamos antes: la aplicación de un método inmerso como filtro de otro (ejemplo: [Car93]).

2.3.2. Según una división del algoritmo en partes

En el fondo, todos los algoritmos de selección de características resuelven un problema de búsqueda en el espacio de subconjuntos. Por esta razón, es posible identificar en ellos partes comunes. La división de los métodos en bloques generales de funcionalidad, viendo qué técnica aplican en cada uno de ellos, es otra forma de analizar los diferentes métodos de selección de características. Así, Langley [Lan94] expone un esquema de construcción general de los algoritmos de selección de características que, posteriormente, es usado en la revisión de Dash y Liu [DL97] para agrupar los algoritmos en categorías. Principalmente, identifica cuatro partes:

- La primera es la selección del **punto de comienzo**. Las posibilidades van desde comenzar a partir del conjunto vacío, con ninguna característica seleccionada, hasta comenzar con el conjunto de todas las características, pasando por múltiples posibilidades como empezar seleccionando un conjunto aleatorio, o algún otro tipo de estrategia más avanzada. Por ejemplo, Kohavi [Koh94] sugiere comenzar con el conjunto de todas las características fuertemente relevantes. Dado que para eso habría que realizar primero una estimación de cuáles son las fuertemente relevantes, en un trabajo posterior, Kohavi y John [KJ97] sugieren comenzar con el conjunto completo de características
- La **organización de la búsqueda**, en la que se presentan infinidad de posibilidades. Desde la más costosa, la búsqueda exhaustiva, pasando por las que usan criterios para seguir haciendo una búsqueda completa sin explorar todo el espacio, hasta todo tipo de heurísticas.
- La función o estrategia de **evaluación**, que indica cómo se valorarán los conjuntos de características. Puede estar basada en algún criterio estadístico o heurística sobre los ejemplos, o bien, usar el rendimiento del propio algoritmo de aprendizaje (estrategia envolvente antes mencionada).
- El **criterio de parada** para determinar cuándo el algoritmo deja de buscar y devuelve la solución encontrada. Aquí hay gran variedad de criterios y depende mucho del problema con el que se trate. Hay algoritmos que están diseñados para buscar un número de características fijado por el usuario. Otros, como Relief [KR92], que sólo proporcionan una valoración de las características y dejan en manos del usuario la elección de un grupo de ellas. De esta forma, los criterios de parada son muy diferentes y pueden basarse, por ejemplo, en buscar hasta alcanzar el número de características deseado, que la búsqueda encuentre un máximo local (o global), alcanzar una medida de precisión deseada, o agotar un tiempo de ejecución establecido.

2.3.3. Según otras propiedades de los métodos

Es posible hacer otras clasificaciones de los métodos. Por ejemplo, la taxonomía de Jain y Zongker [JZ97], expuesta en la figura 2.9, separa los métodos atendiendo a varias de sus propiedades: *i*) si están basados en técnicas estadísticas o redes neuronales, *ii*) si proporcionan una solución óptima o una aproximación, *iii*) si devuelven una única solución o varias, y *iv*) si son determinísticos o aleatorios.

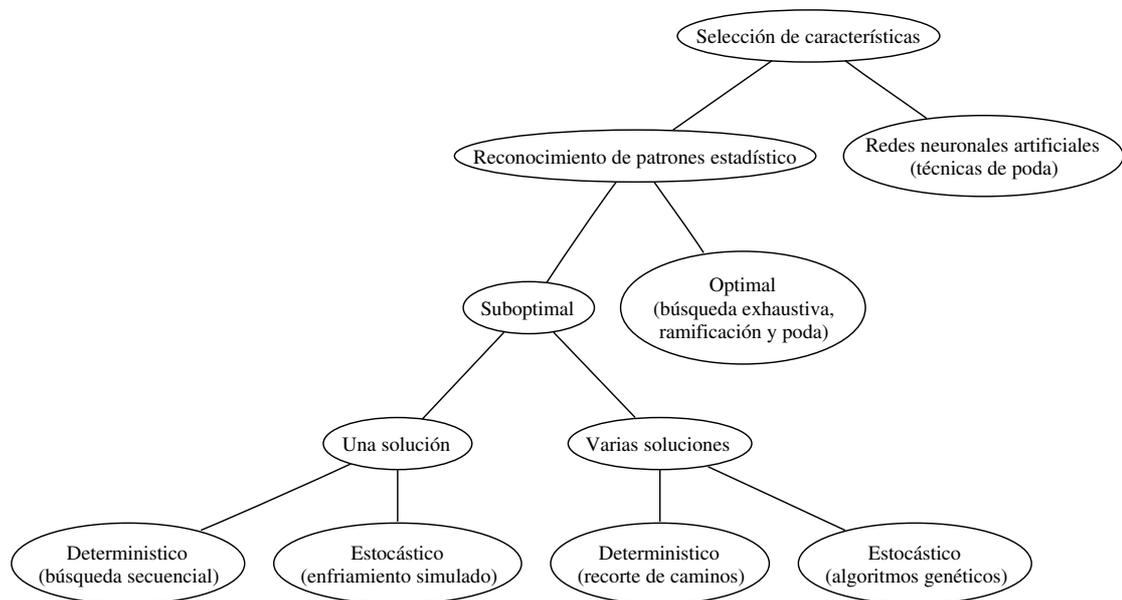


Figura 2.9: Taxonomía de Jain y Zongker para los métodos de selección de características

Además de las propiedades anteriores, y de las consideradas en las clasificaciones de los dos apartados previos, queremos mencionar otras dos que también consideramos importantes a la hora de caracterizar los métodos. La primera es la forma de evaluar las características, que crea dos grandes grupos: los que evalúan cada una de ellas de forma individual, y los que evalúan conjuntos de características. También hay que incluir un grupo, probablemente menos numeroso, para los que aplican ambos tipos de evaluaciones a lo largo del proceso de la selección. La segunda es si pertenecen o no al grupo de los algoritmos de cualquier duración (*anytime algorithms*). Estos algoritmos pueden ser ejecutados durante un tiempo determinado y, si se les permite continuar ejecutándose durante más tiempo, continuarán mejorando la solución tanto como les sea posible.

2.4. Descomposición modular de los métodos de selección de características

Hemos visto como se han identificado diferentes partes funcionales en los algoritmos de selección de características. En este apartado, proponemos una descomposición modular del proceso de selección de características. Está basada en las cuatro funcionalidades identificadas por Langley [Lan94] y descritas en el apartado anterior. Las funciones son similares a las propuestas por Dash y Liu [DL97], con la inclusión de la elección del punto de comienzo, y la eliminación del proceso de validación. Aunque la validación es altamente recomendable, como ya indicamos en la sección 2.1, está fuera de los algoritmos de selección de características, como también se apuntaba en el trabajo mencionado.

A diferencia de estos trabajos, donde se identificaban las funcionalidades con el objetivo de clasificar los métodos en varios tipos, la descomposición modular aquí propuesta está hecha desde un punto de vista más de ingeniería del software, pretendiendo ser un esquema para su programación modular. Nuestro objetivo es que ésta sirva para la implementación de todos los métodos, y que permita combinar partes de unos métodos con otros.

La descomposición modular propuesta se muestra en la figura 2.10. En ella, agrupamos las funciones en dos bloques principales:

- Método de búsqueda en el espacio de conjuntos de características
- Evaluación de las características

El método de búsqueda gobierna el flujo de control del algoritmo, mientras que las funciones de evaluación de características son herramientas usadas por éste. Dentro del método de búsqueda, consideramos tres partes importantes: *i*) la elección del punto de comienzo de la búsqueda, que puede ser inmediata o elegirse mediante un proceso más elaborado; *ii*) estrategia de orientación de la búsqueda, o lo que es lo mismo, elección del siguiente conjunto a evaluar; y *iii*) el criterio de parada, que puede depender de muy diversos factores, como el número de evaluaciones, tiempo de ejecución, o el alcance de alguna condición sobre los resultados.

En la evaluación de las características, creemos importante diferenciar dos tipos de medidas. Por un lado, las que evalúan conjuntos de características completos y, por el otro, las que evalúan características individualmente.

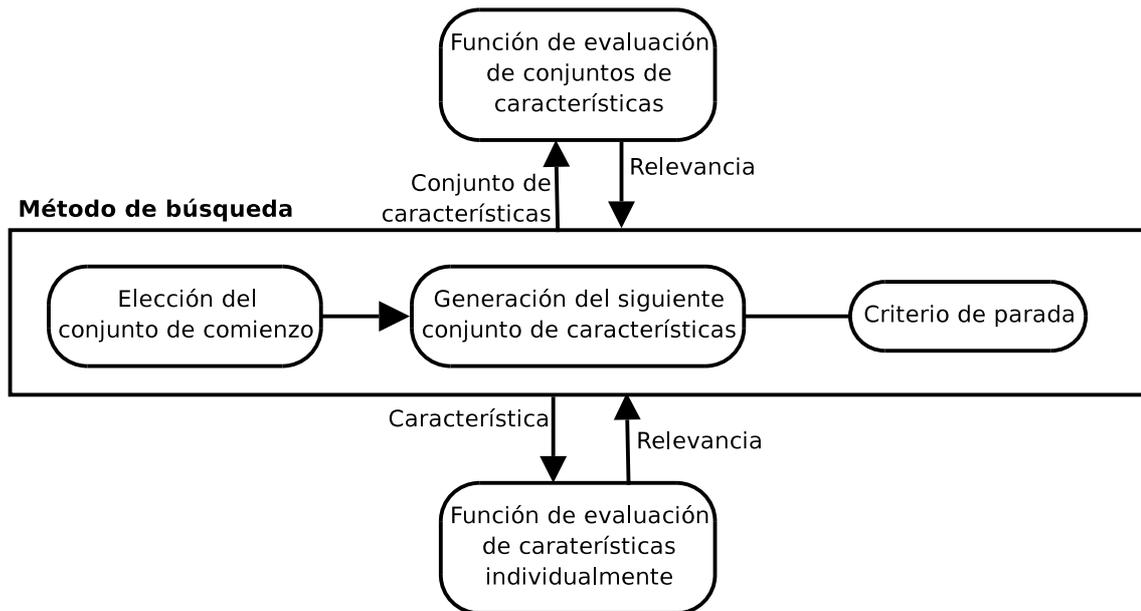


Figura 2.10: Descomposición modular del proceso de selección de características

Un mismo método puede usar una o varias medidas distintas, incluso combinar el uso de medidas de distintos tipos para diferentes propósitos. Dentro del proceso de selección de características, las medidas de relevancia de las características pueden usarse con varios propósitos. Probablemente, los más comunes e importantes son dos: *i*) la elección del mejor conjunto de características entre todos los que han sido evaluados, eligiendo aquél que obtenga el mejor valor de la medida; y *ii*) la orientación de la búsqueda en el espacio de subconjuntos de características. Por ejemplo, en una búsqueda simple de ascensión de colinas, se prueba añadiendo cada una de las características al conjunto de seleccionadas, quedándonos con el conjunto que añadida la nueva característica dé mejor valor para la medida, para continuar la búsqueda por ese camino, el camino indicado por la medida. Aunque no sea tan obvio, este uso de la medida es el mismo que se da en métodos más complejos como los algoritmos genéticos. Es interesante señalar que la medida que permite elegir el mejor conjunto de características no es necesariamente la mejor para guiar la búsqueda.

Todos los métodos de selección de características pueden adaptarse a este esquema. No todos incluyen todas las partes indicadas y, para algunos, puede que situarlos en este esquema no sea demasiado esclarecedor. No obstante, hay una gran familia de métodos que se adaptan muy bien al esquema, y no resulta fácil imaginar un esquema más general. A continuación, a modo de ejemplo, usamos este esquema para describir varios métodos.

Un método que se adapta perfectamente al esquema propuesto es el método

SBS con información mutua (SBS: Sequential Backward Search). Este método utiliza como función de evaluación de conjuntos la medida de información mutua de la teoría de la información, y no utiliza ninguna función de evaluación de características individualmente. Dentro del método de búsqueda, el conjunto de comienzo es el conjunto de todas las características del problema; la estrategia de búsqueda consiste en evaluar todos los conjuntos resultantes de eliminar una de las características del actual y continuar con el mejor de ellos; y como criterio de parada, el algoritmo se detendrá cuando ninguno de los conjuntos del siguiente nivel tengan un valor igual o mejor de la función de evaluación.

Como ejemplo de un algoritmo que no tiene todas las partes del esquema podemos pensar en Relief [KR92]. Relief no realiza ninguna búsqueda, simplemente evalúa la relevancia de cada una de las características y selecciona aquellas que superen un umbral dado como parámetro. Podemos incorporarlo al esquema identificando la función de evaluación de características individuales y considerando que su mecanismo de búsqueda es únicamente la determinación del conjunto de comienzo. El criterio de parada será la función constante a verdadero (devolver el conjunto inicial) y, puesto que no lo va a usar nunca, no tiene mecanismo de generación del siguiente conjunto a evaluar. Colocar Relief en este esquema nos revela que puede ser usado como estrategia de elección del conjunto de comienzo en otros métodos.

La estrategia envolvente [JKP94] también se adapta perfectamente al esquema propuesto. Utiliza un método de búsqueda genérico, que puede ser perfectamente modelado con las partes indicadas en el esquema y, como función de evaluación de conjuntos de características, una medida de la bondad de los resultados que obtiene el algoritmo de aprendizaje con el conjunto dado.

La visión modular del proceso de selección de características presentada nos aporta diversas ventajas. En primer lugar, nos permite alcanzar una mejor comprensión de los métodos de selección de características, al permitirnos comprender su estructura interna de forma más ordenada. En segundo lugar, podemos investigar cada uno de los módulos independientemente de los demás, probando en ellos diferentes opciones de forma aislada del resto. Además, usando este modelo, también es posible crear una gran variedad de métodos de selección de características simplemente combinando diferentes funciones de evaluación y mecanismos de búsqueda. Finalmente, las partes identificadas es posible que puedan ser aprovechadas con otros propósitos ajenos a la selección de características, y viceversa. Por ejemplo, algunas funciones de evaluación son usadas en *discretización*.

2.5. Métodos de búsqueda

El problema de la selección de características puede ser visto como un problema de búsqueda en el conjunto potencia del conjunto de las características disponibles [Koh94, Lan94]. Su objetivo es encontrar un subconjunto de características que nos permita mejorar, en algún aspecto, una actividad de aprendizaje. En esta sección, veremos en primer lugar los aspectos generales de la búsqueda aplicada a la selección de características y, a continuación, algunos de los métodos de búsqueda más usados en este ámbito.

2.5.1. Espacio de búsqueda

A la hora de enfrentar un problema de búsqueda es importante conocer la forma y propiedades del espacio donde se realizará la búsqueda. En el caso de la selección de características, el espacio de búsqueda es el conjunto potencia de las características ($\mathcal{P}(F)$). Éste, considerado bajo el orden parcial de inclusión (\subset), forma un retículo, o lo que es lo mismo, un retículo como estructura algebraica bajo las operaciones unión e intersección (\cup y \cap). Se trata de un retículo acotado, por el conjunto con todas las características y el vacío (F y \emptyset). Como ejemplo, en la figura 2.11 se muestra un diagrama de Hasse del retículo formado por $\mathcal{P}(\{a, b, c, d, e\})$, que es el espacio de búsqueda para la selección de características en un problema de aprendizaje en el que contamos con 5 características.

Para un problema con n características es posible formar 2^n subconjuntos distintos. Así, el tamaño del espacio de búsqueda es 2^n , que crece exponencialmente con el número de características. El número de subconjuntos que se pueden formar con m características viene dado por $\binom{n}{m}$, el tamaño de cada una de las filas del diagrama de Hasse, indicado en el ejemplo de la figura 2.11.

El problema de selección de características clásico establecía encontrar las m mejores características entre las n disponibles. En este caso, el espacio de búsqueda sería solamente una de las filas del diagrama.

Es posible hacerse una idea intuitiva de la forma del espacio de búsqueda, y de su crecimiento a medida que aumenta n , mirando las figuras 2.12 y 2.13. En la primera, hemos representado el tamaño por niveles de m características (que es dos veces el tamaño del perfil del diagrama de Hasse). La gráfica a) es para un tamaño de cinco características, el mismo que el representado en el diagrama 2.11, y en b), c) y

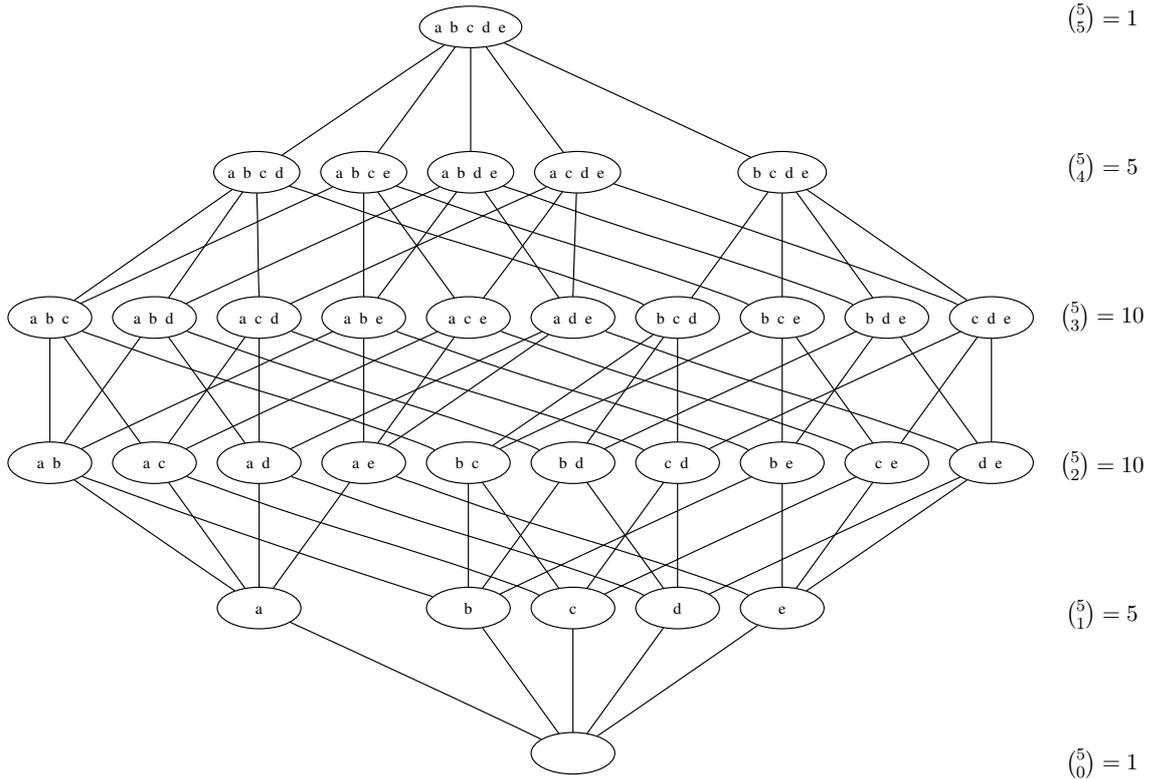


Figura 2.11: Diagrama Hasse del retículo formado por $\mathcal{P}(\{a, b, c, d, e\})$

d) podemos ver el rápido crecimiento al aumentar n a 10, 20 y 50 respectivamente. Como el crecimiento es tan rápido, si las representásemos sobre los mismos ejes, las gráficas correspondientes a los espacios de menor tamaño aparecerían como una línea casi recta pegada al eje x . Por esta razón, en la figura 2.13, hemos representado las mismas gráficas todas juntas con el eje y en escala logarítmica.

Teniendo en cuenta el espacio de búsqueda al que nos enfrentamos, se puede demostrar que el problema de la selección de características es un problema NP-completo [DR94]. Podría incluso ser más duro, si la evaluación de los conjuntos de características no fuese polinomial.

2.5.2. Estrategias de búsqueda

La búsqueda puede realizarse en dos sentidos: *i*) incluyendo características en el conjunto de características seleccionadas, o *ii*) eliminándolas del mismo. Hay sugerencias [KJ97] respecto a comenzar con el conjunto completo de características, ya que parece más fácil eliminar características sin perder información que encontrar las características que proporcionan más información partiendo de cero. También

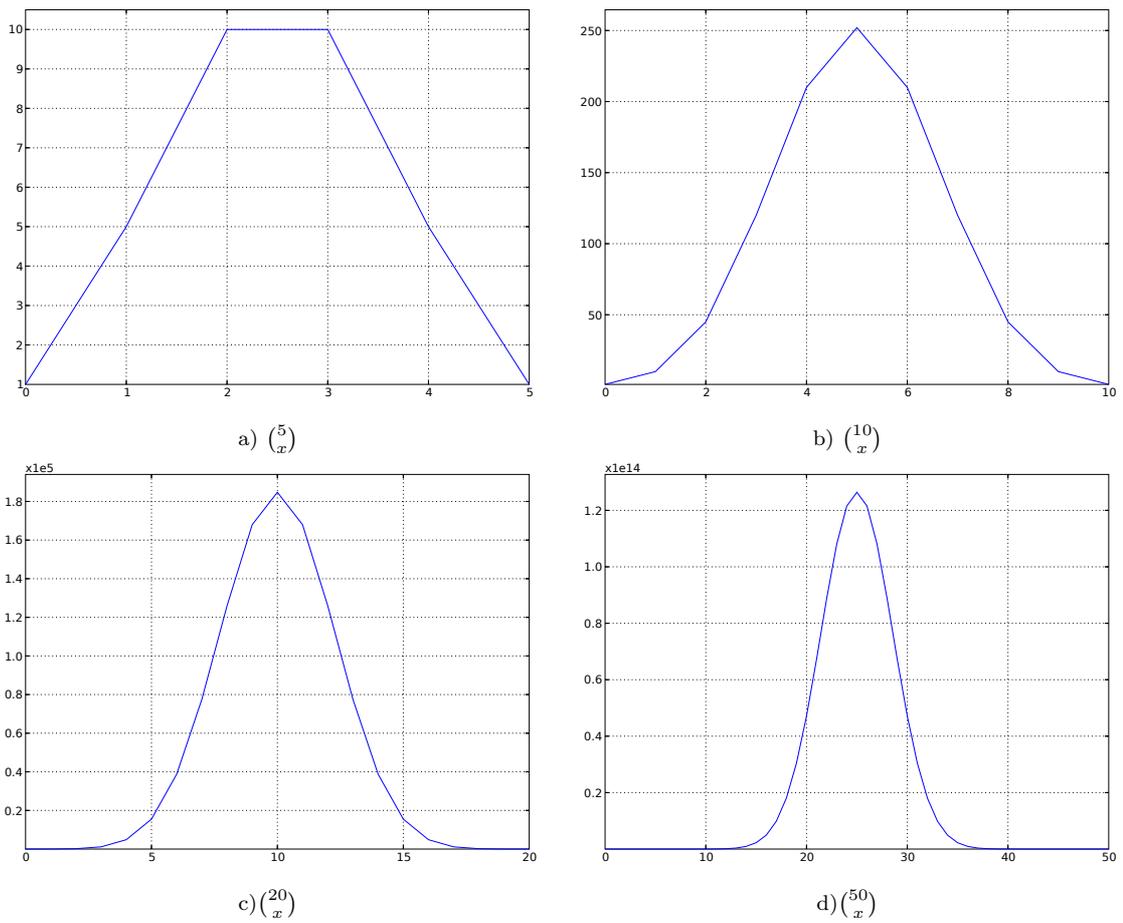


Figura 2.12: Número de conjuntos de tamaño x en el espacio de búsqueda de N características: a) $N = 5$, b) $N = 10$, c) $N = 20$, y d) $N = 50$

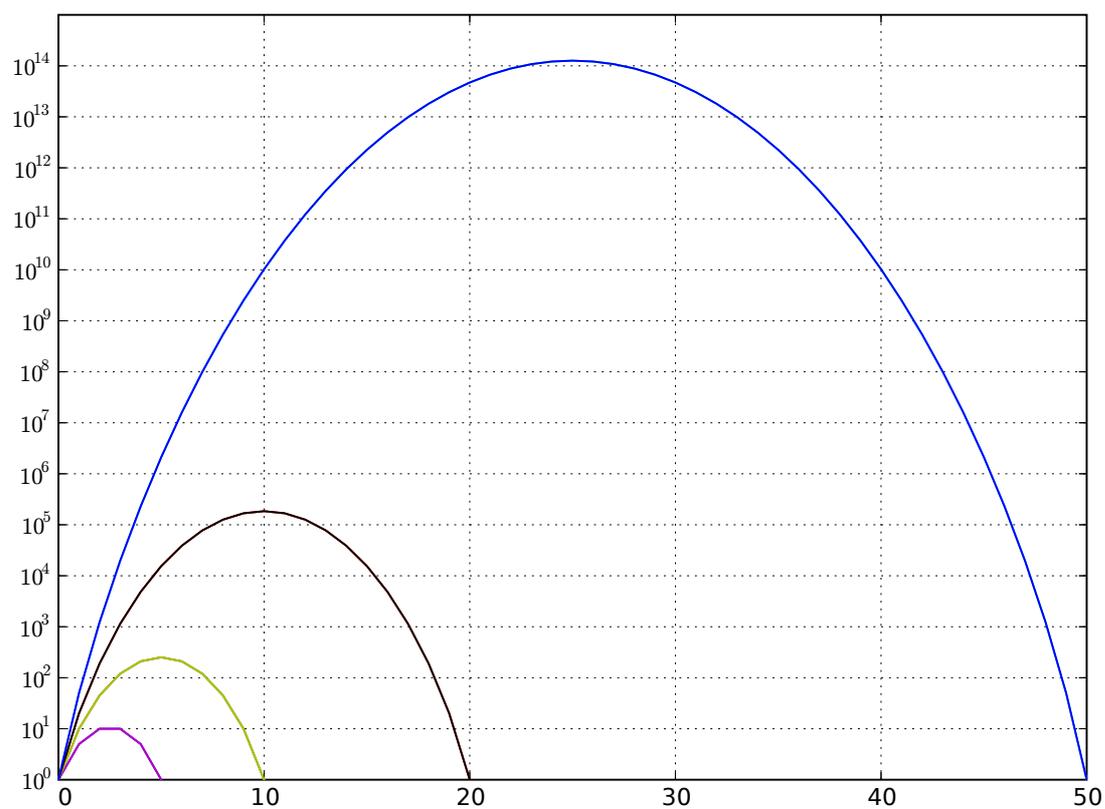


Figura 2.13: Comparación sobre escala logarítmica del número de conjuntos de tamaño x en los espacios de búsqueda de 5, 10, 20 y 50 características

Koller y Sahami [KS96] exponen por qué, en el método que proponen, cuyo objetivo es conservar al máximo la distribución de probabilidad original, es más apropiado ir eliminando características, evitando alejarse así de la distribución con todas las características. Como contrapartida, en algoritmos envolventes, partir del conjunto de todas las características para ir eliminando requiere que el algoritmo de aprendizaje sea capaz de trabajar con números muy elevados de características. Esto no siempre es posible, es más, muchas veces ésta es la razón para aplicar la selección de características.

De todas formas, las estrategias de búsqueda no tienen por qué limitarse a un único sentido. Pueden, por ejemplo, primero añadir características y luego eliminarlas. Tampoco tienen que seguir el orden de inclusión moviéndose únicamente por las aristas de la figura 2.11. Es posible, por ejemplo, eliminar y añadir una característica en el mismo paso y, así, moverse lateralmente. Además, las estrategias también usan otro tipo de mecanismos y saltan a otras zonas completamente distintas del espacio de búsqueda.

2.5.2.1. Búsqueda completa

Estas estrategias de búsqueda garantizan encontrar el conjunto de características óptimo, según el criterio de evaluación utilizado. La más obvia es la búsqueda exhaustiva, que evalúa todos los subconjuntos posibles para quedarse con el mejor.

Para garantizarse encontrar el conjunto óptimo y, por tanto, haber realizado una búsqueda completa, no siempre es necesario evaluar todos los subconjuntos posibles. Por ejemplo, Focus2 [AD94] aprovecha su criterio de evaluación, la búsqueda de consistencia, para dirigir la búsqueda primero por las características que sabe serán imprescindibles para hacer consistente el conjunto. Además, una vez encontrado un conjunto consistente, por el orden seguido, éste será el de menor tamaño, y se podrá detener la búsqueda habiendo evitado evaluar una gran parte del espacio de búsqueda.

Tampoco es necesario restringirse a criterios tan específicos como la búsqueda de consistencia. Para una gran familia de funciones de evaluación, aquellas que cumplen la propiedad de ser monótonas, la técnica de búsqueda ramificación y poda (*Branch and Bound*) [BB97, HS78, NF77, SP04] permite orientar la búsqueda ahorrando explorar gran parte del espacio.

El algoritmo ABB, descrito en la página 77 de [LM98], es un ejemplo de al-

goritmo de ramificación y poda que busca el menor conjunto de características que tenga un valor de la medida de evaluación superior a un umbral dado.

2.5.2.2. Búsqueda secuencial

Los algoritmos voraces (*greedy*) son aquéllos que van tomando decisiones sin replantearse las anteriores y se caracterizan por su simplicidad y eficiencia, por lo que habitualmente se emplean para implementar heurísticas. En el caso de la búsqueda, siguen un camino sin volver nunca hacia atrás. También se les denomina de búsqueda secuencial porque siguen una secuencia de pasos sin vuelta atrás. Son los algoritmos de búsqueda más simples.

Por un lado, tenemos el SFS (*Sequential Forward Selection*) que parte del conjunto vacío de características y va añadiendo la que en mayor medida mejora la selección. También se le denomina de ascensión de colinas. Por otro lado, tenemos el que hace la búsqueda en sentido inverso SBS (*Sequential Backward Selection*), partiendo del conjunto de todas las características y eliminando aquéllas que hagan mejorar el criterio elegido.

2.5.2.3. Búsqueda probabilística

Los algoritmos probabilísticos [BB97] son aquellos que dejan algunas de sus decisiones al azar. En búsqueda podemos ver algoritmos que exploran aleatoriamente el espacio de búsqueda, o por zonas, o siguiendo algún criterio que dependa de algún factor aleatorio.

LVF (*Las Vegas Filter*) [LS96b] es un algoritmo de selección de características probabilístico pensado para medidas de evaluación de tipo filtro monótonas. Explora aleatoriamente conjuntos con igual o menor número de características que el mejor que ha encontrado hasta el momento, y finaliza después de un número de pasos especificados en un parámetro.

LVW (*Las Vegas Wrapper*) [LS96a] es un algoritmo similar, pero pensado para aplicar la estrategia envolvente. También es apropiado para medidas de tipo filtro no monótonas. En este algoritmo, los conjuntos evaluados aleatoriamente pueden tener más o menos características que el mejor encontrado hasta el momento.

2.5.2.4. Búsqueda con meta-heurísticas

Las metaheurísticas [GK03, OK96, VMOR99] son técnicas de propósito general para la orientación una búsqueda o la construcción de soluciones. Se basan en conceptos generales empleados para definir métodos heurísticos. En otras palabras, una metaheurística puede verse como un marco de trabajo general para algoritmos que pueden aplicarse a diversos problemas de optimización.

El enfriamiento simulado (*Simulated Annealing (SA)*) [KJV83, Cer85] es una meta-heurística para problemas de optimización global que se basa en conceptos de la mecánica estadística, concretamente en el enfriamiento de un sólido. Este concepto hace referencia al proceso térmico por el que se obtienen estados de baja energía en un sólido. La evolución de un sólido en un baño térmico puede ser simulado mediante técnicas de Monte Carlo. En este algoritmo, el equilibrio térmico a una determinada temperatura se consigue mediante la generación de un elevado número de transiciones, en cada una de las cuales se pasa de un estado de energía a un nuevo estado, tendiendo cada vez hacia niveles más bajos de energía.

El paso de un estado a otro se realiza según las siguientes reglas: si el estado generado tiene menor energía que el que se tiene actualmente, el generado se acepta como estado actual; en caso contrario, el estado generado se aceptará con una determinada probabilidad. Esta probabilidad es función de la diferencia de energías de los estados y de la temperatura. Cuanto menor sea la temperatura, menor será la probabilidad de transformación en un estado de mayor energía (aceptar peores soluciones) y cuanto mayor sea la energía del nuevo estado, menor será la probabilidad de que sea aceptado. El criterio de parada puede ser que se alcance un estado suficientemente bueno según la evaluación, o hasta que se cumpla cierto tiempo computacional dado. En la figura 2.14, se muestra el algoritmo de este proceso. Hay diversos ejemplos de aplicación de *SA* a selección de características [Xu88, MZ06, DRS99].

Los algoritmos genéticos (*Genetic Algorithms (GA)*) [Gol89] fueron desarrollados por John Holland, quien proporcionó los fundamentos teóricos de los mismos en su libro “Adaptation in Natural and Artificial Systems” [Hol75]. Un *GA* es un procedimiento adaptativo que encuentra soluciones en espacios complejos inspirado por la evolución biológica, es decir, por un proceso de evolución basado en la selección natural.

Estos algoritmos se basan en operadores que tratan de modelar los operadores genéticos existentes en la naturaleza, como el cruce y la mutación, los cuales son aplicados a los individuos que codifican las posibles soluciones.

SimulatedAnnealing($T_0, T_f, \text{enfriamiento}, n\text{Vecinos}$)

$T = T_0$

$S_{actual} = \text{Genera solución aleatoria}$

While $T \geq T_f$ do:

For i in $n\text{Vecinos}(T)$ do:

$S_{candidata} = \text{Genera un vecino}(S_{actual})$

$\lambda = \text{coste}(S_{candidata}) - \text{coste}(S_{actual})$

If $U(0, 1) < e^{-\lambda/T}$ or $\lambda < 0$ then:

$S_{actual} = S_{candidata}$

$T = \text{enfriamiento}(T)$

Return S_{actual}

donde:

- T_0 es la temperatura inicial
- T_f la temperatura final
- *enfriamiento* es el esquema de enfriamiento elegido
- $n\text{Vecinos}(T)$ el número de vecinos generados en cada ciclo según T
- $U(0, 1)$ es un generador de números aleatorios uniformemente distribuidos en $[0, 1]$.

Figura 2.14: Algoritmo de enfriamiento simulado

De igual forma, la terminología que se emplea en los *GAs* también está directamente importada de la Genética y de la Teoría de la Evolución de las Especies. Se habla de individuos de una población denominados cromosomas que están formados por unidades llamadas genes. Cada gen puede manifestarse de forma diferente, es decir, tomar distintos valores, que se conocen con el nombre de alelos o valores del rasgo.

El funcionamiento de un *GA*, a pesar de ser bastante sencillo, presenta una gran potencia en la búsqueda del óptimo. Parte de una población de candidatos a solución generada de forma aleatoria y evoluciona hacia la mejor solución en forma de selección. En cada generación, las soluciones relativamente buenas son reproducidas y las soluciones relativamente malas mueren para ser reemplazadas por otras, que presenten características de las buenas soluciones, en la nueva población; finalmente, las soluciones seleccionadas para formar parte de la nueva población son alteradas a partir de los operadores de cruce y mutación. Para evaluar en qué grado una solución es buena o mala para el problema tratado, se emplea una función de evaluación o función objetivo. En la figura 2.15, se muestra un esbozo de la estructura de un *GA*.

```
Genetic Algorithm()  
  
   $t = 0$   
  
  inicializar  $P(t)$   
  
  evaluar  $P(t)$   
  
  Mientras no se cumpla la condicion de parada hacer:  
     $t = t + 1$   
  
    seleccionar  $P(t)$  desde  $P(t - 1)$   
    recombinar  $P(t)$   
    evaluar  $P(t)$   
  
  donde,  $P(t)$  es la población en la iteración  $t$ .
```

Figura 2.15: Algoritmo Genético

Los *GAs* son muy robustos; destacan por su buen comportamiento en problemas difíciles en los que el espacio de búsqueda es grande. Por ello, es interesante su aplicación en selección de características [LHL05, SP05, ZG04, CQ99]. Aunque no garantizan la obtención de la solución óptima al problema, suelen proporcionar soluciones bastantes aceptables en un tiempo razonable.

Otras aplicaciones de los algoritmos genéticos son la optimización de funciones numéricas y la optimización combinatoria [Gol89, Mic96], la Investigación Operativa

[Nis93], las Redes Neuronales [SWE92], los Sistemas Difusos [Ped97, CHHM01], la Robótica [Dav91], el Aprendizaje Automático [DSG93, Jan93], etc.

La búsqueda tabú (*Tabu Search (TS)*) [Glo89] es un procedimiento meta-heurístico utilizado para guiar un algoritmo heurístico de búsqueda local para explorar el espacio de soluciones más allá de la simple optimalidad local. Se basa en la premisa de que para poder calificar de inteligente la solución de un problema, debe incorporar memoria adaptativa y exploración sensible. Un esquema básico de *TS* es:

1. Sea $f(x)$ una función a ser optimizada en un conjunto X .
2. *TS* procede iterativamente de un punto(solución) a otro hasta satisfacer un criterio dado de terminación.
3. Cada $x \in X$ tiene un entorno asociado $N(x) \subseteq X$, y cada solución $x' \in N(x)$ se puede alcanzar desde x mediante una operación llamada movimiento.

TS mejora la búsqueda local empleando una estrategia de modificación de $N(x)$ por otro entorno $N^*(x)$. Un aspecto clave de *TS* es el empleo de estructuras especiales de memoria que sirven para determinar $N^*(x)$ y así organizar la manera en la cual se explora el espacio. Las soluciones que son admitidas en $N^*(x)$ por estas estructuras de memoria se determinan de varias formas. Una de ellas, que da a la *TS* su nombre, identifica soluciones encontradas sobre un horizonte especificado, y les prohíbe pertenecer a $N^*(x)$ clasificándolas como tabú.

Naturalmente es posible aplicar la búsqueda tabú en selección de características [LTPV02], realizando con esta meta-heurística otro tipo más de búsqueda inteligente.

2.6. Medidas de la utilidad de las características

En todo proceso de selección de características, hay algún momento en que es necesario valorar la utilidad de éstas en la resolución del problema de aprendizaje a abordar. En la mayoría de los métodos, puede hacerse la valoración de forma independiente al proceso de búsqueda. Al separar las medidas del proceso de búsqueda, como se propone en la descomposición modular de la sección 2.4, podemos intercambiar unas por otras y estudiarlas de forma independiente para usar en cada momento la más adecuada.

Hay dos tipos de aproximaciones principales al problema de la valoración de relevancia de las características. La primera es valorar cada una de las características de forma independiente. La ventaja de este tipo de medidas es que suelen ser medidas bastante simples y, como sólo se pueden hacer tantas valoraciones como características tenga el conjunto de datos, apoyarse en estas medidas es muy rápido. Como contrapartida, el inconveniente principal es que estas medidas no aportan ninguna información sobre la posible redundancia que haya entre características. Por ejemplo, si hay una característica relevante duplicada, una medida individual está obligada a darle el mismo valor a ambas, y no será posible descartar ninguna de las dos características. La segunda aproximación a la valoración de características es trabajar con conjuntos de características. En este caso, al valorar conjuntos completos, se puede obtener información sobre las posibles interrelaciones entre características y, de esta forma, averiguar por ejemplo si la inclusión de una característica aporta algo o es completamente redundante a las de un conjunto dado. Otra ventaja es que se valora el conjunto de características que se va a ofrecer como resultado del método, por lo que los métodos que usan este tipo de medidas disponen de una aproximación más exacta a la calidad del resultado que ofrecen. En cambio, el inconveniente es que valorar todos los posibles subconjuntos de características normalmente no es factible.

2.6.1. Medidas sobre características individuales

Hay muchas medidas para valorar la relevancia de una característica. A continuación, se describen algunas de las más comúnmente usadas.

2.6.1.1. Ganancia de información o información mutua

De la teoría de la información, se usa la cantidad de información que aporta una característica sobre la clase a predecir, para valorar la relevancia de dicha característica. Quinlan utilizaba la información mutua para elegir las características que dividirán nodos en generación de árboles [Qui86].

La descripción de esta medida es la misma que la de la medida de conjuntos “Información mutua” en la sección 2.6.2.

2.6.1.2. Gain ratio

La medida anterior de ganancia de información favorece a las características con muchos valores. Puede ocurrir que esta sobre-estimación no sea un comportamiento deseable y para evitarlo se puede usar como medida el ratio entre la ganancia de información y la entropía de la característica.

$$\text{Gain Ratio} = \frac{I(F; C)}{H(F)} \quad (2.1)$$

Esta medida fue usada por Quinlan para el algoritmo C4.5 [Qui93].

2.6.1.3. Índice de Gini

El índice de Gini (*Gini index*) toma su nombre del estadístico italiano Conrado Gini. El índice, más conocido como una medida de la desigualdad usada en economía [Gin06], fue introducido como medida para la generación de árboles de clasificación y regresión por Breiman [Bre98].

La medida se define de acuerdo a la ecuación (2.2), y se puede interpretar como la probabilidad de que dos ejemplos elegidos aleatoriamente tengan una clase diferente.

$$\text{Gini index} = \sum_{i,j \in C; i \neq j} p(i|F)p(j|F) \quad (2.2)$$

2.6.1.4. Relevancia en formación de reglas

Esta medida discrimina las características en función de su potencial de relevancia en la formación de reglas de decisión [DZ04].

2.6.1.5. ReliefF

Originalmente, Kira y Rendell [KR92] propusieron Relief como un método de selección de características. Estaba basado en una valoración novedosa de las carac-

terísticas y en la elección de aquéllas que obtuviesen una valoración mayor que un umbral dado. La valoración original de relevancia de las características sólo estaba definida para problemas lógicos (cuyo resultado sólo puede ser verdadero o falso) pero, posteriormente, Kononenko desarrolló extensiones [Kon94] que pueden trabajar con problemas de clasificación y tolerar valores nulos. La que mejores resultados obtuvo fue Relief-F, que ahora es la versión comúnmente usada para problemas de clasificación. También hay una extensión para problemas de aproximación, RRelief-F [RSK97].

La idea general es medir la habilidad de las características para distinguir entre ejemplos similares. El esqueleto del algoritmo que calcula la valoración de las características en todas las versiones de Relief se muestra en la figura 2.16. Las extensiones varían en los ejemplos vecinos que se buscan y en cómo se realiza el cálculo de la evaluación.

RELIEF(*Dataset*, *m*, ...)

For 1 to *m*:

E_1 = random example from *Dataset*.

Neighbours = Find some of the nearest examples to E_1 .

For E_2 in *Neighbours*:

Perform some evaluation between E_1 and E_2

Return the evaluation

Figura 2.16: Algoritmo general de todas las versiones de Relief

Como Relief-F generaliza el comportamiento de Relief a un ámbito más amplio, aquí describiremos únicamente el primero. Relief-F busca un vecino más cercano a E_1 de cada clase. Con estos vecinos se evalúa la relevancia de cada característica $f \in F$ actualizando su valoración acumulada en $W[f]$ con la ecuación (2.3). El vecino más cercano de la misma clase es un *hit*, H , y los de diferente clase un *miss* de clase C , $M(C)$. Al final del proceso de valoración, $W[f]$ se divide entre m para obtener la valoración media de cada característica en $[-1, 1]$.

$$W[f] = W[f] - \text{diff}(f, E_1, H) + \sum_{C \neq \text{class}(E_1)} p(C) \times \text{diff}(f, E_1, M(C)) \quad (2.3)$$

La función $\text{diff}(f, E_1, E_2)$ calcula el grado en que los valores de la característica f son diferentes en los ejemplo E_1 y E_2 como se indica en la ecuación (2.4), donde $\text{value}(f, E_1)$ denota el valor de la característica f en el ejemplo E_1 y $\text{max}(f)$ el valor máximo que alcanza f .

La distancia entre ejemplos, usada para considerar cuales son los vecinos más cercanos, es la suma de las diferencias, dadas por la función diff , de todas las características.

$$\text{diff}(f, E_1, E_2) = \left\{ \begin{array}{ll} 0 & \text{si } \text{value}(f, E_1) = \text{value}(f, E_2) \\ 1 & \text{en otro caso} \end{array} \right. \left| \begin{array}{l} \mathbf{f \text{ es discreta}} \\ \mathbf{f \text{ es continua}} \end{array} \right. \frac{|\text{value}(f, E_1) - \text{value}(f, E_2)|}{\max(f) - \min(f)} \quad (2.4)$$

A pesar de que esta medida obtiene muy buenos resultados, su eficiencia no es tan buena como la de las demás medidas de este grupo, ya que para buscar los vecinos más cercanos lee los ejemplos $O(m * |\text{Dataset}|)$ veces. No obstante, el parámetro m se puede variar, desde el total de los ejemplos $|\text{Dataset}|$ hasta valores mucho más pequeños, podría estudiarse como afecta la reducción de éste al resultado, lo que posiblemente permita reducir su tiempo de ejecución sin degradarlo. En caso de no reducir m , usando $m = |\text{Dataset}|$, se puede aplicar una optimización usando kd-trees [RS98] que consigue una eficiencia de $O(m \log m)$.

2.6.2. Medidas sobre conjuntos de características

Las medidas sobre conjuntos de características son funciones que, dado un conjunto de datos de entrenamiento ($T \in \mathbb{T}$, denominamos \mathbb{T} a todos los posibles conjuntos de entrenamiento) y un subconjunto de sus características ($S \subset \mathcal{P}(F)$), devuelven una valoración de la relevancia de esas características. El resultado será un número real, normalmente dentro de un intervalo, como $[0, 1]$ o $[-1, 1]$, pero también puede ser un resultado discreto, por ejemplo en $\{0, 1\}$, representando un valor booleano que indique si el conjunto es aceptable o no como resultado de la selección.

$$\text{Función de evaluación} : \mathcal{P}(F) \times \mathbb{T} \longrightarrow \mathbb{R} \quad (2.5)$$

Todas las medidas de conjuntos de características pueden ser usadas como medidas de características individuales, simplemente evaluando conjuntos de una sola característica.

Hay una gran variedad de medidas de relevancia para conjuntos de características. Algunas trabajan tanto con características discretas como continuas, otras sólo

aceptan uno de los dos tipos. Los algoritmos que las calculan pueden ser exactos o aproximados, determinísticos o no. Las medidas cumplen o no diversas propiedades como la monotonía o la invariabilidad a transformaciones lineales. Se muestra una comparación de algunas medidas en la tabla 2.1, donde “=” representa “igual que el algoritmo de aprendizaje usado”. Todas las medidas de la tabla son descritas a continuación, con la excepción de IEP y RFSM que son definidas en el capítulo 3.

<i>Medida</i>	<i>Rango</i>	<i>Monótona</i>	<i>Tipo C.</i>	<i>Eficiencia</i>
Consistencia simple	$[0, 1]$	Sí	Disc.	$O(n)$
Consistencia de Liu	$[0, 1]$	Sí	Disc.	$O(n)$
Consistencia de Rough Set	$[0, 1]$	Sí	Disc.	$O(n)$
IEP	$[0, 1]$	Sí	Disc.	$O(n)$
Información mutua	$[0, H(C)]$	Sí	Disc.	$O(n)$
Incertidumbre simétrica	$[0, 1]$	No	Disc.	$O(n)$
MDL	$[0, +\infty]$	No	Disc.	$O(n^2)$
Criterio discriminante, Narendra	$[0, 1]$	Sí	Cont.	$O(n^2)$
RFSM	$[-1, 1]$	Paramtr.	Ambos	$O(n^2)$
Envolvente (rendimiento)	$[0, 1]$	No	=	=

Tabla 2.1: Medidas de la relevancia de conjuntos de características

Se han usado, y hay disponibles, muchas medidas para selección de características. Normalmente, miden alguna cualidad de las características respecto al conjunto de datos. Una categorización de las medidas basada en el tipo de cualidad que miden está propuesta en [DL97]. Las categorías identificadas son: medidas de distancia, medidas de información, medidas de dependencia, medidas de consistencia y las basadas en el porcentaje de acierto del algoritmo de aprendizaje. Describiremos a continuación las medidas más importantes de cada una de las categorías.

2.6.2.1. Medidas basadas en consistencia

Para poder predecir correctamente la clase asociada a los ejemplos de un conjunto de datos, es necesario que éste sea consistente. Un conjunto de datos se considera consistente siempre que en él no haya ningún par de ejemplos que, perteneciendo a clases distintas, tengan los mismos valores en todas sus características. Si en un conjunto de datos se eliminan algunas características, dejando sólo las seleccionadas, habrá menos valores que diferencien los ejemplos y, por tanto, podrán aparecer más casos de inconsistencia. La idea que persiguen las medidas basadas en consistencia es valorar el nivel de consistencia del conjunto de datos que tiene únicamente las características seleccionadas.

El objetivo de buscar características que hagan consistente el conjunto de datos es equivalente a seleccionar aquellas características que mejor permiten definir hipótesis lógicas consistentes sobre el conjunto de entrenamiento. Se espera que estas hipótesis sean generalizables a ejemplos desconocidos y, por tanto, el algoritmo de aprendizaje se comporte bien con las características seleccionadas.

Como siempre que se aumenta el número de características, aumenta el número de hipótesis consistentes que se pueden definir, el requisito de presentar consistencia suele acompañarse con el de tener un número reducido de características. En cualquier caso, la búsqueda de un conjunto de características pequeño es un objetivo común de todos los métodos de selección de características. Así, esta estrategia no es una particularidad exclusiva de los métodos basados en medidas de consistencia, sino que también es aplicada en algoritmos de búsqueda que se usan con otros tipos de medidas.

En principio, todas las medidas de consistencia por su naturaleza trabajan sólo con características discretas. Para usarlas con características continuas, será necesario *discretizarlas* previamente, aunque luego el algoritmo de aprendizaje use la versión continua de las características seleccionadas.

Hay otros métodos basados en consistencia que, aunque no definen medidas específicas, puede ser interesante tener en cuenta. Schlimmer [Sch93] describe un algoritmo para deducir determinaciones lógicas usando el menor número posible de características, que, de hecho, es un algoritmo de selección de características inmerso. MIFES [OSV92], que puede hacer desde selección de características hasta construcción de características derivadas, presenta el concepto de cubrir pares de ejemplos para alcanzar consistencia con una representación matricial bastante intuitiva. Una aproximación más reciente [BHI⁺00] desarrolla un análisis lógico de datos que incluye selección de características inmersa.

Medida de consistencia básica. La más básica de las medidas de consistencia es la que simplemente averigua si el conjunto de datos restringido a las características seleccionadas es consistente o no. Esta medida fue usada por primera vez en FOCUS [AD91], como lo que ellos llaman “sufficiency test” (comprobación de suficiencia de las características seleccionadas). Tanto el proceso de búsqueda de FOCUS, como el de su versión optimizada FOCUS2 [AD94], usan esta medida para parar la búsqueda en el primer conjunto de características que la medida evalúe positivamente. El algoritmo realiza la búsqueda de tal forma que garantiza encontrar un conjunto minimal de características que haga consistente el conjunto de entrenamiento. De esta forma, se implementa la estrategia que ellos denominan “min-features-bias”.

Aunque se han conseguido buenos resultados usando la medida simple de consistencia, ésta tiene varias limitaciones. En primer lugar, el test de consistencia sólo puede usarse directamente con características discretas. Desarrollar una extensión del algoritmo FOCUS para tratar con otros tipos de características no es directo, ya que son posibles muchas estrategias. Algunas extensiones son CFOCUS, que definiremos en la sección 3.1, para trabajar con características continuas y FFOCUS, definido en la sección 3.2, para incorporar conocimiento experto en forma de características lingüísticas. En segundo lugar, esta medida tiene muy baja tolerancia al ruido. Un cambio en un único valor puede hacer inconsistente el conjunto de datos y forzar la inclusión de otra característica, que puede ser redundante o incluso irrelevante. En tercer lugar, la medida por sí misma es incapaz de aportar información para guiar la búsqueda. Es necesaria una estrategia adicional, como “min-features-bias”, o cualquier otra que sea capaz de dirigir la búsqueda de forma provechosa. Las medidas de consistencia que se describen en las siguientes secciones permiten mejorar la tolerancia al ruido y proveer medios para orientar la búsqueda, al proporcionar un grado de consistencia.

Todas las medidas de consistencia estudiadas pueden emular esta medida convirtiendo su valor de salida a booleano. Es así porque, cuando el conjunto de datos es consistente, estas medidas siempre devuelven un valor concreto (normalmente 1), y un valor distinto en caso contrario. Esta propiedad permite, por ejemplo, implementar FOCUS con cualquier medida de consistencia, aportando algunas ventajas como poder parar la búsqueda antes de alcanzar consistencia completa para gestionar el posible ruido.

Medida de consistencia de Liu. Liu, Motoda y Dash [LMD98] propusieron la primera medida de consistencia definida de forma independiente del proceso de búsqueda para selección de características. Recientemente, han probado esta medida con diversos métodos de búsqueda [DL03].

Esta medida usa un ratio de inconsistencia que se calcula encontrando todos los ejemplos con valores iguales en todas las características seleccionadas, y contando estos ejemplos menos el mayor número de ejemplos que pertenezcan entre ellos a la misma clase. El ratio se calcula dividiendo esta suma por el número total de ejemplos en el conjunto de datos.

Agrupando los ejemplos que tienen los mismos valores en todas las características seleccionadas, si llamamos ejemplos inconsistentes a aquéllos que no pertenecen a la clase mayoritaria de su grupo, la medida de Liu puede ser expresada, con la ecuación (2.6), como la proporción de estos ejemplos inconsistentes en el conjunto

de datos.

$$Inconsistencia = \frac{\text{número de ejemplos inconsistentes}}{\text{número total de ejemplos}} \quad (2.6)$$

Normalmente, el conjunto de las medidas que estamos describiendo son denominadas como “medidas de consistencia”, aunque lo que esta medida realmente mide es la inconsistencia. Para poder comparar las medidas y trabajar con ellas indistintamente, estableceremos la relación entre consistencia e inconsistencia. Como ambas están definidas en el intervalo $[0, 1]$ y parece razonable pensar en el grado de consistencia como el valor opuesto del grado de inconsistencia, en adelante las consideraremos relacionadas por la ecuación (2.7).

$$Consistencia = 1 - Inconsistencia \quad (2.7)$$

Algunos algoritmos de búsqueda, como Branch & Bound, requieren que la medida sea monótona para obtener resultados óptimos o mejores. La propiedad de monotonía requiere que si S_i y S_j son conjuntos de características y $S_i \subset S_j$, entonces $M(S_i, D) \leq M(S_j, D)$, donde M es la medida y D un conjunto de datos. Al igual que todas las otras medidas de consistencia estudiadas, esta medida presenta la propiedad de la monotonía.

Podemos encontrar una equivalencia que, además, nos da una interpretación intuitiva para esta medida. Y es que la medida de Liu es equivalente a la precisión que obtendría un clasificador de memoria* sobre el conjunto de datos con las características dadas. En otras palabras, la probabilidad de que un ejemplo del conjunto de entrenamiento sea clasificado correctamente por el *memory classifier*.

La medida de Liu no está definida para conjuntos de datos con características continuas, pero puede ser usada en combinación con algunos métodos de *discretización*, como ya fue sugerido por sus autores. En un paso previo, se *discretiza* el conjunto de datos, para aplicar posteriormente la selección de características sobre el conjunto *discretizado*. Una vez que las características han sido seleccionadas, el algoritmo de aprendizaje puede usar las características *discretizadas*, o su versión continua del conjunto de datos original. El resto de las medidas de consistencia tampoco tienen una aplicación directa sobre datos continuos.

* *Memory classifier*, también conocido como *table classifier* o *RAM*, es un clasificador que almacena en memoria todos los ejemplos de entrenamiento y clasifica con la clase más frecuente para cada tipo de ejemplo)

Medida de consistencia de la teoría de Rough sets. La presente medida viene de la teoría de los *Rough Sets* [Paw91, KPPS98, PS98], y está descrita en [Paw91](capítulo 7.8). Ha sido usada en *discretización* [CGB96], e incluso ha sido comparada con la medida de Liu [TS02] en un algoritmo de *discretización*, pero no hemos encontrado ningún trabajo previo en el que esta medida haya sido usada para guiar una búsqueda de selección de características.

Hay algunos métodos de selección de características basados en la teoría de los *Rough Sets* como [Mod93]. Un resumen del uso de esta teoría para trabajar con relevancia de características puede ser encontrado en el capítulo 7.1 de [KPPS98]. Zhong et al. [ZDO01] multiplican un factor por la medida de consistencia para seleccionar características que generen reglas simples.

A continuación, se introducen únicamente los conceptos esenciales de la teoría de *Rough Sets* necesarios para describir la medida de consistencia. Denote U el universo de trabajo, esto es, el conjunto de todos los ejemplos del conjunto de datos. Sea F el conjunto de todas las características, $S \subseteq F$ algunas características seleccionadas y $f(x)$ el valor de la característica f en el ejemplo x . La relación de indiscernibilidad se define como:

$$\text{IND}(S) = \{(x, y) \in U \times U, \forall f \in S f(x) = f(y)\} \quad (2.8)$$

Esta relación de equivalencia particiona U en clases de equivalencia, y la partición (conjunto de clases de equivalencia) será denotada $U/\text{IND}(S)$.

Para un subconjunto de ejemplos $X \subseteq U$, por ejemplo el conjunto de ejemplos pertenecientes a una clase dada, la *S lower approximation* (aproximación por debajo) de X se define por:

$$\underline{S}X = \bigcup \{Y \in U/\text{IND}(S) : Y \subseteq X\} \quad (2.9)$$

Si tomamos X como el conjunto de ejemplos de una clase, $\underline{S}X$ representa aquellos ejemplos que pueden ser consistentemente identificados como miembros de esa clase usando las características de S . Podemos aplicar esta idea a cada clase y definir la *positive region* (región positiva) con la siguiente ecuación, donde D denota el conjunto de características dependientes. Éstas son normalmente un único atributo del ejemplo que identifica la clase.

$$POS_S(D) = \bigcup_{X \in U/IND(D)} \underline{S}X \quad (2.10)$$

El grado de consistencia viene dado por la proporción de estos ejemplos consistentemente clasificables en el número total de ejemplos. La medida se muestra en la siguiente ecuación:

$$\gamma(S, D) = \frac{|POS_S(D)|}{|U|} = \sum_{X \in U/IND(D)} \frac{|\underline{S}X|}{|U|} \quad (2.11)$$

Mientras que el resto de la medida de Liu mide lo que le falta a un conjunto de datos para ser consistente, esta medida mide que parte es ya consistente. Intuitivamente, podemos ver esta medida como más conservadora, ya que, en el momento en el que en un grupo ejemplos indistinguibles hay ejemplos pertenecientes a más de una clase, todos los ejemplos son descartados, bajo la suposición de que podrían ser clasificados erróneamente.

Como se ha comentado antes, todas las medidas de consistencia presentan la propiedad de la monotonía. En el caso de esta medida, el cumplimiento de esta propiedad puede ser fácilmente visto con la demostración esbozada en la ecuación (2.12).

$$\begin{aligned} \forall c \in U/IND(S \cup \{f\}) \exists \hat{c} \in U/IND(S) : c \subseteq \hat{c} \longrightarrow \\ \forall X \in \mathcal{P}(F), \underline{S}X \subseteq \underline{S \cup \{f\}}X \longrightarrow \gamma(S, D) \leq \gamma(S \cup \{f\}, D) \end{aligned} \quad (2.12)$$

2.6.2.2. Medidas basadas en la teoría de la información

Las medidas de este apartado se basan en la teoría de la información de Shannon [CT91].

Midiendo la información que aportan las características sobre la clase podemos saber cuales son más informativas, siendo éstas, desde el punto de vista de la teoría de la información, las más apropiadas para la clasificación. Muchos algoritmos de aprendizaje se basan en principios de la teoría de la información, lo que, además de indicar que el uso de estas medidas es prometedor, nos lleva a pensar que habrá una sinergia positiva entre los métodos de selección de características que usen estas medidas y los algoritmos de aprendizaje basados en teoría de la información.

Información mutua. La teoría de la información establece una forma básica de medir la información que aporta el conocimiento de los valores que toman una o más variables sobre otra. Sea C la variable aleatoria que define la clase de un problema de clasificación. La entropía de C viene dada por:

$$H(C) = - \sum_{c \in C} p(c) \log p(c) \quad (2.13)$$

El objetivo del algoritmo de aprendizaje es reducir la incertidumbre sobre el valor de la clase. Para ello, el conjunto de características seleccionadas S aporta la cantidad de información dada por:

$$I(C, S) = H(C) - H(C|S) \quad (2.14)$$

Lo ideal sería encontrar el menor conjunto de características que determine completamente C , esto es, $I(C, S) = H(C)$, pero no siempre es posible. En su defecto, buscaremos el que aporte máxima información.

Esta medida cumple la propiedad de la monotonía. Al añadir una característica más, ésta siempre aportará algo de información o, en el peor de los casos, nada, pero nunca reducirá la información que aportan las características ya seleccionadas.

Los algoritmos de búsqueda que usen esta medida deberán tener en cuenta la propiedad de monotonía, pues en un conjunto de datos no completamente determinado se tenderá a seleccionar todas las características. Así, se debe buscar un conjunto que aporte mucha información, pero con un número reducido de características.

De forma similar a la implementación de las medidas de consistencia, hemos programado un algoritmo que calcula esta medida de forma muy eficiente. La eficiencia de nuestra implementación en el caso promedio pertenece a $O(n)$.

Es posible aplicar una versión continua de esta medida para problemas con características continuas. Podemos ver un ejemplo en este trabajo [SPLC04].

Incetidumbre simétrica. La medida de información mutua es simétrica, siendo

$$I(C, S) = H(C) - H(C|S) = H(S) - H(S|C) = I(S, C) \quad (2.15)$$

Esto es una propiedad deseable para las medidas de correlación entre dos variables. Por otra parte, la medida de información mutua tiende a dar mayor valor a las características con más valores. Si queremos que todas las características sean valoradas equitativamente, podemos usar:

$$U(C|S) = \frac{H(C) - H(C|S)}{H(S)} \quad (2.16)$$

Sin embargo, con esta definición, la medida dejaría de ser simétrica. Por esta razón, se define la medida de incertidumbre simétrica [YL03, PFTV98] como:

$$SU(C, S) = 2 \cdot \frac{H(C) - H(C|S)}{H(S) + H(C)} \quad (2.17)$$

Intuitivamente, esta medida puede interpretarse como el ratio entre la cantidad de información que aportan las características seleccionadas y la cantidad de información total que contienen y, por tanto, la información que podrían aportar. Al ser un ratio, cuyo denominador puede crecer más rápido que el numerador al incluir características, esta medida no cumple la propiedad de monotonía. En cuanto a su eficiencia, dado que es bastante parecida a la medida de información mutua, también hemos podido implementar su cálculo con un algoritmo en el orden de eficiencia promedio de $O(n)$.

MDL. Las medidas con la propiedad de monotonía tienen el inconveniente de no indicar directamente qué características son completamente irrelevantes, pues al añadir una característica más, ésta incrementa la medida aunque sea muy ligeramente. La medida MDL[SDN90] pretende resolver este problema aplicando el criterio MDLC (longitud mínima de la descripción de un modelo). Este criterio sostiene que entre varios modelos de ajuste se debe elegir el que tenga una descripción más corta. Se puede ver como el principio de la navaja de Ockham aplicado a la teoría de la información. De esta forma, se tiene en cuenta que cuantas más características hay, más complejo es el modelo, permitiéndonos discernir cuándo ya no merece la pena usar más características.

Suponiendo que las muestras del conjunto de datos son gaussianas, independientes e idénticamente distribuidas para cada clase, sus autores definen la medida como:

$$\text{MDL}(T) = \left[\sum_{c \in \mathcal{C}} \frac{N_c}{2} \log \frac{|\hat{R}_x(c)|}{|\hat{R}_x|} \right] + h_T \quad (2.18)$$

donde T es el conjunto de entrenamiento proyectando sólo las características seleccionadas, N_c el número de ejemplos que pertenecen a la clase c , $\hat{R}_x(c)$ es la matriz de covarianza de los ejemplos pertenecientes a la clase c , \hat{R}_x la matriz de covarianza de todos los ejemplos (T) y

$$h_T = \frac{1}{2}(m - J)(m + J + 3) \log N + \sum_{c \in \mathcal{C}} J(J + 3) \log N_c \quad (2.19)$$

donde J es el número de características seleccionadas y m el número total de características disponibles.

Esta medida tiene una interpretación intuitiva basándose en que el determinante $|\hat{R}_x(c)|$ puede verse como el volumen ocupado por la nube de los ejemplos de una clase. El ratio $\frac{|\hat{R}_x(c)|}{|\hat{R}_x|}$ representa el volumen ocupado por la nube de una clase en el volumen total del espacio generado por las características útiles. Es fácil ver que cuando las características son útiles y separan claramente la clase, los ratios serán más pequeños.

2.6.2.3. Medidas basadas en distancia

Estas medidas valoran las distancias que hay entre las distribuciones de probabilidad de cada clase. La idea detrás de estas medidas es que aquellas características que hagan mayor la distancia entre las distribuciones las separaran mejor y, por tanto, deben permitir hacer mejor la clasificación.

Un ejemplo de estas medidas es la función discriminante que usaban Narendra y Fukunaga [NF77] para problemas con dos clases:

$$J_d = (M_1 - M_2)^T \left(\sum_1 + \sum_2 \right)^{-1} (M_1 - M_2) \quad (2.20)$$

donde M_i es el vector con los valores medios de cada característica y \sum_i la matriz

de dispersión de cada clase.

2.6.2.4. Medidas basadas en estimaciones sobre el rendimiento del algoritmo de aprendizaje

Una de las técnicas de selección de características más conocidas es el uso de la estrategia envolvente (*wrapper*), mencionada en la sección 2.3.1. En esta estrategia se usa el algoritmo de aprendizaje para evaluar la calidad de los conjuntos de características. En este caso, se usa como medida de relevancia de las características alguna medida de la calidad de las soluciones que obtiene el algoritmo de aprendizaje, aplicándolo únicamente con el conjunto de características a evaluar.

La valoración de los resultados se puede realizar subdividiendo el conjunto de datos en entrenamiento y test, o aplicando alguna técnica como la validación-cruzada (*cross-validation*), o dejar-uno-fuera (*leave-one-out*). La medida de la calidad del resultado puede ser el porcentaje de clasificación, el error promedio, o alguna medida del ajuste del modelo, generado por el algoritmo, a los datos.

Una ventaja del uso de estas medidas es que el algoritmo de selección de características tiene una evaluación de las características en el entorno real en que será aplicado y, por tanto, tiene en cuenta las posibles particularidades del algoritmo de aprendizaje que se use. Esta ventaja se considera tan importante que se tiende a pensar que ésta es la medida óptima, pero hay algunos inconvenientes que la hacen no adecuada en bastantes situaciones.

En primer lugar, hay que tener en cuenta la eficiencia del algoritmo de aprendizaje, que puede hacer que esta medida sea mucho más lenta o consuma más memoria que otras medidas. En segundo lugar, puede que el algoritmo de aprendizaje no sea aplicable con un número elevado de características. De hecho, éste es uno de los motivos para usar selección de características. Estos algoritmos no permiten aplicar la técnica *wrapper* con determinadas estrategias de búsqueda, como comentábamos en la sección 2.5.2. Finalmente, se debe tener en cuenta que en algunas circunstancias otras medidas podrían ser mejores para guiar la búsqueda hacia el conjunto óptimo de características (ver sección 3.4).

Capítulo 3

Mejoras de la selección de características

En el capítulo anterior, hemos estudiado con detalle el problema de la selección de características y los métodos actuales que se aplican al mismo. Partiendo de estos conocimientos ahora, en este capítulo, presentamos y evaluamos algunas propuestas, que incorporamos al conjunto de técnicas actuales para selección de características. Las cuatro aportaciones propuestas mejoran en ciertos aspectos las técnicas actuales, y harán avanzar el estado del arte en selección de características.

En las dos primeras propuestas, planteamos extensiones del algoritmo FOCUS. Este algoritmo es una referencia teórica importante en selección de características. Sin embargo, tiene un ámbito de aplicación muy reducido: únicamente es aplicable a problemas lógicos con características booleanas. La primera extensión propuesta (CFOCUS), de carácter eminentemente práctico, permite aplicar FOCUS a problemas de clasificación reales con características discretas y continuas. En la segunda (FFOCUS), proponemos adoptar un enfoque lingüístico. Pensamos que esta combinación puede aportar soluciones novedosas en la selección de características. Entre otras ventajas, facilita la incorporación de conocimiento experto en el proceso de selección de características, lo que puede resultar muy beneficioso.

Las otras dos propuestas son medidas de la utilidad de conjuntos de características, como las descritas en la sección 2.6.2. Éstas aportan mayor flexibilidad que los métodos anteriores porque pueden ser usadas con cualquier estrategia de búsqueda. La primera medida (IEP) está basada en la consistencia. Como continuación al trabajo desarrollado sobre FOCUS, planteamos una medida que puede cuantificar grados de inconsistencia. Esto permite orientar la búsqueda hacia conjuntos con

una menor inconsistencia, y trabajar sobre conjuntos de datos no completamente consistentes (presencia de ruido o problemas complejos).

Con la segunda medida propuesta (RFSM), pretendemos mejorar la orientación de la búsqueda en el espacio de conjuntos de características. La búsqueda es un aspecto muy costoso de la selección de características. Orientarla adecuadamente, además de reducir el coste computacional, puede mejorar los resultados obtenidos. Basándonos en ideas tomadas de la medida de características individuales Relief, nuestra medida valora mejor, no sólo los conjuntos de características que son buenos para clasificación, sino también aquellos cercanos a estos.

3.1. CFOCUS

3.1.1. El algoritmo FOCUS original

FOCUS pertenece a los algoritmos que aplican la selección de características mediante estrategia de filtro, y su ámbito de aplicación es el de los problemas de clasificación binarios (aprendizaje de un concepto booleano) con características booleanas. Por tanto, los datos de entrada, en adelante referidos como *Muestra*, serán tuplas de valores booleanos para cada una de las n características y un valor booleano que representa el concepto que se desea aprender. El problema que afronta FOCUS es seleccionar características de forma que con ellas baste para determinar una función booleana que represente el concepto.

El algoritmo FOCUS fue expuesto originalmente por Almuallim y Dietterich en [AD91]. Para su desarrollo, siguiendo la sugerencia de otros autores [Bun90, Wol90], usan un proceso algo distinto al más convencional de elegir un esquema de representación y desarrollar un algoritmo que busque soluciones con ese esquema. De esta forma, ellos abogan por adoptar primero una estrategia de tendencia (predisposición, más formalmente, una distribución de probabilidad sobre el espacio de hipótesis). A continuación, seleccionar un esquema para representar las hipótesis y desarrollar el algoritmo que implemente la estrategia previamente definida. Nosotros emplearemos aquí el mismo orden para la explicación del algoritmo. En la descripción original del algoritmo también se incluye la formulación de una función booleana representando el concepto a partir de las características de A . Nosotros nos centraremos únicamente en la selección de características.

Almuallim y Dietterich establecen lo que ellos llaman “MIN-FEATURES bias”

como estrategia de tendencia o predisposición a seguir por el algoritmo. Ésta consiste en, dada un conjunto de ejemplos que representa un concepto, escoger uno cualquiera de los subconjuntos con el mínimo número de características relevantes para el concepto. Una característica i es relevante para el concepto c si existen dos ejemplos $A \in c$ y $B \notin c$ tales que $A_i \neq B_i$ y $\forall_{j \neq i} A_j = B_j$. La idea es que dadas dos funciones, que representan el concepto expresado en la muestra de ejemplos, “MIN-FEATURES bias” elige aquélla con el menor número de características. Además, demuestran que cualquier algoritmo que implemente “MIN-FEATURES bias” requiere un número de ejemplos del orden exacto indicado en la expresión siguiente:

$$\Theta \left(\frac{1}{\epsilon} \ln \frac{1}{\delta} + \frac{1}{\epsilon} [2^p + p \ln(n)] \right) \quad (3.1)$$

siendo p el número de características relevantes del espacio de las n posibles características, con un parámetro de precisión ϵ y δ el de confianza.

Intuitivamente, podemos ver cómo una característica es relevante si nos permite separar entre dos ejemplos de distinta clase. El algoritmo por tanto va a trabajar con pares de ejemplos, especialmente con aquéllos cuya clase (pertenecer o no al concepto) sea diferente; a éstos los llamaremos en adelante “conflictos”. Para simplificar, el conflicto representado por dos ejemplos e_i y e_j con clase diferente puede ser visto como una tupla o vector en el que en cada posición indica si los valores de e_i y e_j para esa característica son diferentes o iguales. La transformación del conjunto de ejemplos de entrada en el conjunto de conflictos puede verse como un cambio para quedarnos con la información que realmente nos interesa de esos ejemplos. El inconveniente es que el número de conflictos es de $O(m^2)$ siendo m el número de ejemplos. En la implementación del algoritmo será aconsejable mantenerlos implícitos en lugar de almacenarlos en memoria.

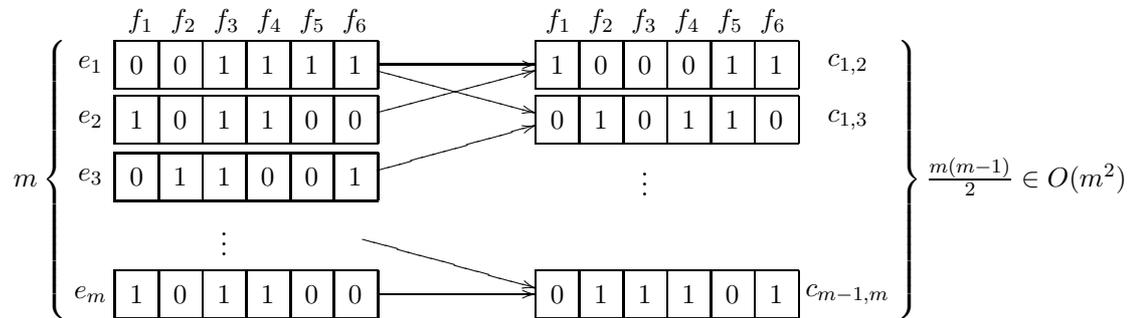


Figura 3.1: Ejemplos de un conjunto de datos y sus respectivos conflictos

A través de una búsqueda exhaustiva, FOCUS es capaz de identificar todas las características fuertemente relevantes y el conjunto mínimo necesario de las débil-

mente relevantes (ver sección 2.2). FOCUS siempre encuentra el conjunto óptimo con el menor número de características a través de la búsqueda completa en el espacio de conjuntos de características. Es, por tanto, un algoritmo ideal para problemas en los que encontrar un conjunto mínimo de características sea crucial.

El algoritmo FOCUS es una implementación simple y exacta de “MIN-FEATURES bias”. Realiza una búsqueda exhaustiva en orden de cardinal creciente de todos los conjuntos. En la figura 3.2 se expone el pseudocódigo que describe el algoritmo.

FOCUS(*Muestra*)

Para $i = 0, 1, 2, \dots, n$ hacer:

Para todo $A \subseteq \{f_1, f_2, f_3, \dots, f_n\}$ de tamaño i hacer:

Si Test-de-suficiencia($A, Muestra$):

devolver A .

Figura 3.2: Algoritmo FOCUS

La prueba de suficiencia de un conjunto de características A para un conjunto de ejemplos *Muestra* indica si con las características en A se puede construir una hipótesis consistente con todos los ejemplos de *Muestra*. Esta comprobación es sencilla. Basta con ver si hay algún par de ejemplos con los mismos valores en todas las características de A pero que sean de diferente clase.

La eficiencia en tiempo del algoritmo FOCUS viene dada por el número de subconjuntos de características que se prueban. Se prueban todos los conjuntos de características de tamaño $0, 1, 2, 3, \dots$ hasta llegar a uno suficiente con p características, así que se prueban:

$$\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \dots + \binom{n}{p} = O(n^p) \quad (3.2)$$

Para cada uno de estos conjuntos se realiza un test de suficiencia, que comprobando todos los pares de m ejemplos, tiene una eficiencia en el peor caso de $O(m^2)$. Así que la eficiencia de FOCUS es $O(m^2 \cdot n^p)$. Esto, en principio, puede parecer muy lento, pero es importante fijarse que el término exponencial es sólo en función del número de características relevantes, de modo que es aplicable a problemas con gran cantidad de características, siempre que con un número reducido de ellas se pueda determinar el concepto objetivo.

La primera versión del algoritmo FOCUS que acabamos de describir es inte-

resante para exponer de una forma clara el funcionamiento de “MIN-FEATURES bias” y ver las propiedades que podemos esperar de un algoritmo que la implementa. No obstante, para la utilización práctica, Almuallim y Dieterich [AD94] desarrollaron una versión optimizada que llamaron FOCUS-2. Esta versión mejora mucho la eficiencia en el caso promedio, aunque la eficiencia teórica en el peor caso sigue siendo de orden exponencial. A continuación, vemos en qué se basa esta optimización.

Podemos observar que si un conflicto sólo se puede resolver con la característica f_i , esta característica debe aparecer obligatoriamente en cualquier conjunto suficiente. Por extensión, si un conflicto sólo puede resolverse con un conjunto de características sabemos que al menos una de ellas debe ser seleccionada. Esta información puede ser aprovechada para orientar la búsqueda y evitar la comprobación de muchos subconjuntos que de entrada sabemos no son viables. Una forma de hacer esto es buscar los conflictos resueltos por el menor número de características y empezar la búsqueda con los conjuntos que incluyen estas características.

La búsqueda usando la idea anterior no es exhaustiva, dado que se evitan muchos posibles subconjuntos, pero sigue siendo completa y garantizando la optimalidad de la solución. El algoritmo resultante es el siguiente:

FOCUS-2(*Muestra*)

Si todos los ejemplos de la *Muestra* tienen la misma clase, devolver \emptyset .

Sea G el conjunto de todos los conflictos presentes en la *Muestra*.

$Cola = \{M_{\emptyset, \emptyset}\}$.

Repetir

$M_{A,B} =$ Cabecera extraída de $Cola$.

$Descartados = B$.

Sea a el conflicto de G no cubierto por ninguna de las características de A , tal que $|Z_a - B|$ sea mínimo, donde Z_a es el conjunto de características que cubren a .

Para cada $f \in Z_a - B$

Si $Suficiente(A \cup \{f\}, Muestra)$, devuelve $A \cup \{f\}$.

Insertar $M_{A \cup \{f\}, Descartados}$ al final de $Cola$.

$Descartados = Descartados \cup \{f\}$.

$M_{A,B}$ representa el espacio de todos los subconjuntos de características que contienen todas las características de A y ninguna de las contenidas en B .

Figura 3.3: Algoritmo FOCUS-2

El test de suficiencia también se puede implementar de forma más eficiente.

En vez de recorrer todos los pares de ejemplos, con una eficiencia de $O(n^2)$, se pueden ir realizando particiones del conjunto de ejemplos en función del valor de una característica en cada paso. En el momento en que no haya en ninguna partición ejemplos de dos clases distintas, podemos decir que ese conjunto de características es suficiente. La eficiencia de este proceso es de $O(n \cdot m)$, donde n es el número de características y m el número de ejemplos.

3.1.2. Extensión continua (CFOCUS)

La idea principal del algoritmo FOCUS original es identificar todos los pares de ejemplos con resultado diferente. Cada uno de estos pares se considera un conflicto. Esto crea una nueva forma de ver el problema de la selección de características donde el objetivo es seleccionar una serie de características que permitan resolver todos los conflictos. Las características que resuelven un conflicto son aquéllas que tienen diferente valor en los dos ejemplos que forman el conflicto. Esta idea es bastante intuitiva y tiene cierta similitud con otros algoritmos como Relief [KR92] que también usan pares de ejemplos.

Nuestra primera propuesta es aplicar las ideas y la forma de funcionamiento del algoritmo FOCUS a problemas más diversos. Pretendemos, de esta manera, ampliar el campo de aplicación de estas técnicas, que presentan buenos resultados en el dominio booleano, para que trabajen con datos que se presentan con frecuencia en los problemas reales.

FOCUS presenta propiedades interesantes. Entre ellas, están su optimalidad, encontrando siempre el conjunto mínimo, y una eficiencia en función de número de características relevantes abordable en muchos de los casos en los que nos interesa la selección de características, que es cuando hay un número pequeño de características relevantes. Además, aunque está diseñado para conjuntos de datos sin ruido, ha mostrado cierta tolerancia a niveles bajos de ruido [DL97].

El principal punto débil del algoritmo FOCUS es que su dominio de aplicación es bastante reducido, ya que se limita sólo a problemas de conceptos lógicos definidos en base a características booleanas. Nos encontramos por tanto ante un algoritmo potente, pero que no puede ser aplicado a la mayoría de los problemas prácticos que nos podemos encontrar.

Hay muchos problemas de aprendizaje cuyas conjuntos de ejemplos contienen características discretas que pueden tomar varios valores, o características con me-

didadas discretas o valores continuos. Un ejemplo ilustrativo puede ser el problema de clasificación de figuras geométricas que se muestra más adelante en la evaluación experimental. La clasificación se debe realizar a partir de algunas de las características de las figuras geométricas, siendo unas discretas y otras de tipo continuo. En otros ejemplos, como pueden ser el problema real de aplicación en Economía que comentaremos, y muchos de los que se pueden encontrar en las bibliotecas de problemas de Aprendizaje Automático como Delve [Tor03], y UCI [HB99], además, es frecuente encontrarse en los datos con valores nulos (valores desconocidos o no aplicables). La aplicación de la selección de características puede ser útil en muchos de estos casos; sin embargo FOCUS, no es aplicable en ellos.

Para poder abordar los problemas comentados, a continuación vamos a proponer una serie de mejoras. En primer lugar, plantearemos la extensión para que FOCUS trabaje con valores discretos y varias clases. Esta extensión es directa como ya apuntaban Almuallim y Dieterich en las conclusiones de su trabajo [AD94], pero no se ha evaluado empíricamente. En segundo lugar, proponemos una extensión para características con valores de tipo real o entero que representen una medida. A continuación, explicamos cómo gestionará nuestra extensión los valores nulos. Finalmente, es interesante que los algoritmos sean capaces de tolerar que aparezca algo de ruido en los datos. El ruido puede venir de errores en las mediciones, o algún otro fallo en el proceso de recogida de datos. La extensión que proponemos tolera un cierto nivel de ruido en las características continuas.

La extensión para características discretas la haremos simplemente considerando que los valores pueden ser discretos en la definición de característica relevante: una característica i es relevante para el concepto c si existen dos ejemplos $A \in c$ y $B \notin c$ tales que $A_i \neq B_i$ y $\forall_{j \neq i} A_j = B_j$. La definición es idéntica a la expuesta anteriormente en el algoritmo FOCUS. La interpretación es lo único que debe variar para considerar ahora que puede haber varios valores discretos en vez de solamente “verdadero” y “falso”. De la misma forma, basta con hacer la misma consideración en el resto del algoritmo para ver que su funcionamiento es correcto con varios valores discretos.

También es directa la extensión para trabajar con varias clases (que la característica que representa el concepto a aprender sea discreta). En la definición de característica relevante, donde vemos que el ejemplo $A \in c$ y la $B \notin c$, basta con considerar que c es una clase en vez de un concepto booleano. Así $B \in d$ donde d es una clase y $d \neq c$. Recordamos que hablamos de problemas de clasificación, y por tanto, cuando hablamos de clase nos referimos a particiones disjuntas en que están clasificados los ejemplos, un ejemplo sólo puede pertenecer a una de ellas. De esta forma, los conflictos considerados en el algoritmo serán todos los pares de ejemplos

que pertenezcan a clases distintas.

Para características continuas (con valores reales o enteros de tipo medida), la extensión requiere un replanteamiento de la estrategia “MIN-FEATURES bias”. Mientras que en los dominios booleano y discreto está claro cuándo dos valores son diferentes y, por tanto, cuándo una característica es relevante y cuándo resuelve un conflicto, para las características con valores continuos esto no es tan simple. Si se toma literalmente la definición de característica relevante, e intentamos aplicarla a valores reales, nos encontraremos con que todos los valores serán diferentes, ya que la probabilidad teórica de que dos valores reales tomados aleatoriamente sean iguales tiende a 0. Con valores enteros que representen medidas o conceptos continuos pasa algo similar, se considerarían valores diferentes algunos valores que a efectos del problema son prácticamente lo mismo.

Un ejemplo que ilustra este problema es el siguiente. Supongamos una característica que representa distancias. En un ejemplo del problema, esta característica toma el valor 3485Km y, en otro, toma el valor 3484Km. No parece que esta diferencia vaya a ser relevante, pero en cambio estrictamente hablando son diferentes. Esto podría hacer que esta característica resuelva un conflicto y sea considerada relevante.

Así, la extensión directa debe ser rechazada porque nos llevaría a la solución absurda de que cualquier característica de tipo real sirve para resolver cualquier problema. En la práctica, podrían encontrarse valores iguales que obligasen a escoger alguna característica más, pero en cualquier caso sería más bien fruto de la casualidad que de una relación cierta con la solución del problema. Dependería de la precisión con la que se ha medido la característica, que no tiene por qué estar relacionada con la relevancia.

Una solución puede ser considerar que los ejemplos son diferentes cuando la distancia (o diferencia en valor absoluto) es mayor que cierta cantidad. Podríamos solicitar que, junto con la muestra que describe el problema, el usuario indicase la cantidad mínima para considerar diferentes los valores de cada característica. Si bien ésta podría ser una idea interesante, y de hecho, en la sección siguiente, se estudia una propuesta más avanzada para incorporar el conocimiento sobre las características que nos pueda aportar el usuario, no siempre es posible disponer de esta información. Puede ocurrir que el usuario no sea experto en el problema o, simplemente, que el problema sea demasiado complejo como para poder deducir qué distancias se pueden considerar relevantes y cuáles no.

Nuestra propuesta consiste en considerar que dos valores son “significativamen-

te iguales” cuando su diferencia en valor absoluto sea relativamente menor de un cierto umbral. Con “relativamente” indicamos que se debe tener en cuenta el rango en el que aparecen valores de esa característica. Esto lo haremos normalizando todas las características para que sus valores estén dentro del intervalo $[0, 1]$. El umbral considerado será un parámetro, que llamaremos U , del algoritmo, y será único para todas las características. Consideraremos que dos valores son “significativamente distintos” cuando no sean “significativamente iguales”.

La extensión de la estrategia “MIN-FEATURES bias” para características con valores continuos, se define igual que la original cambiando en la definición de relevancia las comparaciones igual y distinto respectivamente por los conceptos “significativamente igual” y “significativamente distinto”.

Finalmente, la gestión de valores nulos la incorporaremos de una forma bastante intuitiva. Un conflicto representa un par de ejemplos indicando qué características lo resuelven (distinguen entre los dos ejemplos). Un valor nulo en una característica de alguno de los dos ejemplos es inútil para distinguir entre ellos puesto que no conocemos su valor. Por tanto, indicaremos en el conflicto que esa característica no lo resuelve. Se indicará de la misma forma que si tuviesen valores iguales, que tampoco lo resolvería. De esta forma, una vez definido el conflicto a partir de ejemplos que puedan contener valores nulos, el algoritmo a aplicar es el mismo.

Si el valor nulo aparece en la característica que indica la clase a la que pertenece un ejemplo no podemos extraer información sobre cómo clasificarlo. Estos ejemplos son inútiles y simplemente los ignoraremos.

Denominaremos CFCOCUS al nuevo algoritmo que incluye todas estas extensiones. CFCOCUS mantiene el mismo comportamiento del algoritmo original con las características booleanas. De esta forma, al ser aplicado en problemas cuyos valores sean todos booleanos obtendremos los mismos resultados que con el algoritmo FOCUS.

Hay algunas diferencias adicionales en CFCOCUS. En el caso booleano si los pares de ejemplos (e_1, e_2) y (e_1, e_3) representaban un conflicto sabíamos que el par (e_2, e_3) no lo era. Trabajando con varias clases no ocurre lo mismo. Por otra parte, no es posible hacer particiones en función de las características continuas siguiendo el criterio de distinción, ya que éste no cumple la propiedad transitiva. Por ejemplo, si el umbral se fija en 0.2, los valores 0 y 0.15 se consideran iguales, también consideraremos iguales 0.15 y 0.3, pero no pueden aparecer todos en la misma partición ya que 0 y 0.3 son distintos según nuestro criterio. El inconveniente de estas diferencias es que no se puede aplicar la versión optimizada del test de suficiencia.

Como test de suficiencia nosotros hemos empleado una búsqueda simple en la *Muestra* de dos ejemplos que con valores no distintos, según nuestro criterio de distinción, en las características seleccionadas tengan diferente clase. Si se encuentran el conjunto de características seleccionadas no es suficiente, en otro caso si lo es.

3.1.3. Estudio empírico

Para evaluar el funcionamiento del algoritmo CFOCUS hemos realizado diferentes tipos de pruebas que expondremos a continuación. En primer lugar, veremos la aplicación sobre un problema artificial. Este problema ha sido creado expresamente para ilustrar el funcionamiento de CFOCUS, pero con similitud a posibles problemas reales. Como conocemos las relaciones entre las características, sabemos cuáles son relevantes y podemos ver si el algoritmo escoge las correctas. En segundo lugar, utilizaremos un problema de tipo banco de pruebas de la biblioteca de problemas de Aprendizaje Automático UCI [HB99]. En este caso, a diferencia del anterior, no conocemos *a priori* las características relevantes. Así que para evaluar los resultados utilizaremos un clasificador y los compararemos con los obtenidos sin realizar selección de características y usando otra técnica de selección de características. Finalmente, en tercer lugar, mostraremos la aplicación de CFOCUS y un clasificador basado en redes neuronales a problemas reales con la ayuda de expertos en el problema para evaluar los resultados.

3.1.3.1. Problema de las figuras geométricas

El primer problema sobre el que aplicaremos CFOCUS consiste en clasificar figuras geométricas a partir de algunas de sus características básicas (ver figura 3.4). Es un problema que contiene características discretas y continuas, ilustrativo por su simplicidad, y similar a otros problemas que se podrían encontrar en situaciones reales.

Es interesante ver la aplicación de la selección de características a este problema porque presenta varias características redundantes. Además, incluiremos características irrelevantes con el objetivo de asemejar las que podrían encontrarse en un problema práctico.

Tomamos ejemplos de las siguientes figuras:

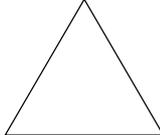
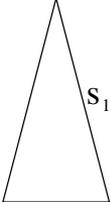
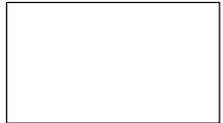
	T. equilátero	T. isósceles	Cuadrado	Rectángulo
				
	s_1	s_2	s_1	s_2
<i>Nº lados</i>	3	3	4	4
<i>Lado mayor</i>	s_1	$\max(s_1, s_2)$	s_1	$\max(s_1, s_2)$
<i>Lado menor</i>	s_1	$\min(s_1, s_2)$	s_1	$\min(s_1, s_2)$
<i>Perímetro</i>	$3 \cdot s_1$	$2 \cdot s_1 + s_2$	$4 \cdot s_1$	$2 \cdot s_1 + 2 \cdot s_2$
<i>Área</i>	$\sqrt{\frac{3}{4}} s_1^2$	$\frac{s_2 \sqrt{4s_1^2 - s_2^2}}{4}$	s_1^2	$s_1 \cdot s_2$
$\frac{\text{Lado menor}}{\text{Lado mayor}}$	1	$\frac{\text{Lado menor}}{\text{Lado mayor}}$	1	$\frac{\text{Lado menor}}{\text{Lado mayor}}$

Figura 3.4: Figuras geométricas y los valores de sus características

- Triángulo equilátero
- Triángulo isósceles
- Cuadrado
- Rectángulo

Y para cada uno de ellos consideramos las siguientes características:

- Número de lados
- Longitud del lado menor
- Longitud del lado mayor
- Perímetro
- Área
- Relación lado menor / lado mayor

La descripción del problema, con las relaciones que hay entre las características consideradas y el tipo de figura, se muestran en la figura 3.4 se muestran los valores de las características para cada una de las figuras utilizadas.

El objetivo de nuestro algoritmo es seleccionar el menor conjunto de características que nos permita clasificar cada ejemplo en uno de los 4 tipos de figura. Nuestro conocimiento del problema nos indica que, de las características disponibles, la combinación que permite clasificar todas las figuras correctamente con menor número de características es: $\{N^{\circ} \text{ Lados}, \frac{\text{Lado menor}}{\text{Lado mayor}}\}$. Otras combinaciones como $\{\text{Lado mayor}, \text{Lado menor}, \text{Area}\}$ también permiten realizar la clasificación correctamente aunque no son soluciones óptimas por incluir un número mayor de características.

Para la generación de los conjuntos de ejemplos se ha seguido el proceso siguiente, repetido N veces para generar N ejemplos:

- Elegir el tipo de figura (Generador aleatorio uniforme 0, 1, 2, 3).
- Mientras no se verifiquen las restricciones:
 - Generar la longitud de los lados (Generador aleatorio uniforme $[0, 1]$).
- Calcular el resto de las características de la figura y almacenar el ejemplo.

Las restricciones a las que se hace mención son: a) que en los triángulos isósceles, por Geometría, la suma de la longitud de los lados iguales debe ser mayor que el lado diferente, y b) que la diferencia de los lados s_1 y s_2 en rectángulos y triángulos isósceles sea mayor del 5%, para evitar su confusión con los cuadrados y triángulos equiláteros, respectivamente.

De esta forma, la aparición de cualquier figura es equiprobable y la distribución de las formas también es equiprobable dentro de los rangos permitidos para la longitud de los lados. Se excluyen únicamente aquellos rectángulos que son casi cuadrados y aquellos triángulos isósceles que son casi equiláteros. Más adelante, comentaremos el porqué de esta exclusión.

Las características elegidas están todas relacionadas más o menos directamente con el problema de clasificación. Para asegurarnos de que nuestro algoritmo es capaz de eliminar todas las características irrelevantes añadiremos también otras características con valores aleatorios; en consecuencia, completamente irrelevantes al problema.

Las pruebas se han realizado con diferentes conjuntos de ejemplos variando su tamaño y el número de características generadas aleatoriamente (irrelevantes). Hemos creado tres tipos de muestras con 1, 10 y 25 características irrelevantes,

para poder estudiar el efecto que puedan tener en el comportamiento del algoritmo. Hemos generado conjuntos de 50, 100, 250 y 500 ejemplos para cada uno de estos tipos de muestra.

Usando estos mismos conjuntos de prueba, el parámetro umbral (U) de CFOCUS ha sido variado en los valores: 0.025, 0.05, 0.1 y 0.2. Los resultados se muestran en las tablas: 3.1, 3.2, 3.3 y 3.4, respectivamente. Las características generadas aleatoriamente, y por tanto irrelevantes, son referidas como “IrrN” donde N es simplemente un número asignado a cada una de ellas para diferenciarlas.

Nº ej.	Nº de características irrelevantes		
	1	10	25
50	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Lmenor, Perímetro
100	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$
250	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$
500	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$

Tabla 3.1: Características seleccionadas en cada muestra de las figuras geométricas por CFOCUS con U=0.025

Nº ej.	Nº de características irrelevantes		
	1	10	25
50	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$
100	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$
250	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$
500	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$

Tabla 3.2: Características seleccionadas en cada muestra de las figuras geométricas por CFOCUS con U=0.05

CFOCUS ha encontrado un conjunto que permite realizar la clasificación correctamente en 41 casos de los 48 estudiados. En otros 5 ha informado que con el umbral elegido no es posible resolverlo. Únicamente en los 2 casos restantes, a partir de las muestras más pequeñas (50 ejemplos), ha devuelto conjuntos de características que no son suficientes para resolver el problema en general.

Esto muestra que el algoritmo exhibe buen comportamiento usando un número muy reducido de ejemplos. no se han utilizado conjuntos con mayor número de ejemplos porque 500 han sido suficientes para que el algoritmo CFOCUS dé buenos resultados.

Nº ej.	Nº de características irrelevantes		
	1	10	25
50	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr0, Irr8	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr19
100	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr0, Área	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr0, Irr2, Irr8	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr4
250	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr0, Irr3, Irr4	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr0, Irr1, Irr21
500	(No resuelto)	(No resuelto)	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr2, Irr10, Irr12

Tabla 3.3: Características seleccionadas en cada muestra de las figuras geométricas por CFOCUS con $U=0.1$

Nº ej.	Nº de características irrelevantes		
	1	10	25
50	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Área, Irr0	Nº Lados, L_{menor} , Irr0, Irr1, Irr4	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr0, Irr2, Irr23
100	(No resuelto)	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, L_{menor} , Irr0, Irr1, Irr6	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, L_{menor} , Irr2, Irr6, Irr12
250	(No resuelto)	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, L_{menor} , Irr0, Irr1, Irr2, Irr3, Irr4	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, Irr0, Irr4, Irr6, Irr7, Irr18
500	(No resuelto)	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$	Nº Lados, $\frac{\text{Lado menor}}{\text{Lado mayor}}$, L_{menor} , Área, Perímetro, Irr2, Irr4, Irr12, Irr17

Tabla 3.4: Características seleccionadas en cada muestra de las figuras geométricas por CFOCUS con $U=0.2$

3.1.3.2. Forest Cover Type

El siguiente problema que hemos elegido como banco de prueba ha sido obtenido de la biblioteca de problemas de Aprendizaje Automático UCI [HB99]. Su nombre es “Forest Cover Type”. Lo hemos elegido porque utiliza un gran número de características booleanas, discretas y continuas. Al disponerse de un gran número de características es un problema en el que parece oportuno aplicar selección de características, y como las características son de diversos tipos es ideal para mostrar la versatilidad de nuestro algoritmo para trabajar con conjuntos de datos variados.

El problema consiste en clasificar el tipo de bosque, entre 7 posibles, de un área de 30×30 metros a partir de un conjunto de 54 características. Se puede encontrar una descripción más detallada del problema en el trabajo de Blackard y Dean [BD99].

Hemos escogido 2000 ejemplos como muestra del conjunto de datos y hemos aplicado sobre la misma CFOCUS con diferentes niveles de umbral. Como estrategia para elegir el umbral, empezamos con un valor de 0.2 y lo fuimos dividiendo por 2 en cada prueba. El primero que dio una selección de características fue 0.0125. En los anteriores CFOCUS indicaba que el conjunto de conflictos era irresoluble con ese nivel de umbral.

Para comprobar si la selección de características obtenida es buena, dado que se trata de un problema real en el que no conocemos *a priori* qué características son relevantes, hemos entrenado redes neuronales para que actúen como clasificador usando las características seleccionadas. Usaremos el porcentaje de clasificación obtenido para valorar la selección de características.

Hemos comparado la calidad de las clasificaciones obtenidas con CFOCUS + redes neuronales, redes neuronales sin selección de características, y Relief-E [Kon94] + redes neuronales.

Se ha elegido Relief como método de selección de características con el que comparar por varias razones. En primer lugar, porque es un algoritmo muy conocido que ha obtenido buenos resultados en estudios comparativos [DL97]. También hemos tenido en cuenta que es capaz de trabajar con características continuas y discretas como el algoritmo propuesto. Además, una versión de Relief similar a la que nosotros hemos escogido, fue usada como representativa de los algoritmos que usan la estrategia de filtro para selección de características en la presentación de la estrategia envolvente [KJ97].

Las características seleccionadas por CFOCUS han sido: Elevation, Aspect, Slope, Horizontal-Distance-To-Hydrology, Vertical-Distance-To-Hydrology, y Horizontal-Distance-To-Roadways.

Como Relief-E sólo asigna una valoración a cada característica, y no indica el número de características que se deben usar, para compararlos en igualdad de condiciones hemos usado, de las más valoradas, el mismo número de características que CFOCUS. Las características seleccionadas por Relief-E han sido: Aspect, Horizontal-Distance-To-Roadways, Horizontal-Distance-To-Fire-Points, Horizontal-Distance-To-Hydrology, Slope, y Hillshade-3pm.

Para estimar la calidad de las características elegidas hemos tomado 4 nuevas muestras de 5000 ejemplos, cada una de ellas la hemos dividido en dos partes, una para entrenamiento con 4000 ejemplos y otra de prueba con 1000 ejemplos. Como clasificador hemos empleado un perceptrón multicapa con una capa oculta, inicializado con pesos aleatorios. El algoritmo de aprendizaje ha sido el método estándar de propagación hacia atrás (backpropagation) con una tasa de aprendizaje de 0.05.

En las tablas 3.5, 3.6 y 3.7 se muestran los resultados obtenidos con cada técnica. El dato indicado es el porcentaje de clasificación, para los 4 conjuntos de prueba y las diferentes arquitecturas del perceptrón multicapa consideradas.

<i>Topología</i>	<i>Conjunto de prueba</i>				<i>Max</i>	<i>Media</i>
	1	2	3	4		
54-4-7	50,4	55,6	57,8	68,4	68,4	58,0
54-5-7	56,8	52,9	54,4	72,0	72,0	59,0
54-6-7	43,4	52,1	56,8	70,8	70,8	55,8
54-7-7	48,9	51,5	51,5	70,4	70,4	55,6
54-8-7	41,7	54,8	57,5	69,5	69,5	55,9
54-9-7	44,5	52,7	57,8	68,1	68,1	55,8
54-10-7	53,0	50,3	52,0	70,7	70,7	56,5
Max	56,8	55,6	57,8	72,0	72,0	60,55
Media	48,4	52,8	55,4	70,0	70,0	56,7

Tabla 3.5: Resultados de clasificación en Forest sin selección de características (% de acierto)

Podemos observar que los resultados son mejores con CFOCUS que sin usar selección de características a pesar de que el clasificador está usando mucha menos información. Se corrobora, en consecuencia, que la selección de características puede ser útil para guiar a los algoritmos de aprendizaje y mejorarlos.

<i>Topología</i>	<i>Conjunto de prueba</i>				<i>Max</i>	<i>Media</i>
	1	2	3	4		
6-4-7	55,6	57,1	64,0	58,7	64,0	58,9
6-5-7	61,8	50,0	64,5	58,2	64,5	58,6
6-6-7	59,5	52,3	69,2	62,2	69,2	60,8
6-7-7	58,7	53,0	60,3	66,2	66,2	59,6
6-8-7	59,8	52,4	71,7	63,9	71,7	62,0
6-9-7	63,8	54,7	67,7	66,3	67,7	63,1
6-10-7	59,5	52,7	60,8	64,5	64,5	59,4
Max	63,8	57,1	71,7	66,3	71,7	64,7
Media	59,8	53,2	65,5	62,8	65,5	60,3

Tabla 3.6: Resultados de clasificación en Forest usando CFOCUS (% de acierto)

<i>Topología</i>	<i>Conjunto de prueba</i>				<i>Max</i>	<i>Media</i>
	1	2	3	4		
6-4-7	21,5	25,5	40,8	41,1	41,1	32,2
6-5-7	25,3	26,6	45,0	41,3	45,0	34,5
6-6-7	25,3	28,9	41,4	45,4	45,4	35,3
6-7-7	23,8	25,5	42,5	48,6	48,6	35,1
6-8-7	32,0	23,8	42,0	42,8	42,8	35,2
6-9-7	28,5	22,8	43,4	46,0	46,0	35,2
6-10-7	25,3	22,3	40,3	47,7	47,7	33,9
Max	32,0	28,9	45,0	48,6	48,6	38,6
Media	26,0	25,1	42,2	44,7	44,7	34,5

Tabla 3.7: Resultados de clasificación en Forest usando RELIEF-E (% de acierto)

Comparando con Relief-E, los resultados obtenidos usando CFOCUS son mucho mejores. Esto puede deberse a que CFOCUS tiene en cuenta las interrelaciones entre características, mientras que Relief-E no.

3.1.3.3. Información contable

Finalmente, contando con la colaboración de expertos en el problema, mostramos la aplicación de CFOCUS a un problema real de predicción de información contable [TMNRDCF⁺02]. Los resultados son comparados con los obtenidos con la técnica empleada habitualmente por los expertos en el problema, la regresión logística.

El problema consiste en predecir la permanencia de empresas en el Índice General de la Bolsa de Madrid. Para ello se ha tomado una muestra de una serie de características de 64 empresas que han pertenecido en algún momento a dicho índice entre 1993 y 1998, clasificándolas según hayan permanecido en él durante todo el intervalo o no. Las características consideradas se muestran en la figura 3.5.

- | | | |
|---------------------------------|--|---------------------------|
| ■ NIVEL DE ENDEUDAMIENTO | ■ ROTACIÓN DE EXISTENCIAS | ■ I+D |
| ■ COMPOSICIÓN DEL ENDEUDAMIENTO | ■ ROTACIÓN DE DEBERCHOS | ■ PUBLICIDAD |
| ■ FONDO DE MANIOBRA | ■ RENTABILIDAD ECONÓMICA DE LA EXPLOTACIÓN | ■ DEMORA |
| ■ DISPONIBILIDAD | ■ MARGEN | ■ DEMANDAS |
| ■ TEST-ÁCIDO | ■ RENTABILIDAD ECONÓMICA TOTAL | ■ INCIDENCIAS |
| ■ SOLVENCIA | ■ RENTABILIDAD FINANCIERA | ■ AUDITORÍA |
| ■ GARANTÍA | ■ COSTE DEL ENDEUDAMIENTO | ■ SOLVENCIA DE CAJA (C/P) |
| ■ TAMAÑO | ■ GASTOS FINANCIEROS | ■ SOLVENCIA DE CAJA (L/P) |
| ■ RECURSOS GENERADOS | | ■ CALIDAD DEL BENEFICIO |
| ■ ROTACIÓN DEL ACTIVO | | ■ RETORNO LÍQUIDO |

Figura 3.5: Lista de características del problema de predicción contable

En este problema hemos usado un modelo híbrido que combina la aplicación de árboles de decisión, selección de características (CFOCUS) y redes neuronales.

En primer lugar se aplicó CFOCUS a la muestra total para ver qué características podrían resultar más interesantes. Entre estas características, con la ayuda de los expertos, se decidió elegir la característica tamaño para aplicar un árbol de decisión dividiendo el problema en dos particiones. Así, el sistema resultante puede usar características y modelos diferentes, según las empresas sean medianas o grandes, dos ámbitos donde el problema se considera diferente. En cada una de las dos particiones del problema que hace el árbol de decisión aplicamos CFOCUS seguido de un perceptrón multicapa como clasificador, con los mismos parámetros que en el problema anterior. En la figura 3.6 se muestra un esquema representando el método global aplicado.

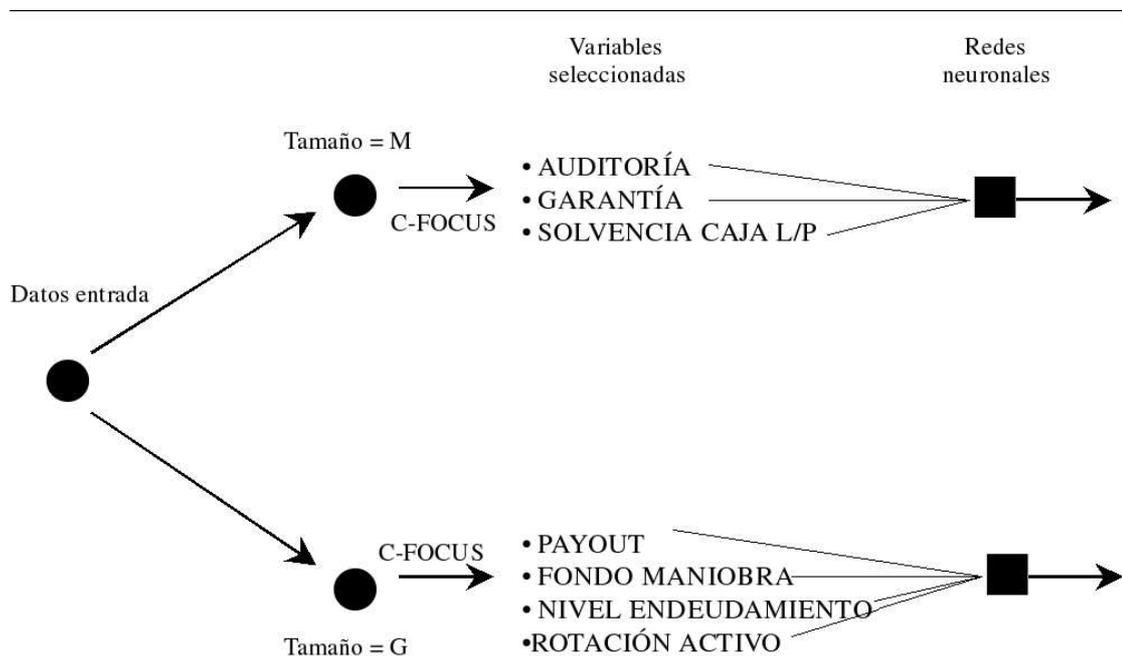


Figura 3.6: Aplicación final de CFOCUS en el proceso de aprendizaje del problema de predicción contable

<i>Característica</i>	<i>Todas</i>	<i>Medianas</i>	<i>Grandes</i>
TAMAÑO	X		
PAYOUT	X		X
GARANTÍA	X	X	
SOLVENCIA DE CAJA (L/P)	X	X	
ROTACIÓN DE ACTIVO	X		X
FONDO DE MANIOBRA	X		X
COSTE DE ENDEUDAMIENTO	X		
NIVEL DE ENDEUDAMIENTO			X
AUDITORÍA		X	

Tabla 3.8: Características seleccionadas para predicción contable por CFOCUS

En la tabla 3.8 se muestran las características que ha seleccionado CFOCUS con la muestra completa y en cada una de las particiones. La valoración de los expertos en el problema sobre las características elegidas es bastante positiva, ya que, según ellos, es muy razonable que con ellas se pueda realizar la clasificación de cada partición.

<i>VALORES OBSERVADOS</i>	<i>VALORES PRONOSTICADOS</i>		<i>PORCENTAJE DE ACIERTOS</i>
	<i>NO PERTENENCIA (0)</i>	<i>PERTENENCIA (1)</i>	
<i>TAMAÑO=M</i>			
NO PERTENENCIA (0)	17	0	100 %
PERTENENCIA (1)	0	5	100 %
<i>TAMAÑO=G</i>			
NO PERTENENCIA (0)	12	2	85,7 %
PERTENENCIA (1)	0	31	100 %
TOTAL			97,01 %

Tabla 3.9: Resultados de clasificación obtenidos con CFOCUS y redes neuronales para el problema de predicción contable

El porcentaje de clasificación para cada una de las particiones del árbol de decisión y el método completo se muestra en el tabla 3.9.

<i>VARIABLE</i>	β	<i>SE</i>	<i>WALD</i>	<i>GL</i>	<i>SIG.</i>
COMPOSICIÓN DEL ENDEUDAMIENTO	0,875	0,374	4,402	1	0,036
PAYOUT	6,872	3,300	4,798	1	0,028
GARANTÍA	1,284	0,512	6,294	1	0,012
TAMAÑO (1)	3,017	1,057	8,149	1	0,004
CONSTANTE	-6,576	1,938	11,510	1	0,001

Tabla 3.10: Modelo obtenido con regresión logística para el problema de predicción contable

<i>VALORES OBSERVADOS</i>	<i>VALORES PRONOSTICADOS</i>		<i>PORCENTAJE DE ACIERTOS</i>
	<i>NO PERTENENCIA (0)</i>	<i>PERTENENCIA (1)</i>	
NO PERTENENCIA (0)	26	3	89,7 %
PERTENENCIA (1)	6	26	81,3 %
TOTAL			85,2 %

Tabla 3.11: Resultados de clasificación obtenidos con regresión logística para el problema de predicción contable

Con el objeto de comparar nuestros resultados con los que se obtienen con una técnica estándar, los expertos en el problema aplicaron regresión logística con esta muestra. El modelo obtenido y los porcentajes de clasificación conseguidos se muestran en las tablas 3.10 y 3.11, respectivamente.

Pese a la coincidencia, en líneas generales, de los resultados obtenidos, nuestro método proporciona niveles de acierto superiores a los correspondientes al análisis mediante regresión logística, lo que hace más aconsejable su uso para resolver el problema planteado.

El tamaño ha resultado ser, en ambos casos, la característica con mayor poder discriminatorio a la hora de clasificar las empresas en el grupo de unidades económicas que han permanecido en el Índice General de la Bolsa de Madrid.

Es de destacar también que, al segmentar la muestra total en función de esta característica, los indicadores más valorados para las empresas grandes y para las de más reducida dimensión difieren. Mostrando de esta forma la idoneidad de la metodología usada.

Podemos concluir que la técnica híbrida basada con CFOCUS no sólo ha mejorado los resultados que se vienen obteniendo con una técnica estándar, sino que además la selección de características realizada por CFOCUS recibe el visto bueno de los expertos.

3.1.3.4. Conclusiones

Hemos desarrollado el algoritmo CFOCUS como extensión del algoritmo FOCUS para dominios no booleanos. Conseguimos de esta forma que se pueda aplicar a un conjunto más amplio de problemas.

Hay muchos problemas de clasificación con valores nulos en algunas características de sus ejemplos. En la evaluación empírica desarrollada nos hemos topado con datos de este tipo y esto nos ha llevado a incorporar el tratamiento de valores nulos en nuestro algoritmo. Se ha hecho aún más amplio el dominio de aplicación de esta técnica de selección de características.

La utilización de este algoritmo está indicada en problemas de clasificación en los que se disponga de un conjunto de entrenamiento sin ruido, y el objetivo prioritario sea reducir al máximo el número de características. Hemos creado un problema de este tipo y comprobado el buen funcionamiento del algoritmo en él.

Sobre un problema banco de prueba hemos comparado los resultados obtenidos usando CFOCUS, Relief y el clasificador sin selección de características. La aplicación de CFOCUS con el clasificador ha obtenido los mejores resultados.

Finalmente, se ha mostrado la aplicación de CFOCUS en dos problemas reales, obteniendo, no sólo el visto bueno de los expertos sobre las características seleccionadas, sino también muy buenos resultados de clasificación.

Podemos concluir que nuestro algoritmo, CFOCUS, permite resolver satisfactoriamente el problema de selección de características en una gran variedad de problemas. Mantiene los resultados del algoritmo en que nos basamos, FOCUS, para los problemas en dominio booleano, y mejora los resultados obtenidos con el algoritmo Relief, de conocida reputación.

Una limitación de nuestra extensión continua es que puede encontrar dificultades al trabajar en conceptos con cambios bruscos respecto a una característica continua. Por ejemplo, si en nuestro problema hubiese rectángulos con lados muy parecidos. Esto podría impedir que CFOCUS detectase la diferencia de esta figura con un cuadrado. Puede ser interesante estudiar formas para eliminar esta limitación.

Para la elección umbral hay varias opciones posibles: basarse en la estabilidad de las soluciones, decrementar hasta conseguir número deseado de variables, o incluso aplicar metodología envolvente.

3.2. FFOCUS

Hasta ahora, partiendo de la base de un método de selección de características booleanas, hemos creado un método que permite trabajar con los tipos más comunes de características: continuas, discretas, y booleanas. En esta sección desarrollamos un método que permite trabajar con otro tipo de características, las lingüísticas. Estas serán características con valores continuos en los ejemplos, pero a las que se incorporara una interpretación lingüística de sus valores.

3.2.1. Motivación

Como mencionamos al presentar CFOCUS, puede ser interesante aprovechar, cuando esté disponible, la información que el usuario nos pueda aportar sobre las características a considerar. De esta forma podemos saber qué valores considera un experto en el problema como distintos. Nuestra propuesta se basará en pedir al experto una descripción lingüística de las características con este fin.

Por otra parte, los métodos de inferencia de reglas lingüísticas comúnmente son un ejemplo de los métodos que sufren de la “maldición de la dimensionalidad creciente”. Estos efectos, como indicábamos en la introducción, se dan en los métodos que, dada su eficiencia de orden exponencial con respecto al número de características, no pueden ser aplicados a partir de un número no muy elevado de las mismas. Son, por tanto, los métodos que más se benefician de las técnicas de selección de características. Y son también los que principalmente se beneficiarán de nuestra propuesta, ya que estos métodos podrían compartir la información lingüística que usará sobre las características.

La principal ventaja que presentan las estrategias de selección de características de tipo envolvente o inmerso es que trabajan más o menos en sincronía con el algoritmo de aprendizaje. Pero esto conlleva una desventaja y es que estos métodos son más dependientes del algoritmo de aprendizaje que los de filtro, como comentábamos en la introducción a la selección de características. La propuesta que presentamos aquí, FFOCUS, cuando se usa con algoritmos de aprendizaje difusos, tiene la ventaja de compartir información sobre las características con el algoritmo de aprendizaje, de lo que se espera un mejor rendimiento. Y, sin embargo, no tiene ninguna ligazón directa con el algoritmo de aprendizaje; así que se puede usar de forma completamente independiente como se muestra en el estudio empírico. En la figura 3.7 se ilustra la forma de aplicación de FFOCUS que acabamos de comentar.

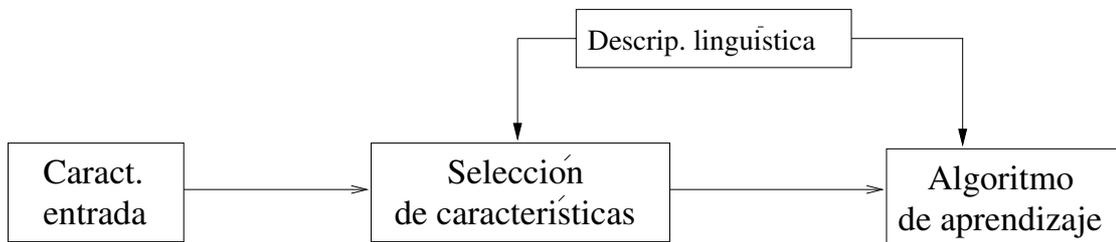


Figura 3.7: Metodología de filtro compartiendo información lingüística

3.2.2. Extensión lingüística (FFOCUS)

Después de exponer los motivos que nos llevan a desarrollar el algoritmo FFOCUS, que presentamos en esta sección, pasamos a detallar su funcionamiento empezando por la forma en que el usuario experto podrá introducir información sobre algunas características.

En la descripción de CFOCUS indicábamos cómo podría ser interesante que, junto con los datos del problema, el usuario indicase una distancia mínima en cada característica para considerar que dos valores son distintos. Esto representaría

una distinción homogénea en todo el dominio de cada característica y, dado que vamos a solicitar información del usuario, hemos creído conveniente aportar una mayor flexibilidad que permita definir varios grupos de valores al usuario según crea conveniente. Para esto emplearemos una descripción lingüística.

Cada característica continua podrá llevar asociada una variable lingüística. El concepto de variable lingüística fue definido por Zadeh en [Zad75d].

Veremos con un ejemplo cómo esto puede ser interesante en los problemas de aprendizaje. Supongamos que queremos determinar si el agua que hay en un cazo está hirviendo o no. Obviamente sería bueno que hubiésemos incluido la temperatura del agua entre las posibles características a considerar. Y si le pidiésemos a un experto que elija las etiquetas que le parezcan más convenientes para resolver el problema seguro que las concentraría alrededor de 100°C, que será el área de mayor interés. Habrá otras características que puedan influir en este problema, como la concentración de sal o la presión, pero es un poco difícil que el agua baje su punto de ebullición a 20°C. La figura 3.8 muestra: a) una elección sin conocer el problema, y b) la opción que elegiría un experto.

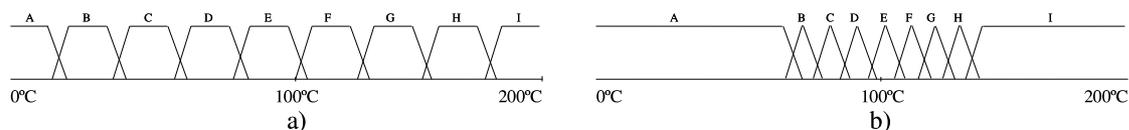


Figura 3.8: Elección de etiquetas: a) a ciegas vs. b) realizada por un experto.

Para aplicar la estrategia del algoritmo FOCUS usaremos, como hasta ahora, la definición de variable relevante adaptándola a la nueva situación, las características continuas con información lingüística. Definiremos, al igual que hicimos en CFOCUS, los conceptos “significativamente igual” y “significativamente distinto”, esta vez para las características lingüísticas.

En primer lugar, definiremos el grado en que dos valores continuos x e y son distintos. Con el objeto de que el grado pertenezca al intervalo $[0, 1]$, y por homogeneidad con el resto de las variables, se normalizarán los valores y conjuntos difusos al intervalo $[0, 1]$. La definición de este grado es la siguiente:

$$d(x, y) = \sum_{L, L' \in \text{Etiquetas difusas}} L(x)L'(y)D(L, L') \quad (3.3)$$

donde $D(L, L')$ es una medida de la separación entre dos conjuntos difusos.

Como medida de separación entre dos conjuntos difusos $D(L, L')$ usaremos la definida por Castro en [Sán00, CSCZ04]. Según ella, se tomará el grado en que dos conjuntos difusos son diferentes como el área de la intersección de los conjuntos difusos que representan los valores mayores que el conjunto de menor valor y el de los menores que el conjunto mayor. Esto se puede ver más claramente en la figura 3.9.

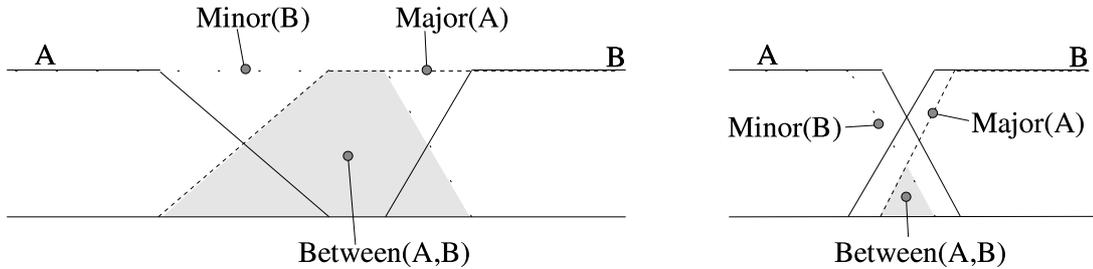


Figura 3.9: Medida de separación de Castro [Sán00, CSCZ04]

Diremos que dos valores x e y son “significativamente iguales” si el grado que acabamos de definir $d(x, y)$ es mayor que un cierto umbral U , el mismo que consideramos para las variables continuas sin información lingüística.

A partir de aquí tendremos definida la estrategia “MIN-FEATURES bias” al igual que en el caso de las variables continuas, y el algoritmo a emplear será el mismo que en la extensión continua. En consecuencia, queda también contemplado el tratamiento de valores nulos y esta extensión sigue siendo compatible con las anteriores, en el sentido de que para problemas con características de los tipos contemplados por FOCUS, o CFOCUS, ofrecerá los mismos resultados.

3.2.3. Estudio empírico

Para evaluar el funcionamiento hemos usado dos conocidos problemas de la misma biblioteca que utilizamos anteriormente, UCI[HB99]. Los problemas son ‘Pima indians diabetes’ y ‘Iris plant’. Las características de ambos problemas son continuas y no disponemos de información lingüística sobre ellas. Por ello, hemos definido cinco grupos de valores (etiquetas lingüísticas) de forma homogénea sobre todo el dominio de definición de cada característica. La forma elegida para las etiquetas, que serán todas iguales, ha sido trapezoidal y hemos hecho que se crucen en su valor de pertenencia 0.5. La suma de los grados de pertenencia de cualquier punto del dominio es 1. Las conjuntos difusos de las etiquetas en los límites se extienden hasta menos infinito y más infinito apropiadamente.

3.2.3.1. Pima Indians Diabetes Database

Hemos usado validación cruzada con 5 dobleces veces para generar diez pares de entrenamiento y prueba. Los resultados se muestran en la tabla 3.12, donde aparece el porcentaje de clasificación obtenido usando como clasificador un perceptrón multicapa con una única capa oculta. Al clasificador se le proporcionan únicamente los datos relativos a las características seleccionadas por FFOCUS, indicadas en la segunda columna junto con estos resultados. Con el objeto de compararlos se muestran los resultados obtenidos por un clasificador similar que usa todas las características en la tercera columna.

FFOCUS ha elegido la característica “Plasma glucose” en todas las muestras y una combinación de dos de las características “Body mass”, “Age” y “Times pregnant”. Usando sólo estas tres características el porcentaje de clasificación no ha disminuido significativamente.

<i>Test</i>	<i>FFOCUS</i>		<i>Todas</i>
0	Times pregnant, Plasma glucose, Age	68,18	77,92
1	Plasma glucose, Times pregnant, Body mass	79,87	79,87
2	Plasma glucose, Body mass, Age	72,73	78,57
3	Plasma glucose, Times pregnant, Age	73,20	81,05
4	Plasma glucose, Body mass, Age	82,35	75,82
5	Plasma glucose, DiastolicBP, Age	71,43	75,32
6	Plasma glucose, Body mass, Age	79,22	75,97
7	Plasma glucose, Body mass, Age	79,87	77,92
8	Plasma glucose, Times pregnant, Body mass	78,43	82,35
9	Plasma glucose, Body mass, Age	74,51	78,43
Media		75,97	78,32

Tabla 3.12: Resultados de clasificación usando FFOCUS en Pima (% de acierto)

3.2.3.2. Iris Plant Database

Este conjunto de datos se ha usado numerosas veces como banco de pruebas. Hemos tomado sus 4 características, y hemos considerado además unas características adicionales que son todos los productos posibles de dos de las 4 características originales. Esto muestra cómo puede ser utilizado nuestro algoritmo para realizar extracción de características derivadas, lo que puede redundar en una mejora en los resultados del algoritmo de aprendizaje.

En este experimento hemos usado la misma metodología que en el conjunto de datos anterior, diez pares de entrenamiento y prueba generados por doble validación cruzada con cinco dobles, y como clasificador un perceptrón multicapa. Los resultados se muestran en la tabla 3.13, con el mismo formato. Los resultados de la clasificación usando sólo las características seleccionadas frente a la clasificación usándolas todas.

FFOCUS ha elegido, en todos los casos, la longitud de “petal” o “sepal” y una característica producto derivada. Como se puede comprobar los resultados obtenidos usando sólo las dos características seleccionadas son algo mejores.

<i>Test</i>	<i>FFOCUS</i>		<i>Todas</i>
0	petalL, petalL × petalW	96,67	86,67
1	sepalL, petalL × petalW	100,00	96,67
2	sepalL, petalL × petalW	93,33	90,00
3	sepalL, petalL × petalW	96,67	96,67
4	sepalL, petalL × petalW	96,67	100,00
5	petalL, sepalW × petalW	96,67	90,00
6	petalL, sepalW × petalW	96,67	100,00
7	petalL, petalL × petalW	96,67	93,33
8	sepalL, petalL × petalW	96,67	90,00
9	petalL, sepalW × petalW	90,00	96,67
Media		96,00	94,00

Tabla 3.13: Resultados de clasificación usando FFOCUS en Iris (% de acierto)

3.2.3.3. Conclusiones

Hemos desarrollado FFOCUS como extensión de nuestra propuesta anterior CFOCUS para permitir la incorporación de conocimiento experto sobre algunas características.

Los problemas en que está indicada la aplicación de este algoritmo son aquéllos en los que, a pesar de que no se conozca la relación de las características con la solución del problema, sí dispongamos de una idea de la forma en que los valores de una característica pueden ser útiles.

En la evaluación empírica se ha demostrado cómo esta idea de agrupar los valores en conjuntos difusos puede ayudar a la clasificación. A pesar de no haber tenido disponible la valoración de un experto se han conseguido buenos resultados.

3.3. Medida de los pares de ejemplos inconsistentes

Las propuestas anteriores de extensiones de FOCUS son algoritmos completos de selección de características. Éstas suponen el uso de una estrategia de búsqueda estrechamente ligada a una estrategia de valoración de características. A la vista de que las técnicas basadas en la consistencia de los conjuntos de datos dan buenos resultados, siguiendo la estructura que hemos descrito en la sección 2.4, proponemos aquí una medida basada en consistencia. Así, esta medida podrá ser utilizada con todas las estrategias de búsqueda genéricas, incluidas las descritas en la sección 2.5.

Un conjunto de datos consistente se hace inconsistente en el momento que contiene dos ejemplos de diferente clase pero con los mismos valores en todas las características. Estos dos ejemplos forman un par de ejemplos inconsistentes. De esta forma, un conjunto de datos puede decirse que es más inconsistente, o que muestra un grado menor de consistencia, a medida que contenga más pares de ejemplos inconsistentes. La medida que proponemos aquí usa la cuenta de estos pares como medida de la inconsistencia. En adelante también denominamos esta medida como “Inconsistent Example Pairs” (IEP).

3.3.1. Antecedentes

Los pares de ejemplos inconsistentes han sido denominados previamente “unsolved conflicts” (conflictos sin resolver). En la terminología usada por FOCUS, un conflicto es un par de ejemplos de diferente clase, o con diferente valor del concepto a aprender. Cuando el par de ejemplos que forma un conflicto tiene valores diferentes en alguna característica, el conflicto se considera resuelto y, en caso contrario, sin resolver.

La cuenta de “unsolved conflicts” ha sido usada como guía de búsqueda en “Simple Greedy” [AD94], y “Set Cover” [Das97], pero, para nuestro conocimiento, no ha sido definida nunca como medida independiente, ni comparada con otras medidas. Hemos considerado por eso importante definir una medida basada en esta cuenta de ejemplos inconsistentes para rellenar un hueco natural en las medidas de consistencia.

3.3.2. Definición de la medida

La cuenta de pares de ejemplos inconsistentes está en el rango entre 0, cuando el conjunto de datos es consistente, y el número de pares de ejemplos con diferente clase, cuando no hay ninguna característica seleccionada y, por tanto, ningún par puede ser distinguido. Esto hace que el rango teórico de la medida sea el intervalo $[0, +\infty]$. En vez de usar esta cuenta directamente como medida de inconsistencia, parece razonable hacerla proporcional al conjunto de datos, con el objetivo de hacer la medida comparable entre diferentes conjuntos de datos y acotada en un intervalo limitado.

<i>Measure</i>	<i>General</i>	<i>Given a Data Set(DS)</i> [<i>All feat.</i> , \emptyset]	<i>Simplest DS</i> \emptyset	<i>Hardest DS</i> \emptyset
Count of IEP	$[0, +\infty]$	[<i>IIP</i> , <i>diffCl</i>]	0	<i>Pairs</i>
$\frac{\text{Count of IEP}}{\text{No. pairs of diff. Class}}$	$[0, 1]$	$[\frac{IIP}{diffCl}, 1]$	$\frac{0}{0}$ (Indet.)	1
$\frac{\text{Count of IEP}}{\text{No. pairs}}$	$[0, 1]$	$[\frac{IIP}{Pairs}, \frac{diffCl}{Pairs}]$	0	1
Liu's measure	$[0, 1[$	$[\frac{IIE}{N}, 1 - Majority]$	0	$\frac{N-1}{N} \rightarrow 1$
Rough Sets	$[0, 1]$	$[1 - \gamma, 1$ (0 if $ Cl = 1$)]	0	1
<i>DS</i> = Data Set. <i>IIE</i> = No. Insolvable Inconsistent Ex. <i>N</i> = $ DS $ (No. examples). <i>IIP</i> = No. Insolvable Inconsistent example Pairs. <i>Cl</i> = Class feature. <i>Pairs</i> = Total no. pairs in data set ($\frac{N(N-1)}{2}$). γ = Rough Sets Consistency. <i>diffCl</i> = No. pairs of different class.				

Tabla 3.14: Rangos, cotas y valores interesantes de las medidas de inconsistencia

La tabla 3.14 muestra algunos valores interesantes de las medidas. Las tres primeras líneas muestran las opciones que hemos considerado:

1. La cuenta del número de pares de ejemplos inconsistentes.
2. Ratio de pares de ejemplos inconsistentes frente el número de pares de ejemplos de diferente clase.
3. Ratio de pares de ejemplos inconsistentes frente el número total de pares de ejemplos.

Las dos últimas filas muestran los mismos valores para las medidas de Liu y *Rought Set Consistency* (ver sección 2.6.2.1), para permitir compararlas.

Como todas estas medidas son monótonas, el rango de la medida, para un conjunto de datos dado, es el intervalo delimitado por los valores de la medida para el conjunto de todas las características y el conjunto vacío. En la primera columna vemos el rango general de cada medida para todos los posibles conjuntos de datos. Las dos opciones alternativas a la cuenta simple están acotadas en el intervalo $[0, 1]$ (la ventaja que buscábamos al definir estos ratios).

En la segunda columna, vemos el rango para un conjunto de datos específico, en función de sus propiedades. Para un conjunto de datos que considerando todas sus características sea consistente, IIP, IIE y γ valdrán cero y, por tanto, todas las medidas coinciden en el mínimo, asignando cero a los conjuntos consistentes. Hay que tener en cuenta que el conjunto con todas las características no tiene por qué ser siempre consistente y, por ejemplo, no lo será en presencia de ruido.

El valor máximo, para la opción que divide por el número de pares de diferente clase, es 1, usando de esta forma el rango más amplio posible en $[0, 1]$ para todos los conjuntos de datos. Sin embargo, la otra opción y la medida de consistencia de Liu proveen un valor que puede ser usado como una medida de la inconsistencia *a priori* (antes de seleccionar ninguna característica), o como la dificultad inherente al conjunto de datos. En el caso de la medida de Liu, este valor es el concepto “Majority” (mayoría) de un conjunto de datos, esto es, la frecuencia de la clase más común. El valor “Majority” es comúnmente usado como el valor mínimo de precisión aceptable para un clasificador. Para ilustrarlo, se muestran en la tabla los valores para dos casos extremos de conjuntos de datos.

En la tercera columna, se muestra el valor de las medidas para el que consideramos el conjunto de datos más simple, áquel en el que todos los ejemplos pertenecen a la misma clase. Es consistente por sí mismo y no hay necesidad de seleccionar ninguna característica; por tanto, parece razonable que se le asigne un 0 como grado de inconsistencia. Todas las medidas satisfacen esto, excepto la que divide por los pares de diferente clase, cuyo valor es indeterminado, y que para permanecer de acuerdo con su rango de valores para cualquier conjunto de datos, debería ser definida a 1. Por otra parte, un conjunto de datos en el que cada ejemplo pertenece a una clase diferente será probablemente el conjunto en el que será más difícil encontrar hipótesis consistentes. Éste es al que hemos llamado el conjunto más difícil; se muestra en la cuarta columna de la tabla. Todas las medidas le asignan el valor máximo.

Para un conjunto de datos dado, la diferencia entre las tres opciones considera-

das para la medida de pares inconsistentes es simplemente una transformación lineal, que hace que las medidas se muevan en los diferentes intervalos identificados. Los efectos de las tres opciones guiando la búsqueda, o eligiendo un conjunto de características, son los mismos. Consideramos que la última opción es la más apropiada, dado que permite comparar con la medida grados de inconsistencia entre conjuntos de datos distintos. Así, definimos la medida como se muestra en la expresión (3.4):

$$\text{Inconsistencia} = \frac{\text{número de pares de ejemplos inconsistentes}}{\text{número de pares de ejemplos}} \quad (3.4)$$

3.3.3. Propiedades

Esta medida es monótona como las demás medidas de consistencia. Puede ser fácilmente deducido de lo siguiente. Un par de ejemplos que es consistente gracias a una característica en S_i seguirá siendo consistente con S_j dado que $S_i \subset S_j$ y, por tanto, esta característica también estará presente en S_j . Por ello, el número de pares de ejemplos inconsistentes sólo puede decrementarse cuando se añaden características y, por tanto, la medida siempre será igual o mayor, cumpliendo así la condición de monotonía.

Otra propiedad teórica interesante de la cuenta de ejemplos inconsistentes, apuntada por Dash [Das97], es que, junto con el algoritmo greedy simple de ascensión de colinas, constituye la aproximación de Jonhson al problema de cobertura de conjuntos [Pap94]. De esta forma, está garantizado que se encontrará un conjunto de características con no más de $O(M \log N)$ características, donde N es el número de características en el conjunto de datos, y M es el tamaño del menor conjunto consistente de características.

Para hacernos una idea intuitiva del funcionamiento de la medida de pares inconsistentes, podemos pensar que ésta representa la probabilidad de poder distinguir, con las características dadas, dos ejemplos escogidos aleatoriamente en el conjunto de datos dado.

3.3.4. Eficiencia e implementación

El hecho de que esta medida trabaje con la combinación de todos los pares de ejemplos no debe hacernos pensar que su computación deba ser necesariamente

costosa en eficiencia. De hecho, mediante el uso de tablas hash, hemos conseguido que la eficiencia de nuestra implementación en el caso promedio esté en $O(n)$, tanto en tiempo, como en espacio.

La descripción del algoritmo se expone a continuación en la figura 3.10. Para ilustrarlo con más claridad, se muestra un ejemplo de su aplicación en la figura 3.11. Primero, se realiza la proyección del conjunto de datos para usar sólo las características a evaluar. Después, todos los ejemplos son insertados en una tabla hash. Los elementos introducidos son los valores de la clase de cada ejemplo, usando como índice los valores de las características seleccionadas. De esta forma, todos los ejemplos están agrupados de acuerdo a sus valores para las características seleccionadas. Finalmente, se cuenta el número de ejemplos que no pertenecen a la clase mayoritaria de su grupo. Es fácil ver que la eficiencia en el caso promedio de este algoritmo está en $O(n)$.

ConsistencyMeasure(*Dataset*, *SelectedFeatures*)

Hash = \emptyset

For each *Example* in *Dataset*:

Insert *class*(*Example*) into *Hash* at $\pi_{SelectedFeatures}(Example)$

InconsistentExamplePairs = 0

For each *ClassList* in *Hash*:

InconsistentExamplePairs += number of all possible pairs of two different class values in *ClassList*

n = |*Dataset*|

Return $1 - \frac{InconsistentExamplePairs}{\frac{n(n-1)}{2}}$

Hash is a hash table in which every element included has a list (initially empty) of class values.

Figura 3.10: Algoritmo para calcular la medida de pares de ejemplos inconsistentes

3.4. Nueva medida de conjuntos de características basada en Relief

Muchos métodos se basan en la búsqueda basada en medidas sobre conjuntos de características como la presentada en el apartado anterior. Por otra parte, los evaluadores de características valoran éstas de forma individual y se quedan con las mejor valoradas. Relief es un evaluador de características conocido, bastante bueno

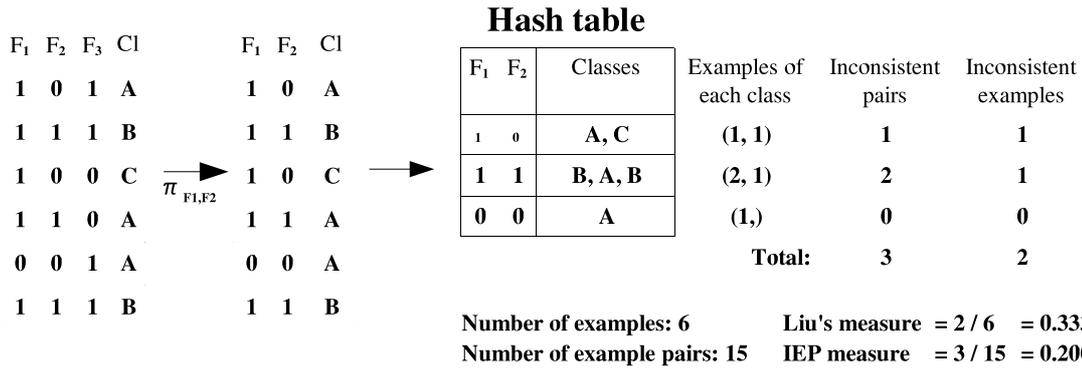


Figura 3.11: Ejemplo de cálculo rápido de las medidas de pares de ejemplos inconsistentes y de Liu

y profundamente estudiado. En muchas ocasiones sus estimaciones son más acertadas que medidas estadísticas habituales, como la correlación o covarianza, porque Relief tiene en cuenta la interrelaciones entre todos los atributos.

Aunque los evaluadores de características son normalmente más rápidos, presentan algunas desventajas frente a los métodos basados en medidas de conjuntos de características (ver sección 2.6.2). La principal es que ningún evaluador puede aportar nada para manejar la redundancia entre características, dado que su respuesta es la evaluación individual de cada una de ellas. La única forma de resolver este problema es evaluar conjuntos de características completos, y realizar una búsqueda entre ellos. En este apartado, se propone y evalúa una nueva medida de conjuntos de características basada en el funcionamiento y las principales ideas de Relief. La nueva medida puede ayudar de forma satisfactoria tanto guiando en proceso de búsqueda como eligiendo el conjunto más apropiado de características.

3.4.1. RFSM (Relief Feature Set Measure)

El algoritmo original Relief, de Kira y Rendell [KR92], proponía un método de selección basado en elegir aquellas características que recibían mejor valoración con la medida que definían en el mismo. La medida de características la hemos descrito de forma completa, incluyendo sus extensiones, en el apartado 2.6.1.5. Describiremos aquí la nueva medida partiendo de la descripción dada en ese apartado.

Dados los extensos trabajos que se han publicado sobre Relief [Kon94, RSK97, RS97, RSK99, RSK03], y los buenos resultados que se obtienen, pensamos que es interesante aplicar sus ideas principales al diseño de una medida para evaluar con-

juntos de características. Desde nuestro punto de vista, estas ideas principales son:

- Elevar el grado de relevancia de aquellas características que tienen valores diferentes en los pares de ejemplos de diferente clase.
- Penalización de características. En paralelo a la idea anterior, Relief reduce el grado de relevancia de aquellas características con valores diferentes en pares de ejemplos de la misma clase.
- Los pares de ejemplos son seleccionados entre ejemplos cercanos. Dado un ejemplo, Relief toma entre su vecindario otros ejemplos con la misma y diferente clase. Éste es, probablemente, el punto sobre el que gira todo el éxito de Relief. La estrategia de considerar ejemplos cercanos le hace trabajar de forma no miope [SK96], y le permite considerar interrelaciones entre características.
- Se usa muestreo aleatorio para escoger los ejemplos usados en la evaluación. De esta forma, el tiempo de ejecución se reduce de forma significativa mientras que el rendimiento no se degrada de forma significativa. No obstante, si el conjunto de datos es pequeño como para que nos lo podamos permitir, es recomendable usar todos los ejemplos del conjunto de datos.

Como cada ejemplo es tomado en un paso independiente, si hay más tiempo disponible, Relief puede mejorar sus valoraciones usando más ejemplos sobre la marcha. Esto hace de Relief un “any-time algorithm” (algoritmo de cualquier duración) como se menciona en [RSK03] (section 6.2).

Basándonos en estas ideas proponemos una medida para conjuntos de características, que pueda ser usada como función de evaluación en el proceso de búsqueda (figura 2.10). En adelante, nos referiremos a la medida como RFSM (Relief Feature Set Measure). El algoritmo para calcular la medida seguirá el esquema común de todas las ampliaciones de Relief, mostrado en la figura 2.16. La valoración dada como resultado será ahora un único valor W_S para el conjunto de características en evaluación S , en vez del vector de valores $W[f]$ con la valoración para cada característica.

La nueva medida usará la función diffS para valorar la diferencia que hay en las características del conjunto S para distinguir los pares de ejemplos. La función diffS se define en la expresión (3.5) como una agregación de los valores de diff (expresión 2.4) para las características seleccionadas.

$$\text{diffS}(S, E_1, E_2) = \text{operator}_{f \in S} \{ \text{diff}(f, E_1, E_2) \} \quad (3.5)$$

El *operador* debe ser un operador de agregación de los valores de *diff*. En características discretas, éstos son valores booleanos, transformados a $\{0, 1\}$, indicando si cada característica distingue los ejemplos. Un conjunto de características es capaz de distinguir dos ejemplos si alguna de sus características puede hacerlo. Por tanto consideramos que el operador más apropiado para características discretas es el *or* lógico. Cualquier t-conorma puede ser empleada para generalizar el comportamiento del *or* para las características continuas. Como parece razonable pensar que el grado en el cual un conjunto de características diferencia dos ejemplos es el grado de la característica que más los diferencia, hemos usado el operador *max*.

En la parte de la penalización, usamos la misma función de evaluación pero podemos elegir otro operador de agregación. En este caso, hemos tenido en cuenta que en la realización de varios experimentos informales usando el operador *max* se elegían muy pocas características. Además, intuitivamente, hemos pensado que una característica podría tener valores que sean irrelevantes en un par de ejemplos y marcarlos como distintos pero ser necesaria para distinguir otros. Eligiendo un operador que asemeje el *and* lógico penalizamos sólo los conjuntos en los que todas las características diferencien un par de ejemplos que sean de la misma clase. Por tanto, creemos adecuado elegir una t-norma y, en este caso, hemos elegido el operador *min*.

Además, la elección de los operadores *max* y *min* hacen que la medida cumpla la propiedad de monotonía (ver sección 2.6.2.1). Se puede ver fácilmente que si $S_i \subset S_j$ las características adicionales de S_j sólo podrían aumentar o mantener igual la relevancia, y disminuir o mantener igual la penalización.

En cada paso i , con el objeto de evaluar el conjunto de características completo para un par de ejemplos en cualquier problema de clasificación, usaremos la expresión (3.6) que aplica la misma ponderación a las clases que Relief-F.

$$W_i(S) = W_{i-1}(S) - \text{diffS}(S, E_1, H) + \sum_{C \neq \text{class}(E_1)} p(C) \times \text{diffS}(S, E_1, M(C)) \quad (3.6)$$

Con el diseño de esta medida esperamos que sea capaz de dar medidas más altas no sólo a los conjuntos de características que sean mejores, sino también a aquellos que sean potencialmente mejores. Éstos son aquellos conjuntos que, aun no siendo mejores que otros para clasificar, estén más cerca de los conjuntos que son mejores. De esta forma esperamos que guíe la búsqueda de forma más efectiva. Por ejemplo, en un conjunto de datos booleano, con valores aleatorios equiprobables de

las características, y siendo la función objetivo $(x_1 \oplus x_2) \oplus x_3$ (\oplus denota el ó lógico exclusivo), cualquiera de las características relevantes sola es estadísticamente independiente del objetivo. Cualquier algoritmo de aprendizaje no podría deducir nada de una sola característica, ni siquiera de un conjunto con dos de las características relevantes. Por tanto, ni la estrategia envolvente, ni ninguna otra medida basada sólo en las características a evaluar, podrían guiar una búsqueda comenzando en el conjunto vacío de características, y comenzar la búsqueda en el conjunto de todas las características no siempre es posible o recomendable, como comentábamos en la sección 2.5.

3.4.2. Estudio empírico

El objetivo de este estudio es determinar el rendimiento de la nueva medida Relief Feature Set Measure (RFSM) en el proceso de selección de características. Exploraremos los dos usos principales de las medidas identificados en la sección 2.4.

Por una parte, evaluamos el rendimiento de la medida guiando la búsqueda, usando un método voraz de búsqueda. La ascensión simple de colinas es un método que depende fuertemente de la medida para guiar la búsqueda, dado que sólo explora un camino sin posible vuelta atrás. Creemos que un método de búsqueda que es tan sensible a la medida es una buena forma de comprobar si la medida es buena para guiar la búsqueda. Además, la búsqueda por ascensión de colinas es muy eficiente.

El método de búsqueda, compuesto por las tres partes descritas en en la sección 2.4 será el siguiente. Como punto de comienzo se tomará el conjunto vacío de características. En cada paso, se elegirá añadir la característica que dé lugar al conjunto con mejor valor de la medida, y la búsqueda parará cuando no sea posible mejorar la medida añadiendo ninguna característica.

Por otra parte, también se estudia el uso de la medida para elegir el mejor conjunto de características. En los conjuntos de datos artificiales comprobamos si los conjuntos con todas las características relevantes obtienen el valor máximo de la medida y, en conjuntos de datos del mundo real, comparamos el rendimiento de la clasificación con las características seleccionadas.

El rendimiento del proceso completo de selección de características está incluido en el estudio. Los resultados se compararán con la estrategia envolvente, la medida de consistencia de Liu (ver apartado 2.6.2), y el rendimiento del algoritmo de aprendizaje sin selección de características.

3.4.2.1. Conjuntos de datos artificiales

Hemos usado dos conjuntos de datos artificiales ampliamente conocidos. Estos conjuntos presentan dos dificultades: características redundantes en Parity3+3, y una característica que parece relevante pero es innecesaria en CorrAL. Parity3+3 es la paridad de tres características booleanas ($x_1 \oplus x_2 \oplus x_3$), con las tres características relevantes duplicadas y otras seis irrelevantes aleatorias incluidas. CorrAL representa la función $(x_1 \wedge x_2) \vee (x_3 \wedge x_4)$, con una característica aleatoria adicional y otra correlacionada con la función concepto aleatoriamente en el 75% de los casos.

	<i>CorrAL</i>	<i>Parity3+3</i>
<i>Liu's measure</i>	5.0 \pm 0.00	4.7 \pm 0.21
<i>Wrapper (tree)</i>	5.2 \pm 0.13	6.5 \pm 0.65
<i>Wrapper (knn)</i>	5.0 \pm 0.15	5.9 \pm 0.85
<i>RFSM</i>	4.0 \pm 0.00	3.0 \pm 0.00

Tabla 3.15: Comparación del número de características seleccionadas usando tres medidas diferentes en conjuntos de datos artificiales (\pm Err.std.)

La tabla 3.15 muestra el número medio de características seleccionadas por cada método en una validación cruzada con 10 dobleces. Hemos verificado que en todos los casos los conjuntos de características seleccionados contienen todas las características necesarias, por lo que la diferencia entre los métodos radica en ver cual es capaz de elegir menos características innecesarias. Mejor de lo que nos esperábamos, la nueva medida RFSM ha sido capaz de guiar la búsqueda de forma perfecta en todos los casos, mientras que los demás métodos incluyen alguna característica innecesaria.

3.4.2.2. Conjuntos de datos del mundo real

Se han utilizado cinco conjuntos de datos de problemas reales extraídos de la biblioteca UCI KDD [HB99]. Tres de ellos son problemas con únicamente características discretas, y dos de ellos combinan características continuas y discretas. Para aplicar la medida de Liu en características continuas, hemos utilizado el método de *discretización* por intervalos de igual frecuencia.

Los algoritmos de aprendizaje utilizados para comprobar la calidad de las características seleccionadas son: k NN con $k = 21$, y árboles de clasificación con poda. La implementación y todos los detalles sobre estos algoritmos están disponibles

en el sistema de minería de datos Orange [DZ04]. Para todos los parámetros no especificados aquí, se han tomado los que da este sistema por defecto.

	<i>zoo</i>		<i>lung-cancer</i>		<i>house-vot.84</i>		<i>breast-canc.</i>		<i>wine</i>	
	<i>Acc.</i>	<i>NoF</i>	<i>Acc.</i>	<i>NoF</i>	<i>Acc.</i>	<i>NoF</i>	<i>Acc.</i>	<i>NoF</i>	<i>Acc.</i>	<i>NoF</i>
<i>tree</i>	96.0	16.0	43.3	56.0	96.09	16.0	67.8	9.0	93.79	13.0
<i>Liu+tree</i>	95.0	4.9	50.8	4.5	96.55	10.0	66.7	7.9	94.93	5.1
<i>Wrapper+tree</i>	95.0	4.9	48.3	8.9	95.60	8.2	72.1	3.1	93.90	4.0
<i>RFSM+tree</i>	94.0	6.0	50.8	3.5	95.87	11.5	66.7	8.1	94.35	11.0
<i>kNN</i>	94.1	16.0	44.2	56.0	93.32	16.0	72.0	9.0	97.19	13.0
<i>Liu+kNN</i>	83.1	4.9	45.8	4.5	94.48	10.0	71.7	7.9	94.97	5.1
<i>Wrapper+kNN</i>	89.0	9.9	41.7	16.8	94.50	5.4	67.9	5.2	94.30	6.0
<i>RFSM+kNN</i>	88.1	6.1	44.2	3.9	95.39	11.8	73.8	8.2	97.75	11.5

Tabla 3.16: Comparación de RFSM con otros métodos en clasificación de problemas reales (% de acierto)

La tabla 3.16 muestra los resultados de los métodos de selección combinados con los algoritmos de aprendizaje. Se indica la precisión media alcanzada en la validación cruzada de 10 dobles en porcentaje, y el número medio de características usadas de los 10 dobles.

3.4.2.3. Conclusiones

Podemos observar que la selección de características usando RFSM es útil, pues el porcentaje de clasificación es mantenido o mejorado, mientras que el número de características se reduce. Comparativamente, los resultados son incluso mejores que la estrategia envolvente en algunos problemas. La combinación del clasificador kNN con RFSM en general ha dado mejores resultados que usando la medida de Liu, mientras que con los árboles de clasificación los resultados son similares.

Capítulo 4

Evaluación empírica de métodos de selección de características

Aplicando la descomposición modular del proceso de selección de características propuesto en la sección 2.4, nos proponemos evaluar una gran diversidad de métodos de selección de características. De esta forma pretendemos obtener una visión general del panorama de la selección de características. Siendo nuestro objetivo obtener la mayor información posible sobre la aplicación práctica de los diferentes métodos y, así, poder determinar el método o la familia de métodos más adecuada para un problema dado.

El estudio lo dividiremos en dos partes, evaluando dos grandes grupos de métodos, según el tipo de evaluación de características que utilizan (ver sección 2.6). En primer lugar, veremos los métodos de selección de características que usan algún tipo de evaluación de conjuntos de características completos. El segundo grupo lo forman aquéllos que valoran las características individualmente.

4.1. Métodos basados en evaluación de conjuntos de características

Con el objetivo de evaluar la mayor diversidad de métodos de selección de características posibles, combinaremos métodos de búsqueda representativos de los tipos de búsqueda más importantes con medidas de relevancia, también agrupadas según la teoría en que se basan.

<i>Métodos Búsqueda</i>		<i>20 conjuntos \leq 16 características</i>				<i>Todos</i>		<i>Todos</i>		<i>Todos</i>		
		<i>Completa</i>				<i>Secuencial</i>		<i>Probabilísticos</i>		<i>Meta-heurísticas</i>		
<i>Medidas Relevancia</i>		Exhausti.	FOCUS 2	B & B	ABB	SFS	SBS	LVF	LVW	SA	GA	TS
<i>Consistencia</i>	Consistencia básica	✗	[AD94] ✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
	Ejemplos inconsistentes (Liu)	[DL03] ✓	✗	✓	[DL03] ✓	[DL03] ✓	✓	[DL03] ✓	✗	✓	✓	✓
	Consistencia de la Teoría de Rough Sets	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓
	Pares de ejemplos inconsistentes (IEP)	✓	✗	✓	✓	[Das97, AD94] ✓	✓	✓	✗	✓	✓	✓
<i>Teoría Información</i>	Información mutua	✓	✗	✓	✓	[SPLC04] ✓	✓	✓	✗	✓	✓	✓
	Incertidumbre simétrica	✓	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓
	Longitud mínima de descripción (MDLC)	[SDN90] ✓	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓
<i>Distancia</i>	Basada en Relief (RFSM)	✓	✗	✓	✓	(sec. 3.4) ✓	✓	✓	✗	✓	✓	✓
<i>Rendimiento</i>	Envolvente usando validación cruzada de 10dobl. (Wrapper) - adult	✓	✗	✗	✗	✓	✓	✗	✓	✓	✓	✓

Tabla 4.1: Métodos de selección de características ordenados por tipo de búsqueda y medida de relevancia

La tabla 4.1 muestra las combinaciones que evaluaremos. Los métodos de búsqueda han sido agrupados según el tipo de búsqueda que realizan: completa, secuencial, probabilística o basada en meta-heurísticas. Las medidas han sido agrupadas según el principio en que se inspiran: consistencia de un conjunto de datos, teoría de la información, distancia entre ejemplos, o aplicación de la estrategia envolvente.

Sin embargo, no todas las combinaciones son apropiadas. En la tabla aparecen marcadas con un aspa las rechazadas. Las limitaciones consideradas son las siguientes: *i*) los métodos de búsqueda basados en ramificación y poda (BB y ABB) requieren que la medida sea monótona, *ii*) los dos métodos probabilísticos (LVF y LVW) están diseñados de forma complementaria para trabajar con medidas monótonas y no monótonas, respectivamente, *iii*) el método de búsqueda de FOCUS2 sólo tiene sentido si se considera una medida de consistencia en su criterio de parada y, además, desprecia la información extra que aportan las medidas de consistencia que no son la básica, *iv*) la medida de consistencia básica puede ser mejorada por cualquier otra medida de consistencia en los demás métodos de búsqueda. En total tienen sentido 67 combinaciones, que son los métodos que evaluaremos. Además, incluiremos en la evaluación un método ficticio que siempre elija todas las características, con el objeto de comparar las selecciones de características frente al resultado que se obtendría sin realizarlas.

Hay métodos previamente definidos y ampliamente conocidos. Para éstos, se ha incluido en su casilla la cita a la referencia bibliográfica donde se describen. En estos métodos, su implementación es fiel a la definición original, y servirán para comparar los resultados obtenidos con los demás métodos. En algunos métodos, como los basados en metaheurísticas, estamos seguros que hay una gran variedad de propuestas similares, pero es difícil precisar una cita al más parecido. Finalmente, algunos son propuestas novedosas por alguna de las siguientes razones: *i*) porque la medida que usan ha sido propuesta en este trabajo (RFSM), *ii*) porque, hasta donde hemos podido determinar en nuestra búsqueda bibliográfica, la medida no había sido usada previamente en selección de características (consistencia de *Rough Sets*), *iii*) o porque la combinación de la medida con el método de búsqueda no haya sido evaluada previamente.

En las tablas 4.2 y 4.3, se describen varias propiedades de los métodos de búsqueda: su eficiencia, dada como el orden del número de evaluaciones que necesita realizar; y la descripción de las tres partes de los métodos de búsqueda identificadas en la descomposición modular propuesta en el apartado 2.4. Cuando se exponen varias opciones en alguna parte del método, la implementada en nuestra evaluación es la indicada como habitual.

Tabla 4.2: Principales características de los métodos de búsqueda (1/2)

Método	Eficiencia (peor caso) (en n° de evaluaciones)	Punto de comienzo	Criterio de parada	Cada paso
Exhaustivo orden natural (usado por FOCUS)	$O(2^n)$	\emptyset	Llegar a F	N° eval./paso = 1 Creciente, añade 1 característica o reemplaza.
FOCUS2 (req. med. consistencia)	$O(2^n)$ (optimizado)	\emptyset	$M(S)=1$	N° eval./paso = 1 Elimina o añade características.
Branch & Bound (req. med. monótona)	$O(2^n)$ (optimizado)	F	Explorar todos los S tales que $ S < p$ (p puede ser dado como porcentaje)	N° eval./paso = 1 Elimina 1 en avance, añade características en backtraking.
ABB (req. med. monótona)	$O(2^n)$ (optimizado)	F	Explorar todos los S tales que $M(S) > u$ (u puede ser dado como porcentaje del máximo)	N° eval./paso = 1 Elimina 1 característica en avance, añade características en backtraking.
SFS, búsqueda secuencial hacia adelante (escalado de colinas) (SetCover)	$O(n^2)$	Posible: $S \in P(F)$ (cualquiera) Habitual: \emptyset No tienen sentido: F o cercanos a F	$M(S)=u$	N° eval./paso = $O(n)$ Estrictamente creciente, añade 1 característica. Del paso anterior necesita: S
SBS, búsqueda secuencial hacia atrás (eliminando características)	$O(n^2)$	Posible: $S \in P(F)$ (cualquiera) Habitual: F No tienen sentido: \emptyset o cercanos a \emptyset	Llegar a \emptyset	N° eval./paso = $O(n)$ Estrictamente decreciente, elimina 1 característica. Del paso anterior necesita: S
LVF (req. med. monótona)	$O(k)$	F	k pasos	N° eval./paso = 1 Elimina varias. Del paso anterior necesita: S
LVW (pensado para medidas NO monótonas)	$O(k_1)$	Posible: $S \in P(F)$ (cualquiera) Habitual: aleatorio ó \emptyset	k_1 pasos ó k_2 pasos sin mejora	N° eval./paso = 1 Añade o elimina varias. Del paso anterior necesita: S

Tabla 4.3: Principales características de los métodos de búsqueda (2/2)

Método	Eficiencia (peor caso) (en nº de evaluaciones)	Punto de comienzo	Criterio de parada	Cada paso
SA, enfriamiento simulado	$O(k)$	Posible: $S \in P(F)$ (cualquiera) Habitual: aleatorio	$k = k_1 \cdot k_2$ pasos k_1 : número de enfriamientos k_2 : número de vecinos	Nº eval./paso = 1 Puede añadir o eliminar una o varias características según el resultado criterio de Metropoli.
GA, algoritmos genéticos (consideramos tipo generacional)	$O(k)$ $k = k_1 \cdot k_2$	Posible: $\{S_i : S_i \in P(F), i = 1..k_1\}$ (varios conjuntos de características) Habitual: aleatorios	k_2 pasos k_1 : número de individuos k_2 : número de generaciones	Nº eval./paso = k_1 Puede añadir y eliminar una o varias características. Del paso anterior necesita: $\{S_i : i = 1..k_1\}$
TS, búsqueda tabú	$O(k)$	Posible: $S \in P(F)$ (cualquiera) Habitual: aleatorio	$k = k_1 \cdot k_2$ pasos k_1 : número de iteraciones k_2 : número de vecinos	Nº eval./paso = 1 Se añaden/eliminan varias características. Del paso anterior necesita: S y $\{S_i : i = 1..k_1\}$ soluciones tabú.

F = conjunto con todas las características del problema.

$n = |F|$ = número de características

$S \in P(F)$ = un conjunto cualquiera de características (P conjunto potencia)

$M(S)$ = valor de la medida para el conjunto S

m = número entero que depende de algún parámetro del algoritmo e independiente del problema (tanto de n como del número de ejemplos)

u = umbral, número real, un parámetro del algoritmo

k = número entero, parámetro que determina el número de evaluaciones del algoritmo

p = número entero, parámetro del algoritmo

4.1.1. Parámetros de los métodos

Se indican, a continuación, los parámetros usados en los métodos de selección de características elegidos. Dada una medida, los parámetros usados han sido los mismos independientemente de la combinación medida-búsqueda de la que formase parte. Lo mismo se aplica a los parámetros de los métodos de búsqueda.

4.1.1.1. Parámetros de las medidas

Todas las medidas están completamente determinadas a excepción de RFSM y Envolvente. RFSM tiene dos parámetros: el número de muestras a tomar, cuanto mayor sea mejor será la estimación, lo fijamos en 64 para que la evaluación no sea muy costosa en los conjuntos grandes; y el número de vecinos de cada clase a considerar, que fijamos en 4, valor con el que se obtienen resultados razonables. El método envolvente se puede aplicar con diversas técnicas para valorar los resultados del aprendizaje. El más comúnmente aplicado es la validación cruzada. Para no exigir un esfuerzo de computación excesivo y, a la vez, obtener resultados razonables, se ha utilizado con cinco dobleces.

4.1.1.2. Parámetros de la búsqueda

Los métodos de búsqueda Exhaustivo, FOCUS2, SFS y SBS están completamente determinados y, por tanto, no tienen parámetros que fijar. El resto de los métodos tienen al menos un parámetro variable. A continuación, se describe la elección tomada para cada uno.

Los dos métodos de búsqueda del tipo ramificación y poda (*Branch and Bound*) se han escogido porque cubren dos formas distintas de enfocar la selección de características. BB exige una reducción determinada de características, mientras que ABB exige mantener una determinada relevancia.

BB es un algoritmo que elige el mejor conjunto de un número de características previamente determinado. Para probarlo en todos los conjuntos de datos, lo hemos hecho relativo al número total de características. De esta forma, el algoritmo eliminará siempre el 40% de las características de un conjunto de datos.

ABB está diseñado para elegir el mejor conjunto que mantiene la medida de

evaluación por encima de un cierto umbral. Para determinar este umbral de una forma consistente, que sirva para todas las medidas y conjuntos de datos, hemos establecido que permita conjuntos cuyo valor de la medida sea un 1% menor al valor máximo de la medida en el conjunto de datos en que se está aplicando. Como las medidas que se pueden aplicar con este método son monótonas, el valor máximo viene dado por el conjunto con todas las características. Dicho de otro modo, reduce todas las características que pueda manteniendo el 99% de relevancia para la clasificación.

Los dos algoritmos probabilísticos de tipo Las Vegas son similares. Su parámetro principal es el número límite de evaluaciones, que hemos fijado para ambos en 1000 conjuntos de características. Sin embargo, cada uno de ellos es apropiado para un tipo distinto de medidas lo que condiciona el otro de sus parámetros.

LVF necesita limitar la reducción de relevancia de las características que, por homogeneidad con ABB, hemos fijado también en un 1%.

LVW incluye un parámetro para evitar perder tiempo de ejecución cuando no es previsible ninguna mejora. Para ello, limita el número de evaluaciones que realiza sin mejora. Este valor lo hemos fijado en 100 que parece un límite razonable en las 1000 evaluaciones.

De los métodos empleados, las metaheurísticas son los métodos con mayor flexibilidad. La primera restricción que hemos impuesto sobre sus parámetros es que hagan 1000 evaluaciones, con el objeto de que puedan ser aplicadas en todos los conjuntos de datos elegidos y darles las mismas oportunidades que a los métodos probabilísticos. Dentro de esa restricción, los parámetros se han determinado mediante experimentación sobre dos conjuntos de datos, usando diferentes valores de los mismos según recomendaciones de expertos. Se han probado en torno a 20 combinaciones diferentes para cada metaheurística.

En SA (*Simulated Annealing*) es preciso fijar:

- Representación: La solución es representada como un vector de bits, con tantos bits como características tiene el problema. Cada bit se corresponde con una característica y su valor indica la presencia o ausencia de la característica asociada en el conjunto seleccionado.
- Temperatura inicial: Para que SA no sea muy dependiente de la solución inicial, partimos de una temperatura inicial alta. De esta forma, al principio se aceptarán soluciones lejanas a la actual. Utilizamos $T_0 = (-\nu / -\ln(\phi))(S_0)$,

que permite una probabilidad ϕ de aceptar una solución que sea un ν por uno peor que la solución inicial S_0 . Tomamos $\nu = 0,3$, y $\phi = 0,3$.

- Generación de vecinos: La generación de nuevos vecinos se realiza mediante la adición o eliminación de una característica del conjunto actual. Se generan 20 vecinos en cada enfriamiento, es decir, para cada valor de la temperatura.
- Esquema de enfriamiento: Hemos utilizado el de Cauchy ($T = T_0/(1 + i)$). Se realizan 50 enfriamientos.

En GA (*Genetic Algorithms*):

- Representación: Vector de bits, igual al usado en SA.
- Tipo generacional, con una población de 30 individuos y 34 generaciones.
- El cruce es tipo simple en un punto, sobre la representación binaria del conjunto de características seleccionadas. Probabilidad de cruce: 0,6.
- La mutación añade o elimina una característica. Probabilidad de mutación: 0,001.

En TS (*Tabu Search*):

- Representación. Vector de bits, igual al usado en SA.
- Realizaremos 12 reinicializaciones de la búsqueda.
- Los vecinos se generan por adición o eliminación de una característica. En cada iteración se visitarán 84 vecinos.

4.2. Métodos basados en la evaluación de características individuales

Estos métodos utilizan medidas que asignan una valoración de la relevancia a cada característica. En la figura 4.1 se muestra una representación de un vector de las características con sus valoraciones, y su versión ordenada según la valoración. No todos los métodos necesitan ordenar las características. El objetivo fundamental de

esta figura es aclarar la notación que se usará en las figuras siguientes, donde la permutación $\sigma(i)$ dispone las características por orden decreciente de su valoración:

$$M(f_{\sigma(i)}) > M(f_{\sigma(j)}) \forall i < j \quad (4.1)$$

f_1	f_2	f_3	f_4		f_n
$M(f_1)$	$M(f_2)$	$M(f_3)$	$M(f_4)$. . .	$M(f_n)$

a) valoración de las características.

$f_{\sigma(1)}$	$f_{\sigma(2)}$	$f_{\sigma(3)}$	$f_{\sigma(4)}$		$f_{\sigma(n)}$
$M(f_{\sigma(1)})$	$M(f_{\sigma(2)})$	$M(f_{\sigma(3)})$	$M(f_{\sigma(4)})$. . .	$M(f_{\sigma(n)})$

b) características ordenadas según su valoración.

Figura 4.1: Valoración de características con una medida individual

Los métodos de selección de características individuales, una vez evaluadas las características, siempre escogerán las que obtengan valores mayores de la medida. Para completar la selección, necesitan determinar cuántas características seleccionarán y cuántas serán descartadas. Existen muchos criterios de corte y, para nuestro conocimiento, no hay estudios que determinen cuál es el más adecuado. En muchas propuestas, simplemente se determina que debe usarse algún tipo de umbral como corte y se deja al usuario del método la determinación del mismo.

4.2.1. Criterios de corte

En el estudio que presentamos aquí, hemos intentado fijar criterios de corte que se puedan aplicar de forma genérica a todos los problemas. Uno de los objetivos que nos planteamos es determinar si es posible recomendar un criterio que funcione bien en la mayoría de los casos.

Para la aplicación de los criterios de corte todas las medidas serán trasladadas al intervalo $[0, 1]$. De esta forma, los criterios a los que les afecta el ranking de aplicación de la medida serán independientes de la medida aplicada.

Cada uno de los criterios que evaluaremos tiene un parámetro. En primer lugar,

mediante experimentación fijamos un valor para este parámetro, escogiendo aquel valor del parámetro que dé mejores resultados en ranking promedio de acierto en su aplicación sobre todos los conjuntos de datos. Los criterios de corte seleccionados se describen a continuación, junto con los detalles de la elección de su parámetro. Entre paréntesis se indica el código por el que aparecerá el método en las tablas de resultados. Este código incluye el valor elegido de su parámetro.

Los seis criterios de corte evaluados son los siguientes:

- Las m mejores características (n17). Simplemente escoge las m características con mayor valoración. Se ilustra en la figura 4.2. En la experimentación se ha probado con $m \in \{5, 8, 11, 14, 17\}$.
- Una fracción de las características disponibles (p0.8). Escoge las $m = np$ características con mayor valoración. Se ilustra en la figura 4.3. En la experimentación se ha probado con $m \in \{0,4, 0,5, 0,6, 0,7, 0,8\}$.
- Valoración por encima de un umbral (t0.1). Se escogen las características cuya valoración supere el umbral dado. Se ilustra en la figura 4.4. En la experimentación se ha probado con umbral: $u \in \{0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8\}$.
- Valoración por encima de una fracción del ranking de las valoraciones (pm0.8). Se escogen las características cuya valoración sea superior a un umbral dado como fracción de la variación máxima de la medida en las características disponibles. Se ilustra en la figura 4.5. En la experimentación se ha probado con $pm \in \{0,1, 0,2, 0,3, 0,4, 0,5, 0,6, 0,7, 0,8\}$.
- Bajada de valoración absoluta (d0.2). Sobre las características ordenadas de mayor a menor por la valoración de la medida, vamos escogiendo características hasta que la diferencia con la siguiente sea superior al valor dado como parámetro. Se ilustra en la figura 4.6. En la experimentación se ha probado con $d \in \{0,05, 0,10, 0,15, 0,20, 0,25\}$.
- Bajada de valoración relativa (s1.5). Sobre las características ordenadas de mayor a menor por la valoración de la medida, vamos escogiendo características hasta que pendiente entre una valoración y la siguiente sea superior al valor dado como parámetro. Se ilustra en la figura 4.7. En la experimentación se ha probado con $s \in \{0,50, 0,75, 1,00, 1,25, 1,50\}$.

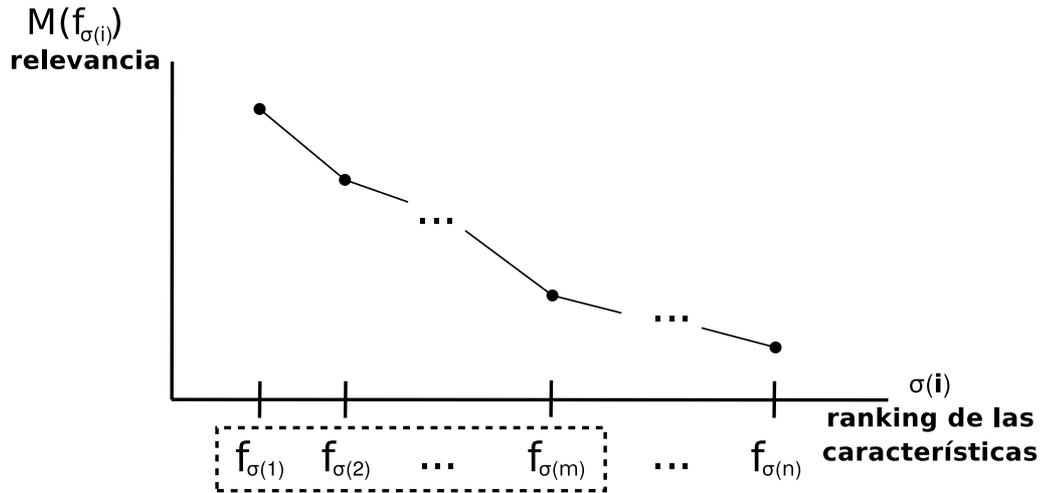
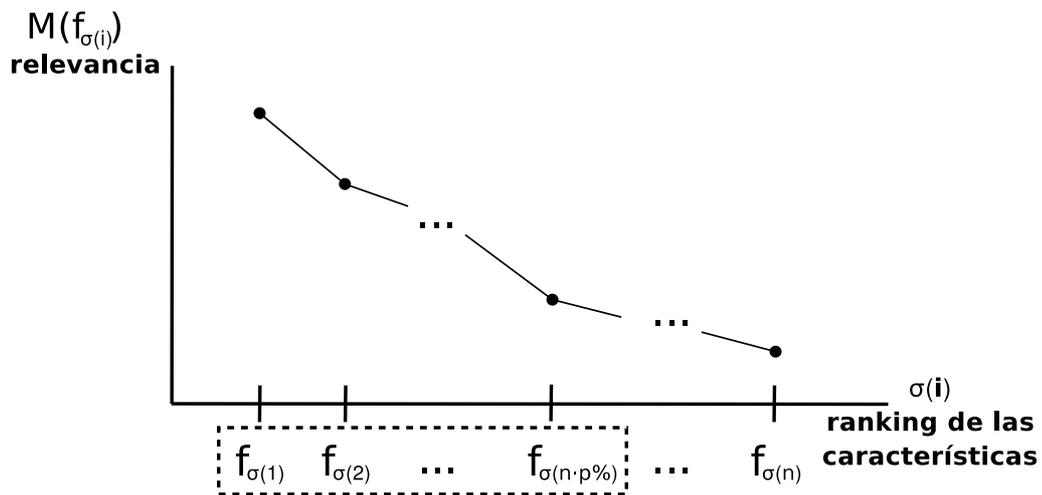
Figura 4.2: Criterio de corte: las m características más relevantes

Figura 4.3: Criterio de corte: un porcentaje de las características disponibles

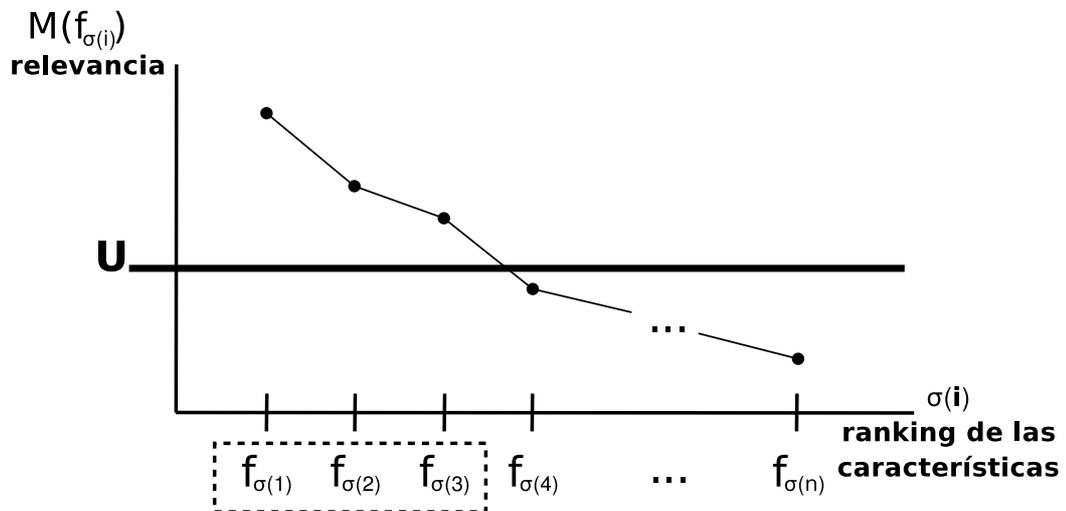


Figura 4.4: Criterio de corte: características cuya valoración sea superior a un umbral

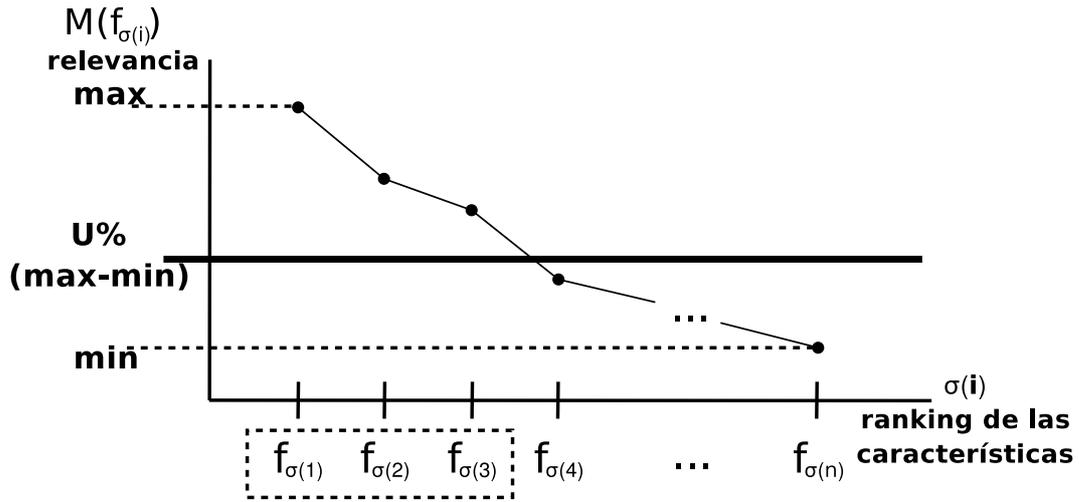


Figura 4.5: Criterio de corte: umbral como porcentaje de la variación máxima

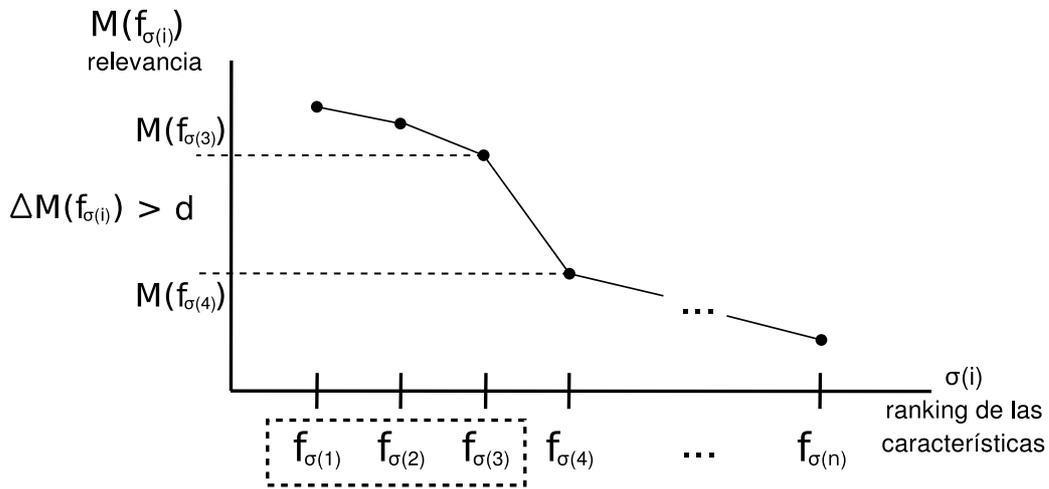


Figura 4.6: Criterio de corte: hasta una diferencia superior a un umbral dado

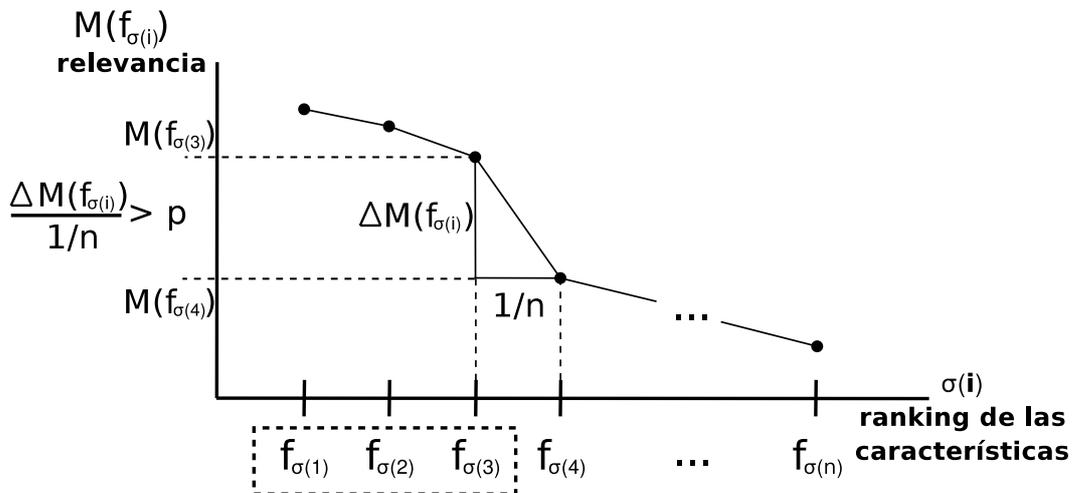


Figura 4.7: Criterio de corte: hasta una pendiente superior a un umbral dado

4.2.2. Combinación con las medidas de características individuales

En total evaluaremos 30 métodos, formados por la combinación de los tipos de criterios de corte descritos con cada una de las cinco medidas descritas en el apartado 2.6.1:

- Información mutua (info)
- Gain ratio(gain)
- Índice de Gini(gini)
- Valoración de Relief-F (reli)
- Valor potencial en la formación de reglas de decisión (rele)

4.3. Metodología experimental

4.3.1. Conjuntos de datos

Habitualmente los métodos de Aprendizaje Automático actuales se han presentado mostrando una mejora sobre los resultados de métodos anteriores. Esta mejora se puede acreditar sobre un conjunto de datos que representa un problema concreto o, más eficazmente, sobre varios conjuntos de datos. Si bien la búsqueda de un algoritmo que funcione mejor que otro en absolutamente todos los problemas, se ha demostrado inútil por el teorema de *No-free-lunch* [HP02], sí que puede ser interesante buscar métodos que funcionen bien en los problemas con los que nos encontramos habitualmente, o algoritmos que funcionen mejor en una clase concreta de problemas.

Los conjuntos de datos pueden ser artificiales, cuando se crean a partir de la definición de un problema conocido, o reales, cuando son datos experimentales tomados para resolver un problema dado en una situación real. Para mostrar la efectividad de los algoritmos de selección de características, se han usado ambos tipos de conjuntos de datos. Los artificiales han sido ampliamente usados en la bibliografía de selección de características, porque permiten tener perfectamente identificadas las

características irrelevantes y las redundantes, y averiguar fácilmente si el método de selección está acertando en la respuesta que da.

Por otra parte, el uso de conjuntos de datos artificiales, sin saber si se parecen o no a los problemas que pretendemos abordar, nos aporta poco sobre el rendimiento de los métodos en el contexto real de aplicación. Por esta razón, en nuestra experimentación únicamente hemos incluido dos conjuntos artificiales. Estos conjuntos, *parity3+3* y *led24*, los hemos considerado interesantes porque pueden formar parte de otros problemas reales más complejos.

Para realizar la experimentación, hemos usado 36 conjuntos de datos que han sido elegidos entre los usados en diversas publicaciones. Hemos empleado una gama suficientemente variada y representativa de problemas de clasificación, la mayoría disponibles en el repositorio de la UCI. Todos ellos han sido descritos previamente en diversos trabajos publicados en revistas científicas de prestigio. En las tablas descritas a continuación se indica de dónde se ha tomado cada conjunto.

Las tablas 4.4, 4.5, 4.6 y 4.7 muestran las características más importantes de los conjuntos de datos que hemos utilizado en la experimentación:

- *Nombre* indica el nombre por el que se suele conocer el conjunto de datos aunque, a veces, éste varía entre las distintas fuentes.
- *Nº ej.* es el número de ejemplos (tuplas) que tiene el conjunto de datos.
- *Nº car.* es el número de características del problema incluyendo la clase.
- *Tipo* indica de qué tipos son las características:
 - *Discrete*, todas las características son discretas.
 - *Continuous*, todas las características son continuas.
 - *Mixed*, hay características discretas y continuas mezcladas.
- *Mayo.* es el porcentaje de ejemplos que pertenecen a la clase mayoritaria. Se usa como umbral mínimo del porcentaje de clasificación aceptable para un clasificador aplicado a ese problema.
- *Desc* es el número de valores desconocidos en el conjunto de datos.
- *Desc(ca)* es el número de características que contienen valores desconocidos.
- *Desc(ej)* es el número de ejemplos que contienen valores desconocidos.

Tabla 4.4: Conjuntos de datos (1/4)

Nombre	Nºej.	Nºcar.	Tipo	Nºcl.	Mayo.	Desc	Desc(ca)	Desc(ej)	Origen	Descripción
adult	32561	14	Mixed	2	75.92	4262	3	2399	UCI	Datos del censo de Estados Unidos. El objetivo es predecir si los ingresos son inferiores o superiores a 50000\$.
anneal	898	38	Mixed	5	76.17	22175	29	898	UCI	Annealing.
audiology	226	69	Discrete	24	25.22	319	8	222	UCI	Audiology database.
balance-scale	625	4	Discrete	3	46.08	0	0	0	UCI	Resultados de experimentos psicológicos.
breast-cancer	286	9	Mixed	2	70.28	9	2	9	UCI	Cáncer de pecho, datos de University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.
bupa	345	6	Continuous	2	57.97	0	0	0	Org	Desórdenes del hígado debidos a la ingestión de alcohol, de BUPA Medical Research Ltd.
car	1728	6	Discrete	4	70.02	0	0	0	UCI	Base de datos de evaluación de coches, originalmente creada desde un modelo simple de decisión jerárquico.
credit	690	15	Mixed	2	55.51	67	7	37	UCI	Solicitudes de tarjetas de crédito.
echocardiogram	131	10	Mixed	2	67.18	101	9	70	UCI	Datos de ecocardiogramas de pacientes que han sufrido algún ataque al corazón. Predecir si vivirán después de un año.
horse-colic	368	26	Mixed	2	63.04	1927	21	361	UCI	Cólicos en caballos. Usando “surgical lesion” como clase.
house-votes84	435	16	Discrete	2	61.38	392	16	203	UCI	Registros de voto del congreso de los EEUU en 1984.

Nombre	Nºej.	Nºcar.	Tipo	Nºcl.	Mayo.	Desc	Desc(ca)	Desc(ej)	Origen	Descripción
ionosphere	351	32	Continuous	2	64.10	0	0	0	UCI	Datos de radar en Goose Bay, Labrador. Tomados con 16 antenas de alta frecuencia, el objetivo es detectar electrones libres en la ionosfera.
iris	150	4	Continuous	3	33.33	0	0	0	UCI	Clasificación de plantas de lirios.
labor-neg	57	16	Mixed	2	64.91	326	16	56	Sgi	Acuerdos establecidos en negociaciones laborales del sector industrial de Canadá.
led24	1200	24	Discrete	10	10.58	0	0	0	UCI	Averiguar el dígito de un display led de 7 segmentos. Problema artificial incluyendo 16 características irrelevantes y ruido.
lenses	24	4	Discrete	3	62.50	0	0	0	UCI	Determina para un paciente si son adecuadas las lentes de contacto y de qué tipo.
lung-cancer	32	56	Discrete	3	40.62	5	2	5	UCI	Cáncer de pulmón, distinción entre tres tipos patológicos.
lymphography	148	18	Discrete	4	45.27	0	0	0	UCI	Uno de los tres conjuntos del University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.
mushrooms	8416	22	Discrete	2	53.33	2480	1	2480	UCI	Registros de setas tomados de <i>The Audubon Society Field Guide to North American Mushrooms</i> .
parity3+3	500	12	Discrete	2	53.20	0	0	0	UCI	Paridad de 3 bits. Problema artificial con 3 características redundantes y 6 irrelevantes.

Tabla 4.5: Conjuntos de datos (2/4)

Nombre	Nºej.	Nºcar.	Tipo	Nºcl.	Mayo.	Desc	Desc(ca)	Desc(ej)	Origen	Descripción
pima	768	8	Continuous	2	65.10	0	0	0	UCI	Diabetes en mujeres indias Pima.
post-operative	90	8	Mixed	3	71.11	3	1	3	UCI	Pacientes en post-operatorio.
primary-tumor	339	17	Discrete	21	24.78	225	5	207	UCI	Tumores, datos de University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.
promoters	106	57	Discrete	2	50.00	0	0	0	UCI	E. coli promoter gene sequences
saheart	462	9	Mixed	2	65.37	0	0	0	ESL	Datos de varones de una zona del sur de Africa con alto riesgo de enfermedad coronaria del corazón.
shuttle-landing-control	253	6	Discrete	2	57.31	0	0	0	Org	Control de aterrizaje automático de un cohete espacial.
soybean	307	35	Discrete	19	13.03	712	34	41	UCI	Base de datos grande con datos de enfermedades de la soja.
splice	3190	60	Discrete	3	51.88	0	0	0	Del	Unión de secuencias de ADN de primates.
tic-tac-toe	958	9	Discrete	2	65.34	0	0	0	UCI	Posibles situaciones del juego “tic-tac-toe” (tres en raya), determinar si ganará.
vehicle	846	18	Continuous	4	25.77	0	0	0		Cuatro tipos de vehículos clasificados a partir de características deducidas de su silueta. Turing Institute, Glasgow, Scotland.
vowel	990	10	Continuous	11	9.09	0	0	0	Org	Reconocimiento del sonido de vocales en inglés independientemente del hablante.

Tabla 4.6: Conjuntos de datos (3/4)

Nombre	Nºej.	Nºcar.	Tipo	Nºcl.	Mayo.	Desc	Desc(ca)	Desc(ej)	Origen	Descripción
wdbc	569	20	Continuous	2	62.74	0	0	0	Org	Diagnóstico del cáncer de pecho, datos de Wisconsin.
wine	178	13	Continuous	3	39.89	0	0	0	UCI	Análisis químicos de vinos de tres cultivos distintos de una misma región de Italia.
yeast	1484	8	Continuous	10	31.20	0	0	0	UCI	Localización de emplazamientos de proteínas.
yeast-class-RPR	186	79	Continuous	3	65.05	214	69	116	Org	n/d
zoo	101	16	Discrete	7	40.59	0	0	0	UCI	Identificación de animales por características simples. Base de datos Zoo.

Tabla 4.7: Conjuntos de datos (4/4)

- *Origen* indica de donde se han recogido los conjuntos de datos, aunque, al estar repetidos algunos en varios repositorios, puede que su fuente original sea otra. Las fuentes de los conjuntos de datos son:
 - UCI: Biblioteca de conjuntos de datos de la Universidad de California, Irvine [HB99].
 - ESL: *Elements of Statistical Learning* [HTF01].
 - Org: Conjuntos de datos del sitio web de Orange [DZL06].
 - Sgi: Biblioteca de conjuntos de datos de Silicon Graphics [Gra06].
 - Del: Delve (*Data for Evaluating Learning in Valid Experiments*) [Tor03]

4.3.2. Clasificadores

Como medida de rendimiento del proceso de selección de características empleamos la calidad de la clasificación alcanzada con cuatro clasificadores distintos. Estos clasificadores se han seleccionado por su representatividad. Los más simples, naive-bayes y k -NN, están basados en la teoría de la probabilidad y en la medida de distancias entre ejemplos, respectivamente, y representan la base de muchos otros, más complejos, entre los que sería difícil elegir. Los árboles de clasificación, asemejando la clasificación humana, representan a los algoritmos que proveen una fácil interpretación de sus resultados de aprendizaje. Finalmente, las redes neuronales artificiales son uno de los métodos más avanzados y que obtiene mejores resultados actualmente.

4.3.2.1. Naive-bayes

Naive-bayes es un clasificador simple basado en la aplicación del teorema de Bayes. Para simplificar, usa una suposición muy fuerte: que las características son todas independientes entre sí. Por esta razón, se le denomina *naive* (ingenuo). Simplemente, dadas las probabilidades del conjunto de datos como muestra, aplica la expresión (4.2) sobre el ejemplo a clasificar y lo asigna a aquella clase que obtenga mayor probabilidad.

$$p(C|F_1, \dots, F_n) = \frac{1}{p(F_1, \dots, F_n)} p(C) \prod_{i=1}^n p(F_i|C) \quad (4.2)$$

Hemos considerado que es un método de aprendizaje interesante para incluirlo en este estudio por diversas razones. Primero, porque este algoritmo de aprendizaje constituye una base para comparar con otros algoritmos. Así, naive-bayes fija un nivel mínimo que deberían superar todos los sistemas más elaborados para ser tenidos en cuenta. No obstante, tiene aplicaciones directamente en problemas reales donde funciona bien [Ris01] y es posible combinar varios de estos clasificadores en estrategias tipo *bagging* y *boosting* [BK99] obteniendo mejores resultados. Creemos que al ser combinado con selección de características también puede dar resultados interesantes.

4.3.2.2. Los k -vecinos más cercanos (k -NN)

El algoritmo de aprendizaje k -NN para clasificación se basa en la suposición de que los ejemplos cercanos pertenecen a la misma clase. Su fase de aprendizaje es muy simple, pues se limita a almacenar los ejemplos del conjunto de entrenamiento. La fase de clasificación también es simple, aunque más costosa en eficiencia. El clasificador busca los k ejemplos más cercanos al dato que se quiere clasificar y le asigna la clase más frecuente entre ellos.

4.3.2.3. Árboles de clasificación

Los árboles de decisión son estructuras en forma de árbol que permiten representar decisiones en función de los valores de ciertas características. En cada nodo del árbol, se sigue el camino por alguna de sus ramas según el valor de la característica correspondiente. Los nodos hoja representan la decisión a tomar. Cuando se aplican a un problema de clasificación, los árboles se llaman *de clasificación* y la decisión que se toma es la clase a asignar al ejemplo. Al representar de esta forma sus decisiones, estos algoritmos de aprendizaje son fáciles de interpretar y, por ello, su uso está muy extendido. Hay muchos tipos de algoritmos de aprendizaje que construyen árboles de clasificación, y numerosa bibliografía sobre ellos [Bre98]. El tipo más utilizado en la actualidad es el árbol C4.5 de Quinlan[Qui93].

4.3.2.4. Redes neuronales artificiales

Inspiradas en las redes neuronales reales, las redes neuronales artificiales están formadas por elementos sencillos de cálculo (modelos simplificados de una neurona real)

que se interconectan formando una red. En conjunto pueden exhibir comportamientos complejos y resolver problemas difíciles. Hay muchos tipos de redes y algoritmos de entrenamiento para realizar el aprendizaje [Hay99]. El tipo de red más conocido y utilizado es el perceptrón multicapa, que emplearemos en este estudio.

4.3.2.5. Parámetros elegidos en cada tipo de aprendizaje

Con el objeto de que la experimentación sea abordable y centrarnos en la selección de características, usaremos los métodos de aprendizaje con una asignación de parámetros fija para todos los conjuntos de datos. En la aplicación práctica a un problema concreto, los parámetros deberían ser ajustados al conjunto de datos pero, en nuestro estudio, esto alteraría la valoración de algunos métodos. Por ejemplo, en el caso de los métodos envolventes (*wrappers*), los parámetros afectan a la selección, y la selección a los parámetros. No cabe duda de que sería interesante investigar las interacciones entre los parámetros del aprendizaje y la selección, pero eso queda fuera del ámbito del estudio que planteamos aquí.

Para la elección de los parámetros, hemos realizado un experimento reducido, utilizando todos los conjuntos de datos elegidos, pero sin selección de características. Excepto para Naive-bayes que no tiene parámetros, se detallan a continuación las combinaciones probadas para cada método y los parámetros finalmente elegidos.

De los árboles de clasificación probamos un método básico de construcción de árboles [DZ04] con diferentes valores de su parámetro de poda: 1, 2, 3 y 4; y el método C4.5. Los árboles de clasificación que mejor funcionaron fueron los generados con el valor 4 del método básico y con C4.5, sin diferencia significativa entre ellos. Elegimos C4.5 porque éste es un método más conocido y se incluye habitualmente en las comparativas entre métodos de aprendizaje.

En el algoritmo k -NN, el parámetro k lo elegiremos impar para reducir la probabilidad de empates. Así, decidimos probar los siguientes valores: 3, 9, 15 y 21. Aunque las diferencias no eran significativas entre los valores 9, 15 y 21, el que mayor porcentaje de clasificación correcta obtuvo fue $k = 15$.

Para las redes neuronales fijamos el tipo de salidas para clasificación, la estructura y el algoritmo de entrenamiento. En clasificación, necesitamos que la salida de la red determine una clase. Hay varias formas de hacerlo. Nosotros usaremos una salida para cada clase, que la red valorará entre 0 y 1, y determinaremos que la clase sea aquella cuyo valor sea mayor. En cuanto a la estructura, dado que se ha

demostrado que con sólo una capa oculta el perceptrón multicapa es capaz de calcular lo mismo que con más capas [HSW89] y, por tanto, prefiriendo los sistemas más simples, usaremos sólo una capa oculta. Para establecer el tamaño de la capa oculta, aplicaremos una regla heurística que sugiere que el número de neuronas en la capa oculta sea la media entre el número de entradas y de salidas de la red.

En relación con el algoritmo de entrenamiento a utilizar, consideramos las tres opciones más habituales: retropropagación estándar, Rprop y BPprop. Los valores estudiados para el número de ciclos fueron 250, 500 y 1000. Y finalmente, la tasa de aprendizaje entre 0,1 y 0,2. Realizamos experimentos preliminares con todas estas combinaciones. La red que mejor se comportó y, en consecuencia, la que hemos utilizado para los experimentos de selección de características, fue la que utilizaba retropropagación estándar con tasa de aprendizaje 0,1 y era entrenada durante 500 ciclos.

4.3.2.6. Clasificadores sin selección de características

Como valores de referencia, en la tabla 4.8, indicamos los valores promedio de la eficacia que los clasificadores de cada tipo produjeron. Estos valores son el resultado promedio de aplicar una validación cruzada con parámetro 10 en los 36 conjuntos de datos considerados. En la tabla A.2 del Apéndice A se indican los datos para cada problema.

<i>Método de selección</i>	<i>%Acierto promedio</i>			
	<i>NBayes</i>	<i>C4.5</i>	<i>k-NN</i>	<i>RNA</i>
Sin SC	77,17	80,81	78,70	83,36

Tabla 4.8: Rendimiento medio de los métodos de clasificación sin selección de características

4.3.3. Transformaciones de los datos

4.3.3.1. Discretización

La *discretización* es necesaria para aplicar aquellos métodos que sólo trabajan con características discretas a los problemas que contienen características continuas. Esta *discretización* sólo la aplicaremos cuando sea imprescindible, y sólo para el método de selección de características. Una vez seleccionadas las características, el método de aprendizaje tendrá acceso a las originales, no a las transformadas.

Los métodos de *discretización* más conocidos son: la división en intervalos de espacios equidistantes, división en intervalos de frecuencias equitativas, el método de Fayyad Irani que usa teoría de información para elegir el tamaño y número de intervalos, y Chi2 [LHTD02].

Para elegir el método de *discretización* a usar, en primer lugar, se descartaron los métodos elaborados como Fayyad Irani y Chi2 porque pueden dejar algunas características sin intervalos, haciendo de esta forma su propia selección de características y, por tanto, distorsionar los resultados de nuestros experimentos (como comentamos en la sección 2.1, la *discretización* puede ser usada para realizar selección de características).

A continuación, para elegir entre la *discretización* en intervalos equidistantes o de frecuencias de aparición iguales (percentiles), y el número de intervalos, nos planteamos el siguiente experimento: probar todas las combinaciones entre los dos tipos de división de intervalos y número de intervalos a dividir entre 3, 4, 5, 6 y 7. Como métodos de selección de características emplearemos todas las medidas que necesitan *discretización* combinadas con el método de búsqueda exhaustiva para asegurarnos que la búsqueda no distorsiona el resultado. La evaluación se realiza en todos los conjuntos donde consideramos que es abordable la búsqueda exhaustiva (< 16 características).

El mejor en *ranking* promedio (media de la posición alcanzada al ordenar los métodos según su resultado en cada experimento) de los experimentos con cada conjunto de datos ha sido la división en 6 intervalos de la misma frecuencia de aparición. Esta conclusión concuerda con el artículo [LHTD02] que afirma que el método de *discretización* por intervalos de frecuencia es preferible al de intervalos del mismo tamaño.

4.3.3.2. *Continuización*

En los métodos que sólo trabajan con características continuas, es necesaria la transformación contraria a la *discretización*. En este caso, las características discretas son *continuizadas* transformando sus valores discretos en puntos equidistantes en el intervalo $[0, 1]$.

4.3.3.3. Valores nulos o desconocidos

Algunos métodos de selección de características toleran la presencia de valores nulos en los datos de entrada, pero otros no. Para aquellos métodos que no pueden gestionar los valores nulos, aplicaremos un filtro para eliminarlos.

Nos planteamos tres opciones: *i*) eliminar todos los ejemplos en los que aparezca un valor nulo, *ii*) sustituir los valores nulos por la media de su característica, *iii*) eliminar las características en las que aparezcan valores nulos.

En primer lugar, descartamos la última opción por distorsionar la selección de características y eliminar en muchos casos demasiada información. A continuación, realizando un experimento similar al de los métodos de *discretización*, se ha elegido sustituir por la media. La razón principal es que en dos conjuntos de datos (labor-neg, y anneal) no es posible eliminar los ejemplos con valores nulos, pues todos o casi todos los ejemplos tienen algún valor nulo, mientras que, en los datos que sí es posible eliminarlos, no se ha apreciado ninguna diferencia significativa en los resultados de las dos opciones.

4.3.4. Entorno de desarrollo y ejecución

4.3.4.1. Entorno software

Hemos elegido **Orange** [DZ04] como entorno software donde implementar los algoritmos y ejecutar los experimentos. Es un entorno para minería de datos especialmente bueno para investigación y enseñanza en temas de Aprendizaje Automático. **Orange** está desarrollado en los lenguajes de programación **python** y **C++** combinando lo mejor de ambos: interpretabilidad y fácil uso de **python** para la interfaz con el usuario programador, y la eficiencia de **C++** para la ejecución de los algoritmos internamente.

En el momento de la elección, **Orange** era, además, el único paquete de software libre para Aprendizaje Automático lo suficientemente completo para los experimentos que se han realizado. No obstante, adolecía de la falta de una implementación de redes neuronales artificiales. Por esta razón, hubo que desarrollar el módulo **OrangeSNNS** para permitir el uso del software **SNNS** [oS95] desde **Orange**. Puede encontrarse más información en B.4.

4.3.4.2. Implementación de los métodos

Los métodos de selección de características han sido todos implementados en `python`. La elección de este lenguaje está motivada por su facilidad para el desarrollo rápido y condicionada por la elección del entorno `Orange`.

El diseño realizado de los métodos ha sido orientado a objetos, y se comenta en el Apéndice B. Durante el diseño se ha prestado especial atención a propiedades fundamentales como generalidad, extensibilidad y eficiencia.

4.3.4.3. Servidores de cálculo

Para la ejecución de toda esta carga experimental, se han usado las siguientes plataformas hardware:

- Cluster de la E.T.S.I. Informática, con 8 nodos cada uno con 2 procesadores Xeon Dual de 2.8GHz y 1GB de RAM
- Cluster del grupo de investigación, con 8 nodos cada uno con 1 procesador Xeon Dual de 3.2GHz y 1GB de RAM
- Un AMD Athlon XP 2800+ con 2GB de RAM
- Un Intel Pentium 4 CPU 3.20GHz con 2GB de RAM

Ha sido necesario utilizar varios servidores de cálculo para poder ejecutar todos los experimentos en un tiempo razonable, teniendo en cuenta además que no todos los servidores han estado disponibles durante toda la experimentación. Incluyendo el tiempo perdido por cortes de luz, sobrecalentamiento de los servidores y tareas de administración relacionadas con estos eventos, la duración de toda la ejecución de esta fase de experimentación ha sido de ocho meses.

Lo ideal para la medición del tiempo de ejecución de los algoritmos hubiese sido la utilización siempre del mismo servidor. No obstante, hemos comprobado empíricamente que los tiempos de ejecución (medidos en tiempo de uso de la CPU) no son muy distintos entre los diferentes servidores para un mismo algoritmo. Dado que el tiempo que tardan en la CPU más lenta nunca es más del doble de lo que tardan en la más rápida, hemos establecido como observación conservadora, a la hora de comparar tiempos, que sólo consideremos distintos aquellos tiempos que

se diferencien en más del doble, lo que para nuestros análisis sobre el tiempo de ejecución de los algoritmos es más que suficiente, ya que suele haber diferencias de varios órdenes de magnitud.

4.3.5. Configuración de los experimentos

Los datos que se obtienen en la validación de un método de selección de características son: el porcentaje de acierto de clasificación, el número de características, y el tiempo que ha tardado cada una de las partes del proceso: la selección, el aprendizaje y la aplicación sobre los datos de prueba. En los experimentos realizados, las variables controladas que actúan como factores son: el algoritmo de aprendizaje, el conjunto de datos, y el método de selección, que puede dividirse en dos factores: medida y búsqueda.

El diseño del experimento global es completo, con el objetivo de posteriormente poder estudiar todas las interacciones de los factores, de forma aislada o en conjunto. En consecuencia, se probarán todas las combinaciones posibles entre los factores, a excepción de algunas de ellas. Estas excepciones vienen impuestas por las dos limitaciones siguientes. En primer lugar, los métodos que usan un tipo de búsqueda completa tienen órdenes de eficiencia exponenciales respecto al número de características, por lo que no es factible aplicarlos en conjuntos de datos con muchas características. Estos métodos los aplicaremos sólo en conjuntos con 16 o menos características. La otra limitación es sobre el conjunto de datos *adult*. Dado su gran tamaño, resulta muy costosa la aplicación de la estrategia envolvente, por lo que en este conjunto no se ha probado ningún método con la medida de rendimiento envolvente. Estas limitaciones aparecen reflejadas de forma ilustrativa en la tabla 4.1, usando un tipo de letra *Sans* cursiva, en la parte superior de la tabla y en la tabla de la medida envolvente.

En cada experimento aplicamos validación cruzada con diez dobleces. Hemos elegido esta técnica como equilibrio entre resultados fiables y tiempo de cálculo, siendo un término medio entre la estrategia dejar-uno-fuera (*leave-one-out*) y la división del conjunto de datos una única vez en entrenamiento y prueba.

Para los algoritmos no determinísticos repetiremos el experimento tres veces sobre las mismas particiones, tomando como resultado de cada partición la media de estas tres ejecuciones. De esta forma, nos aseguramos que el resultado sea más estable.

En cada partición mediremos:

- Porcentaje de clasificación correcta en datos de entrenamiento
- Porcentaje de clasificación correcta en datos de prueba
- Número de características seleccionadas
- Tiempo de selección de características
- Tiempo de aprendizaje
- Tiempo de evaluación

y para cada uno de ellos se ha tomado la media y la desviación típica de las diez particiones, que serán los datos con los que trabajaremos. Se puede ver parte de una versión condensada de esta tabla de datos en el Apéndice A.

4.3.5.1. Tests estadísticos

La comparación entre métodos de selección de características se realiza habitualmente analizando diversas medidas de su rendimiento. Entre las medidas consideradas están: el porcentaje de acierto que obtenemos en la clasificación, como medida del éxito del proceso de aprendizaje; el tiempo o el espacio de memoria usado, para medir el uso de recursos que requiere el algoritmo; y el número de características seleccionadas, que nos aporta una idea de la reducción de la complejidad y la posible mejora de la interpretabilidad de las soluciones alcanzadas por los algoritmos de aprendizaje.

Para saber si las diferencias en las medidas de los resultados son significativas, aplicaremos los test estadísticos que describimos a continuación. El que una diferencia sea significativa nos garantizará que la diferencia apreciada entre los métodos se debe a un mejor o peor funcionamiento de uno de ellos, y no a otros factores aleatorios que puedan aparecer en el experimento.

En primer lugar, tendremos en cuenta que nos encontramos ante datos pareados. En cada experimento, el conjunto de datos tiene una influencia muy fuerte y, por tanto, nos interesa comparar los resultados obtenidos agrupados por conjuntos de datos. Podríamos usar el test de Student (t -test) sobre la diferencia media entre

dos métodos en estudio, pero este test tiene varios inconvenientes. En nuestro caso, el principal inconveniente es que requiere que las medidas sean commensurables. Además, es sensible a valores fuera de ranking.

Pensamos que esto no es adecuado sobre dominios tan diferentes, influenciados por los conjuntos de datos, donde mientras que en unos es fácil alcanzar el 100 %, en otros es difícil superar el 70 % de aciertos en clasificación. Por otra parte, requiere que se cumpla la hipótesis de normalidad sobre las poblaciones en las que se va a aplicar si el tamaño de éstas es pequeño (< 30). Aunque en principio nuestra población tiene más de 30 conjuntos de datos, no nos distancia del límite en una cantidad importante y, además, no siempre dispondremos de valores para todos los conjuntos de datos, quedándonos en algunas ocasiones con sólo 20 valores por las limitaciones indicadas en el apartado 4.3.5. Todo esto, unido a que tenemos serias dudas sobre la corrección de suponer normalidad en las medidas planteadas (especialmente en el caso del porcentaje de clasificación), nos hace descartar el t -test y optar por tests no paramétricos.

En vez de usar la medida que se compara directamente, es posible contar el número de veces que un método supera a otro y la posición relativa en que quedan los resultados. De esta forma, se elimina el problema de las medidas no commensurables y la necesidad de normalidad. Esto es lo que aplica el test de los rankings con signo de Wilcoxon, que será el utilizado cuando se quieran comparar únicamente dos métodos.

Sin embargo, al hacer una comparación con un test estadístico, como el de Wilcoxon, con un nivel de significación α tenemos una probabilidad α de cometer un error de Tipo 1 (descartar la hipótesis nula H_0 cuando realmente es cierta). Cuando se hacen varias comparaciones, la probabilidad de cometer al menos un error en una de ellas aumenta $(1 - (1 - \alpha)^n)$. Esto es lo que se conoce como error de todo el experimento en conjunto (*experimentwise error*). Al hacer muchas comparaciones, la probabilidad de error puede ser tan alta que el uso de los test estadísticos individuales deje de tener sentido.

El test Chi-cuadrado de Friedman nos permite hacer una comparación de varios métodos simultáneamente. Como hipótesis nula consideraremos que todos los métodos se comportan igual. Si ésta es rechazada, sabremos que alguno de los métodos se comporta de forma significativamente distinta respecto a la medida considerada. Se basa en el estadístico de la expresión (4.3), donde k es el número de grupos a evaluar, n el número de muestras por grupo y R_i^2 es la suma de los rankings del grupo i . A partir de valores razonables de $k(5)$ y $n(10)$, éste se puede estimar por una distribución χ^2 con $k - 1$ grados de libertad.

$$\chi_r^2 = \frac{12}{nk(k+1)} \sum_{i=1}^k R_i^2 - 3n(k+1) \quad (4.3)$$

El poder del test de Friedman, cuando se cumplen las condiciones de normalidad requeridas para aplicar el ANOVA paramétrico, es desde un 73% (para $k = 3$) hasta un 95% ($k = \infty$) del poder del ANOVA. Podemos ver entonces que, aunque se cumplieren las condiciones de normalidad necesarias, la elección del test de Friedman no representa una rebaja demasiado grande en poder de discriminación.

36 Muestras pareadas (conj. datos):	4 factores (medidas)							
	Liu		IEP		RSC		Inf	
	N°C.	Rank.	N°C.	Rank.	N°C.	Rank.	N°C.	Rank.
adult	13.00	(2.5)	12.80	(1.0)	13.00	(2.5)	13.40	(4.0)
anneal	9.20	(3.0)	9.10	(2.0)	8.20	(1.0)	30.50	(4.0)
audiology	12.50	(1.0)	12.70	(2.0)	15.00	(3.0)	50.20	(4.0)
balance-scale	4.00	(2.5)	4.00	(2.5)	4.00	(2.5)	4.00	(2.5)
breast-cancer	7.90	(1.0)	8.20	(2.0)	8.40	(4.0)	8.30	(3.0)
bupa	6.00	(2.5)	6.00	(2.5)	6.00	(2.5)	6.00	(2.5)
⋮	⋮		⋮		⋮		⋮	
yeast	3.00	(2.0)	3.00	(2.0)	3.00	(2.0)	5.80	(4.0)
zoo	4.90	(2.5)	4.90	(2.5)	5.10	(4.0)	1.00	(1.0)
Averages	7.19 (2.750)		6.94 (1.917)		7.27 (2.750)		8.85 (2.583)	

Tabla 4.9: Ejemplo de aplicación de los tests estadísticos para comparar 4 medidas en reducción de características

Iman y Davenport mostraron que el estadístico de la expresión (4.4) sigue una distribución F de Snedecor con grados de libertad $k - 1$ y $(k - 1)(n - 1)$, y es más poderoso que el test de Friedman original. Por tanto, usaremos esta extensión en lugar del de Friedman.

$$F_F = \frac{(n-1)\chi_r^2}{n(k-1) - \chi_r^2} \quad (4.4)$$

Una vez que se ha hecho la comparación de todos los métodos juntos y ha salido que al menos uno se comporta de forma significativamente diferente, nos interesa saber concretamente cuáles de ellos se comportan significativamente mejor que los otros. Esto se resuelve con los métodos de comparación múltiple, entre ellos hemos elegido el test de Nemenyi (versión no paramétrica del método Tukey). El estadístico

usado se muestra en la expresión (4.5). Éste sigue la distribución Q “Studentized range statistic”.

$$\frac{\bar{R}_B - \bar{R}_A}{\sqrt{\frac{k(k+1)}{12n}}} = q_{\alpha, \infty, k} \quad (4.5)$$

Con este test podemos comparar cada uno de los algoritmos del grupo con el resto. Para ello, calcularemos la distancia crítica (DC) según la expresión (4.6) y consideraremos significativas las diferencias entre los métodos cuyos rankings medios difieran en una cantidad mayor a DC .

$$DC = q_{\alpha, \infty, k} \cdot \sqrt{\frac{k(k+1)}{12n}} \quad (4.6)$$

En la tabla 4.9, se muestra un ejemplo de la aplicación de estos tests. Se realiza la comparación de 4 medidas de relevancia de conjuntos cuando se usan en combinación con el método de búsqueda *SFS*. El criterio de comparación usado en este ejemplo es el número de características seleccionadas, que consideramos mejor cuanto más reducido sea. Para cada conjunto de datos, se toma el número de características seleccionadas, se comparan ordenándolos y se les asigna el valor de su ranking, indicado entre paréntesis. A modo de orientación, en la última fila se indican los valores medios del número de características y el ranking de cada medida.

En el ejemplo, el valor del estadístico de Friedman es 14,6295, lo que nos da una probabilidad de 0,0022 de que la hipótesis nula (que todos las medidas den resultados similares) sea cierta. La aplicación del test de Iman-Davenport, con valor de estadístico 5,4839 y probabilidad 0,0015, nos da una seguridad mayor de que al menos una medida tiene un comportamiento diferente a las demás. Para extraer mayor información, aplicaremos el método de Nemenyi en un nivel de significación del 0,05. Éste nos indica que una distancia en el valor medio del ranking superior a 0,7817 es significativa. Por tanto, podemos decir con una seguridad del 95 % que *IEP* reduce más características que *Liu* o *RSC*, y no podemos saber si reduce más que *Inf* o si la diferencia observada es debida a algún factor aleatorio.

La metodología empleada ha sido recomendada por Demsar [Dem06] para la comparación de algoritmos de clasificación. En la descripción realizada aquí, de los métodos no se incluye la gestión de los empates en los rankings, que sí ha sido implementada y tenida en cuenta. Para una explicación con mayor detalle sobre los

métodos estadísticos empleados puede consultarse [Zar99], o en español [AdLdC94] (sólo algunos de los métodos empleados).

4.4. Análisis de resultados

Los experimentos descritos en las secciones precedentes suponen una importante carga de trabajo. La cantidad de resultados también es elevada y un análisis correcto es muy importante.

Para exponer de forma más clara los resultados los vamos a presentar agrupados según el siguiente esquema:

- Resultados correspondientes a métodos que evalúan subconjuntos
 - Agrupados por medidas
 - Agrupados por métodos de búsqueda
 - Globales
- Resultados correspondientes a métodos que evalúan características
 - Agrupados por medidas
 - Agrupados por criterios de corte
 - Globales
- Comparativa entre métodos basados en subconjuntos y métodos basados en valoración de características individuales

Descripción de las gráficas y tablas de comparación incluidas en este capítulo. Hemos usado como medidas de eficacia, el porcentaje promedio de aciertos que resulta de aplicar validación cruzada a los clasificadores obtenidos con cuatro modelos de clasificación distintos, y número de características empleadas. En las primeras tablas nos proponemos comparar las medidas y los métodos de búsqueda. Considerando que hemos empleado 36 problemas y hay unos 10 métodos de búsqueda, las tablas de datos resultantes tendrían 360 líneas. Para mostrar los datos de forma resumida, en cada comparación, hemos elaborado una única tabla que recoge los rankings promedio que cada método de búsqueda ha obtenido para cada clasificador. Para una visualización más clara y rápida de los resultados, junto con la tabla, se muestran dos gráficas reflejando los mismos (ver como ejemplo la tabla 4.10).

En las gráficas, el eje X mide la posición en que queda cada método respecto a los demás. Aparecen los métodos o medidas ordenados según el ranking promedio de su aplicación en todos los conjuntos de datos.

En la gráfica superior se muestran cuatro rankings referidos al porcentaje de acierto en clasificación, uno para cada modelo de clasificación empleado. De esta forma, podemos ver el efecto diferente de la selección de características según el clasificador y, además, nos aseguramos que las muestras son independientes para la aplicación de los tests estadísticos.

En la gráfica inferior, se muestra el ranking sobre la reducción de características. Aparece únicamente un ranking que se refiere al número de características seleccionadas para el primero de los modelos de clasificación ($C45$). Se muestra sólo este porque en la mayoría de las gráficas (todas menos las que incluyen la medida Wra) los resultados son muy similares*. Las variaciones de la medida envolvente (Wra) se pueden observar fácilmente mirando su fila o columna en la tabla. El resto sólo varían por influencia de la variación del puesto en el ranking de Wra .

En la tabla con los resultados numéricos se muestra el ranking en aciertos ($R.Ac.$) y el ranking sobre el número de características seleccionadas ($R.n^o c.$). En ambos casos, un ranking más cercano a 1 es una mejor solución: un mayor porcentaje de acierto, o un número menor de características (mayor reducción).

Debajo de la tabla se muestra el valor de la distancia crítica según el test de Nemenyi. Esta distancia crítica se refleja también en la gráfica en un recuadro rojo. Sólo se indica para el mejor método para no emborronar la gráfica, pero es fácilmente extrapolable a las diferencia entre los demás métodos que aparecen en la gráfica y, así, ver si son significativamente distintos.

*Serían idénticos si en los métodos no determinísticos hubiésemos usado exactamente la misma selección para todos los modelos de clasificación en vez de repetir el proceso (necesario sólo para Wra). Cada experimento ha usado una semilla para el generador aleatorio, pero se ha repetido para los cuatro clasificadores uno detrás de otro. De esta forma, el experimento es reproducible y ha usado las mismas particiones de los datos, pero la selección de características puede variar.

4.4.1. Métodos que evalúan subconjuntos

4.4.1.1. Resultados agrupados por medidas

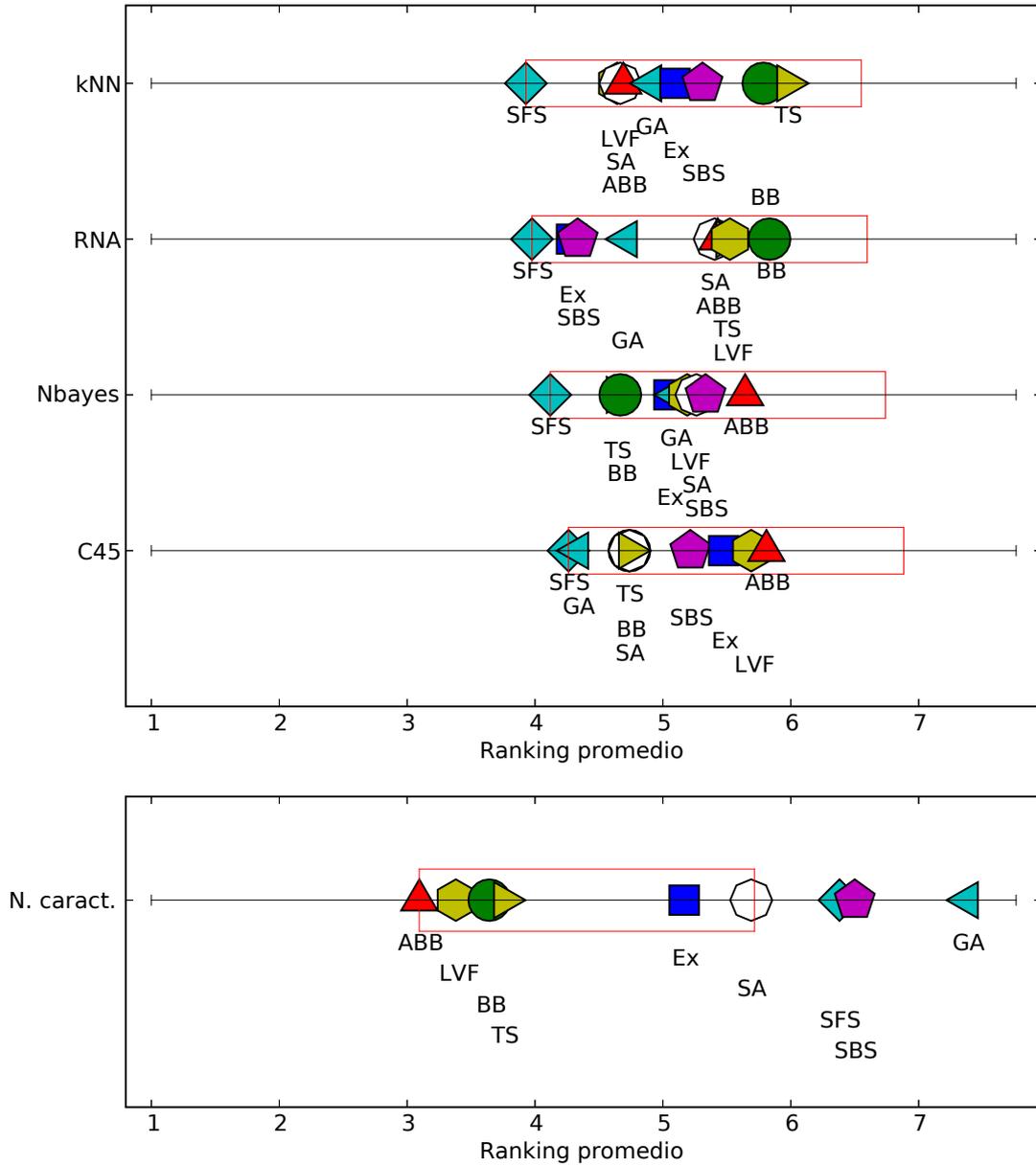
Con estas comparaciones pretendemos evaluar la eficacia relativa de los distintos métodos de búsqueda. Realizamos las comparaciones por cada medida de relevancia, con la idea de separar los efectos que tienen las medidas en la búsqueda y estudiarlos en cada método.

Ejemplos inconsistentes (Liu). Como se puede observar en la tabla 4.10, sobre el porcentaje de clasificación no se han detectado diferencias significativas entre los métodos de búsqueda. En cambio, sí podemos observar cómo SFS, SBS y GA son peores en reducción de características. LVF y TS obtienen buenas soluciones en ambos objetivos.

Consistencia de Rough Sets (RSC). Los resultados son similares a Liu, aunque la reducción en características, en este caso, es más similar entre los diversos métodos de búsqueda. Es posible que se deba a que esta medida es más conservadora a la hora de considerar que un conjunto de características sirve para clasificar correctamente.

Pares de ejemplos inconsistentes (IEP). En este caso ABB y LVF destacan por una mayor reducción del número de características empleadas. Desafortunadamente, esto va acompañado por unos resultados significativamente peores en porcentaje de clasificación. Sin embargo, es interesante observar que el método que obtiene mejores resultados con todos los clasificadores, SFS, no es el peor en reducción y no hay diferencia significativa salvo con los dos primeros en reducción.

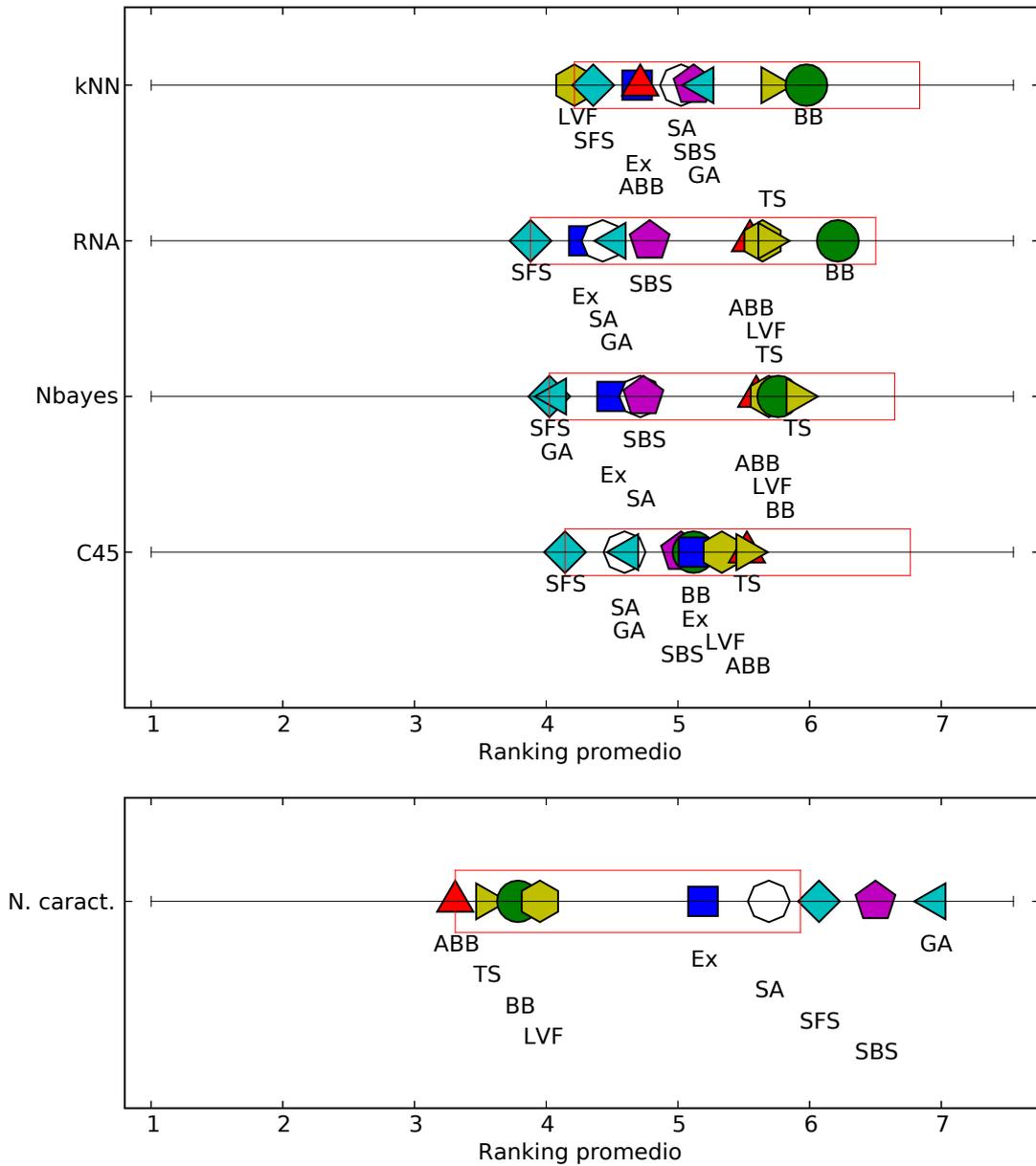
Tanto ABB como LVF permiten reducciones en la medida, de esta forma pueden terminar encontrando conjuntos de características más pequeños, pero no completamente consistentes. SFS ha guiado bien la búsqueda hasta un conjunto consistente (no escoge demasiadas características innecesarias). Podemos concluir que la medida IEP es buena para guiar la búsqueda creciente en características, pero no parece buena para encontrar conjuntos que obtengan buenos resultados sin ser consistentes.



Método de búsqueda	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Ex	5.48	5.17	5.05	5.12	4.29	5.21	5.10	5.21
BB	4.74	3.64	4.67	3.69	5.83	3.76	5.79	3.62
ABB	5.81	3.10	5.64	3.05	5.43	3.07	4.69	3.14
SFS	4.26	6.38	4.12	6.24	3.98	6.45	3.93	6.38
SBS	5.21	6.50	5.33	6.36	4.33	6.52	5.31	6.45
LVF	5.69	3.38	5.19	3.31	5.52	3.33	4.64	3.50
SA	4.74	5.69	5.26	5.69	5.40	5.81	4.67	5.67
GA	4.33	7.38	5.10	7.76	4.71	7.02	4.90	7.02
TS	4.74	3.76	4.64	3.79	5.50	3.81	5.98	4.00

Nemenyi critical distance (p=0.05): 2.62173395457

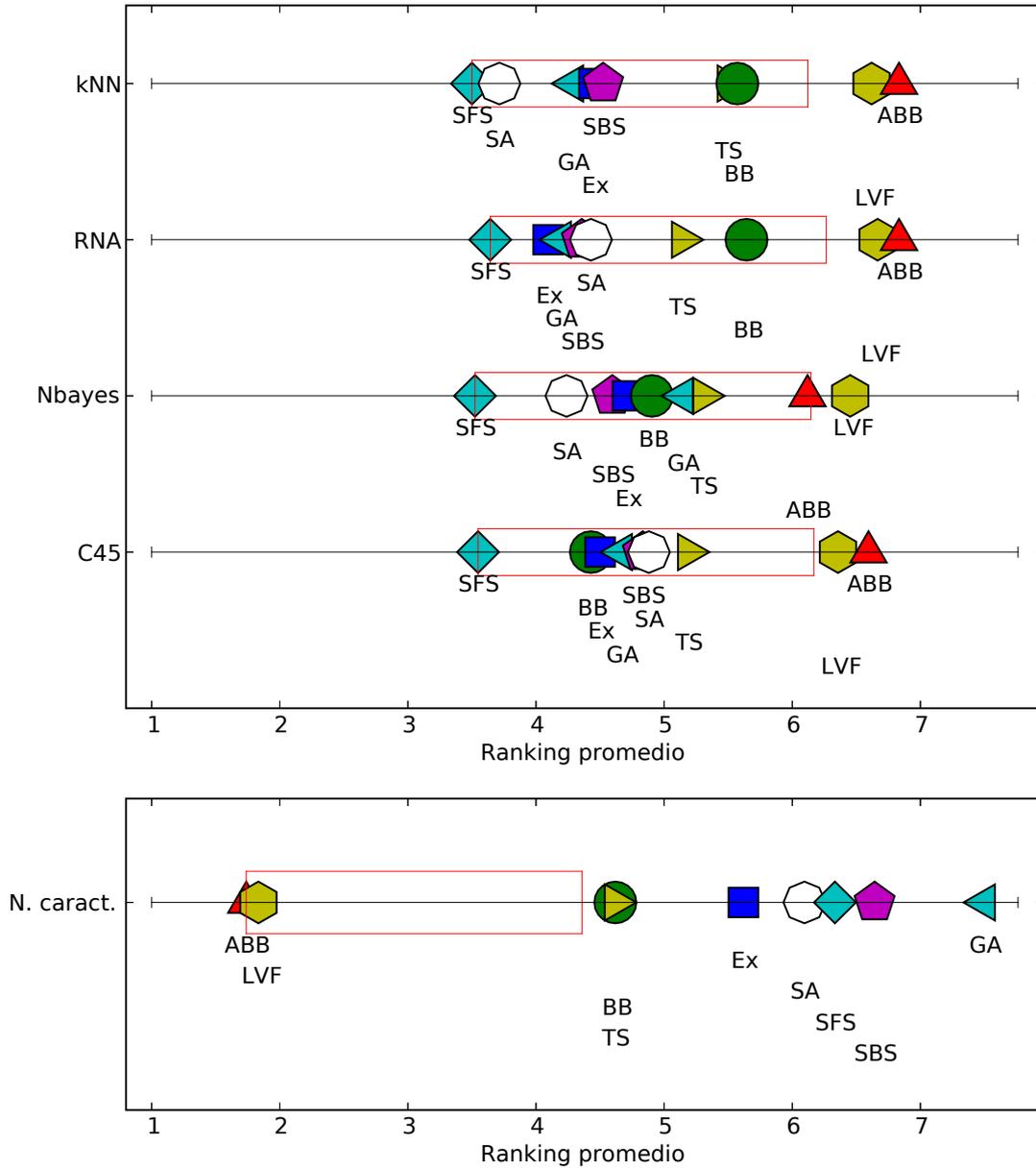
Tabla 4.10: Comparación de métodos de búsqueda para la medida Liu



Método de búsqueda	C45		Nbayes		RNA		knn	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
Ex	5.12	5.19	4.50	5.12	4.29	5.14	4.69	5.05
BB	5.12	3.79	5.76	3.67	6.21	3.57	5.98	3.69
ABB	5.52	3.31	5.60	3.43	5.55	3.33	4.71	3.33
SFS	4.14	6.07	4.02	6.10	3.88	6.12	4.36	5.90
SBS	5.02	6.50	4.74	6.50	4.79	6.45	5.12	6.36
LVF	5.33	3.95	5.69	4.00	5.64	3.90	4.21	3.86
SA	4.60	5.69	4.71	5.60	4.43	5.64	5.02	5.79
GA	4.62	6.95	4.07	6.98	4.52	6.93	5.19	7.55
TS	5.52	3.55	5.90	3.62	5.69	3.90	5.71	3.48

Nemenyi critical distance (p=0.05): 2.62173395457

Tabla 4.11: Comparación de métodos de búsqueda para la medida RSC



Método de búsqueda	C45		Nbayes		RNA		knn	
	R.Ac.	R.nºc.	R.Ac.	R.nºc.	R.Ac.	R.nºc.	R.Ac.	R.nºc.
Ex	4.50	5.62	4.71	5.48	4.10	5.57	4.45	5.55
BB	4.43	4.62	4.90	4.62	5.64	4.64	5.57	4.52
ABB	6.60	1.74	6.12	1.79	6.83	1.74	6.83	1.79
SFS	3.55	6.33	3.52	6.19	3.64	6.29	3.50	6.17
SBS	4.83	6.64	4.60	6.57	4.36	6.62	4.52	6.60
LVF	6.36	1.83	6.45	1.83	6.67	1.79	6.62	1.83
SA	4.88	6.10	4.24	6.12	4.43	6.26	3.71	6.00
GA	4.67	7.50	5.14	7.76	4.19	7.55	4.29	7.52
TS	5.19	4.62	5.31	4.64	5.14	4.55	5.50	5.02

Nemenyi critical distance (p=0.05): 2.62173395457

Tabla 4.12: Comparación de métodos de búsqueda para la medida IEP

Tras analizar los resultados obtenidos por todas las medidas de consistencia podemos concluir que:

- En general, no se detectan diferencias significativas estadísticamente entre la eficacia de los métodos para ninguno de los cuatro clasificadores. Dentro de esta falta de diferencias sí se observa un liderazgo claro del SFS, que además por su carácter voraz** es más simple y eficiente. Resulta curioso que este método de búsqueda preceda a otros que, en teoría, son mucho más potentes, como por ejemplo los algoritmos genéticos.
- Sí se detectan diferencias significativas en la reducción del número de características. ABB suele encabezar la lista de los métodos que más reducen.
- Se observa una contraposición entre los dos principales objetivos en Selección de características: obtener un buen nivel de clasificación y una reducción de características elevada. Los métodos que dan mejores resultados para uno de los objetivos dan peores resultados para el otro.

Información mutua (Inf). Con esta medida, otra vez, SFS queda el primero en resultados de clasificación. Sin embargo, no se detecta una diferencia que pueda considerarse significativa con los demás, pero tampoco la hay en reducción de características donde no queda mal posicionado.

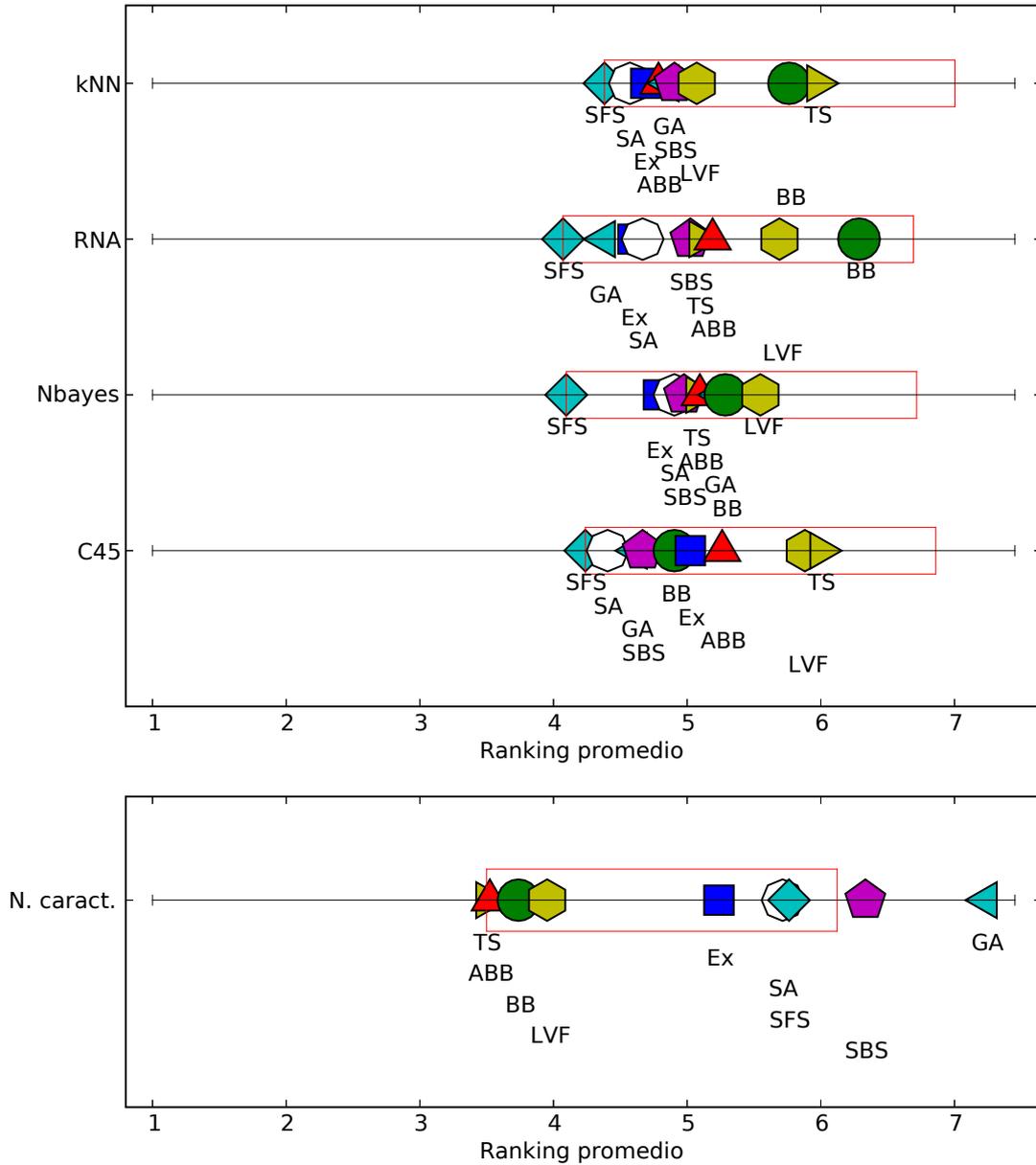
ABB resulta ser la opción más adecuada para reducir al máximo el número de características sin perder demasiado en aciertos de clasificación.

Incertidumbre simétrica (SU). Esta es la medida en que se observa menor diferencia entre los métodos de búsqueda, ya que todos se encuentran dentro de la mitad de la distancia crítica. No hay métodos que destaquen, todos se comportan igual de bien o mal.

El hecho de que la búsqueda exhaustiva (Ex) sea el método que más reduce nos indica que la medida SU da valores más altos a conjuntos con menos características, lo que concuerda con su definición.

Longitud mínima de la descripción (MDL). La combinación de esta medida con TS y LVW produce reducciones de características superiores a los demás

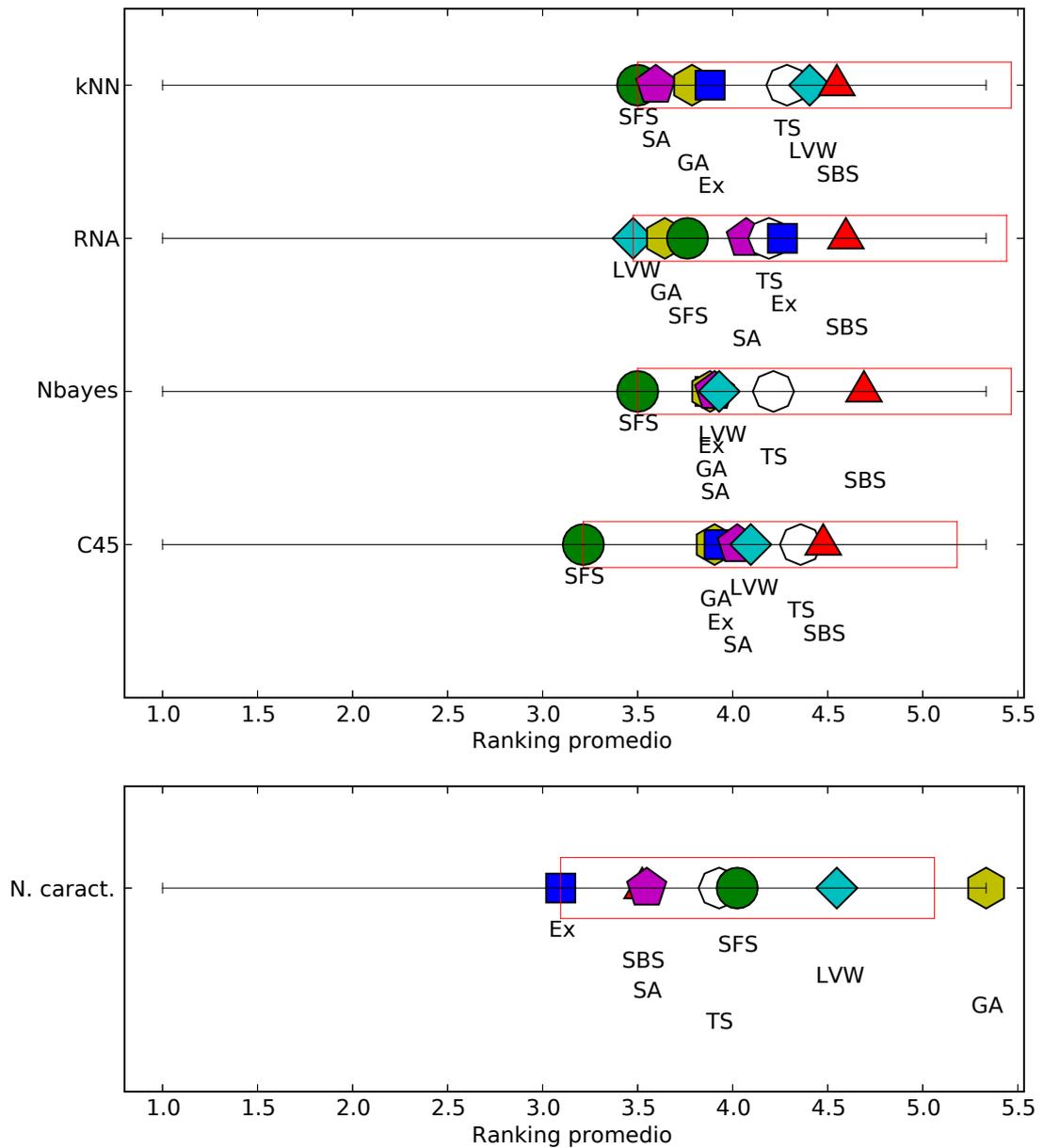
** *greedy*, en inglés.



Método de búsqueda	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Ex	5.02	5.24	4.79	5.17	4.60	5.31	4.69	5.19
BB	4.90	3.74	5.29	3.62	6.29	3.60	5.76	3.81
ABB	5.26	3.52	5.10	3.38	5.19	3.48	4.79	3.40
SFS	4.24	5.76	4.10	5.60	4.07	5.67	4.38	5.64
SBS	4.67	6.33	4.98	6.26	5.02	6.43	4.90	6.33
LVF	5.88	3.95	5.55	3.86	5.69	4.02	5.07	3.90
SA	4.40	5.71	4.90	5.69	4.67	5.76	4.57	5.76
GA	4.62	7.24	5.24	7.45	4.38	7.26	4.86	7.38
TS	6.00	3.50	5.07	3.98	5.10	3.48	5.98	3.57

Nemenyi critical distance (p=0.05): 2.62173395457

Tabla 4.13: Comparación de métodos de búsqueda para la medida Inf



Método de búsqueda	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Ex	3.93	3.10	3.88	3.05	4.26	3.05	3.88	2.93
SFS	3.21	4.02	3.50	4.00	3.76	3.98	3.50	3.93
SBS	4.48	3.52	4.69	3.52	4.60	3.48	4.55	3.38
LVW	4.10	4.55	3.93	4.69	3.48	4.88	4.40	4.62
SA	4.02	3.55	3.90	3.60	4.07	3.67	3.60	3.60
GA	3.90	5.33	3.88	4.79	3.64	4.76	3.79	4.79
TS	4.36	3.93	4.21	4.36	4.19	4.19	4.29	4.76

Nemenyi critical distance (p=0.05): 1.9657568517

Tabla 4.14: Comparación de métodos de búsqueda para la medida SU

métodos de búsqueda. Sin embargo, pierden en calidad de resultados.

En relación con las medidas basadas en Teoría de la información siguen siendo válidas todas las conclusiones establecidas para las medidas basadas en consistencia.

Medida de conjuntos basada en Relief (RFS). Es interesante ver que, pese a no apreciarse gran diferencia entre los métodos con tres clasificadores, al usar redes neuronales (RNA) las diferencias son más importantes, destacando GA y SA.

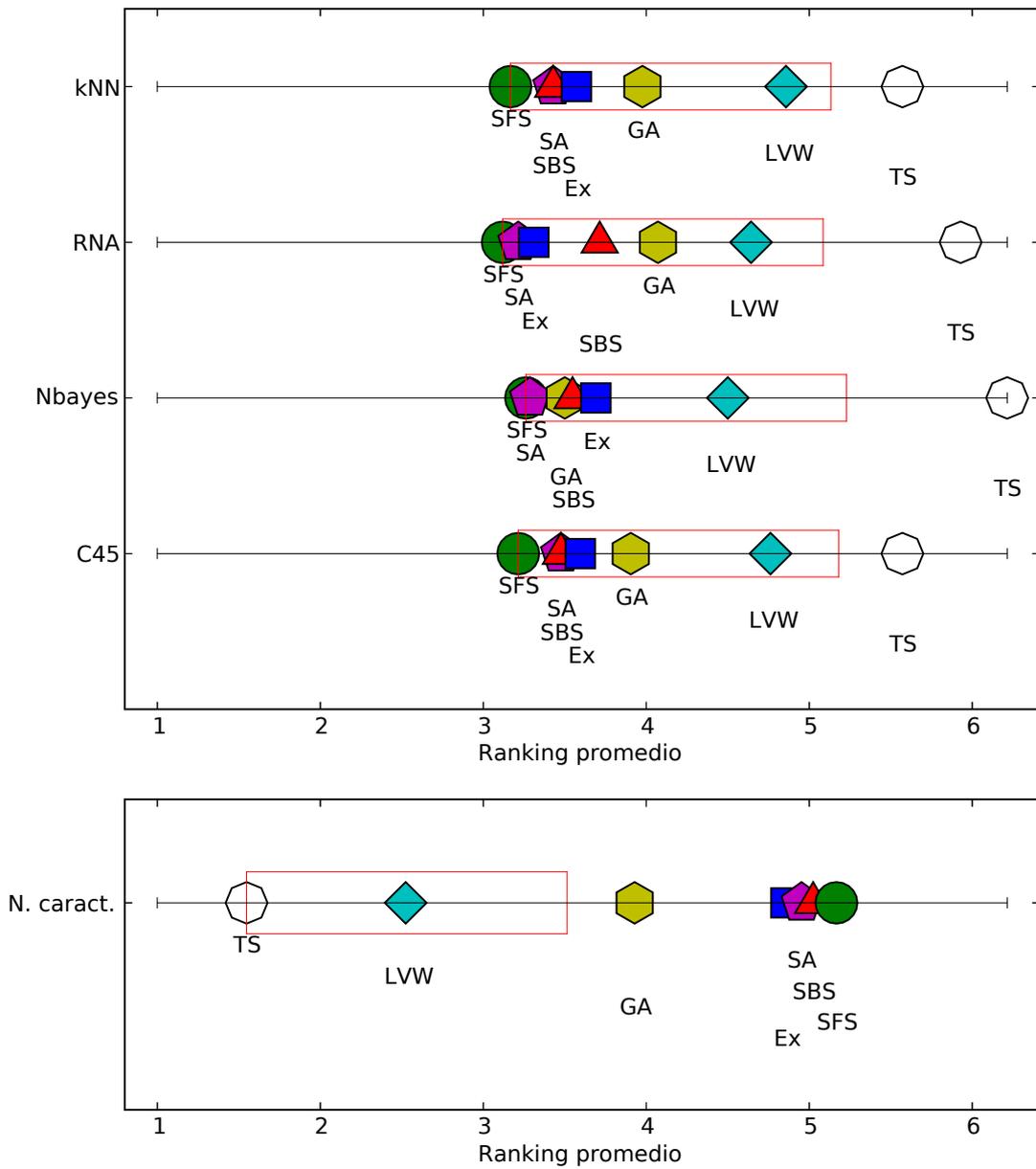
Las diferencias en ranking de ejecución entre los diversos métodos de búsqueda son bastante variables. Lo que nos indica que los métodos se comportan de formas bastante diferentes entre ellos. El hecho de que la búsqueda Ex obtenga un ranking medio indica que RFS valora mejor algunos conjuntos que reducen que otros que tienen más características.

Envolvente (Wra). Se aprecia una mayor diferencia de los resultados entre clasificadores como es lógico, ya que esta medida se adapta al clasificador. El hecho de que Ex no obtenga en todos los casos el mejor resultado puede ser debido a que se produzca sobreaprendizaje. Lo cual nos lleva a pensar que es posible que en muchos casos no merezca la pena usar esta medida frente a otras menos costosas que pueden dar aproximaciones similares.

Destaca el hecho de que el método SBS obtiene mejores puestos que los que obtenía con otras medidas, tanto en resultados de clasificación como en reducción de características. La explicación puede ser que es más fácil acertar el camino correcto quitando características cuando la valoración es el método que da el resultado. En cambio si intentamos ir averiguando el camino añadiendo características, las interacciones entre características pueden hacer que la medida envolvente haga que la búsqueda se pierda, como veíamos en el ejemplo del apartado 3.4.

4.4.1.2. Resultados agrupados por métodos de búsqueda

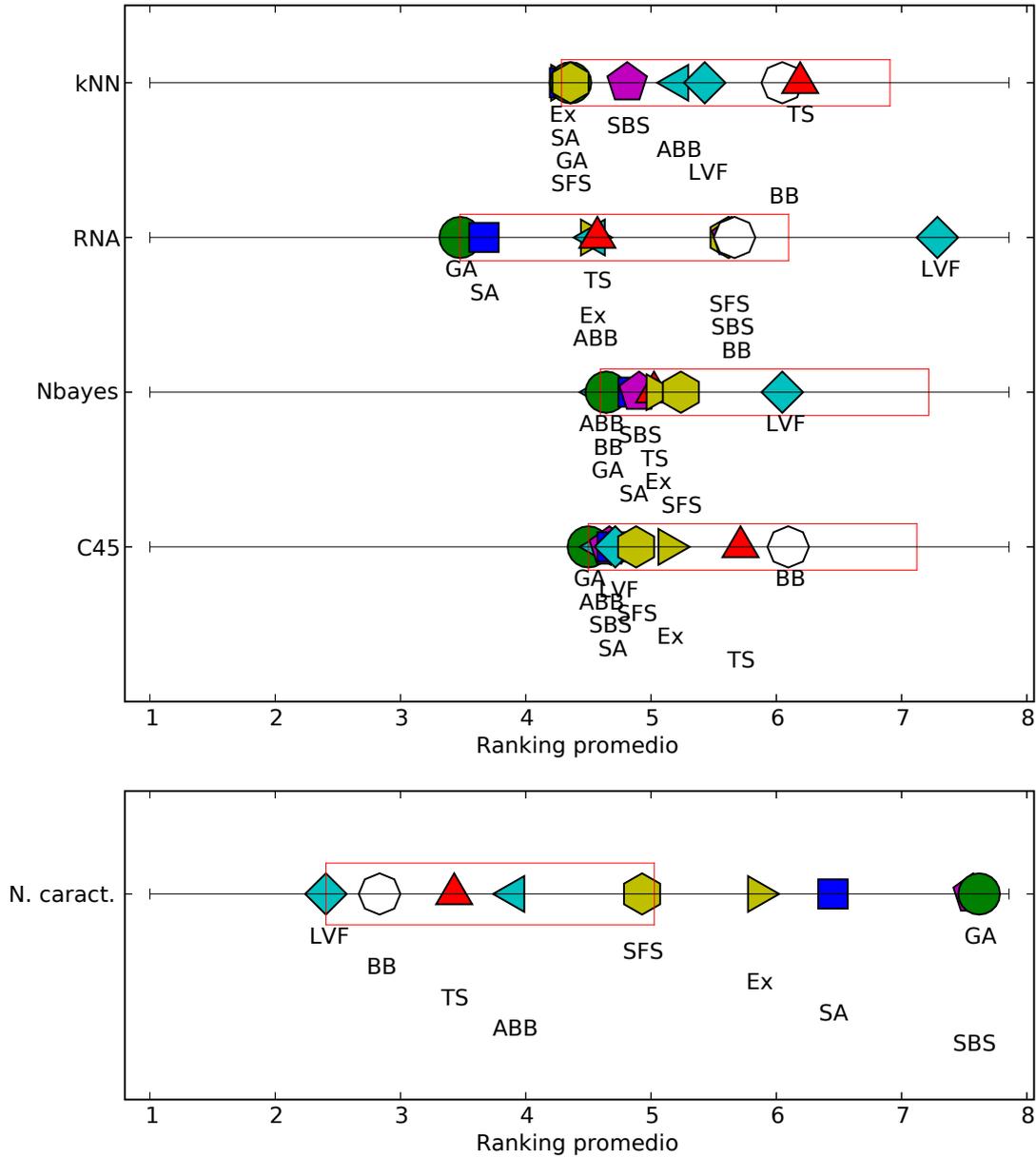
De forma similar a como llevamos a cabo el estudio de la eficacia relativa a los métodos de búsqueda, estudiamos ahora los méritos de las distintas medidas de evaluación, presentando los resultados agrupados por métodos de búsqueda.



Método de búsqueda	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Ex	3.60	4.86	3.69	4.93	3.31	4.95	3.57	4.81
SFS	3.21	5.17	3.26	5.31	3.12	5.29	3.17	5.17
SBS	3.48	5.02	3.55	5.10	3.71	5.07	3.43	4.93
LVW	4.76	2.52	4.50	2.48	4.64	2.71	4.86	2.76
SA	3.48	4.95	3.29	5.14	3.21	5.02	3.43	5.07
GA	3.90	3.93	3.50	3.57	4.07	3.29	3.98	3.48
TS	5.57	1.55	6.21	1.48	5.93	1.67	5.57	1.79

Nemenyi critical distance ($p=0.05$): 1.9657568517

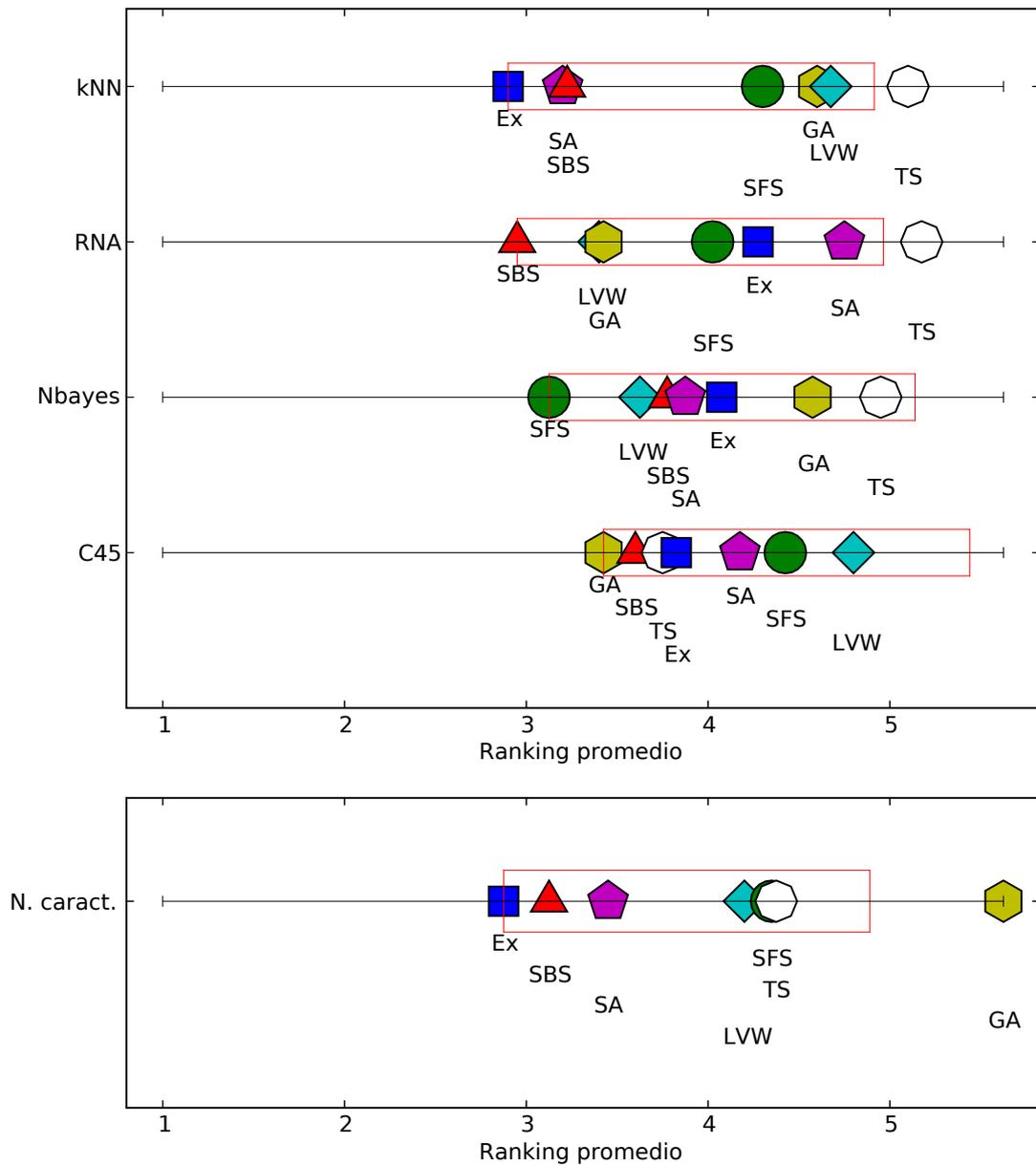
Tabla 4.15: Comparación de métodos de búsqueda para la medida MDL



Método de búsqueda	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
SA	4.69	6.45	4.86	6.36	3.67	6.79	4.31	6.48
GA	4.50	7.62	4.64	7.48	3.48	7.48	4.36	6.95
TS	5.71	3.43	5.02	3.45	4.57	3.40	6.19	3.17
LVF	4.71	2.40	6.05	2.19	7.29	2.21	5.43	2.21
SBS	4.67	7.57	4.90	7.86	5.64	7.55	4.81	7.57
SFS	4.88	4.93	5.24	4.52	5.62	5.31	4.36	6.00
BB	6.10	2.83	4.64	2.93	5.67	2.81	6.05	2.76
ABB	4.60	3.90	4.60	4.33	4.55	3.93	5.21	4.00
Ex	5.14	5.86	5.05	5.88	4.52	5.52	4.29	5.86

Nemenyi critical distance (p=0.05): 2.62173395457

Tabla 4.16: Comparación de métodos de búsqueda para la medida RFS



Método de búsqueda	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Ex	3.83	2.88	4.08	3.58	4.28	3.40	2.90	2.90
SFS	4.42	4.35	3.12	3.15	4.03	5.08	4.30	4.45
SBS	3.60	3.12	3.77	3.83	2.95	3.77	3.23	4.50
LVW	4.80	4.20	3.62	4.17	3.40	4.05	4.67	4.35
SA	4.17	3.45	3.88	4.00	4.75	3.90	3.20	3.62
GA	3.42	5.62	4.58	5.05	3.42	4.55	4.60	4.58
TS	3.75	4.38	4.95	4.22	5.17	3.25	5.10	3.60

Nemenyi critical distance (p=0.05): 2.01430136772

Tabla 4.17: Comparación de métodos de búsqueda para la medida Wra

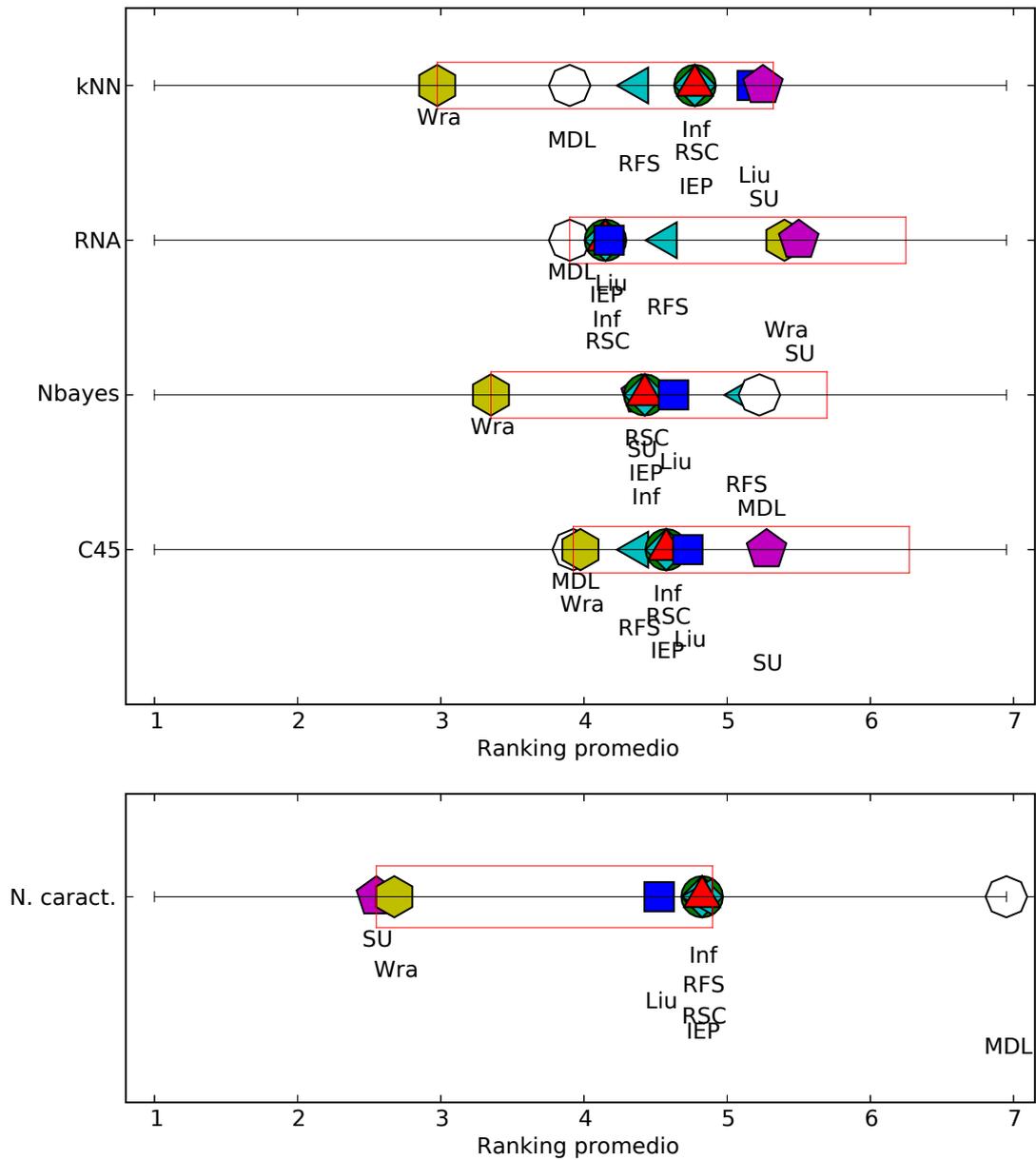
Búsqueda exhaustiva (Ex). La búsqueda exhaustiva, al hacer una búsqueda completa eligiendo el mejor conjunto de características según la medida en cuestión, nos permite saber si el conjunto mejor valorado por cada medida es mejor que el mejor valorado de las demás. Sin embargo, aunque los resultados nos dan alguna pista, no hay diferencias significativas que nos determinen diferencias claras, excepto en el caso de la medida MDL cuya elección del mejor conjunto tiene un número de características superior al elegido por SU o Wra.

Ramificación y poda (BB). Este método fija un número de características para cada conjunto de datos independientemente de la medida usada. Por tanto, las medidas que dan buenos resultados son apropiadas para la elegir entre conjuntos con el mismo número de características. Aunque no hay diferencias muy significativas, destacan en resultados Liu, RFS e Inf.

ABB. Todas parecen comportarse de forma parecida, destacando quizá Inf para ser usada en combinación con redes neuronales, y Liu si para los mismos resultados se prefiere una mayor reducción de características. IEP todavía permite una mayor reducción, a costa de perder porcentaje de aciertos, aunque no hemos podido determinar que la pérdida sea significativa salvo para k NN.

En general, para los métodos de búsqueda completa no se detectan diferencias significativas estadísticamente para los resultados de clasificación, y sí para la reducción de características. No obstante, el grupo de indistinguibles con el mejor, sigue siendo el más numeroso. Se mantiene la contraposición entre los objetivos, con la excepción notable de la medida Wra para el método exhaustivo, donde produce los mejores resultados para ambos objetivos. Hecho que no se puede corroborar con los otros dos métodos, porque Wra no es una medida monótona y no se puede combinar con BB y ABB. Los buenos resultados el Wra se deben a que es la medida que utiliza más información del problema que se está resolviendo.

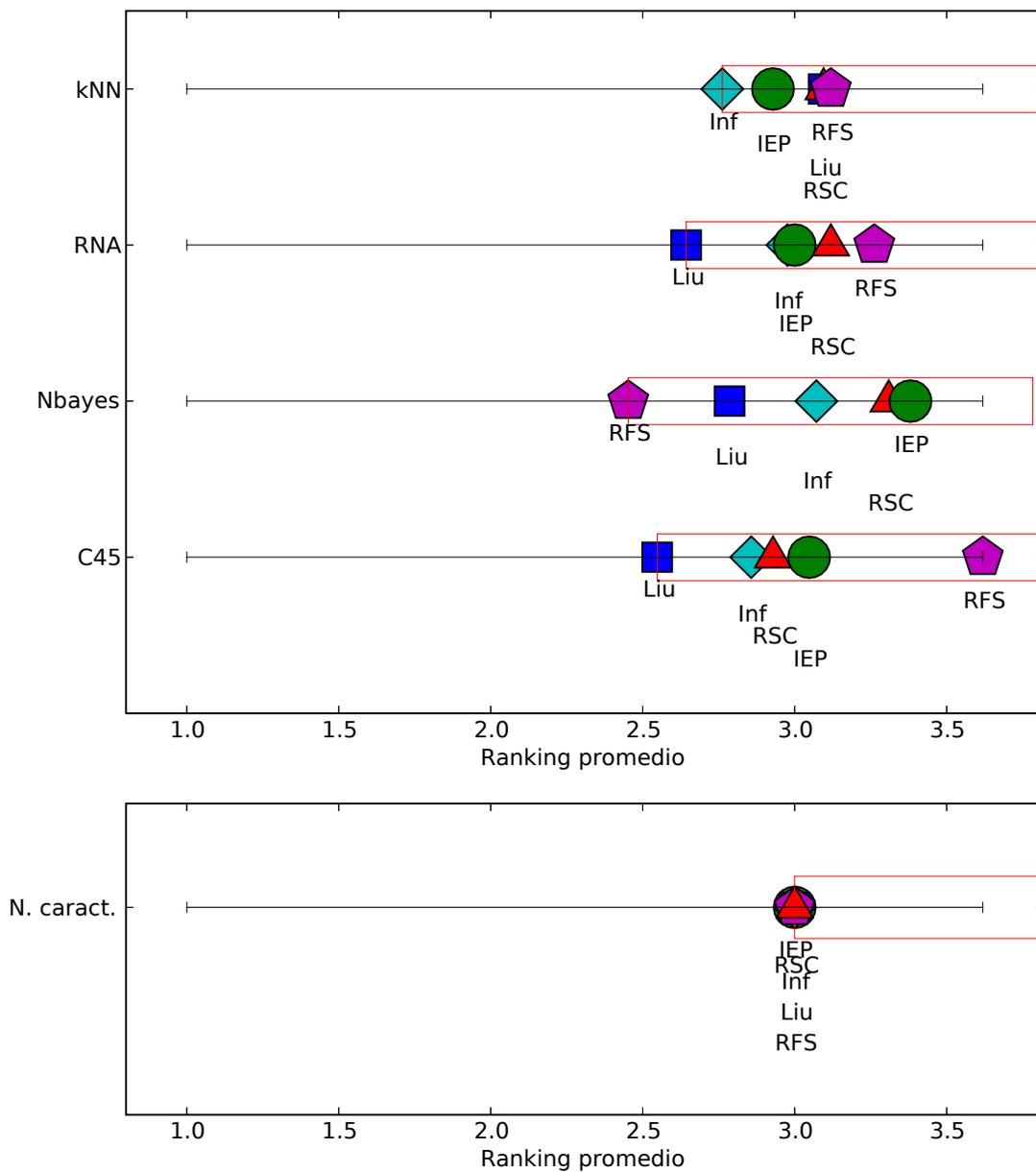
Ascensión de colinas (SFS). Destacan positivamente en aciertos Wra y MDL. MDL lo hace a costa de la más baja reducción de características. Mientras que Wra consigue los primeros puestos en reducciones. RSC, Liu, IEP e Inf consiguen también buenos resultados sin distinción significativa, destacando IEP con la mejor reducción entre ellos. SU consigue todavía mayor reducción, pero sus resultados se ven afectados empezando a ser significativamente peores dependiendo del clasificador.



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Liu	4.72	4.53	4.62	4.45	4.17	4.45	5.17	4.50
IEP	4.58	4.83	4.42	4.75	4.15	4.75	4.78	4.75
RSC	4.58	4.83	4.42	4.75	4.15	4.75	4.78	4.75
Inf	4.58	4.83	4.42	4.75	4.15	4.75	4.78	4.75
SU	5.28	2.55	4.40	2.50	5.50	2.60	5.25	2.50
Wra	3.98	2.67	3.35	3.17	5.40	3.02	2.98	3.23
MDL	3.92	6.95	5.22	6.90	3.90	6.88	3.90	6.88
RFS	4.38	4.83	5.12	4.72	4.58	4.80	4.38	4.65

Nemenyi critical distance (p=0.05): 2.34753888147

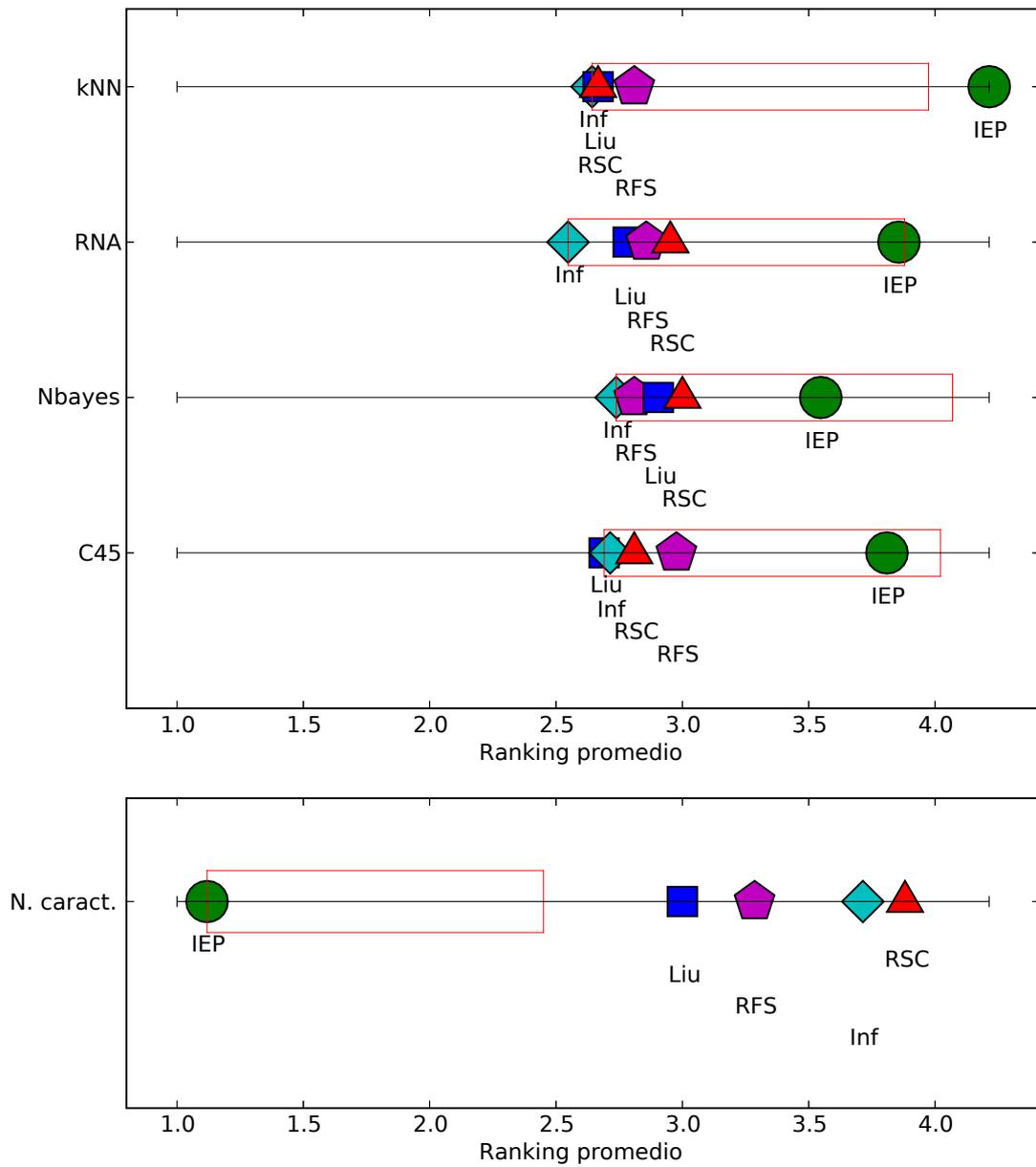
Tabla 4.18: Comparación de medidas para el método de búsqueda Ex



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
Liu	2.55	3.00	2.79	3.00	2.64	3.00	3.10	3.00
IEP	3.05	3.00	3.38	3.00	3.00	3.00	2.93	3.00
RSC	2.93	3.00	3.31	3.00	3.12	3.00	3.10	3.00
Inf	2.86	3.00	3.07	3.00	2.98	3.00	2.76	3.00
RFS	3.62	3.00	2.45	3.00	3.26	3.00	3.12	3.00

Nemenyi critical distance ($p=0.05$): 1.33113646397

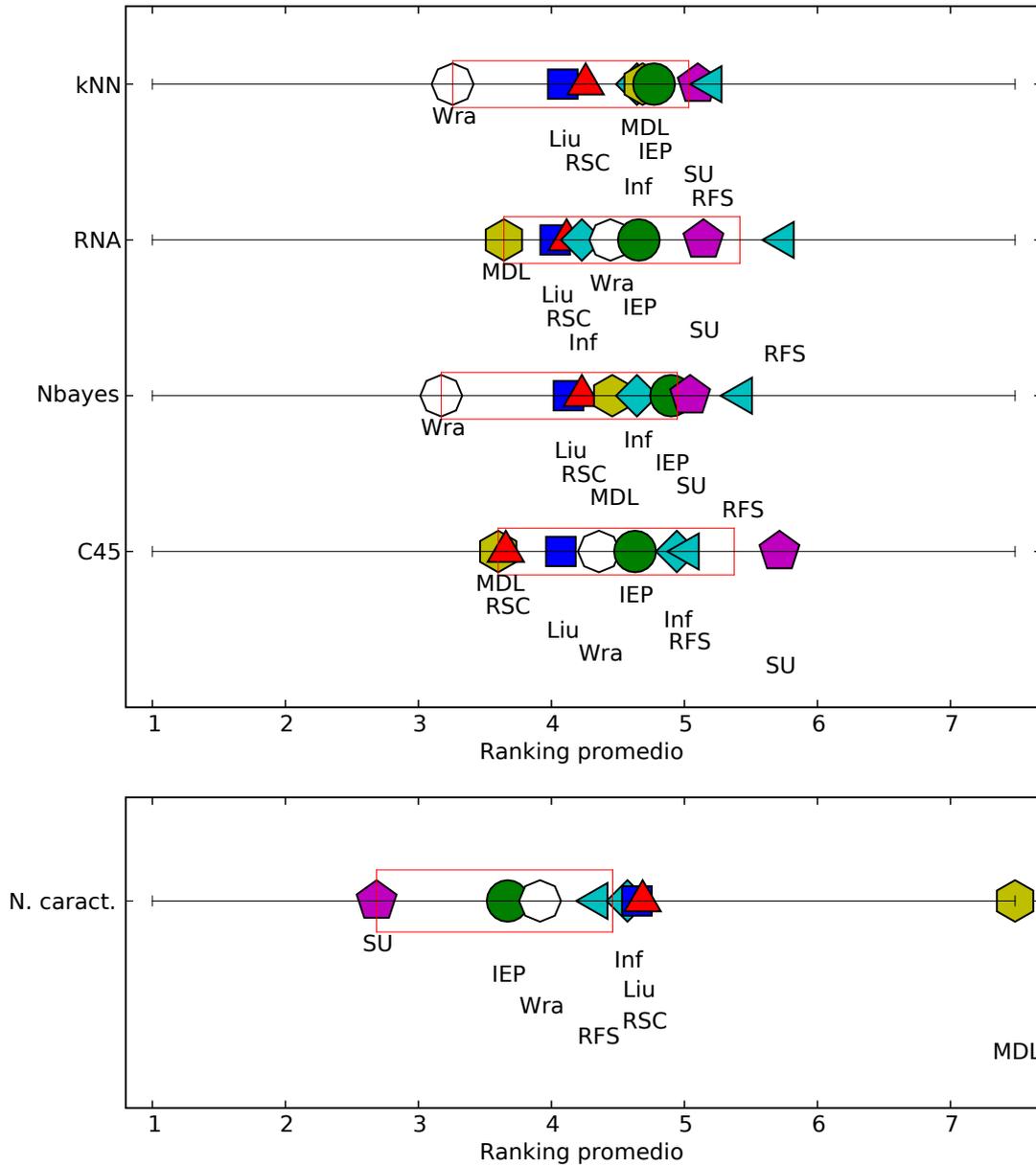
Tabla 4.19: Comparación de medidas para el método de búsqueda BB



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.nºc.	R.Ac.	R.nºc.	R.Ac.	R.nºc.	R.Ac.	R.nºc.
Liu	2.69	3.00	2.90	2.93	2.79	2.98	2.67	3.00
IEP	3.81	1.12	3.55	1.10	3.86	1.10	4.21	1.10
RSC	2.81	3.88	3.00	3.74	2.95	3.81	2.67	3.86
Inf	2.71	3.71	2.74	3.57	2.55	3.64	2.64	3.69
RFS	2.98	3.29	2.81	3.67	2.86	3.48	2.81	3.36

Nemenyi critical distance (p=0.05): 1.33113646397

Tabla 4.20: Comparación de medidas para el método de búsqueda ABB



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Liu	4.07	4.64	4.13	4.61	4.03	4.59	4.09	4.54
IEP	4.63	3.67	4.90	3.73	4.66	3.67	4.77	3.70
RSC	3.66	4.69	4.23	4.63	4.11	4.64	4.26	4.66
Inf	4.94	4.57	4.64	4.56	4.23	4.51	4.64	4.51
SU	5.71	2.69	5.04	2.80	5.14	2.77	5.10	2.69
MDL	3.60	7.49	4.46	7.46	3.64	7.43	4.69	7.44
Wra	4.36	3.91	3.17	3.89	4.44	4.06	3.26	4.07
RFS	5.03	4.34	5.43	4.33	5.74	4.33	5.20	4.39

Nemenyi critical distance (p=0.05): 1.7745725924

Tabla 4.21: Comparación de medidas para el método de búsqueda SFS

SBS En aciertos, destacan Wra y MDL con diferencias significativas respecto al resto de las medidas. Sin embargo, MDL falla en reducción de características, mientras que Wra consigue los segundos mejores resultados en reducción. Los demás métodos coinciden en resultados de los dos tipos, excepto SU que consigue la mayor reducción. Es el primer método para el que los métodos estadísticamente no distinguibles del mejor ya no son la mayoría, sino que el grupo “de cabeza” es más reducido.

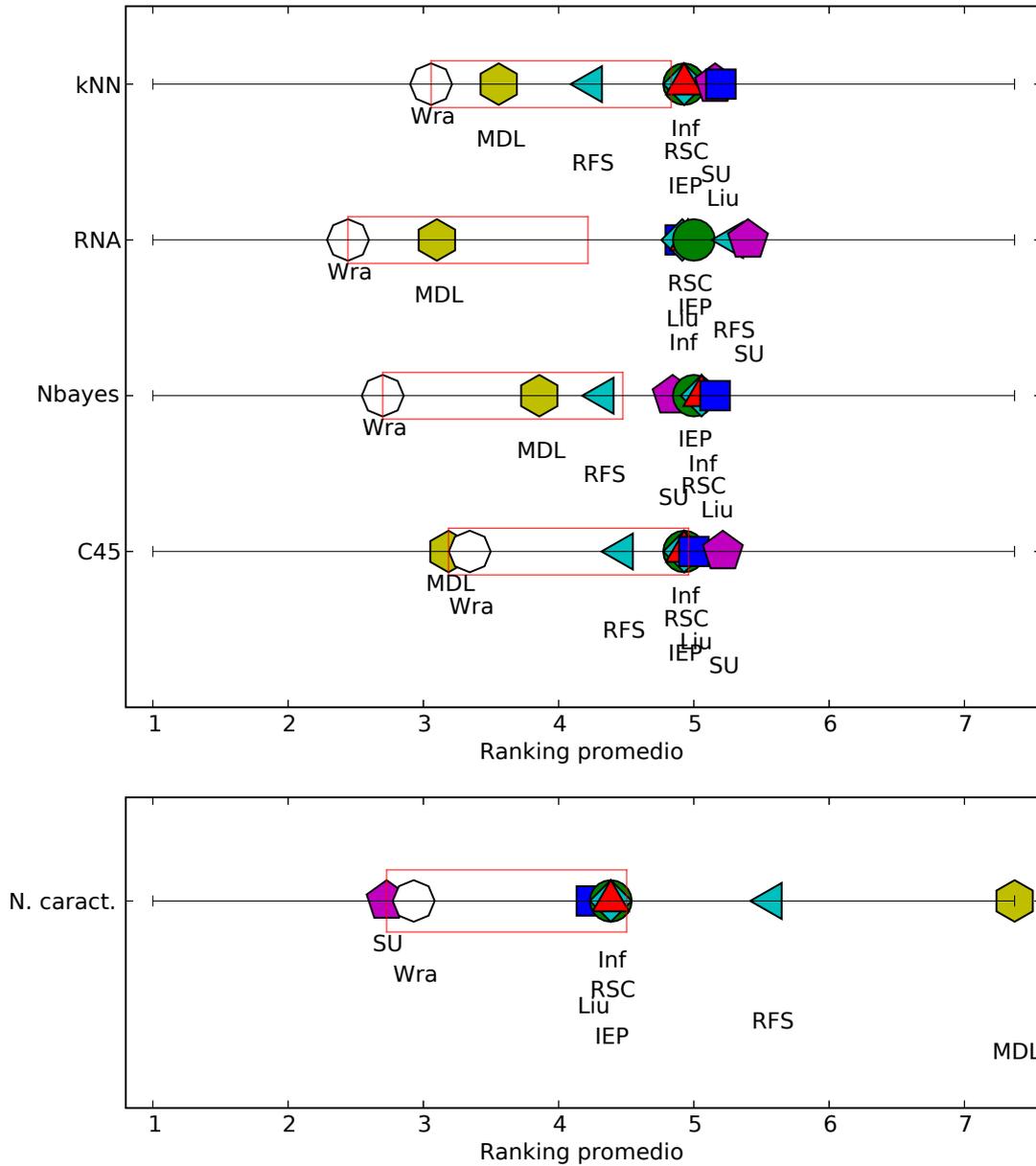
Para los métodos de búsqueda secuencial sí comienzan a detectarse diferencias significativas entre las medidas tanto en reducción de características, como ocurría hasta ahora, como (y aquí está la novedad) en los porcentajes de aciertos (para todos los clasificadores). Se vuelve a comprobar la incompatibilidad de los dos objetivos (reducción de características y error de clasificación) con la excepción, de nuevo, de Wra.

Filtro de Las Vegas (LVF). Liu y RFS obtienen buenos resultados tanto en acierto como en reducción, salvo a la hora de aplicar RFS con redes neuronales. Los mejores resultados en acierto se consiguen con RSC, pero es a costa de obtener los peores resultados en reducción. Esto se corresponde con la teoría que indica que RSC es una medida más conservadora. Por su parte, IEP consigue las mayores reducciones, pero a costa de obtener resultados significativamente peores.

Envolvente de Las Vegas (LVW). Para este método de búsqueda los resultados son muy claros. La medida Wra supera a las otras dos claramente. MDL ofrece resultados significativamente peores en aciertos y reducción. SU aunque parece ofrecer una reducción superior, no es significativa, mientras que el porcentaje de acierto sí se reduce significativamente.

El subgrupo de métodos de búsqueda probabilísticos es de lo más peculiar, porque en realidad los dos métodos son complementarios y cada uno se combina con medidas con que no se puede aplicar el otro. Se observa el mismo fenómeno de contraposición de objetivos, sólo aunados en el caso de Wra.

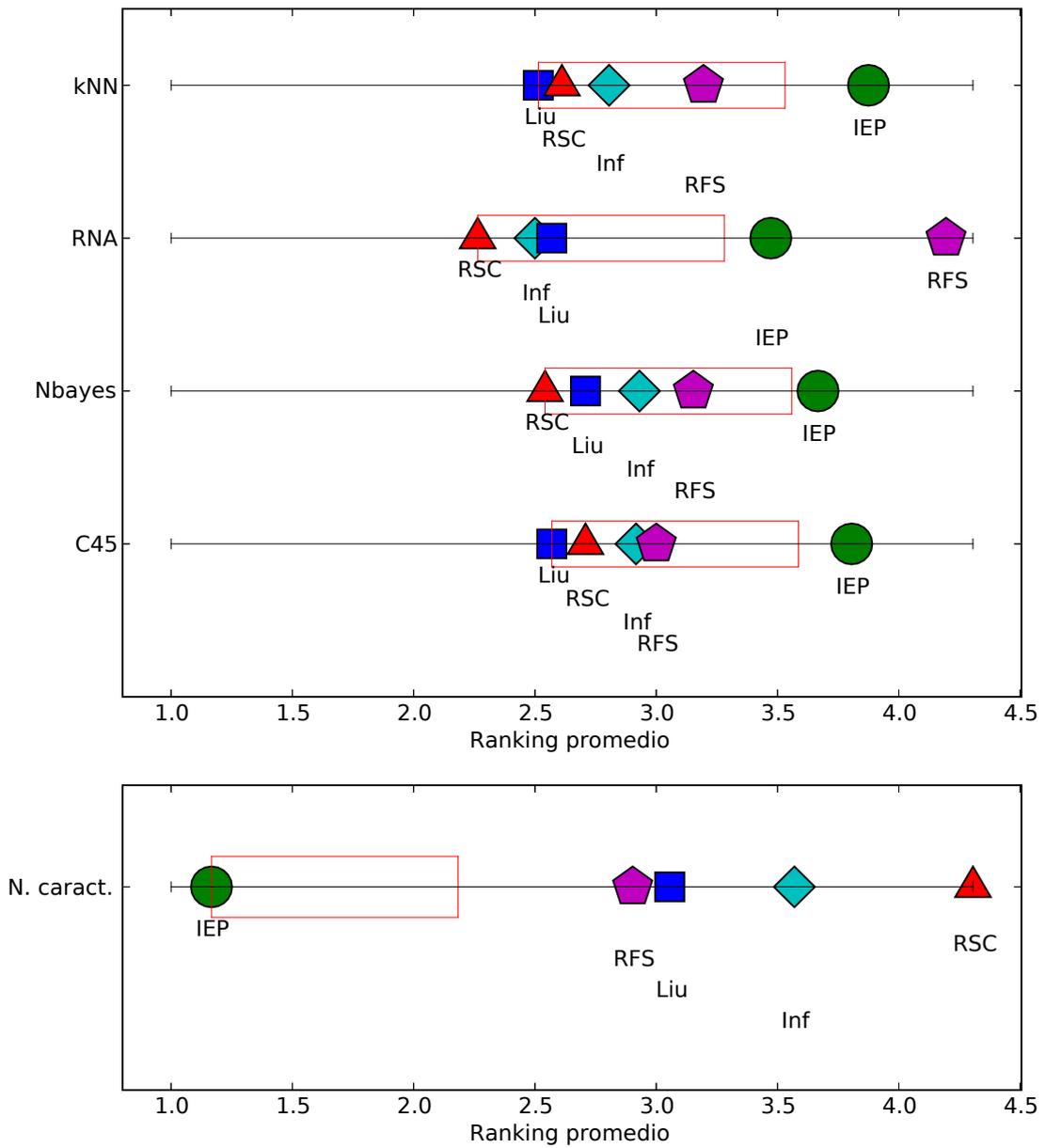
Enfriamiento simulado (SA). Al igual que en los dos métodos anteriores Wra destaca con los mejores resultados en acierto y muy buenos en reducción. MDL rivaliza en aciertos, pero sin reducir características.



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Liu	5.00	4.24	5.16	4.06	4.90	3.99	5.20	4.04
IEP	4.93	4.39	5.00	4.21	5.00	4.14	4.93	4.20
RSC	4.93	4.39	5.06	4.21	4.96	4.14	4.93	4.20
Inf	4.93	4.39	5.06	4.21	4.91	4.14	4.93	4.20
SU	5.21	2.73	4.84	2.61	5.40	2.57	5.16	2.59
MDL	3.19	7.37	3.86	7.37	3.10	7.31	3.56	7.36
Wra	3.34	2.93	2.70	3.70	2.44	4.21	3.06	3.77
RFS	4.47	5.57	4.33	5.61	5.29	5.49	4.24	5.64

Nemenyi critical distance (p=0.05): 1.7745725924

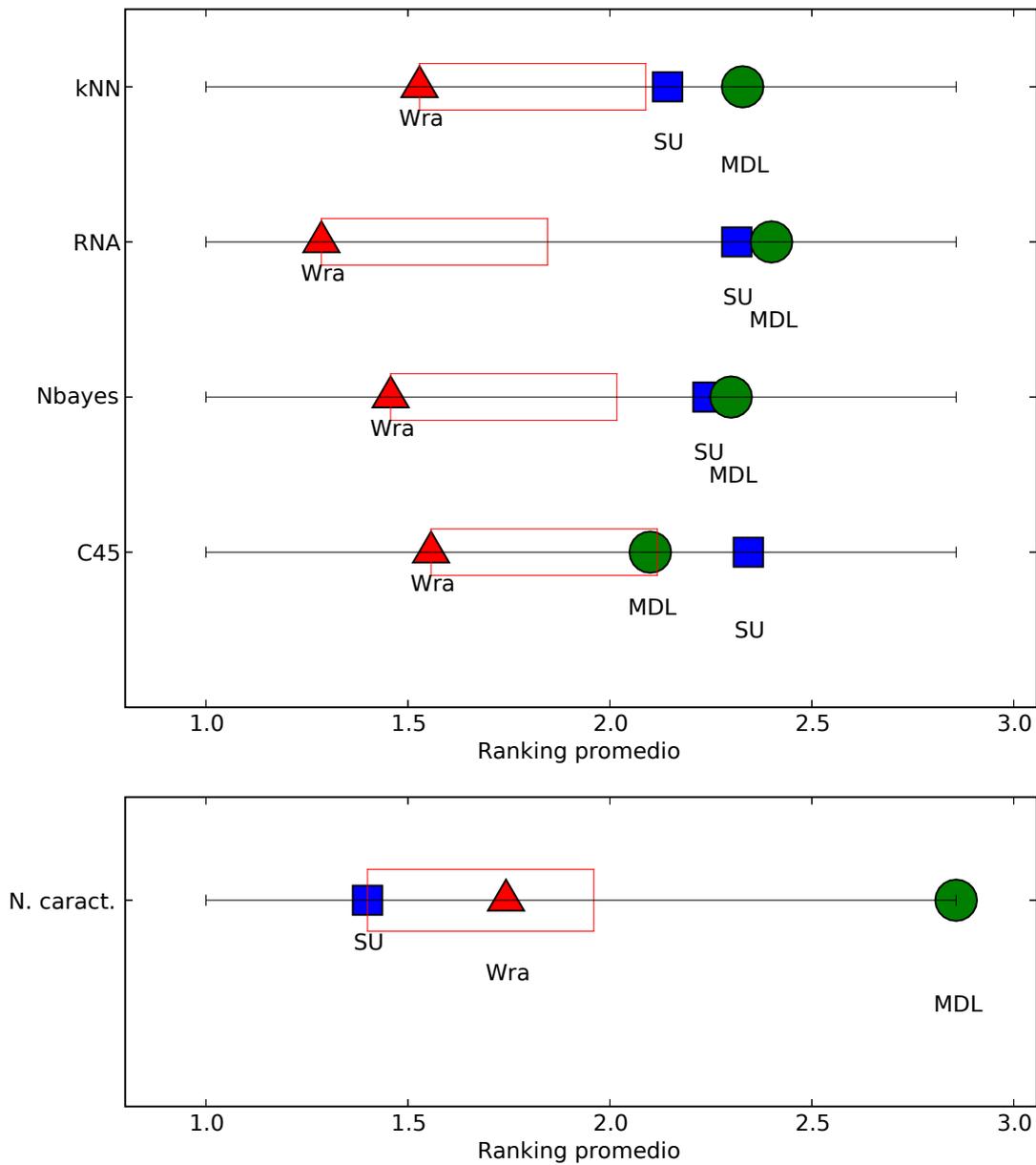
Tabla 4.22: Comparación de medidas para el método de búsqueda SBS



<i>Medida de relevancia</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
Liu	2.57	3.06	2.71	3.01	2.57	3.01	2.51	3.10
IEP	3.81	1.17	3.67	1.22	3.47	1.22	3.88	1.22
RSC	2.71	4.31	2.54	4.22	2.26	4.22	2.61	4.17
Inf	2.92	3.57	2.93	3.74	2.50	3.78	2.81	3.76
RFS	3.00	2.90	3.15	2.81	4.19	2.76	3.19	2.75

Nemenyi critical distance ($p=0.05$): 1.01667226774

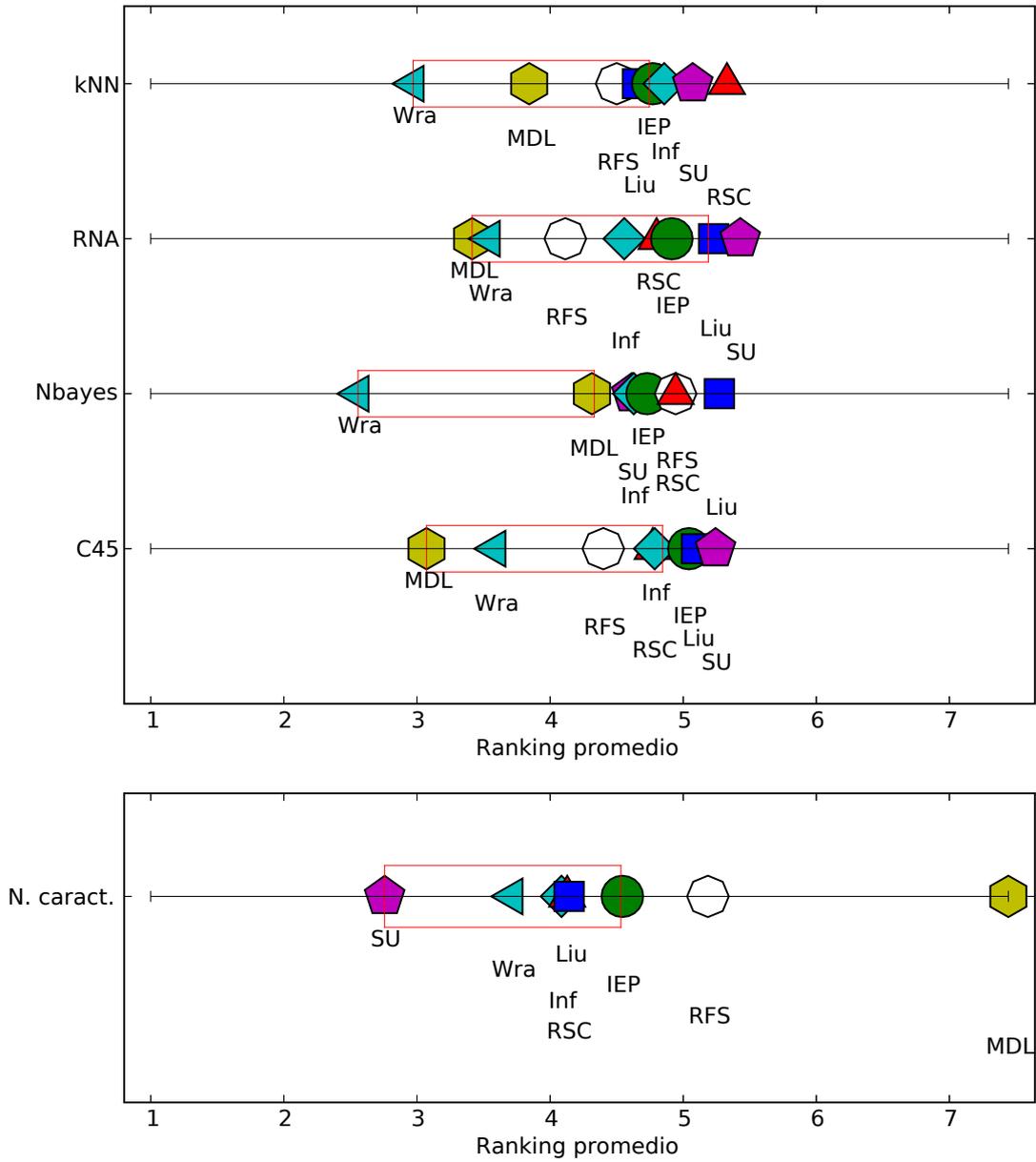
Tabla 4.23: Comparación de medidas para el método de búsqueda LVF



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
SU	2.34	1.40	2.24	1.40	2.31	1.40	2.14	1.43
MDL	2.10	2.86	2.30	2.86	2.40	2.76	2.33	2.81
Wra	1.56	1.74	1.46	1.74	1.29	1.84	1.53	1.76

Nemenyi critical distance (p=0.05): 0.560168240034

Tabla 4.24: Comparación de medidas para el método de búsqueda LVW



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Liu	5.10	4.14	5.27	4.23	5.23	4.30	4.66	3.90
IEP	5.04	4.54	4.73	4.53	4.91	4.61	4.77	4.53
RSC	4.77	4.13	4.94	4.06	4.80	3.71	5.33	4.13
Inf	4.79	4.09	4.63	4.14	4.56	3.74	4.86	4.00
SU	5.24	2.76	4.61	2.79	5.43	2.73	5.07	2.67
MDL	3.07	7.44	4.31	7.41	3.41	7.40	3.84	7.40
RFS	4.40	5.19	4.94	4.91	4.11	5.27	4.50	5.27
Wra	3.59	3.71	2.56	3.93	3.54	4.23	2.97	4.10

Nemenyi critical distance (p=0.05): 1.7745725924

Tabla 4.25: Comparación de medidas para el método de búsqueda SA

Algoritmos genéticos (GA). Wra destaca con diferencia significativa de todos los demás métodos, tanto en aciertos como en reducción.

Búsqueda tabú (TS). Wra obtiene mejores resultados, sin diferencia significativa en acierto, pero sí en reducción. SU consigue todavía mejores reducciones.

Los potentes métodos de búsqueda basados en metaheurísticas no alteran la validez de las conclusiones establecidas previamente. Se mantiene la contraposición de los dos objetivos con la excepción y liderazgo de Wra. También destaca la capacidad de reducción de características de SU.

4.4.1.3. Comparativa global de métodos de subconjuntos

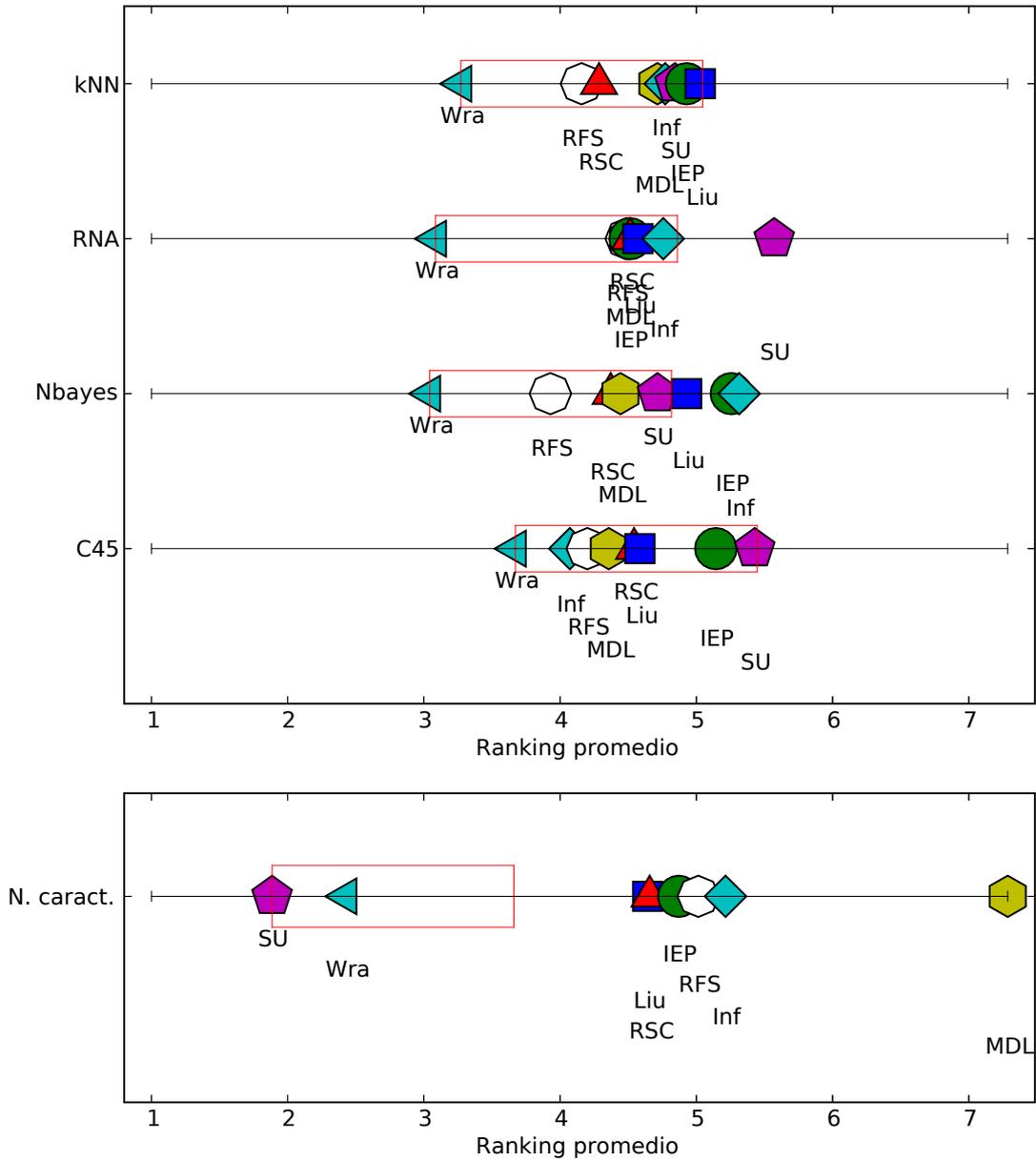
En el análisis de datos realizado en la subsección previa hemos ido derivando las conclusiones que establecían los datos de forma parcial. Para poder disponer de una perspectiva global es necesario considerar todos los métodos conjuntamente. Además, en esta comparativa se incluirán, tanto el método *Focus2* que al tener una sola medida no aparecía, como el método *NoFS* que nos sirve para comparar con los resultados que se obtendrían sin realizar selección de características.

En la tabla 4.29 se muestra la comparación en ranking de todos los métodos basados en subconjuntos, pero sólo sobre los 20 conjuntos de datos de menos de 16 características (limitación búsqueda completa, ver 4.3.5).

En comparación con los demás métodos *Focus2* obtiene unos resultados aceptables. Sin embargo, entre todos los métodos hay unos cuantos que, además de obtener algo mejores resultados, consideramos más interesantes por su menor carga computacional: SFS con alguna medida de consistencia para C45, SFS con cualquier medida para Nbayes, SFS-Inf o SFS-RSC para RNA, y SFS-IEP para knn.

En la tabla 4.28, se muestran los resultados de los métodos que se pueden aplicar en conjuntos de más de 16 características. Estos datos son más relevantes pues tienen en cuenta todos los conjuntos de datos, a excepción de *adult* sobre el que no pudimos aplicar la estrategia envolvente (*Wra*) por su elevado tiempo de ejecución.

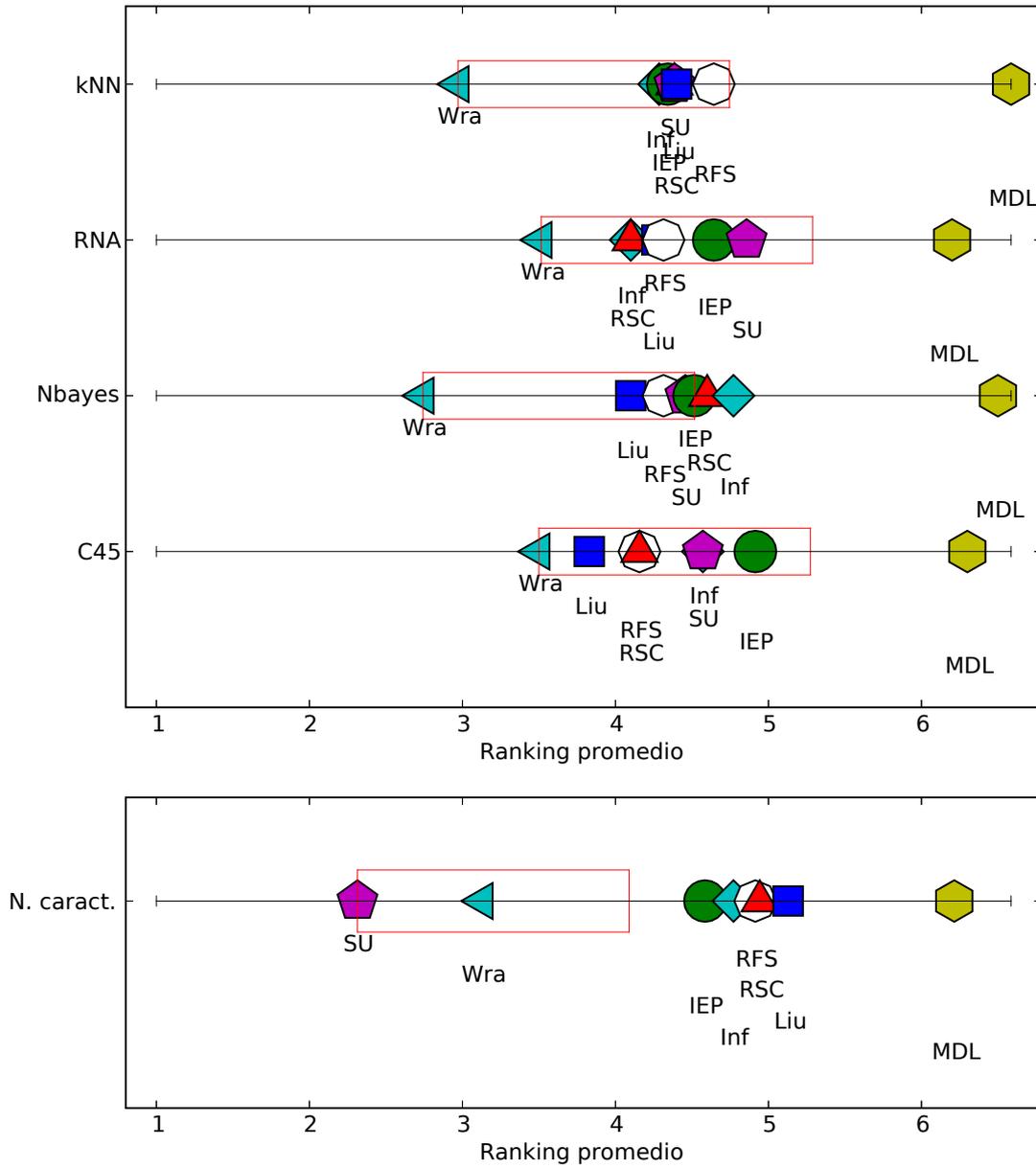
En general, los resultados en aciertos sin selección de características (*NoFS*) son mejores que los que obtienen la mayoría de los métodos evaluados. En bastantes



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
Liu	4.59	4.64	4.93	4.80	4.57	4.73	5.03	4.26
IEP	5.14	4.87	5.26	4.61	4.51	4.69	4.93	4.86
RSC	4.54	4.66	4.37	4.39	4.51	4.37	4.29	5.29
Inf	4.07	5.21	5.31	5.00	4.76	5.03	4.77	4.67
SU	5.43	1.89	4.71	2.00	5.57	1.86	4.84	1.81
MDL	4.36	7.29	4.44	7.17	4.50	7.10	4.71	7.07
RFS	4.20	5.01	3.93	5.36	4.49	5.24	4.16	4.96
Wra	3.67	2.43	3.04	2.67	3.09	2.99	3.27	3.09

Nemenyi critical distance (p=0.05): 1.7745725924

Tabla 4.26: Comparación de medidas para el método de búsqueda GA



Medida de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
Liu	3.83	5.13	4.10	4.99	4.27	4.50	4.40	5.01
IEP	4.91	4.59	4.51	5.14	4.64	4.84	4.34	4.97
RSC	4.16	4.94	4.60	4.86	4.10	5.11	4.39	4.44
Inf	4.57	4.77	4.77	4.63	4.10	5.06	4.29	4.47
SU	4.57	2.31	4.46	2.26	4.86	2.39	4.39	2.54
MDL	6.30	6.21	6.50	5.86	6.20	6.04	6.59	6.41
RFS	4.16	4.91	4.31	5.34	4.31	5.10	4.64	5.36
Wra	3.50	3.13	2.74	2.93	3.51	2.96	2.97	2.79

Nemenyi critical distance (p=0.05): 1.7745725924

Tabla 4.27: Comparación de medidas para el método de búsqueda TS

casos, la selección implica una mejora y cuando se producen pérdidas en la eficacia son de pequeñas. Por ejemplo, considerando los resultados obtenidos en todos problemas incluidos en nuestro estudio el promedio indica una mejora en el porcentaje de aciertos de 0,26 (véase la Tabla A.4). Lo interesante no es lo pequeño de la diferencia, sino que reduciendo el número de características (a veces, mucho) se mantiene el nivel de exactitud en el clasificador. Por otro lado, no hay que olvidar todas las demás ventajas adicionales asociadas a la selección de características mencionadas en la sección 1.1.

También puede ser interesante considerar la tolerancia de los algoritmos de aprendizaje a características irrelevantes o redundantes. Si analizamos los resultados con C45, podemos ver como ningún método de selección le hace mejorar (en ranking promedio). La razón puede ser que C45 lleva su propia selección de características inmersa. Sin embargo, según la tabla 4.8, éste no es el clasificador que mejores resultados obtenía sin selección de características, sino que era la RNA. Sobre la tabla A.1 de resultados detallados, podemos ver un ejemplo ilustrativo de como la selección de características inmersa en C45 puede no ser una ventaja. El resultado obtenido en el conjunto de datos *anneal* con C45 sin selección es de un 92.3% de acierto, mientras que, usando un buen método de selección de características como SBS-RFS, se puede obtener un 93.0%. Es más, podemos ver cómo C45 sin selección de características es mejorado por un algoritmo tan simple como Nbayes usando SBS-RFS.

En el resto de los clasificadores sí que hay algunos métodos de selección que obtienen mejoras en promedio. Nbayes es mejorado por la combinación de cualquier método de búsqueda con la medida Wra. RNA es mejorado por SBS-Wra y LVF-Wra. Finalmente, knn es mejorado por SFS-Liu y la mayoría de los métodos de búsqueda combinados con Wra.

Sin tener en cuenta nada más que el porcentaje de acierto, la aplicación de la estrategia envolvente, casi con cualquier mecanismo de búsqueda, supera a las demás. Desafortunadamente esta medida tiene problemas de eficiencia y no es aplicable en muchos de los casos donde la selección de características es útil. Por esta razón es interesante considerar otros métodos.

Se constata una contraposición entre los dos objetivos fundamentales: precisión en la clasificación y reducción en el número de características. Se observa en los resultados una tendencia a obtener mejores resultados en acierto cuanto menor reducción es la reducción en el número de características, y viceversa. De modo gráfico, se puede constatar este hecho en las figuras 4.8 y 4.9

Con el objeto de ver la distribución de los métodos relacionando la reducción con la calidad de los resultados obtenidos, representamos gráficamente la envolvente convexa de los métodos, donde cada método está representado por un punto en el plano, cuyas coordenadas son sus rankings en porcentaje de aciertos y reducción.. En estas gráficas, cuanto más concentrada esté un área indica una mayor similitud entre los métodos, ya que estamos comparando rankings. Los mejores métodos son los que quedan más cercanos al origen de coordenadas (0,0).

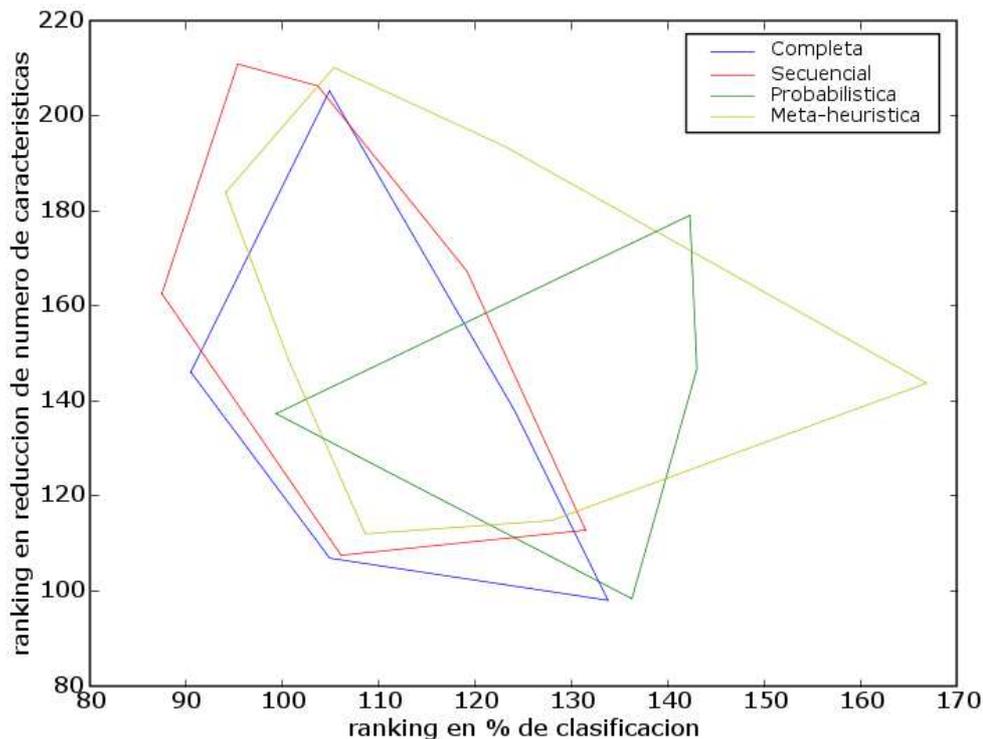


Figura 4.8: Envolvente convexa según métodos de búsqueda para C45

Por otro lado, hemos de considerar que partimos de que queremos realizar selección de características y por tanto reducir su número. Si nos fijamos en aquellos métodos que obtienen un ranking menor que 23 en reducción de características (los métodos usando Wra oscilan entre 17,5 y 23), los mejores métodos sin usar Wra son:

- SFS-Liu
- SFS-IEP
- SFS-RSC

Es curioso que las metaheurísticas no superen en resultados a un algoritmo voraz simple como SFS. Puede ser debido a que en este estudio se han considerado

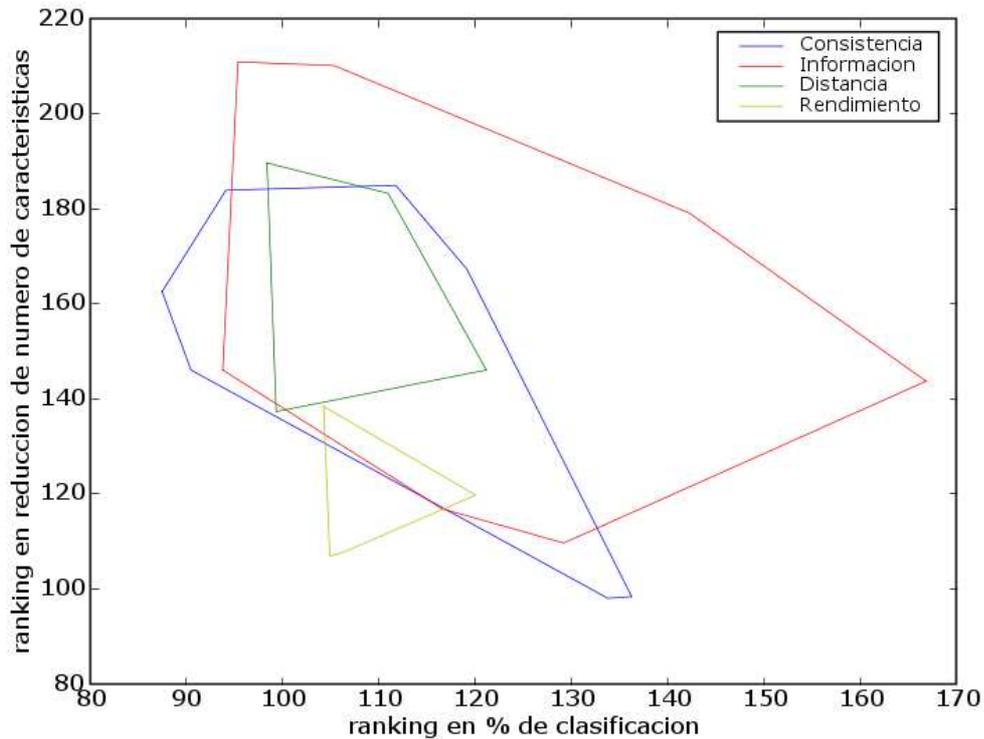


Figura 4.9: Envolverte convexa según tipo de medida para C45

pocas evaluaciones, y las metaheurísticas necesitan más para funcionar bien. En este estudio se han usado tan sólo 1000 evaluaciones para poder abordar todos los conjuntos de datos, y con la idea en mente de que la selección de características debe ser un paso previo más simple que el aprendizaje posterior.

<i>Método de selección</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
NoFS-	15.92	46.45	17.88	46.41	14.00	46.39	18.44	46.36
SFS-Liu	19.35	21.35	18.77	21.11	19.70	21.11	18.32	20.94
SFS-IEP	21.70	19.03	23.15	18.77	22.68	18.79	21.27	18.70
SFS-RSC	17.42	21.09	20.42	20.64	20.06	20.77	19.83	20.56
SFS-Inf	24.20	23.45	22.12	22.92	21.15	23.05	21.36	23.06
SFS-SU	27.48	12.62	23.62	12.91	26.30	12.64	24.26	12.61
SFS-MDL	16.33	44.45	19.38	44.32	17.03	44.27	20.15	44.23
SFS-RFS	24.62	22.17	25.73	21.27	30.97	21.42	25.02	23.32
SFS-Wra	20.58	15.73	12.14	17.91	22.68	18.79	14.67	17.48
SBS-Liu	31.21	22.05	31.15	21.88	29.53	21.85	30.15	21.56

continúa en la página siguiente

viene de la página anterior

<i>Método de selección</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
SBS-IEP	30.68	22.82	30.09	22.55	29.94	22.55	28.44	22.32
SBS-RSC	30.68	22.82	30.15	22.55	29.89	22.55	28.44	22.32
SBS-Inf	30.68	22.82	30.15	22.55	29.82	22.55	28.44	22.32
SBS-SU	32.48	12.29	29.91	12.26	32.14	12.17	28.83	12.03
SBS-MDL	16.82	43.55	20.47	43.44	18.83	43.41	19.82	43.29
SBS-RFS	23.55	34.68	25.98	35.05	29.55	34.70	23.94	35.08
SBS-Wra	18.74	12.15	13.45	17.64	12.83	22.30	13.76	18.27
LVF-Liu	28.83	15.03	27.98	14.98	31.68	14.89	26.68	15.27
LVF-IEP	34.20	4.47	35.08	4.36	35.45	4.41	35.95	4.23
LVF-RSC	27.20	18.83	28.50	18.83	31.00	18.56	28.05	18.56
LVF-Inf	32.86	16.33	30.52	16.64	30.79	16.80	28.59	16.92
LVW-SU	32.09	18.64	27.53	19.44	25.76	19.20	28.38	18.58
LVW-MDL	26.14	36.95	26.76	36.85	26.70	36.89	29.47	37.30
LVF-RFS	27.70	15.12	30.58	14.82	42.35	14.02	30.95	14.50
LVW-Wra	20.55	19.59	13.36	20.82	13.27	22.24	18.18	20.85
SA-Liu	26.71	24.52	27.20	23.94	27.76	24.65	25.42	23.67
SA-IEP	27.45	25.70	25.59	25.47	25.85	26.14	24.79	25.38
SA-RSC	25.64	24.27	26.77	23.68	24.91	23.36	27.74	24.33
SA-Inf	24.94	24.20	25.32	23.91	24.09	23.53	24.50	24.03
SA-SU	27.12	16.86	25.20	16.65	27.92	16.59	27.83	16.58
SA-MDL	17.65	43.73	20.95	43.73	17.42	43.45	21.38	43.56
SA-RFS	24.33	32.59	25.80	31.45	21.55	32.98	24.24	32.80
SA-Wra	20.06	17.76	13.36	20.71	17.62	22.59	14.92	20.98
GA-Liu	23.23	35.45	26.42	35.91	22.71	34.70	25.94	33.52
GA-IEP	25.68	35.89	27.59	34.79	23.24	34.62	25.80	34.89
GA-RSC	23.17	35.32	23.42	34.21	23.65	33.70	22.58	36.59
GA-Inf	21.20	36.06	27.56	35.73	22.68	34.64	23.85	35.95
GA-SU	27.65	19.20	24.92	18.83	28.39	18.73	25.39	18.85
GA-MDL	23.24	40.91	23.45	40.48	24.45	39.33	24.98	40.20
GA-RFS	21.88	36.64	21.47	36.77	21.97	35.64	21.67	34.85
GA-Wra	20.06	23.24	16.12	22.48	16.11	23.65	16.88	23.00
TS-Liu	24.08	24.38	28.36	24.03	25.58	22.58	29.03	24.45
TS-IEP	26.91	24.17	29.36	24.33	27.14	22.95	30.41	24.50
TS-RSC	25.83	24.58	30.03	23.97	26.15	23.86	29.26	22.85
TS-Inf	28.89	23.80	30.26	23.44	24.65	23.09	29.59	23.29
TS-SU	27.03	18.23	27.71	18.15	26.33	17.48	28.03	18.41

continúa en la página siguiente

viene de la página anterior

<i>Método de selección</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
TS-MDL	36.05	28.00	38.42	26.21	34.95	26.59	39.17	27.55
TS-RFS	25.61	24.65	28.17	25.62	25.24	24.44	29.50	25.08
TS-Wra	18.58	20.38	16.64	19.59	20.52	19.39	20.70	19.05

Tabla 4.28: Comparación de métodos basados en evaluación de conjuntos de características sobre 35 conjuntos de datos

<i>Método de selección</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
NoFS-	27.47	63.03	29.83	62.89	20.31	62.92	23.81	62.86
Ex-Liu	34.31	33.67	35.19	32.81	28.08	33.28	36.42	33.36
Ex-IEP	33.42	35.47	33.11	34.42	28.36	34.89	32.31	34.97
Ex-RSC	33.42	35.47	33.11	34.42	28.36	34.89	32.31	34.97
Ex-Inf	33.42	35.47	33.11	34.42	28.36	34.89	32.31	34.97
Ex-SU	38.28	20.44	33.47	19.83	36.19	20.14	37.00	19.56
Ex-Wra	30.03	15.11	22.83	22.72	36.56	19.61	18.44	21.14
Ex-MDL	30.97	55.61	36.72	55.31	26.61	55.44	28.69	55.11
Ex-RFS	32.92	42.86	36.67	42.25	35.86	42.31	28.31	41.92
Focus2-	32.08	47.33	35.81	46.36	28.22	46.67	33.94	46.50
BB-Liu	29.33	29.97	36.06	29.25	40.39	28.14	42.47	28.39
BB-IEP	32.31	29.97	40.44	29.25	42.22	28.14	44.03	28.39
BB-RSC	32.31	29.97	45.06	29.25	42.61	28.14	43.83	28.39
BB-Inf	31.08	29.97	40.58	29.25	42.36	28.14	41.56	28.39
BB-RFS	37.50	29.97	32.00	29.25	43.00	28.14	38.83	28.39
ABB-Liu	40.44	23.97	40.47	23.81	38.28	23.47	36.61	23.83
ABB-IEP	43.50	8.86	42.64	8.47	46.72	8.50	47.75	8.17
ABB-RSC	36.53	27.58	40.92	27.25	38.94	26.92	35.92	27.36
ABB-Inf	35.89	27.17	37.94	26.83	35.97	26.58	37.31	26.97
ABB-RFS	34.36	31.72	31.81	36.06	37.28	31.97	33.31	33.42
SFS-Liu	27.14	41.58	27.28	41.11	27.14	41.44	28.17	40.81
SFS-IEP	27.64	38.97	26.50	38.28	27.14	38.78	27.31	38.31
SFS-RSC	27.44	40.86	28.31	40.00	26.69	40.58	28.75	39.83
SFS-Inf	33.03	39.03	28.47	38.22	26.64	38.44	30.53	38.25
SFS-SU	33.33	24.39	27.47	24.42	32.47	24.33	31.92	23.83
SFS-MDL	28.00	57.92	33.58	57.58	27.42	57.58	27.83	57.47

continúa en la siguiente página

viene de la página anterior

<i>Método de selección</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
SFS-RFS	32.17	38.08	35.06	34.94	44.97	38.00	30.89	42.56
SFS-Wra	33.22	20.44	17.72	21.89	30.94	32.64	25.44	25.00
SBS-Liu	37.11	40.89	36.39	40.42	34.39	40.58	37.81	40.06
SBS-IEP	36.14	42.75	34.11	42.06	35.47	42.31	33.64	41.83
SBS-RSC	36.14	42.75	34.11	42.06	35.47	42.31	33.64	41.83
SBS-Inf	36.14	42.75	34.11	42.06	35.47	42.31	33.64	41.83
SBS-SU	40.25	20.69	35.33	20.17	37.28	20.42	38.64	19.72
SBS-MDL	30.31	55.94	35.86	55.64	27.19	55.67	27.44	55.39
SBS-RFS	34.19	49.06	34.50	49.56	45.72	48.78	31.75	49.22
SBS-Wra	30.08	15.53	20.03	22.64	16.47	26.81	22.14	26.61
LVF-Liu	37.17	25.08	35.44	24.81	35.33	24.50	33.92	25.47
LVF-IEP	42.50	8.94	44.47	8.58	45.50	8.61	47.17	8.28
LVF-RSC	35.94	30.22	41.56	29.72	39.44	29.44	31.36	29.86
LVF-Inf	43.39	28.97	37.47	28.89	37.83	29.50	37.58	29.08
LVW-SU	39.33	22.86	32.67	23.17	35.83	23.81	41.25	22.42
LVW-MDL	41.58	47.47	41.81	47.42	40.44	47.69	42.86	47.81
LVF-RFS	30.78	24.89	41.81	24.42	56.39	23.50	36.39	24.22
LVW-Wra	34.83	20.17	18.33	23.89	19.31	26.64	27.61	25.61
SA-Liu	31.92	37.39	34.81	37.28	38.14	38.06	33.06	36.67
SA-IEP	37.61	40.44	30.08	40.00	33.72	41.00	28.75	39.47
SA-RSC	33.11	38.42	35.08	37.36	30.17	38.14	34.53	38.97
SA-Inf	33.25	38.61	33.17	38.03	29.69	38.36	31.67	38.78
SA-SU	39.08	21.89	33.97	21.33	36.33	21.94	35.97	21.17
SA-MDL	31.36	56.47	36.67	56.39	27.69	55.89	31.11	56.11
SA-RFS	32.78	47.86	37.03	46.22	26.17	48.81	28.61	47.64
SA-Wra	32.61	17.25	20.33	23.92	34.61	24.11	20.50	24.39
GA-Liu	31.08	49.92	36.42	50.17	32.72	46.56	36.22	46.00
GA-IEP	34.39	50.53	37.64	50.08	31.61	48.89	36.67	47.89
GA-RSC	29.42	49.81	29.67	45.75	33.08	46.72	32.81	50.33
GA-Inf	29.64	49.83	36.31	50.92	30.06	48.69	34.28	50.67
GA-SU	39.89	24.31	30.33	24.00	35.42	23.72	35.86	23.81
GA-MDL	37.28	52.81	36.39	51.61	36.81	49.22	33.56	51.31
GA-RFS	29.11	52.69	31.89	50.58	27.36	50.83	28.00	48.92
GA-Wra	31.08	25.89	26.67	25.19	23.31	27.11	25.42	25.97
TS-Liu	32.31	30.08	37.11	28.97	39.64	28.39	43.83	30.44
TS-IEP	38.50	28.53	42.81	30.47	39.75	27.28	43.83	31.17

continua en la siguiente página

viene de la página anterior

<i>Método de selección</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
TS-RSC	36.97	28.75	45.39	29.97	40.33	30.00	43.28	26.36
TS-Inf	40.78	29.03	39.17	30.56	35.17	27.14	44.81	26.44
TS-SU	39.47	19.14	37.61	20.11	38.22	18.81	39.58	20.42
TS-MDL	49.94	30.31	56.61	27.17	50.78	28.36	52.50	31.67
TS-RFS	36.03	31.56	34.39	33.33	36.36	30.47	43.53	29.61
TS-Wra	30.97	20.61	26.31	20.58	33.19	19.67	34.83	19.44

Tabla 4.29: Comparación de todos los métodos basados en evaluación de conjuntos de características sobre 20 conjuntos de datos

4.4.2. Métodos que evalúan características individuales

4.4.2.1. Resultados agrupados por criterio de punto de corte

Las n mejores características (n17). Escogiendo un número fijo de características y tan elevado como ha sido el resultado de la elección de parámetros (ver sección 4.2.1), las diferencias entre las medidas son prácticamente inapreciables, y en cualquier caso no significativas. Aparentemente info consigue resultados ligeramente mejores.

Una fracción de las características disponibles (p0.8). Al escoger un porcentaje de las características disponibles, aunque sin ser significativas hay mayor variación que con un número fijo. Destaca la medida de gini en 3 ocasiones y reli en 2.

Valoración por encima de un umbral (t0.1). Destaca reli en aciertos, pero a costa de la peor reducción. En reducción destaca gini, pero con resultados de acierto significativamente peores que las demás. La medida gain consigue un punto de equilibrio con reducción en segundo puesto, y aciertos no significativamente peores que reli.

Valoración por encima de una fracción del ranking de las valoraciones (pm0.8). Las medidas gain, info y gini destacan por estar entre los mejores aciertos

y las mayores reducciones, pero sin diferencias que puedan considerarse significativas en ambos casos.

Bajada de valoración absoluta (d0.2). Con este criterio de corte parece que *reli* se ve perjudicada, obteniendo peores aciertos que las demás, aunque la diferencia no es significativa. Los demás métodos obtienen resultados similares entre ellos, *gini* con menores reducciones de características.

Bajada de valoración relativa (s1.5). Usando este criterio de corte no hay diferencias significativas ni destacables entre las medidas.

4.4.2.2. Resultados agrupados por medidas

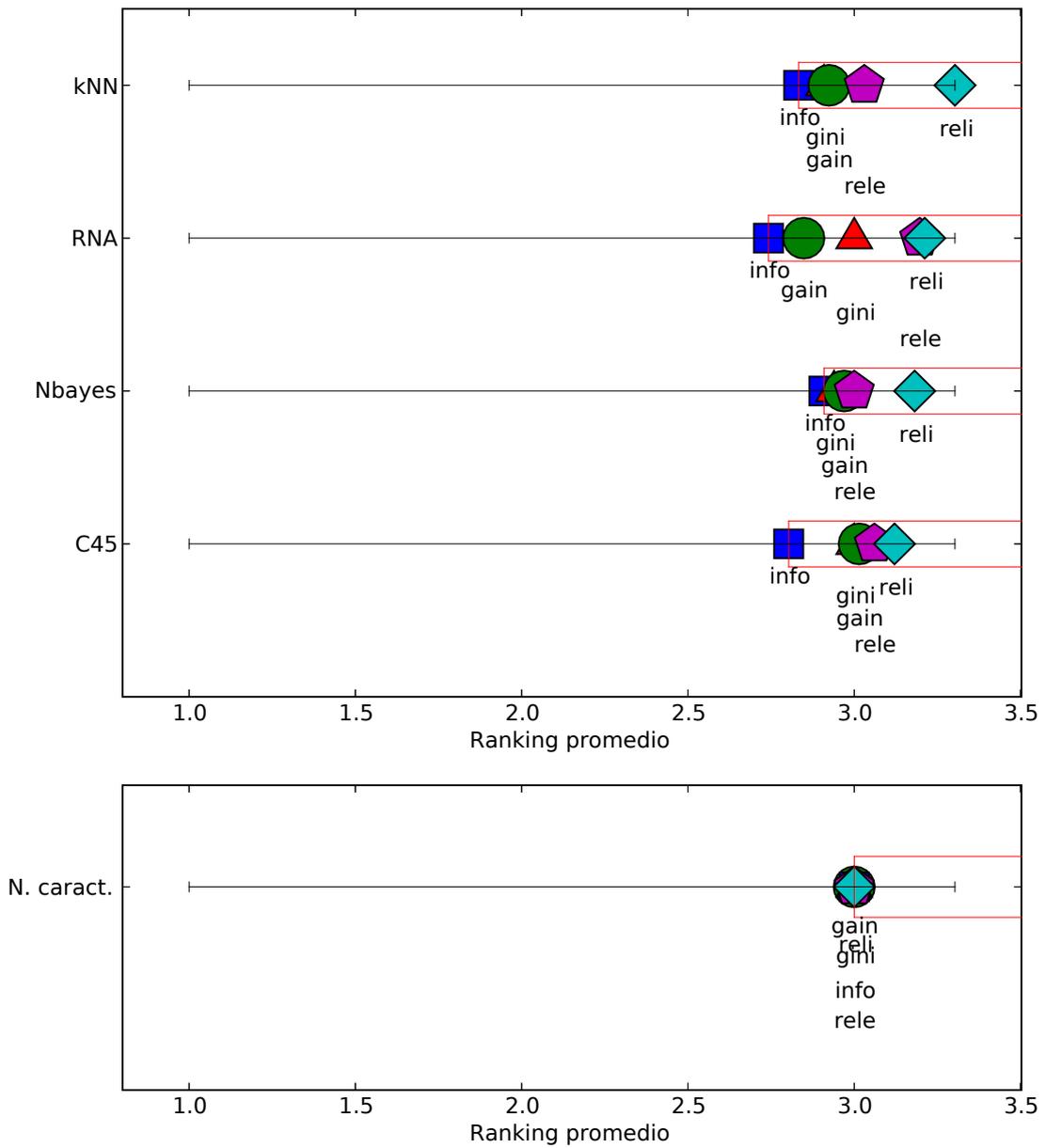
Información mutua (info) Por aciertos descartaríamos los criterios de corte *pm0.8* y *s1.5*, sin embargo, estos son los métodos que aportan mayores reducciones. De nuevo, aparece la relación inversa entre reducción y aciertos.

Gain ratio (gain) Se repite la observación de relación inversa entre reducción y aciertos. Podríamos destacar *t0.1* como un criterio que permite el equilibrio entre aciertos no significativamente peores que el mejor y reducciones no significativamente peores que la mejor.

Índice de Gini (gini) Nuevamente se observa relación inversa entre reducción y aciertos. En este caso se puede destacar a *pm0.8* como método de equilibrio.

Valoración de Relief-F (reli) Los métodos *d0.2* y *p0.8* pueden ser el mejor equilibrio entre aciertos y reducción para esta medida. Por lo demás se observa la relación inversa entre aciertos y reducción.

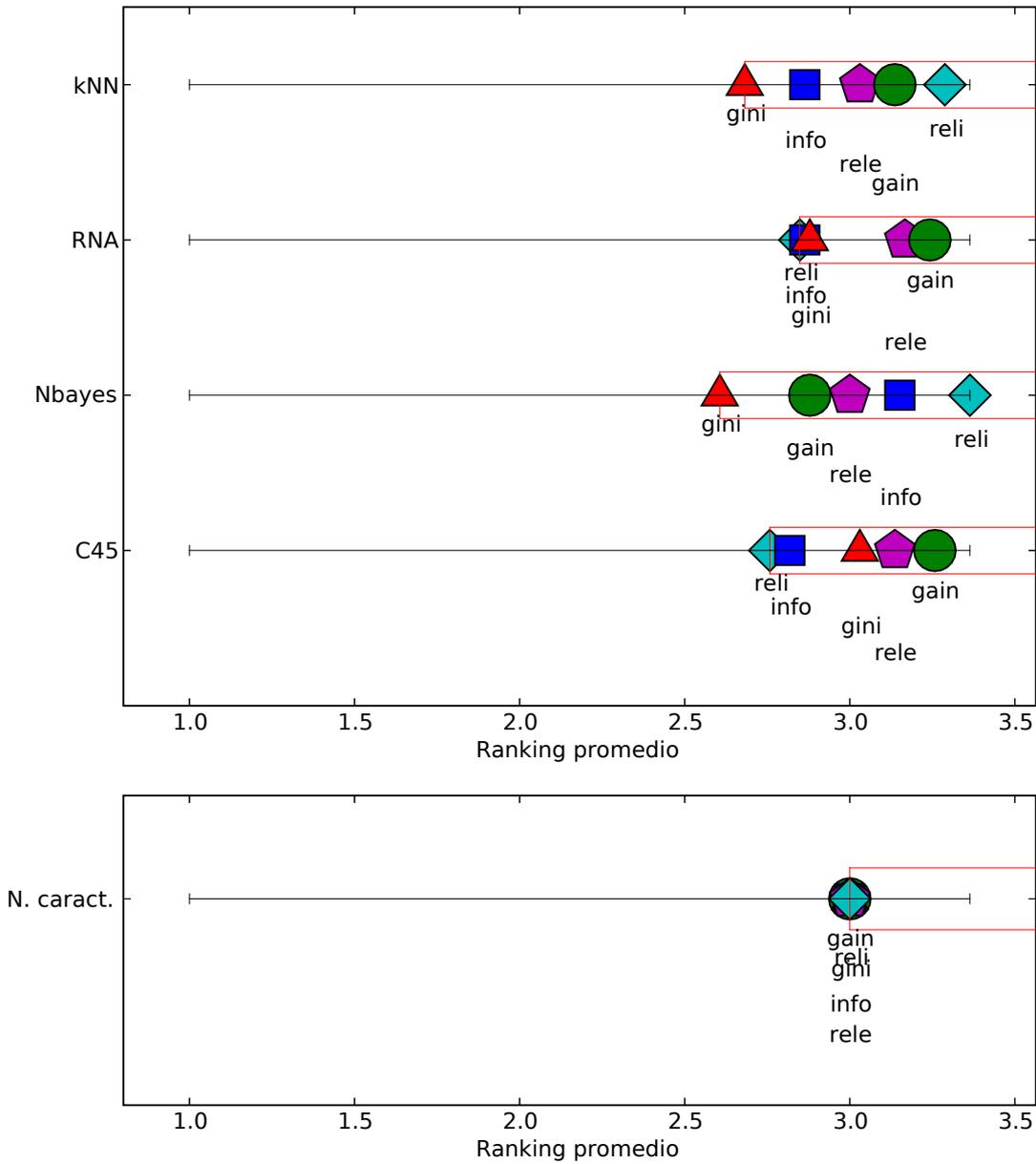
Potencial en la formación de reglas de decisión (rele) En este caso los métodos de equilibrio son *p0.8* y *t0.1*. Relación inversa entre aciertos y reducción como en las anteriores.



<i>Medida indiv. de relevancia</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
info	2.80	3.00	2.91	3.00	2.74	3.00	2.83	3.00
gain	3.02	3.00	2.97	3.00	2.85	3.00	2.92	3.00
gini	3.00	3.00	2.94	3.00	3.00	3.00	2.91	3.00
reli	3.12	3.00	3.18	3.00	3.21	3.00	3.30	3.00
rele	3.06	3.00	3.00	3.00	3.20	3.00	3.03	3.00

Nemenyi critical distance (p=0.05): 1.06187955146

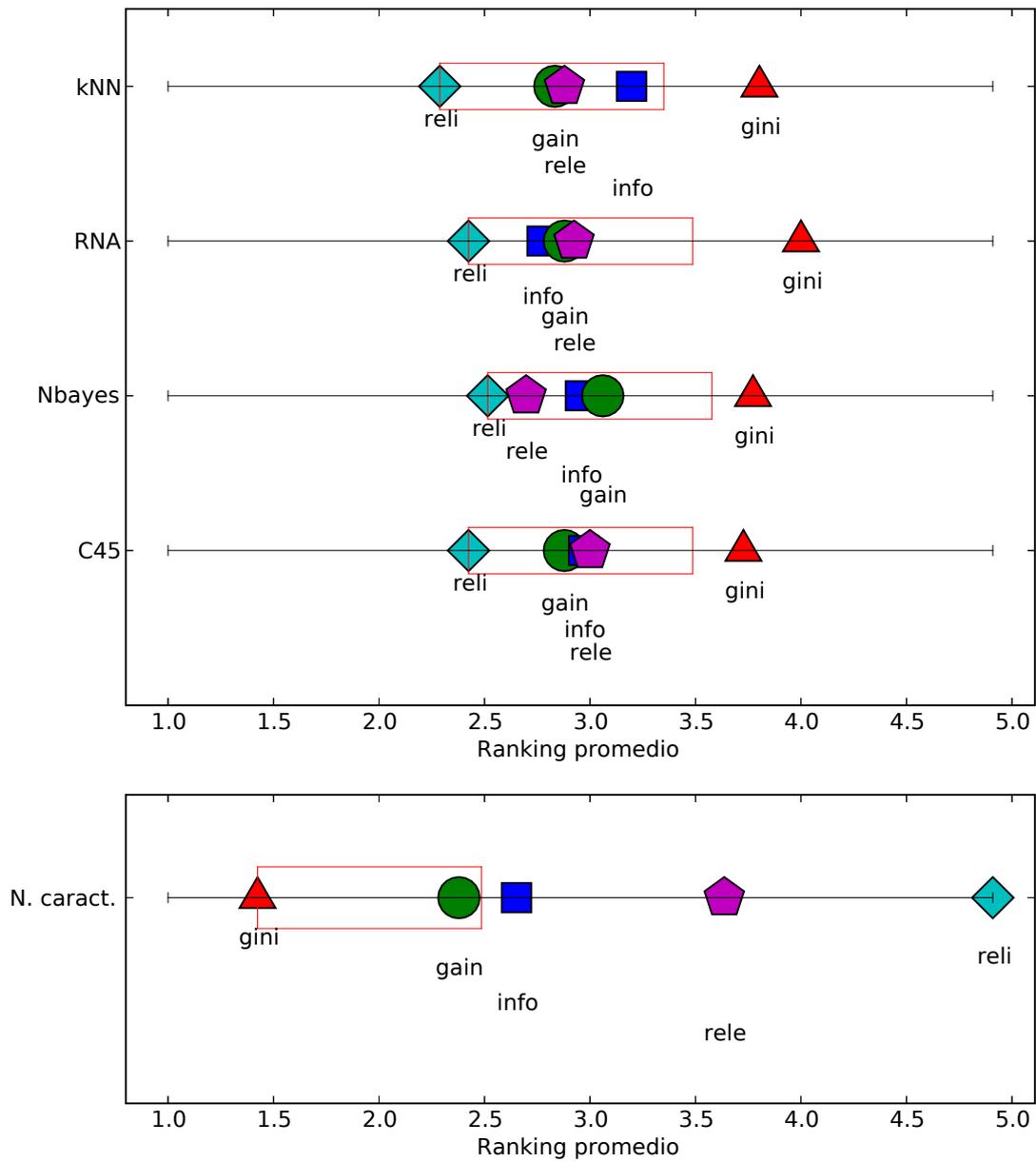
Tabla 4.30: Comparación de medidas para el método de corte n17



Medida indiv. de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
info	2.82	3.00	3.15	3.00	2.86	3.00	2.86	3.00
gain	3.26	3.00	2.88	3.00	3.24	3.00	3.14	3.00
gini	3.03	3.00	2.61	3.00	2.88	3.00	2.68	3.00
reli	2.76	3.00	3.36	3.00	2.85	3.00	3.29	3.00
rele	3.14	3.00	3.00	3.00	3.17	3.00	3.03	3.00

Nemenyi critical distance (p=0.05): 1.06187955146

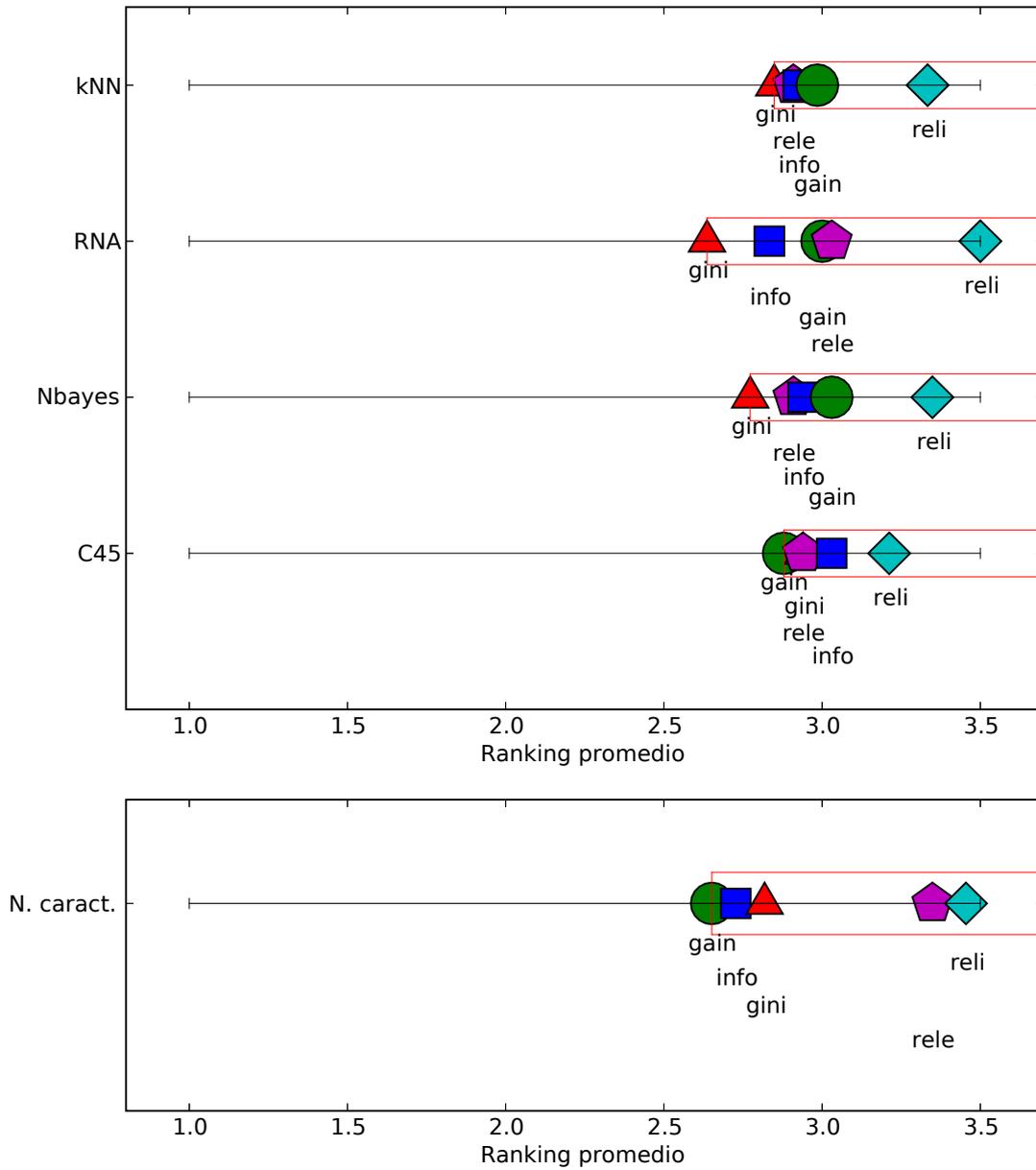
Tabla 4.31: Comparación de medidas para el método de corte p0.8



<i>Medida indiv. de relevancia</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
info	2.97	2.65	2.95	2.65	2.77	2.65	3.20	2.65
gain	2.88	2.38	3.06	2.38	2.88	2.38	2.83	2.38
gini	3.73	1.42	3.77	1.42	4.00	1.42	3.80	1.42
reli	2.42	4.91	2.52	4.91	2.42	4.91	2.29	4.91
rele	3.00	3.64	2.70	3.64	2.92	3.64	2.88	3.64

Nemenyi critical distance (p=0.05): 1.06187955146

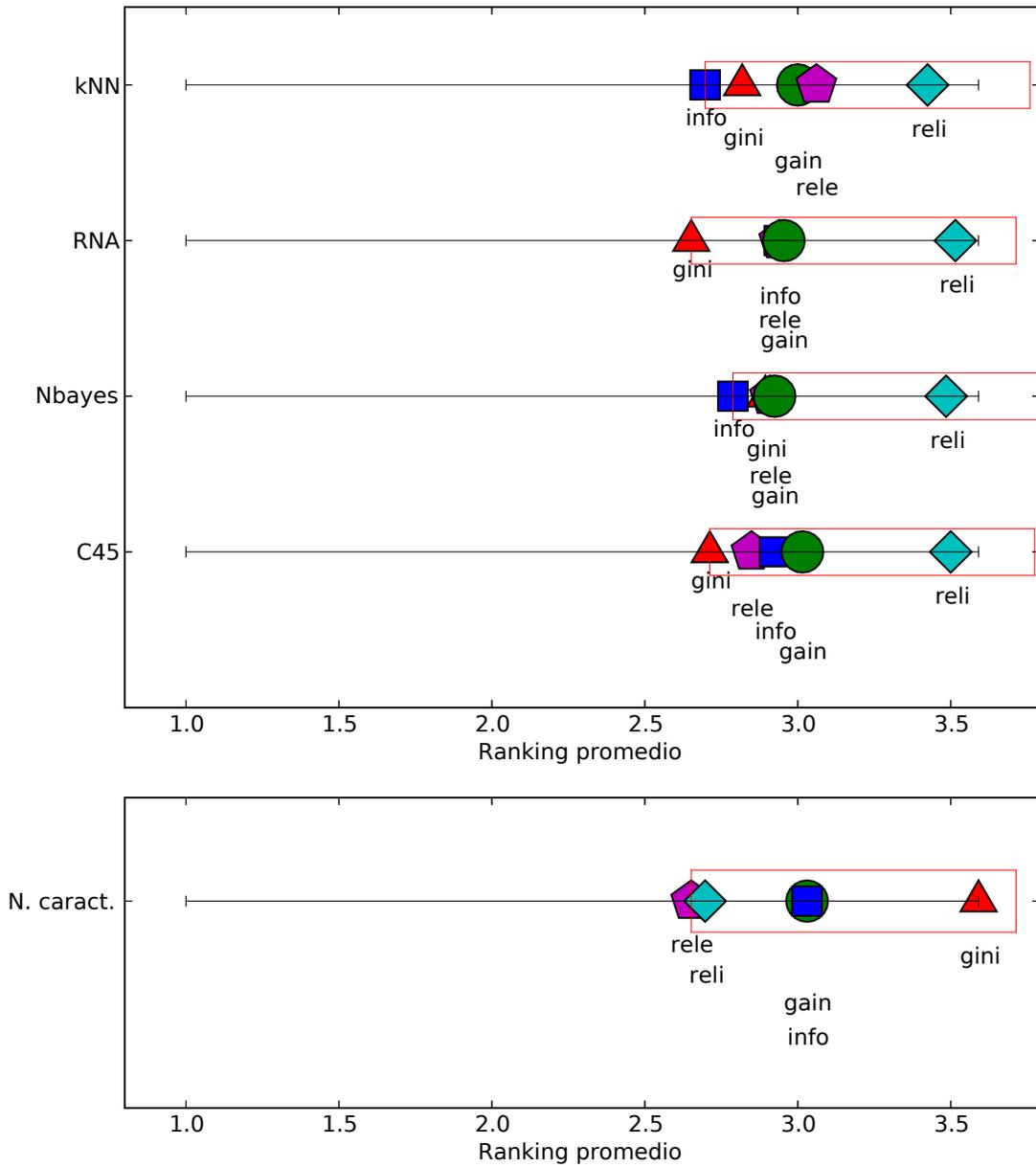
Tabla 4.32: Comparación de medidas para el método de corte t0.1



Medida indiv. de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.nºc.	R.Ac.	R.nºc.	R.Ac.	R.nºc.	R.Ac.	R.nºc.
info	3.03	2.73	2.94	2.73	2.83	2.73	2.92	2.73
gain	2.88	2.65	3.03	2.65	3.00	2.65	2.98	2.65
gini	2.94	2.82	2.77	2.82	2.64	2.82	2.85	2.82
rele	2.94	3.35	2.91	3.35	3.03	3.35	2.91	3.35
reli	3.21	3.45	3.35	3.45	3.50	3.45	3.33	3.45

Nemenyi critical distance (p=0.05): 1.06187955146

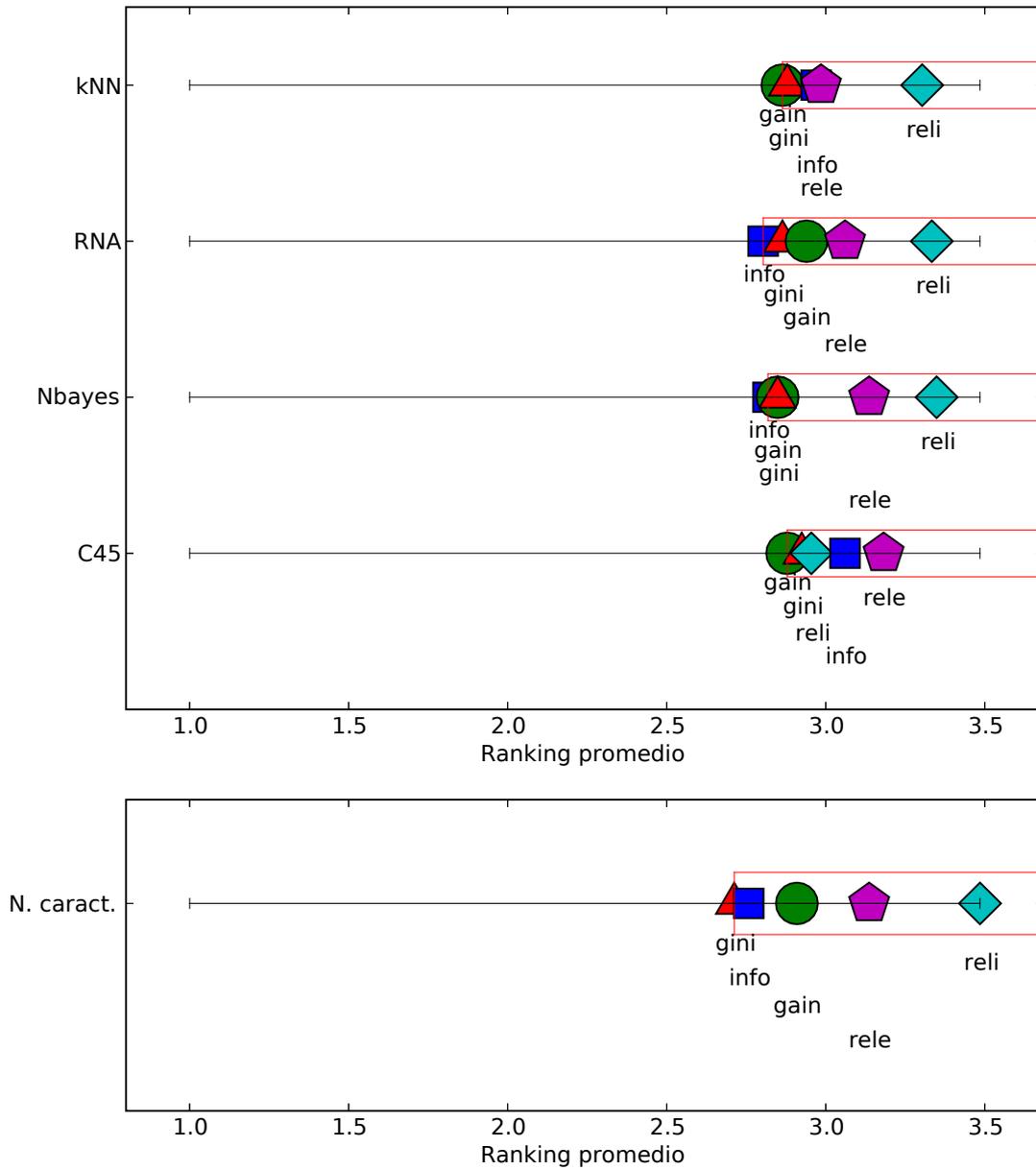
Tabla 4.33: Comparación de medidas para el método de corte pm0.8



<i>Medida indiv. de relevancia</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
info	2.92	3.03	2.79	3.03	2.94	3.03	2.70	3.03
gain	3.02	3.03	2.92	3.03	2.95	3.03	3.00	3.03
gini	2.71	3.59	2.89	3.59	2.65	3.59	2.82	3.59
rele	2.85	2.65	2.91	2.65	2.94	2.65	3.06	2.65
reli	3.50	2.70	3.48	2.70	3.52	2.70	3.42	2.70

Nemenyi critical distance (p=0.05): 1.06187955146

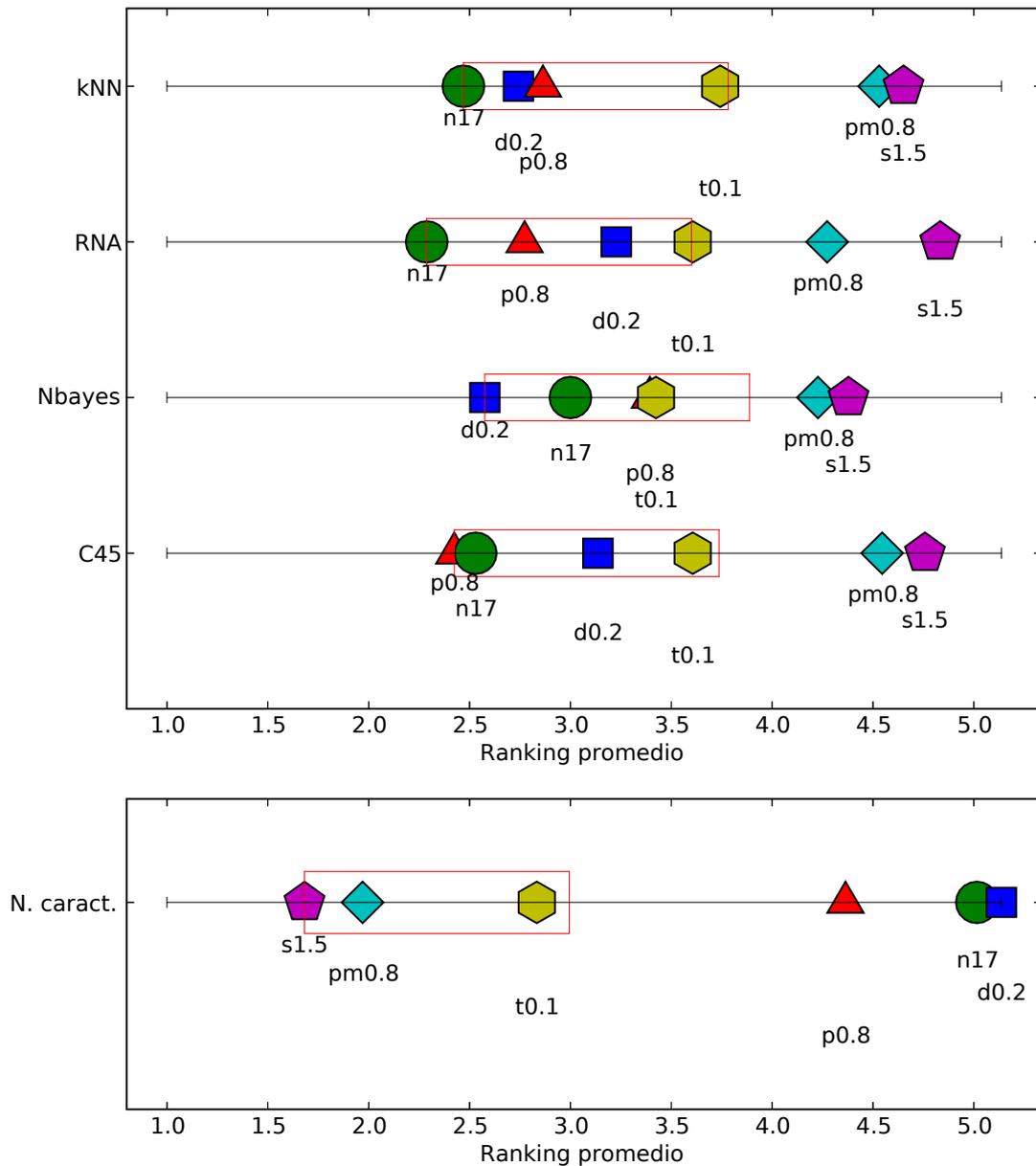
Tabla 4.34: Comparación de medidas para el método de corte d0.2



Medida indiv. de relevancia	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
info	3.06	2.76	2.82	2.76	2.80	2.76	2.97	2.76
gain	2.88	2.91	2.85	2.91	2.94	2.91	2.86	2.91
gini	2.92	2.71	2.85	2.71	2.86	2.71	2.88	2.71
reli	2.95	3.48	3.35	3.48	3.33	3.48	3.30	3.48
rele	3.18	3.14	3.14	3.14	3.06	3.14	2.98	3.14

Nemenyi critical distance (p=0.05): 1.06187955146

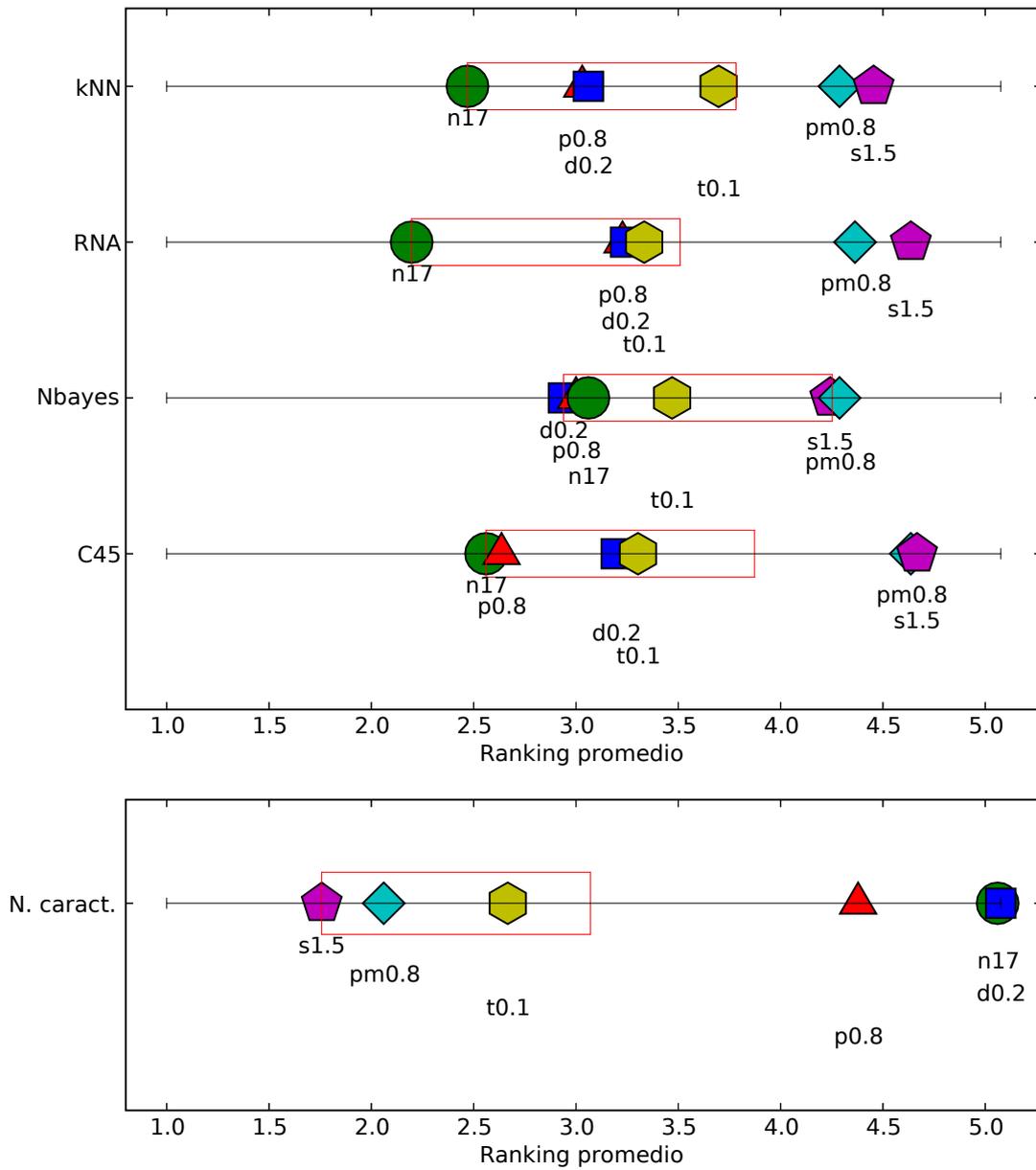
Tabla 4.35: Comparación de medidas para el método de corte s1.5



Método de corte	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
d0.2	3.14	5.14	2.58	5.14	3.23	5.14	2.74	5.14
n17	2.53	5.02	3.00	5.02	2.29	5.02	2.47	5.02
p0.8	2.42	4.36	3.39	4.36	2.77	4.36	2.86	4.36
pm0.8	4.55	1.97	4.23	1.97	4.27	1.97	4.53	1.97
s1.5	4.76	1.68	4.38	1.68	4.83	1.68	4.65	1.68
t0.1	3.61	2.83	3.42	2.83	3.61	2.83	3.74	2.83

Nemenyi critical distance ($p=0.05$): 1.31244797877

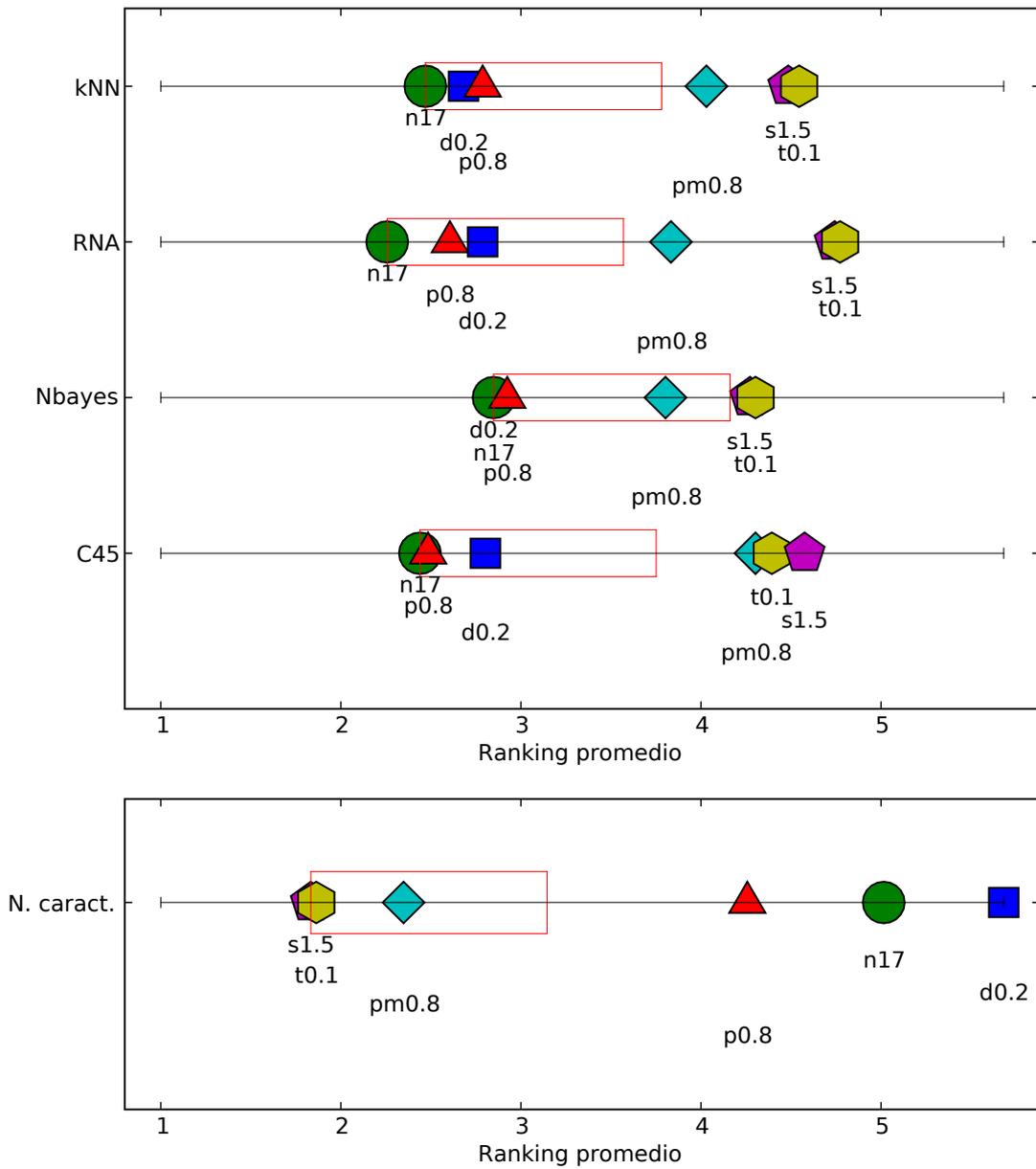
Tabla 4.36: Comparación de métodos de búsqueda para la medida info



Método de corte	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
d0.2	3.20	5.08	2.94	5.08	3.24	5.08	3.06	5.08
n17	2.56	5.06	3.06	5.06	2.20	5.06	2.47	5.06
p0.8	2.64	4.38	3.00	4.38	3.23	4.38	3.03	4.38
pm0.8	4.64	2.06	4.29	2.06	4.36	2.06	4.29	2.06
s1.5	4.67	1.76	4.24	1.76	4.64	1.76	4.45	1.76
t0.1	3.30	2.67	3.47	2.67	3.33	2.67	3.70	2.67

Nemenyi critical distance (p=0.05): 1.31244797877

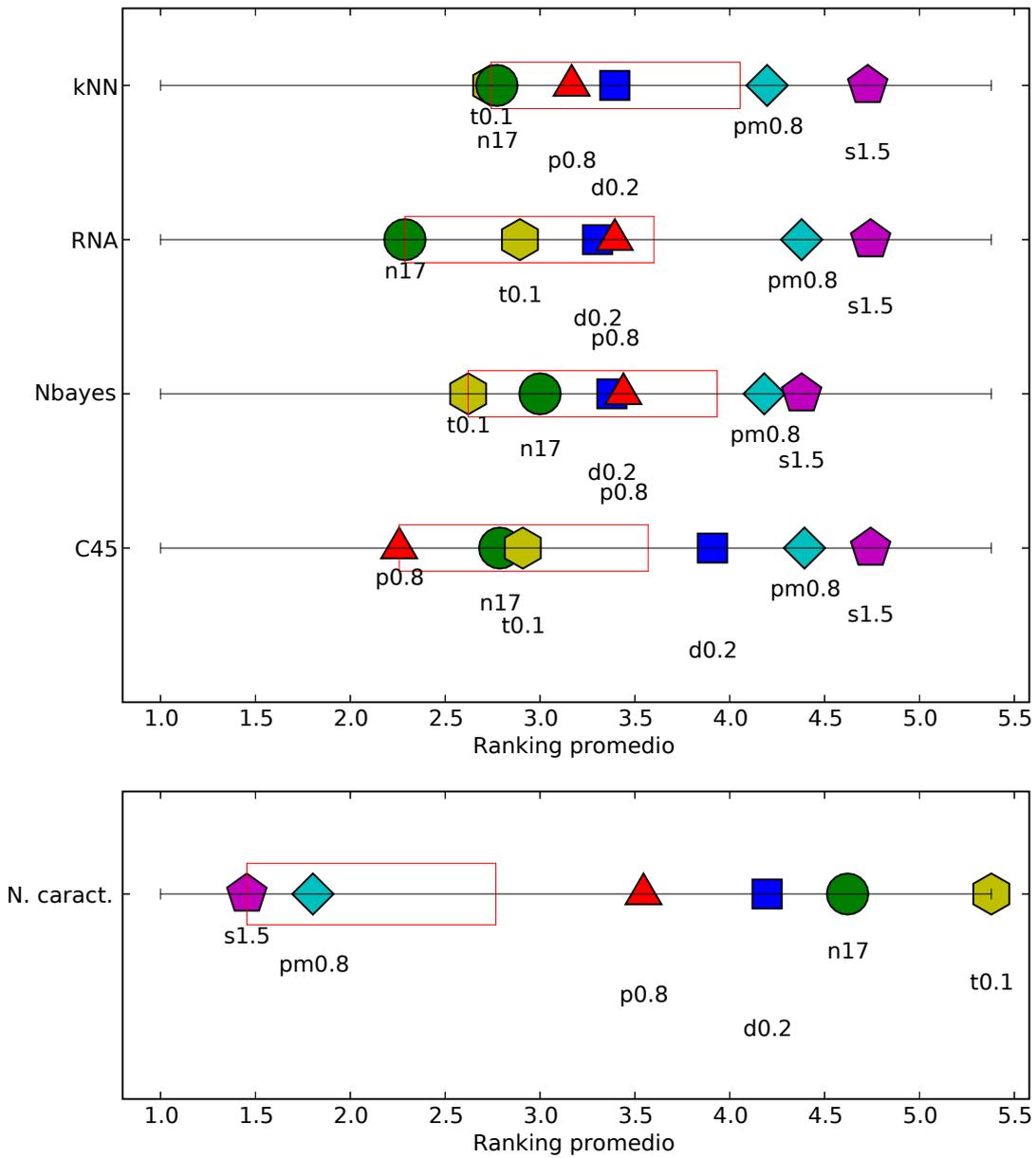
Tabla 4.37: Comparación de métodos de búsqueda para la medida gain



Método de corte	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
d0.2	2.80	5.68	2.85	5.68	2.79	5.68	2.68	5.68
n17	2.44	5.02	2.85	5.02	2.26	5.02	2.47	5.02
p0.8	2.48	4.26	2.92	4.26	2.61	4.26	2.79	4.26
pm0.8	4.30	2.35	3.80	2.35	3.83	2.35	4.03	2.35
s1.5	4.58	1.83	4.27	1.83	4.74	1.83	4.48	1.83
t0.1	4.39	1.86	4.30	1.86	4.77	1.86	4.55	1.86

Nemenyi critical distance (p=0.05): 1.31244797877

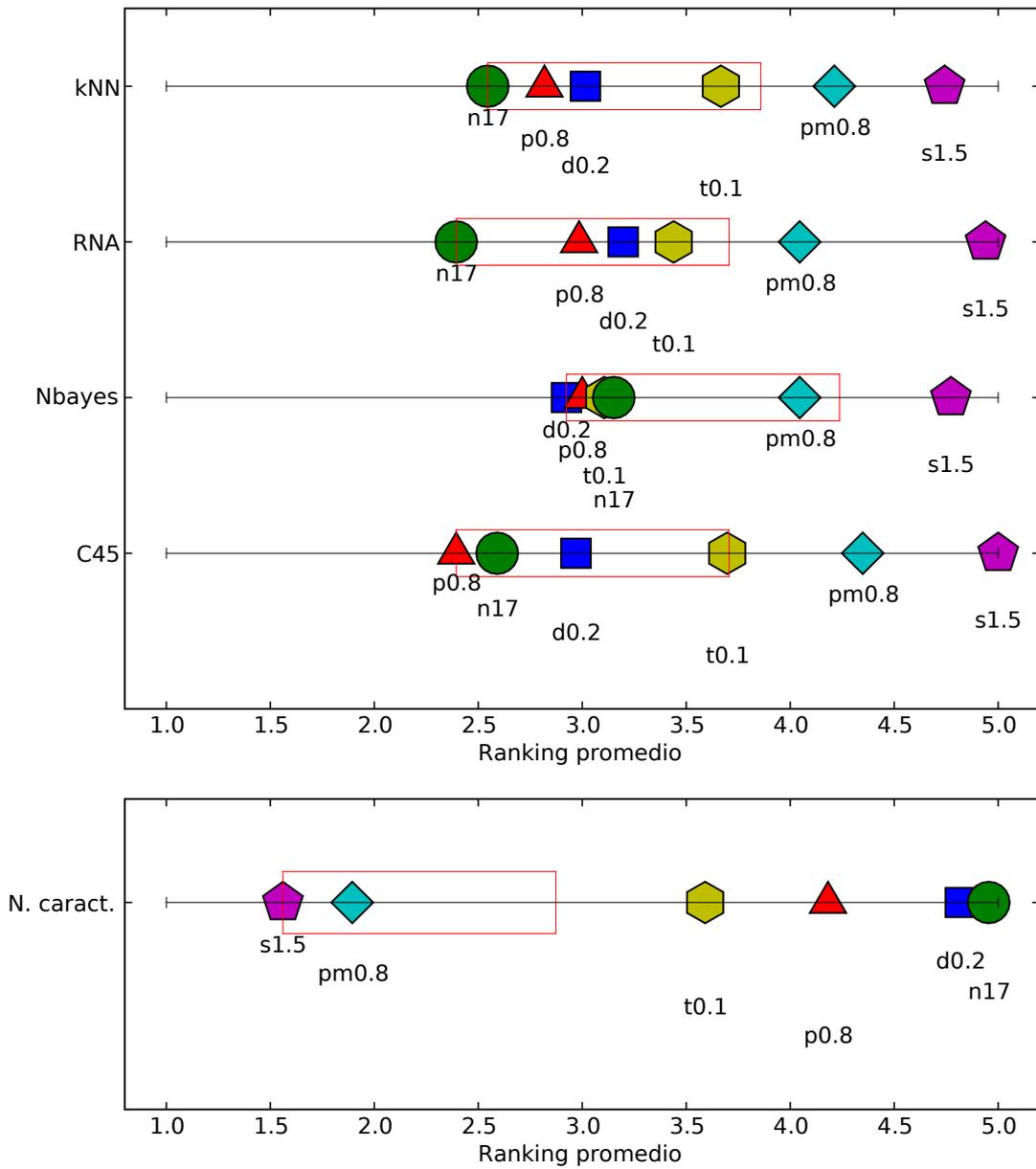
Tabla 4.38: Comparación de métodos de búsqueda para la medida gini



Método de corte	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
d0.2	3.91	4.20	3.38	4.20	3.30	4.20	3.39	4.20
n17	2.79	4.62	3.00	4.62	2.29	4.62	2.77	4.62
p0.8	2.26	3.55	3.44	3.55	3.39	3.55	3.17	3.55
pm0.8	4.39	1.80	4.18	1.80	4.38	1.80	4.20	1.80
s1.5	4.74	1.45	4.38	1.45	4.74	1.45	4.73	1.45
t0.1	2.91	5.38	2.62	5.38	2.89	5.38	2.74	5.38

Nemenyi critical distance (p=0.05): 1.31244797877

Tabla 4.39: Comparación de métodos de búsqueda para la medida reli



Método de corte	C45		Nbayes		RNA		knn	
	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.	R.Ac.	R.n°c.
d0.2	2.97	4.82	2.92	4.82	3.20	4.82	3.02	4.82
n17	2.59	4.95	3.15	4.95	2.39	4.95	2.55	4.95
p0.8	2.39	4.18	3.00	4.18	2.98	4.18	2.82	4.18
pm0.8	4.35	1.89	4.05	1.89	4.05	1.89	4.21	1.89
s1.5	5.00	1.56	4.77	1.56	4.94	1.56	4.74	1.56
t0.1	3.70	3.59	3.11	3.59	3.44	3.59	3.67	3.59

Nemenyi critical distance (p=0.05): 1.31244797877

Tabla 4.40: Comparación de métodos de búsqueda para la medida rele

4.4.2.3. Comparativa global de métodos individuales

En estos métodos el efecto de los distintos criterios de puntos de corte sobre la reducción de características, como era de esperar, es muy acusado. En algunos casos el número de características elegido es forzado. En los casos en los que no es forzado, ninguno de los criterios probados se puede considerar definitivo para todas las medidas.

En la tabla 4.41 se pueden ver todos los resultados de los métodos que valoran características individualmente, incluyendo en la comparación la no realización de selección de características. Se aprecia una relación entre aciertos y menor reducción mucho más fuerte que en los métodos basados en valoración de conjuntos.

Los métodos n17-info, n17-gain, y n17-gini obtienen resultados que mejoran la no selección de características pero con una reducción inferior a los demás métodos. El método p0.8-gini permite una reducción algo mayor sin degradar demasiado los aciertos y, permitiéndonos una pérdida algo mayor, podemos reducir aun más el número de características con t0.1-info o t0.1-gain.

4.4.3. Comparativa global de todos los métodos

Finalmente, hacemos algunas consideraciones globales sobre todos los métodos considerados. La tabla 4.42 muestra la comparativa de una selección de 9 métodos basados en evaluación de conjuntos y 5 basados en evaluación individual.

El mejor método en aciertos resulta ser SBS-Wra, además, con resultados bastante aceptables en reducción, pero, como hemos comentado antes, no siempre es posible usar este método por su costo computacional.

Aunque la mayor reducción la consigue t0.1-gain, es a costa de obtener un peor puesto en aciertos. En general, como esperábamos, los resultados de los métodos de evaluación individual son peores, en el sentido de que, si consiguen buenos resultados en acierto, es a costa de mucho peores resultados en reducción. Por otra parte los métodos individuales son bastante más rápidos.

En la figura 4.10 podemos ver que el comportamiento de los métodos entre clasificadores es similar. Se observa la dependencia entre reducción y porcentaje de acierto, y como para los métodos es difícil superar una línea para acercarse al óptimo

<i>Método de selección</i>	<i>C45</i>		<i>Nbayes</i>		<i>RNA</i>		<i>knn</i>	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
NoFS-	12.88	26.91	12.61	26.91	10.02	26.91	12.35	26.91
d0.2-info	14.12	23.58	11.58	23.58	14.26	23.58	11.45	23.58
d0.2-gain	14.61	23.09	13.14	23.09	14.64	23.09	13.80	23.09
d0.2-gini	12.80	26.91	12.61	26.91	12.98	26.91	12.35	26.91
d0.2-reli	17.35	21.21	16.41	21.21	16.11	21.21	16.03	21.21
d0.2-rele	13.67	21.71	12.39	21.71	14.55	21.71	14.15	21.71
n17-info	10.89	22.98	13.24	22.98	9.65	22.98	10.39	22.98
n17-gain	11.27	22.98	13.39	22.98	9.18	22.98	10.62	22.98
n17-gini	11.03	22.98	13.17	22.98	9.98	22.98	11.17	22.98
n17-reli	12.33	22.98	14.62	22.98	10.59	22.98	12.70	22.98
n17-rele	11.65	22.98	13.27	22.98	10.32	22.98	11.27	22.98
p0.8-info	10.09	18.73	15.38	18.73	12.65	18.73	12.42	18.73
p0.8-gain	11.79	18.73	14.09	18.73	14.73	18.73	13.26	18.73
p0.8-gini	10.83	18.73	13.53	18.73	12.50	18.73	12.35	18.73
p0.8-reli	9.35	18.73	16.08	18.73	14.56	18.73	14.45	18.73
p0.8-rele	10.73	18.73	12.94	18.73	13.65	18.73	13.35	18.73
pm0.8-info	21.79	7.17	19.35	7.17	19.55	7.17	20.73	7.17
pm0.8-gain	21.85	7.35	19.88	7.35	20.42	7.35	20.33	7.35
pm0.8-gini	21.42	7.59	18.50	7.59	18.70	7.59	19.38	7.59
pm0.8-reli	20.02	9.18	20.53	9.18	22.00	9.18	20.32	9.18
pm0.8-rele	19.76	8.65	18.35	8.65	18.95	8.65	19.14	8.65
s1.5-info	22.74	6.00	19.61	6.00	22.03	6.00	21.50	6.00
s1.5-gain	22.17	6.48	19.97	6.48	21.86	6.48	20.94	6.48
s1.5-gini	22.71	5.91	20.08	5.91	22.55	5.91	21.62	5.91
s1.5-reli	21.26	8.06	21.21	8.06	23.02	8.06	22.41	8.06
s1.5-rele	23.18	7.08	21.70	7.08	22.80	7.08	22.20	7.08
t0.1-info	17.00	11.33	15.21	11.33	16.17	11.33	17.18	11.33
t0.1-gain	15.15	10.36	16.09	10.36	15.47	10.36	16.65	10.36
t0.1-gini	21.39	5.59	20.62	5.59	22.97	5.59	21.83	5.59
t0.1-reli	12.65	26.91	12.61	26.91	13.23	26.91	12.35	26.91
t0.1-rele	17.52	16.36	13.86	16.36	15.92	16.36	17.30	16.36

Tabla 4.41: Comparación de los métodos de selección de características que usan valoración individual de las características para cada algoritmos de aprendizaje

Método de selección	C45		Nbayes		RNA		knn	
	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.	R.Ac.	R.n ^o c.
NoFS-	7.39	13.85	8.23	13.77	6.79	13.68	8.11	13.73
SFS-Liu	7.70	7.18	8.18	7.21	8.58	6.97	8.20	6.86
SFS-IEP	8.30	6.30	9.21	6.32	9.30	6.14	8.76	6.14
SFS-RSC	7.38	7.20	8.36	7.11	8.58	6.92	8.38	6.92
SFS-Wra	8.33	5.92	5.82	5.56	8.82	6.44	6.77	6.53
SBS-Wra	7.71	4.67	6.26	5.80	6.48	7.06	5.67	6.53
LVW-Wra	8.79	7.14	6.86	7.21	7.29	6.97	8.47	7.20
SA-Wra	8.47	6.39	6.61	6.91	7.56	7.71	7.03	7.02
GA-Wra	8.67	8.45	7.97	8.12	7.53	7.67	8.21	7.83
TS-Wra	8.45	7.35	8.21	7.17	9.12	6.33	8.94	6.76
n17-info	6.23	11.92	8.52	11.82	6.74	11.68	7.02	11.74
n17-gain	6.89	11.92	8.52	11.82	6.55	11.68	7.12	11.74
n17-gini	6.74	11.92	8.58	11.82	6.92	11.68	7.50	11.74
t0.1-info	9.97	5.38	9.36	5.32	10.03	5.12	10.45	5.11
t0.1-gain	8.97	4.39	9.32	4.05	9.71	3.94	9.38	4.15

Tabla 4.42: Comparación conjunta de los mejores métodos de selección de características que usan valoración individual o de subconjuntos

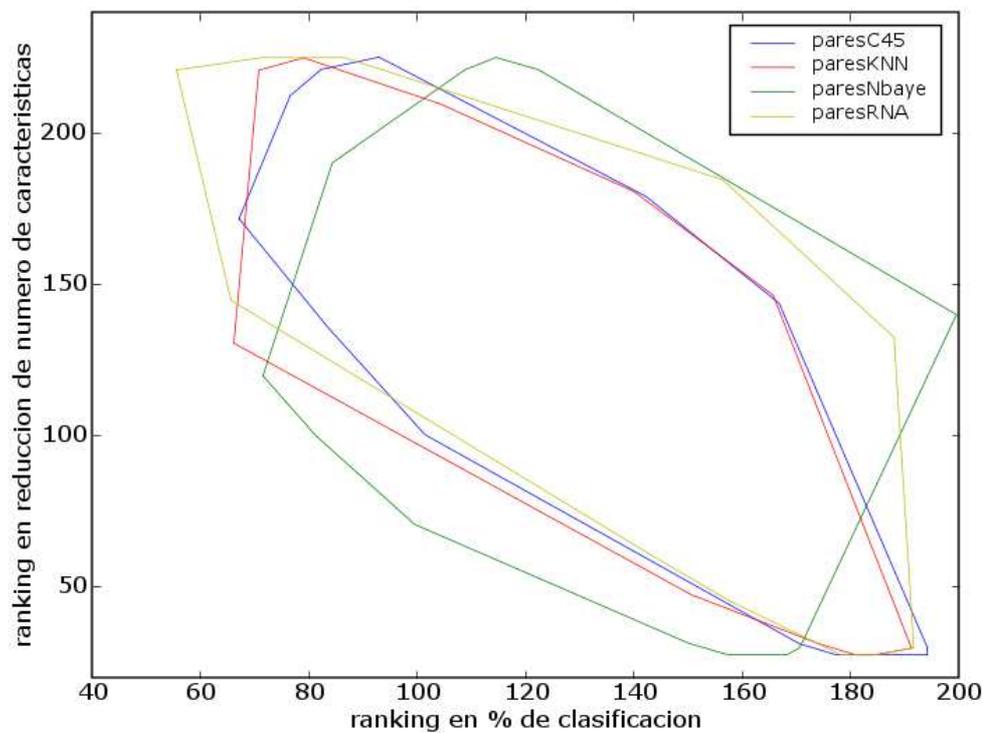


Figura 4.10: Envolvente convexa de todos los métodos para cada clasificador

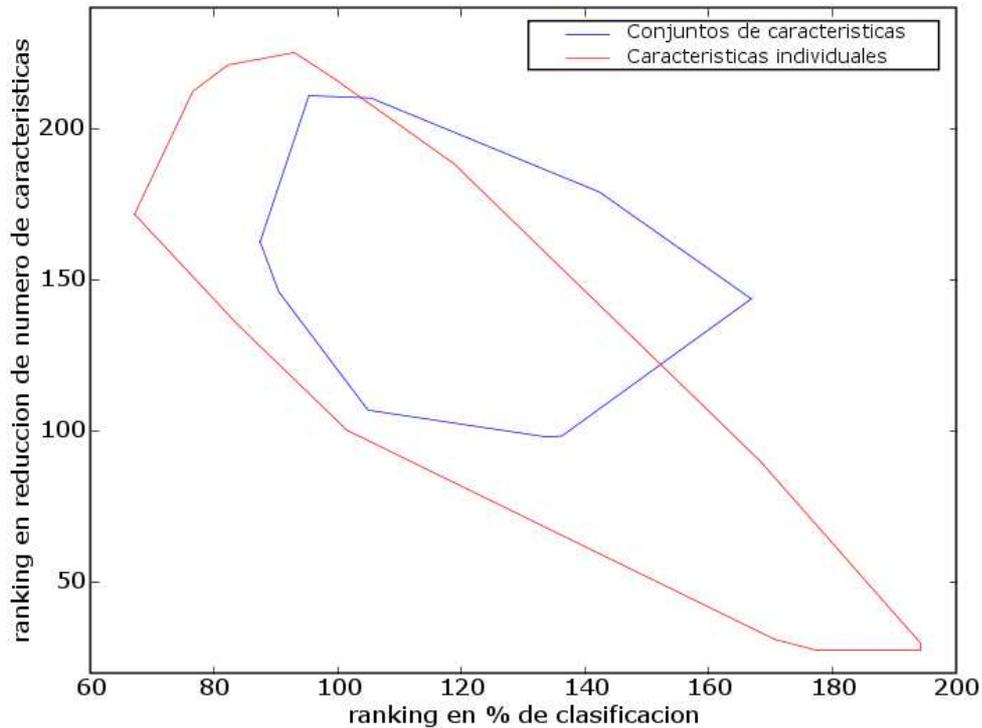


Figura 4.11: Envolverte convexa según tipo de método para C45

(o reducen o mejoran, no es factible todo a la vez). En la figura 4.11, podemos ver como los métodos de selección individual presentan una mayor variación con respecto a la reducción de características.

4.4.4. Mejora del porcentaje de acierto al aplicar selección de características

Ante la cantidad de resultados que nos indican que a mayor reducción de características se obtienen peores resultados, parece interesante corroborar que, además de las otras ventajas ya descritas, también se puede obtener una mejora en porcentaje de acierto, de la aplicación de selección de características. En la tabla A.4, se muestran los resultados sobre el porcentaje de acierto que se obtiene con cada uno de los conjuntos de datos. Mientras que ciertamente en algunos no se obtiene mejora, se puede apreciar como hay muchos conjuntos de datos en los que se produce una mejora. Aunque los resultados no son commensurables, el hecho que la media indique una mejora del 0.2% nos da una pista de que sí se produce una mejora,

4.4.5. Diferentes resultados según tipos de problemas

Si analizamos los métodos de selección en grupos de problemas atendiendo a sus propiedades, como: número de características, tipo de las características o número de ejemplos; podemos ver si los resultados varían dependiendo del grupo del que forma parte el problema. En general los resultados presentan similitudes con el resultado global, pero aparecen algunas variaciones.

Aunque hay muchas variaciones que no se pueden considerar significativas, si que hay algunas que aconsejan o descartan métodos en un grupo de problemas mientras que en otro no. Como ejemplo, se pueden comparar las figuras 4.12, 4.13 y 4.14, que representan los resultados para tres subgrupos de problemas, con la figura de la tabla 4.10 que incluye todos los conjuntos de datos. En ellas podemos ver como podemos descartar usar *BB* con *RNA* en conjuntos de menos de 8 características, mientras que no se aprecia una diferencia significativa en conjuntos de datos de entre 9 y 20 características. Por otra parte, en conjuntos con más de 20 características, se pueden descartar, con algunos clasificadores, los métodos de búsqueda *SBS* y *LVF*.

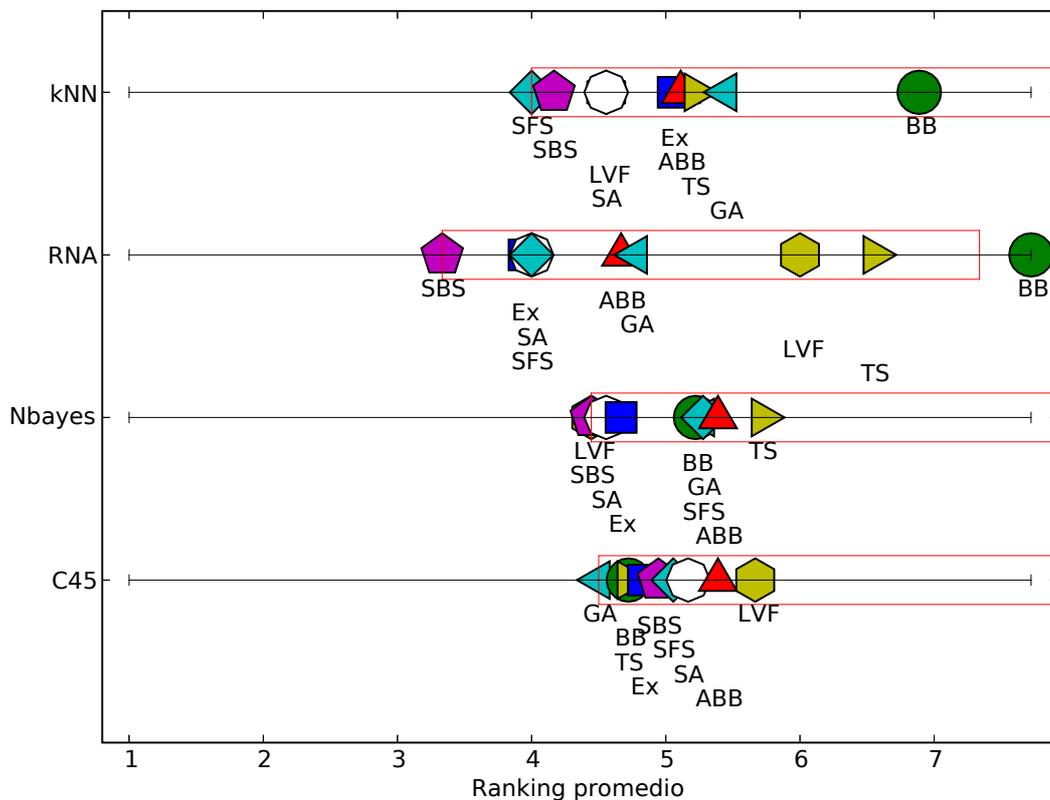


Figura 4.12: Comparación de métodos de búsqueda para la medida Liou en problemas con menos de 8 características

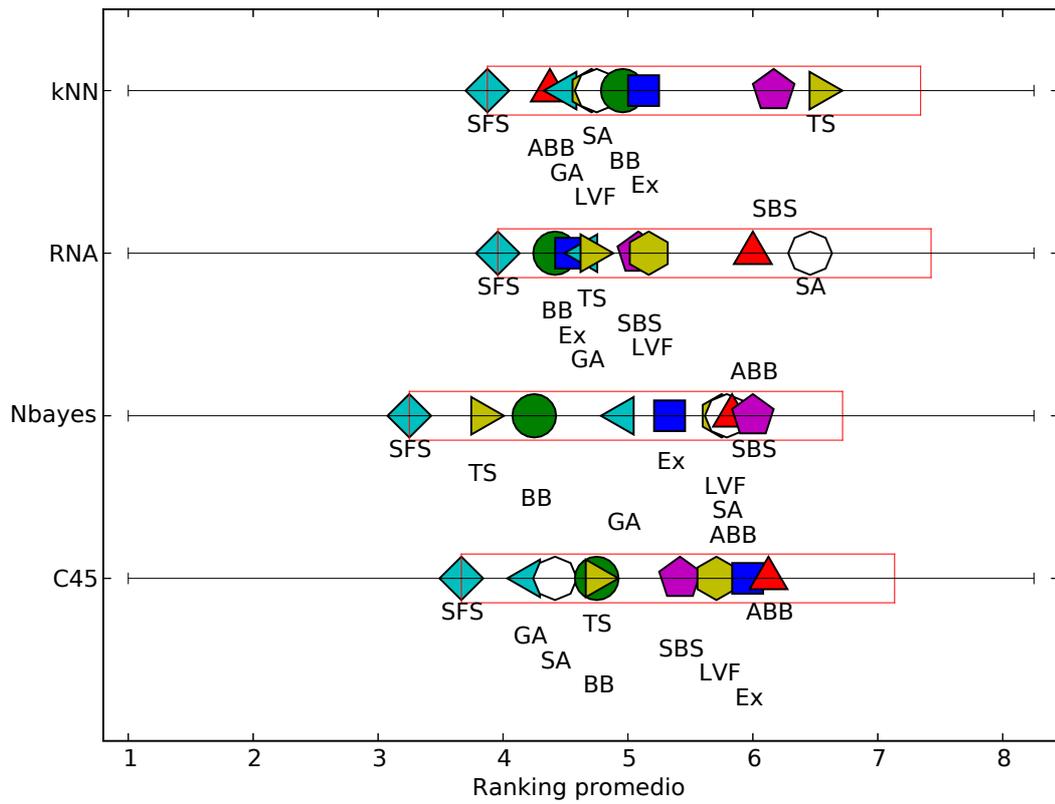


Figura 4.13: Comparación de métodos de búsqueda para la medida Liu en problemas con entre 9 y 20 características

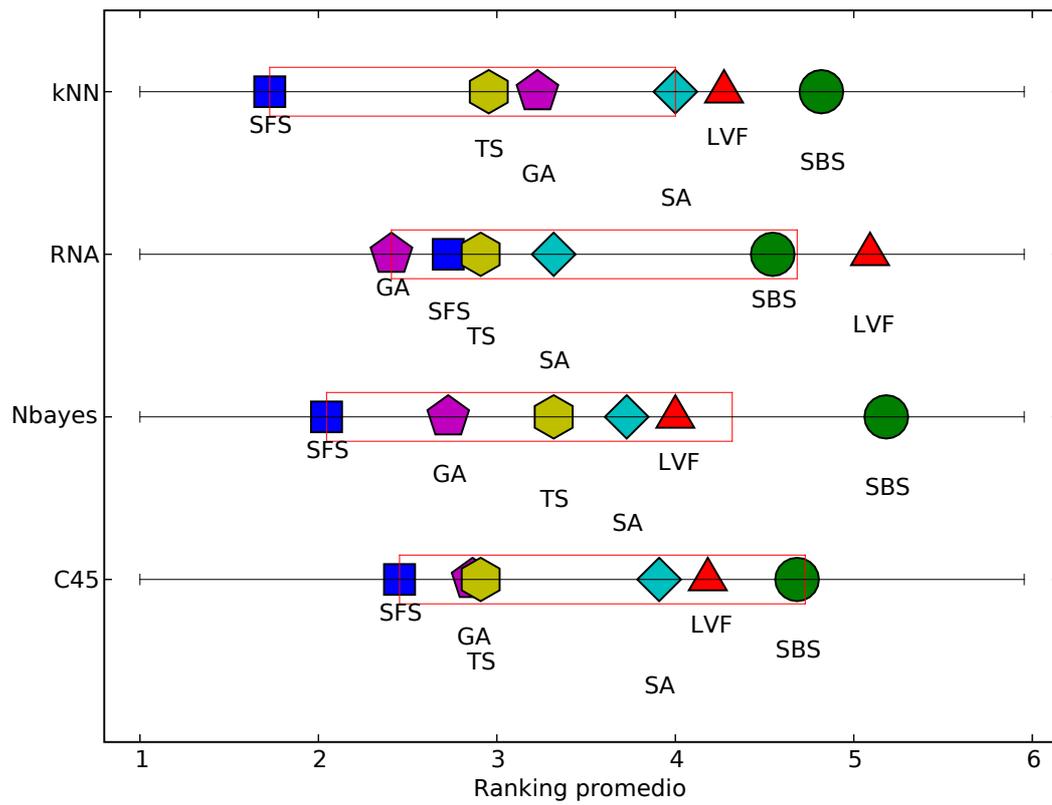


Figura 4.14: Comparación de métodos de búsqueda para la medida Liu en problemas con más de 20 características

Capítulo 5

Sistema inteligente para selección de características

Los resultados experimentales obtenidos en el desarrollo de este trabajo, y otros recogidos en la bibliografía, nos permiten asegurar que no existe un único algoritmo universalmente válido para realizar una selección de características acertada. Así, ante un problema nuevo, se plantea la cuestión de decidir qué método es el más idóneo para aplicar. En este capítulo, pretendemos aportar una respuesta a esta cuestión mediante una combinación de conocimiento experto y experimentación. Por tanto, la solución se podrá implementar en términos de un sistema experto que asesore en esta decisión.

Siguiendo en esta línea, para abordar la solución del problema de selección de características, se plantea la conveniencia de aprovechar todo el esfuerzo de desarrollo realizado y ofertar un servicio extendido. Si además de las propiedades del problema se proporciona un conjunto de datos representativos, el sistema no sólo asesorará, sino que ejecutará la propia recomendación. Es decir, aplicará el método o métodos propuestos y dará como respuesta el conjunto de características finalmente seleccionado.

Con estas dos ideas en mente, diseñamos un sistema inteligente asesor y selector. Planteamos la creación de este sistema experto en selección de características aplicando una metodología combinada entre Ingeniería del Conocimiento e Ingeniería del Software. Los pasos seguidos son:

- Análisis de los requisitos del sistema

- Diseño del sistema
- Adquisición del conocimiento
 - Extracción automática
 - Inclusión de conocimiento experto
- Implementación
- Validación

A continuación, describimos con detalle la aplicación de cada uno de estos pasos en la resolución del problema que nos ocupa.

5.1. Requisitos del sistema

En esta sección, describimos el conjunto de requisitos funcionales y no funcionales del sistema a crear. En primer lugar, delimitamos claramente el cometido del sistema en función de las necesidades que se pretenden cubrir. El usuario potencial de este sistema sería cualquier investigador o ingeniero que precise resolver un problema de clasificación, ya que en cualquier contexto puede ser relevante realizar una selección de características eficaz. Este potencial cliente espera una respuesta que le indique qué método de selección de características usar y, con ese método, obtener una selección adecuada para resolver su problema de clasificación.

El objetivo principal del sistema es asesorar sobre el método de selección de características más adecuado para aplicar a un problema de clasificación. Las respuestas que proporcione deberán estar debidamente justificadas en base al conocimiento incorporado. En una fase posterior, se permitirá usar los métodos recomendados para realizar la selección de características.

El sistema empleará como entrada un conjunto de datos de ejemplos sobre el problema. Como respuesta, dará una lista ordenada de métodos, o combinaciones de métodos que sugiere como procedimientos más efectivos para determinar las características más relevantes del problema.

Como entradas del sistema, consideramos las que hemos estado manejando en la descripción de los problemas empleados en el estudio empírico. Son datos fácilmente obtenibles sobre cualquier problemas de clasificación que se vaya a abordar. La relación es:

- Número de características
- Tipo de las características: categóricas, medidas continuas o una mezcla de ambas.
- Número de ejemplos
- Número de clases del problema
- Complejidad del problema: podría usarse el porcentaje de clase mayoritaria o el porcentaje de acierto de naive-Bayes.
- Si está disponible un conjunto de datos sobre el problema, número de ejemplos.

Además interesa conocer la importancia relativa de los siguientes objetivos para la selección de características:

- Calidad de los resultados de clasificación
- Reducción de características
- Tiempo de cálculo

Se desea hacer lo más extensivo posible el uso del sistema, por lo que el objetivo final será que la interfaz de usuario sea una página o aplicación web. El tipo de información a proporcionar es simple, por lo que una interfaz estándar tipo formulario es más que suficiente. Los requisitos para la interfaz de usuario serán: simplicidad, intuitiva para alguien habituado a usar un navegador e interactiva, de forma que durante la ejecución de los procesos el sistema no parezca quedarse bloqueado.

Dos son las fuentes de conocimiento principales a considerar para este problema:

- Conocimiento experto
- Ejemplo: resultados de casos previos

Ambas fuentes han de usarse para obtener el conocimiento disponible y conseguir el mejor sistema posible. En este caso, existen conceptos que no pueden definirse de modo totalmente preciso y, por tanto, es conveniente usar mecanismos que permitan manejar tales fuentes de vaguedad y de falta de precisión.

Respecto a la parte de procesamiento del sistema, los requisitos son: *i)* generalidad, para trabajar con la mayor parte de problemas de clasificación posibles, y *ii)* eficiencia. Al margen de ello son requisitos deseables en cualquier sistema software, la facilidad de extensión y mantenimiento, y la documentación efectiva.

Además de lo anterior, es importante tener en cuenta que el problema de la selección de características continúa abierto a nuevos enfoques y propuestas que mejoren siquiera parcialmente el estado actual. El sistema debe ser extensible, preparado para incorporar tanto nuevos métodos de selección de características como conocimiento relativo al uso más eficaz de los métodos disponibles.

5.2. Diseño

A partir de los requisitos planteados en el apartado anterior, describimos una solución en forma de sistema experto. La estructura general del mismo se corresponde con la estándar de sistemas basados en conocimiento [Ign91] y se muestra en la figura 5.1. A continuación, detallamos los componentes del sistema experto y su desarrollo.

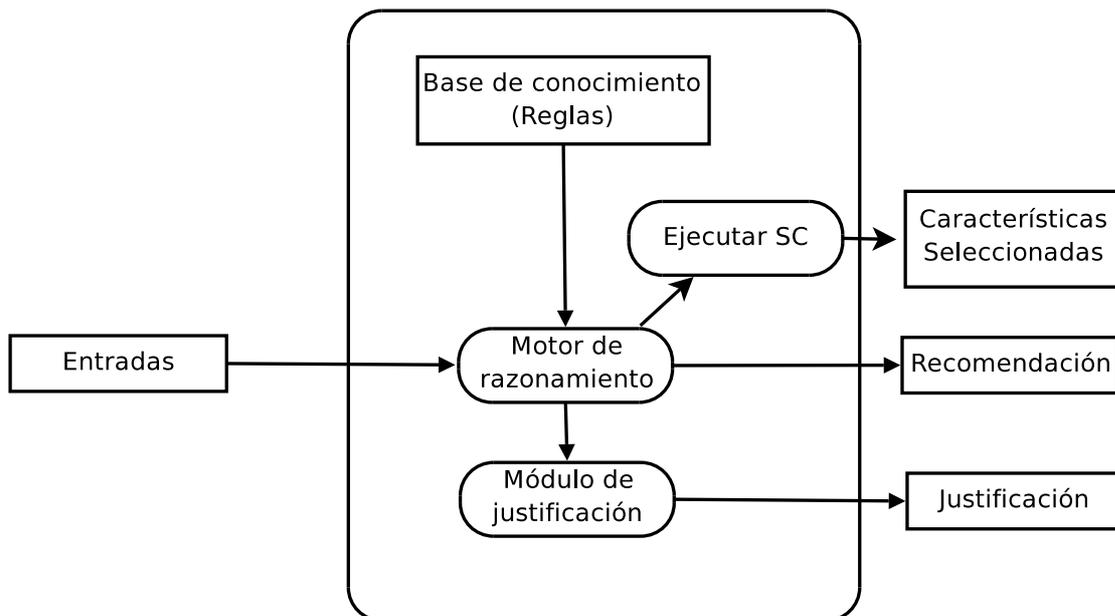


Figura 5.1: Esquema del sistema experto desarrollado

5.2.1. Base de conocimiento

Un aspecto fundamental en el diseño de cualquier sistema experto es la representación del conocimiento a usar. La representación está directamente relacionada con los procedimientos para manejar el conocimiento, en particular, para hacer inferencias. Por el tipo de conocimiento disponible (relaciones causales) y la necesidad de gestionar la falta de precisión en algunos de los conceptos y términos usados, nos decantamos por el uso de reglas lingüísticas [Zad75a, Zad75b, Zad75c] fundamentadas en la Teoría de los Conjuntos Difusos [Zad65, KY95].

Para el dominio de cada variable de entrada cuantificable se ha realizado una partición de etiquetas lingüísticas. La partición se basa en nuestro conocimiento experto sobre el problema de la selección de características. Evidentemente, las características de los problemas considerados en el capítulo 4 ha tenido influencia sobre la definición particular de las etiquetas.

- *Nº de características*, con tres etiquetas: *Grande*, *Medio* y *Bajo*, representadas en la figura 5.2.

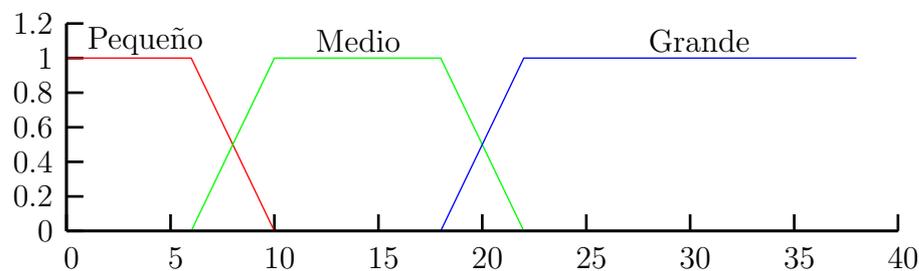


Figura 5.2: Etiquetas lingüísticas de la variable que indica el número de características del problema

- *Tipo de las características* es una variable discreta con tres etiquetas: *Discretas*, cuando todas las características son categóricas; *Continuas*, cuando todas son medidas; y *Mezcla*, cuando hay ambos tipos de características. El valor de pertenencia a cada una de estas etiquetas está completamente determinado en todos los casos como 0 o 1 (no es difusa), y es disjunto (no puede pertenecer a más de una).
- *Nº de ejemplos*, con tres etiquetas: *Grande*, *Medio* y *Bajo*, representadas en la figura 5.3.

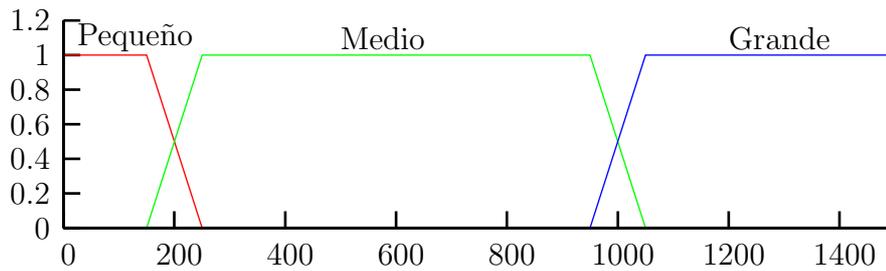


Figura 5.3: Etiquetas lingüísticas de la variable que indica el número de ejemplos del problema

- Importancia de obtener mejores *Resultados*, una única etiqueta *Importante*, cuyo grado de pertenencia indica la relevancia que el usuario da a este factor. Se representa en la figura 5.4.
- Importancia de la *Reducción* de características, igual que la anterior.
- Importancia de un menor *Tiempo* de ejecución, igual que las dos anteriores.

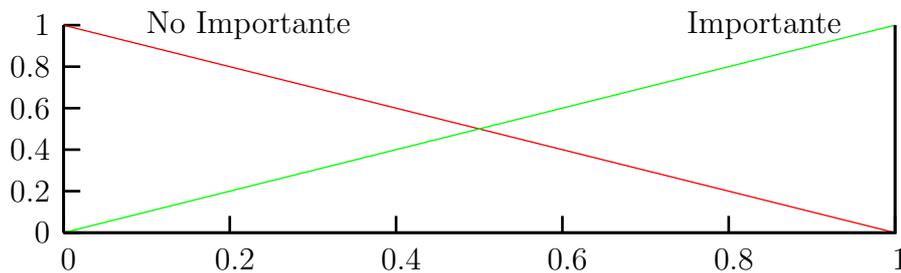


Figura 5.4: Etiqueta lingüística de las variables que indican la importancia de cada uno de los objetivos de la selección de características

La variable de salida será la recomendación de uso de cada método. Se expresa como conjunto difuso sobre el dominio (discreto) de todos los métodos de selección disponibles.

Habrà una por cada método a recomendar: $\{\mu_1/met_1, \mu_2/met_2, \dots, \mu_n/met_n\}$.

Las reglas de la base de conocimiento serán de este tipo:

SI x_1 es A **Y** x_2 es B **Y** ...
ENTONCES y_1 es C (con grado μ_C) **Y** y_2 es D (con grado μ_D) **Y** ...

El antecedente de la regla estará formado por la conjunción de proposiciones (lingüísticas o no) sobre las variables de entrada. El consecuente de la regla será una asignación, con un grado de pertenencia, a cada una de las variables de salida. Hemos incluido este grado de pertenencia en las salidas de la regla para que nuestro sistema pueda recomendar varios métodos con diferentes grados.

Un ejemplo de regla de nuestro sistema podría ser:

SI *reducción es Importante* **Y** *nº características es Bajo*
ENTONCES *SBS-Wra es Usar*(con grado 1) **Y** *SFS-IEP es Usar*(con grado 0.8)

5.2.2. Motor de razonamiento

El motor de razonamiento será simple, el habitual en los sistemas difusos tipo TSK [KY95] adaptado para el tipo de salidas de las reglas descritas. En primer lugar, se calcula el grado de disparo (μ_i) de cada regla (i) con la fórmula (5.1), denotando por A_k a cada una de las etiquetas y por $\mu_{A_k}(x_k)$ el grado de pertenencia del valor de la variable x_k a dicha etiqueta.

$$\mu^i = \mu_{A_1^i}(x_1)\mu_{A_2^i}(x_2) \dots \mu_{A_N^i}(x_N) \quad (5.1)$$

Para calcular la etiqueta de salida, se calcula el grado de pertenencia de cada método (y_j) con la fórmula (5.2). K es el número de reglas cuyo μ^i es distinto de 0 y para las que el grado de pertenencia del método i al conjunto de salida de la regla es distinto de 0.

$$y_j = \frac{\sum_{i=1}^K \mu^i \mu_{Usar_j}^i}{\sum_{i=1}^K \mu^i} \quad (5.2)$$

5.2.3. Módulo de justificación

El módulo de justificación mostrará las reglas disparadas que han llevado a obtener la recomendación de métodos de selección. De esta forma, podrá verse cuales son los factores que la han determinado y la influencia de cada uno de los requisitos impuestos por el usuario: mejora de resultados, reducción de características, tiempo de proceso.

Las reglas se presentará ordenadas considerando dos criterios. Primero en orden decreciente de su grado de disparo. En segundo lugar se considerará el orden decreciente del valor de pertenencia de los métodos recomendados a sus respectivos conjuntos difusos.

5.3. Adquisición del conocimiento

La adquisición del conocimiento, para el sistema que diseñamos, tiene una doble vertiente en función de la fuente del mismo: aportado por el experto, o extraído automáticamente de ejemplos de problemas resueltos. En nuestro caso la base de ejemplos la constituyen los experimentos realizados en el capítulo 4.

5.3.1. Conocimiento de expertos

La primera fuente de conocimiento que consideramos es el experto. Durante el desarrollo de este trabajo de investigación, hemos estudiado detenidamente el problema de la selección de características; hemos aplicado métodos diversos a un nutrido y variado conjunto de problemas; hemos recabado información sobre múltiples casos prácticos descritos en la literatura; hemos reflexionado y mantenido largas discusiones de investigación sobre el problema; finalmente, haber realizado el análisis experimental descrito en el capítulo 4 nos ha ayudado a estructurar el conocimiento disponible acerca del problema. En definitiva, hemos estado trabajando sobre este problema con bastante intensidad durante los últimos años. Esto nos ha permitido alcanzar un elevado conocimiento sobre el problema para actuar como expertos en el desarrollo de este sistema. Existe, en este caso, una ventaja: el experto y el ingeniero del conocimiento coinciden, por lo que es más fácil el entendimiento entre *ambos*.

El proceso de extracción de conocimiento se ha realizado siguiendo una metodología simplificada basada en rejilla. El resultado de este proceso es una base inicial de reglas con el aspecto de las que se muestran a continuación:

SI *resultados es Importante* **Y** *nº de características es Bajo*
ENTONCES *SBS-Wra es Usar*(con grado 1)

SI *reducción es Importante* **Y** *tiempo es Importante*
ENTONCES *SFS-IEP es Usar*(con grado 1) **Y** *SFS-RSC es Usar*(con grado 0.7)

SI *resultados es Importante* **Y** *n° de características es Alto*
ENTONCES *SFS-Liu es Usar(con grado 1)* **Y** *LVF-Liu es Usar(con grado 0.8)*

Esta base de reglas aportadas por el experto podrá ser ampliada y perfeccionada conforme se vaya adquiriendo mayor conocimiento sobre el problema.

5.3.2. Aprendizaje Automático

El conocimiento experto sobre el problema no es completo, y la precisión con la que se pueden expresar los hechos conceptuales es limitada. Para ayudar a mejorar, depurar y, ante nueva experimentación, ampliar fácilmente el conocimiento, es interesante considerar ejemplos prácticos. El objetivo es complementar y matizar el conocimiento experto con información y conocimiento extraído de resultados experimentales. Por ello, aplicamos métodos de extracción de reglas basados en técnicas de Computación flexible.

En los deseos del cliente del sistema, identificamos tres objetivos diferentes que condicionan la elección de un método de selección de características:

- Obtener buenos resultados
- Reducir el número de características empleadas
- Emplear un menor tiempo de proceso

Desafortunadamente, estos objetivos se contraponen en muchas ocasiones. En general, para un menor tiempo de proceso, se encuentran peores soluciones tanto en resultados como en reducción y, basándonos en la experimentación realizada, hemos visto que se pueden conseguir mayores reducciones sacrificando los resultados y viceversa. Por ello, consideramos útil que el usuario indique la importancia relativa que asigna a cada uno de ellos, con el objetivo de encontrar una solución equilibrada para su problema de selección. Con la idea de cumplir este objetivo, se han incluido las variables: *resultados*, *reducción* y *tiempo* en la base de conocimiento.

Aquí generaremos reglas que nos indiquen una recomendación para cada uno de los objetivos identificados. Para la extracción de estas reglas, proponemos utilizar una división en regiones similar a la usada en el método de Wang y Mendel [WM92].

Para cada objetivo, nos planteamos generar una batería de reglas. Para ellas usamos las dos variables que consideramos más relevantes y la variable del objetivo

correspondiente. En la figura 5.5, se muestra la división de regiones tomada para las reglas que tendrán como objetivo el mejor resultado.

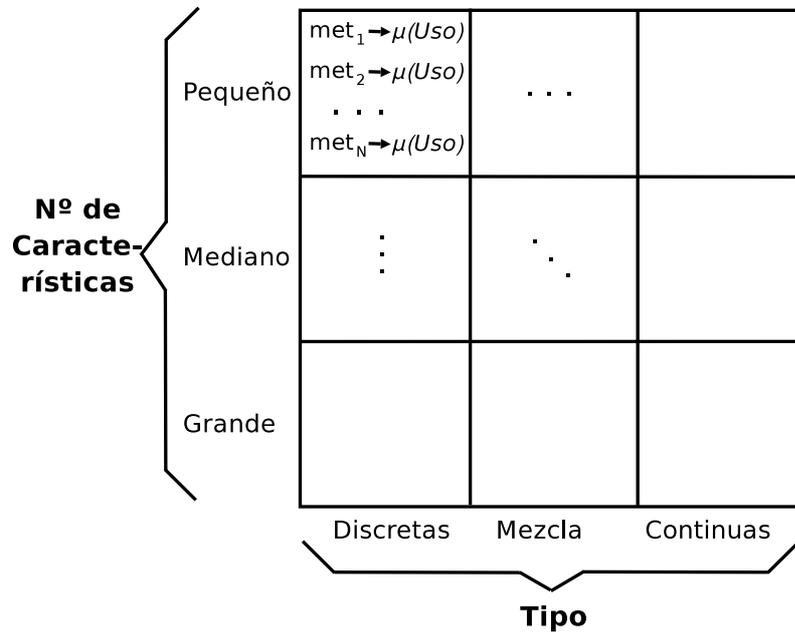


Figura 5.5: Regiones para la generación de reglas basadas en los mejores resultados obtenidos

En cada región, se generará una regla. Ésta tendrá como antecedentes los que determinan la región y, además, uno que la condicione al objetivo concreto, en este caso: “*resultados es Importante*”. La salida de esa regla se basará en los resultados obtenidos en la experimentación con problemas que pertenezcan a la región. La salida es un conjunto difuso sobre el conjunto referencial de métodos de clasificación. Para determinar el grado de pertenencia asignado a cada método, tomaremos la posición en el ranking promedio que el método ha obtenido en los problemas de la región considerada y lo dividiremos entre el número total de problemas.

La batería de reglas generada para el objetivo de reducción se basa en las mismas regiones. En la figura 5.6, se muestran las regiones consideradas para el objetivo de emplear un menor tiempo de proceso, donde se consideran otras variables, que son las que tienen influencia más directa en el tiempo de proceso: el número de características y el número de ejemplos.

Una limitación importante en el proceso de extracción de reglas es el número, relativamente pequeño, de ejemplos de que se dispone. El elevado coste computacional hace difícil disponer de un conjunto de ejemplos mayor. Nuestra propuesta para este sistema es que sea dinámico y “vivo”. Va a quedar accesible en Internet a disposición de cualquier usuario interesado. El objetivo es ir recopilando ejemplos

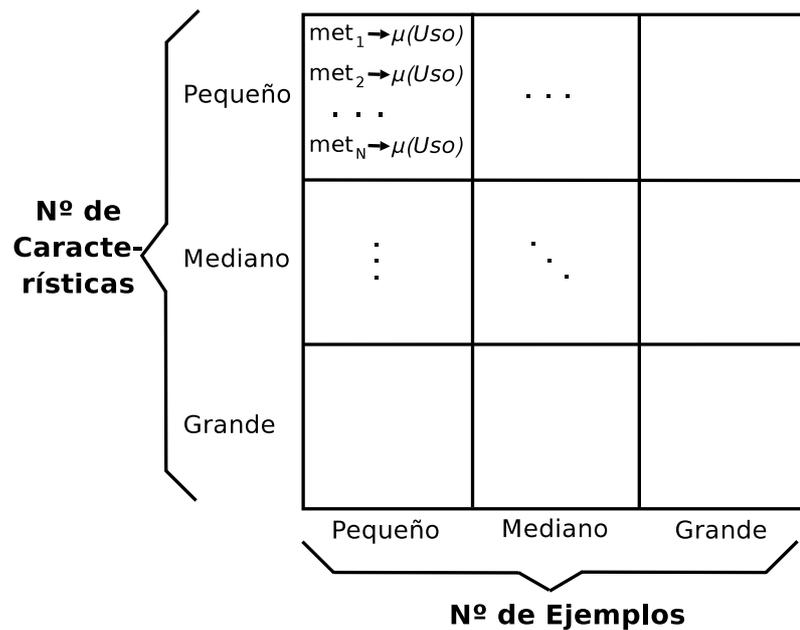


Figura 5.6: Regiones para la generación de reglas basadas en el menor tiempo de ejecución

derivados de las consultas de los usuarios e ir ampliando la base de ejemplos. Los problemas que se vayan incorporando serán evaluados de forma completa, según el diseño experimental del capítulo previo. Programaremos estas ejecuciones en los tiempos ociosos de nuestros servidores de cálculo. Y periódicamente, lanzaremos el proceso de extracción de reglas para ampliar o mejorar la base de reglas.

Cuando dispongamos de más datos podremos aplicar un método que extraiga reglas en espacios con más de dos dimensiones, utilizando todas las variables consideradas. Con el número actual de ejemplos, ese enfoque daría lugar a un espacio con muchas regiones y la mayoría vacías.

5.4. Implementación y prueba

A partir del diseño, se seguirá una metodología de programación ágil (basada en prototipos incrementales) para desarrollar el sistema. Los algoritmos de cálculo y modelado, así como el control de ejecuciones, captación de resultados, e incorporación de los mismos a la base de datos de ejecuciones se han implementado en `python` y reservado `C++` para los procedimientos que requieren una implementación más eficiente. La parte de interfaz y gestión de datos se implementará sobre la plataforma

Django. Este entorno de trabajo lo compone una biblioteca extensa de clases y plantillas en `python` para el desarrollo de aplicaciones web. El Sistema Gestor de Bases de Datos a emplear es MySQL. Todo el desarrollo de los sistemas de cómputo se han integrado en la plataforma **Orange**.

La integración y prueba del sistema software se llevan a cabo de forma incremental según la metodología basada en prototipos indicada. El lenguaje de programación `python` soporta este tipo de programación permitiendo implementar código para verificar los requisitos que debe cumplir una clase (y sus métodos) de forma que se automatice la verificación de los requisitos impuestos. Esto acelera claramente la detección temprana de errores.

5.5. Validación del modelo

Para validar el modelo y contrastar su eficacia, lo hemos aplicado a un problema real. Se ha elegido un problema entre aquéllos donde el número de características es elevado y una selección de características adecuada es un requisito para la resolución eficaz del mismo. Además de aconsejar el método de selección de características más adecuado, el sistema ejecutará también la selección de características. Evaluamos esta selección de características.

Clasificación de aceites de oliva El problema considerado versa sobre la caracterización de aceites de oliva. El aceite de oliva es un producto natural componente esencial de la beneficiosa dieta mediterránea. Además de eso, este producto es una pieza clave en la economía agrícola española. No en vano, España es el primer productor mundial de este producto. La producción del aceite se distribuye a lo largo de diversas áreas de la geografía española. Aparte de su origen, otros factores determinantes de su calidad son la variedad de la planta productora, los cuidados aplicados durante el cultivo etc.

El objetivo fundamental de un distribuidor de aceite es asegurarse que el aceite comprado cumple determinadas características de calidad. Tradicionalmente, un aspecto importante de la calidad ha sido la valoración organoléptica realizada por un panel de cata. Éste es un proceso aquejado de diversos inconvenientes que se tratan de evitar automatizando el proceso. Para ello, se realiza un análisis químico riguroso de las muestras. Esto genera gran cantidad de valores para una misma muestra. El objetivo final es determinar a qué nivel de calidad se puede asociar la muestra

de aceite. Se definen varios niveles, por lo que puede considerarse un problema de clasificación.

Este problema se formula con 64 características, todas de tipo continuo. Los aceites se catalogan en dos categorías (intenso y suave). Estas dos clases están bastante parejas, apareciendo la clase mayoritaria en el 52.1% de las ocasiones. No hay valores nulos o desconocidos. Para la resolución de este problema se nos proporcionó un conjunto de datos con 68 ejemplos.

De cara a la selección se consideró muy importante obtener buenos resultados, asignándole un grado de 0.8, y se valoró ligeramente que se use un conjunto más reducido de características, con un grado 0.2. En la figura 5.7 se muestran algunas de las reglas que se activan en el sistema para este caso. La recomendación ordenada de métodos sugerida finalmente se muestra en la figura 5.8.

Regla 1	Activación
SI	
<i>resultados es Importante</i>	0.8
<i>nº de características es Bajo</i>	1.0
<i>tipo es Continuas</i>	1.0
ENTONCES	
<i>GA-IEP es Usar</i> (con grado 0.92)	0.74
<i>GA-Wra es Usar</i> (con grado 0.89)	0.72
<i>SA-Wra es Usar</i> (con grado 0.87)	0.70
...	
Regla 2	Activación
SI	
<i>reducción es Importante</i>	0.2
<i>nº de características es Bajo</i>	1.0
<i>tipo es Continuas</i>	1.0
ENTONCES	
<i>GA-IEP es Usar</i> (con grado 0.41)	0.08
<i>GA-Wra es Usar</i> (con grado 0.37)	0.07
<i>SA-Wra es Usar</i> (con grado 0.42)	0.08
...	

Figura 5.7: Reglas disparadas en el sistema experto para el problema de clasificación de aceites

Para verificar si la recomendación es adecuada se lanzaron los dos primeros métodos recomendados. Para facilitar la valoración del efecto de la selección reco-

<i>Método</i>	Grado de recomendación
<i>GA-IEP</i>	0.82
<i>GA-Wra</i>	0.79
<i>SA-Wra</i>	0.78
...	...

Figura 5.8: Métodos recomendados por el sistema experto para el problema de clasificación de aceites

mendada, facilitamos los resultados obtenidos sin selección de características y con Relief-F, como método de selección conocido que podría ser el elegido por alguien que no dispone de la información del sistema. Como algoritmo de aprendizaje hemos empleado el mismo perceptrón multicapa usado en el estudio del capítulo 4.

<i>Método</i>	$\%AC \pm se$	$nCar \pm se$	tSC
IEP-GA	79.76 ± 5.693	33.0 ± 1.15	17.5
Wra-GA	76.90 ± 5.265	32.4 ± 1.01	1288.2
Relief-F	76.90 ± 6.066	32.0 ± 0.00	0.0
Sin selección	84.05 ± 4.970	64.0 ± 0.00	0.0

Tabla 5.1: Resultados obtenidos con el sistema experto en un problema de clasificación de aceites

Los resultados obtenidos se muestran en la tabla 5.1. En porcentaje de acierto, ninguno de los métodos consigue superar a la aplicación de redes neuronales sin selección de características. Sin embargo, se puede ver, como el primer método recomendado por el sistema consigue un porcentaje de acierto no muy inferior usando poco más de la mitad de las características. El segundo método recomendado está en la línea de lo que habría elegido alguien que usase Relief-F. El tiempo empleado en la selección de características por el segundo método es mayor, pero la respuesta del sistema experto es correcta ya que el usuario no ha indicado importancia sobre el tiempo de ejecución.

Capítulo 6

Conclusiones y líneas futuras de investigación

A modo de resumen, recogemos en este capítulo las principales conclusiones derivadas de la labor de investigación descrita a lo largo de la memoria. El trabajo no sólo ha permitido obtener resultados muy interesantes, sino que también ha abierto nuevos caminos cuya exploración parece ser muy prometedora. Incluimos una descripción de varios de estos posibles caminos, cuya explotación queda propuesta para desarrollos futuros.

6.1. Conclusiones

Este trabajo se ha centrado en el estudio de la selección de características en problemas de clasificación. Se trata de resolver un problema con muchas aplicaciones y muy estudiado, pero no por ello resuelto. Y con razón, debido a la complejidad que esto implica. Complejidad tanto desde el punto de vista computacional, pues es un problema NP-Completo como de abstracción, ya que es difícil llegar a conocerlo en profundidad.

Esta dificultad pone al problema en el contexto de muchos otros para los que la única solución globalmente satisfactoria sólo se ha alcanzado mediante la aplicación de técnicas de Inteligencia Artificial y más concretamente, de Inteligencia Computacional. Nos hemos planteado, pues, buscar una solución a este problema siguiendo un enfoque híbrido entre científico y de ingeniería.

El objetivo global lo descomponíamos en cinco objetivos concretos, todos los cuáles han sido alcanzados satisfactoriamente tal y como ilustramos a continuación.

El primer paso para abordar cualquier problema es conocer bien el problema a resolver. En un trabajo de investigación, esto se extiende hasta comprender a fondo el problema objeto de estudio. Esto se ha traducido en una extensa revisión bibliográfica y en el estudio de este material. Con esto, obtenemos el conocimiento plasmado por otros autores que previamente se habían enfrentado a este problema. La mayoría lo habían hecho tan sólo de una forma parcial y sesgada, y sólo unos pocos se han enfrentado a la complejidad total que implica este problema. Nosotros hemos avanzado sobre las contribuciones de todos ellos llegando a una comprensión detallada del mismo. La aplicación de diversos métodos de selección de características en un amplio conjunto de problemas reales ha cimentado el conocimiento sobre el tema.

En el camino seguido, hemos estudiado muchos de los métodos previamente propuestos, algunos de los cuales se han convertido en algoritmos de referencia por sus características. Uno de los más distinguidos es el método FOCUS. A pesar de sus virtudes teóricas y prácticas, este algoritmo tiene un punto débil importante: el reducido campo de aplicación, al restringirse a características lógicas. Para solventar la situación, hemos desarrollado un algoritmo que, basado en las mismas ideas y estrategias de búsqueda, pueda ser aplicado a un conjunto de problemas más amplio. Como primer paso, realizamos una extensión directa de FOCUS, ya propuesta por los autores originales, para incorporar características discretas. El segundo paso ha sido la extensión para poder ser aplicado a problemas con características continuas o enteras. Hemos denominado a este método C-FOCUS, y comprobado experimentalmente su gran capacidad en la selección de características. El algoritmo se ha completado dotándolo de cierta inmunidad frente a niveles bajos de ruido y de capacidad para manejar conjuntos de datos incompletos y con valores nulos. Finalmente, se amplía el ámbito de aplicación a problemas que incluyan variables lingüísticas, obteniendo así el método FFOCUS, que también ofrece un buen rendimiento.

En la misma senda de mejorar métodos de selección de características ya propuestos, hemos hecho una revisión extensiva de todas las medidas de consistencia y hemos propuesto una nueva, que llamamos IEP. Las virtudes de esta nueva medida se centran en que, con métodos de búsqueda muy simples, alcanza con facilidad conjuntos mínimos de características consistentes. Además, se ha desarrollado un algoritmo para calcular la medida con un orden de eficiencia en $O(n)$, lo que permite usarla en conjuntos de datos muy grandes.

Por otro lado, hemos encontrado y empleado algunas medidas de consistencia

que se aplicaban en otros contextos, pero cuya aplicación en selección de características nunca se había considerado. Es el caso de la medida de consistencia de la teoría de Rough Sets.

Una medida exitosa, en la evaluación de características individuales, es la propuesta en el método Relief (y sus variantes). Hemos explotado las buenas ideas del diseño de esta medida para extenderla a una medida que permita evaluar conjuntos de características en lugar de características individuales. Esto permite hacer una selección de características más avanzada, fundamentalmente detectando características redundantes.

Del estudio detallado del problema de selección de características y de los diversos métodos propuestos, hemos derivado un esquema modular en el que tengan cabida todos los métodos de selección de características. Este esquema se vertebra en dos puntos fundamentales: método de búsqueda y medida usada. Así, se pueden representar en un mapa bidimensional las distintas combinaciones de métodos de búsqueda y medidas que dan lugar a los distintos algoritmos de selección de características. Esta representación permite detectar qué tipo de combinaciones aún no han sido consideradas y sugerir nuevos algoritmos de selección de características basados en esas combinaciones.

A partir de esta estructuración, y para desarrollar un conocimiento más completo del problema, hemos realizado un ambicioso estudio empírico. Son destacables la minuciosidad y el rigor con que se han llevado a cabo el diseño experimental, ejecución y análisis de resultados. Este último permite derivar conclusiones que establecen guías claras sobre la aplicación de los métodos de selección de características.

Los resultados del estudio corroboran la utilidad de la selección de características, mostrando que, en muchos problemas, se mejoran los resultados que se obtienen sin selección, o se mantienen en niveles similares reduciendo el número de características empleadas. Hemos detectado que, en general, hay una fuerte relación inversa entre buenos resultados y reducción de características, mostrándose estos como objetivos contrapuestos que es necesario ponderar a la hora de aplicar selección de características. Tan sólo los métodos que usan medidas envolventes se escapan de esta contraposición, y cuando es viable su aplicación da resultados que están entre los mejores tanto en porcentaje de aciertos como en reducción de características. Sin embargo, su elevado coste computacional los hace inviables para problemas con un número grande de ejemplos. Por ello es conveniente considerar otras medidas como IEP o RSC.

Habitualmente, no es fácil encontrar diferencias significativas estadísticas en

eficacia entre los métodos de selección de características, mientras que sí hay mayores diferencias en términos de reducción.

También queda establecida la supremacía de los métodos basados en selección de subconjuntos de características frente a los de evaluación individual, debido a que los primeros sí consideran las posibles interrelaciones entre las distintas características.

No parece posible encontrar un método globalmente mejor para todos los problemas. No obstante, el estudio permite destacar algunos en circunstancias concretas. Por ejemplo, *SBS-Wra* es muy recomendable siempre que el tamaño y la eficiencia del algoritmo de aprendizaje lo permitan, y los métodos de ascensión de colinas combinados con medidas de consistencia (*SFS-Liu*, *SFS-IEP* y *SFS-RSC*) son adecuados para problemas más grandes.

Finalmente, como solución global al problema de selección de características desde la perspectiva de la Inteligencia Computacional, hemos diseñado e implementado un sistema inteligente. El sistema es dinámico con capacidad para ir ampliando su base de conocimiento mediante la incorporación de nuevos ejemplos de problemas de selección de características resueltos. La eficacia de este sistema se ha validado con su aplicación exitosa en un problema real.

6.2. Líneas de trabajo futuras

La investigación realizada ha sido fructífera en cuanto a resultados logrados, pero también ha abierto nuevos caminos cuya exploración promete obtener nuevos resultados interesantes. Entre estos caminos destacamos las siguientes cuatro líneas de investigación.

- **Estudio de la tolerancia de los algoritmos de aprendizaje a características innecesarias.** Los diferentes algoritmos de aprendizaje no trabajan igual ante la presencia de características redundantes y/o irrelevantes. Algunos llevan selección de características inmersa, otros pueden tolerar datos irrelevantes sin perjudicar su rendimiento pero sí su tiempo de cálculo o viceversa. Creemos que puede ser interesante, para orientar la investigación y avanzar en selección de características, un estudio de la tolerancia a características redundantes e irrelevantes de los diversos algoritmos de aprendizaje, tanto los clásicos, como los más recientes.

- **Combinación de métodos de selección de características.** Al igual que se han obtenido buenos resultados mediante la combinación de métodos de clasificación [BK99], es posible combinar métodos de selección de características para realizar mejores selecciones. Hay algunas propuestas [ÖUT06], no obstante, la cantidad de posibilidades de combinación es enorme. Nos proponemos explorar algunas como: la agregación de las salidas de varios métodos simples de selección de características, conectar en serie métodos de evaluación individual de características que hagan una preselección para luego refinar la búsqueda con un método que haga selección de subconjuntos o, a la vista de la descomposición modular propuesta en este trabajo, hibridar los diferentes métodos de búsqueda y medidas.
- **Selección de características en problemas de regresión.** En este trabajo, nos hemos centrado en problemas de clasificación. Sin embargo, la selección de características es útil en un espectro más amplio de problemas. Creemos interesante realizar un estudio, similar al aquí presentado, sobre la aplicación de la selección de características en problemas de regresión o aproximación.
- **Selección de características para series temporales.** La selección de características está resultando de sumo interés en el ámbito del modelado y predicción de series temporales, tanto de sus variables endógenas como exógenas. Sin embargo, el efecto de la variable tiempo hace que este tipo de problema requiera una adaptación de técnicas especiales, ya que una aplicación directa de los métodos habituales no produce resultados tan buenos como se podría esperar. Por ello, nos parece conveniente investigar estas adaptaciones.

Apéndice A

Tablas con los resultados experimentales

En este apéndice se exponen algunas tablas de resultados experimentales. Se ha considerado que la inclusión de estas tablas es interesante para describir toda la experimentación realizada en el capítulo 4 e ilustrar algún ejemplo, pero incluirlas dentro del mismo capítulo entorpecería en exceso su lectura.

A.1. Muestra de los resultados experimentales

En primer lugar, queremos detallar todos los datos tomados en la ejecución de los diferentes métodos de selección de características evaluados. La combinación de todas las posibilidades consideradas de métodos, factores y conjuntos de datos hace que el número de experimentos sea elevado. Concretamente hay 8644 resultados de los métodos que usan medidas de conjuntos de características y 4320 de los que usan medidas individuales. Por esta razón, no es apropiado incluir una tabla con todos los resultados. A modo de muestra, incluimos en la tabla A.1 los datos tomados de los métodos basados en valoración de conjuntos para dos de los problemas considerados.

En la tabla, la precisión está reducida a un sólo decimal para que quepa, los datos originales tienen mayor precisión*. En general, la selección de características se ha repetido en cada experimento. Sin embargo, la selección debe ser la misma para los

*La precisión de los números en coma flotante estándar de Python, o la máxima permitida por el contador de tiempo de uso de la CPU en el caso de los tiempos.

cuatro algoritmos de aprendizaje en los métodos de selección que son determinísticos y no dependen del aprendizaje (todos excepto *Wra*). Por esta razón, en los más costosos hemos decidido reaprovecharla y por esta razón el tiempo de selección que aparece es exactamente el mismo.

Los datos incluidos en cada una de las columnas de la tabla son los siguientes:

- Conjunto de datos del problema sobre el que se prueba (*Datos*).
- Método de selección:
 - Método de búsqueda (*Bús.*) o criterio de corte empleado en la selección, respectivamente si el método es de valoración de conjuntos o individual.
 - Medida empleada en la selección (*Med.*). Individual o de conjuntos según el método. En blanco si no procede.
- Método de aprendizaje usado para la clasificación (*Apr.*).
- Porcentaje medio de acierto en la validación cruzada y su error estándar ($\%AC \pm se$).
- Número de características medio empleado en las particiones de la validación cruzada y su error estándar ($\%nCar \pm se$).
- Tiempo empleado por las diferentes partes del proceso de aprendizaje. Se indica la media en cada una de las particiones de la validación cruzada y su error estándar.
 - En realizar la selección de características ($\%tSC \pm se$).
 - En realizar el aprendizaje ($\%tAprd. \pm se$).
 - En realizar la evaluación sobre las particiones de prueba ($\%tTest. \pm se$).
- Porcentaje medio de acierto sobre las particiones de entrenamiento en la validación cruzada y su error estándar ($Entr. \pm se$).

<i>Datos</i>	<i>Bús.</i>	<i>Med.</i>	<i>Apr.</i>	$\%AC \pm se$	$nCar \pm se$	$tSC \pm se$	$tAprd. \pm se$	$tTest \pm se$	$Entr. \pm se$
adult	ABB	IEP	C45	78.0 ± 0.2	3.0 ± 0.0	27939.3 ± 94.5	0.9 ± 0.0	0.3 ± 0.0	78.0 ± 0.0
adult	ABB	IEP	kNN	75.0 ± 0.2	3.0 ± 0.0	28460.5 ± 125.2	0.1 ± 0.0	219.8 ± 1.3	86.0 ± 0.0
adult	ABB	IEP	Nba.	78.0 ± 0.2	3.0 ± 0.0	37598.7 ± 2730.4	1768.5 ± 259.9	0.3 ± 0.0	78.0 ± 0.0
adult	ABB	IEP	RNA	78.0 ± 0.3	3.0 ± 0.0	21324.0 ± 53.8	7.6 ± 0.0	1.0 ± 0.0	78.2 ± 0.0
adult	ABB	Inf	C45	86.0 ± 0.2	11.0 ± 0.0	320.6 ± 7.5	5.5 ± 0.1	0.3 ± 0.0	87.0 ± 0.0
adult	ABB	Inf	kNN	83.0 ± 0.2	11.0 ± 0.0	326.1 ± 7.5	0.1 ± 0.0	257.3 ± 1.3	90.0 ± 0.0
adult	ABB	Inf	Nba.	83.0 ± 0.2	11.0 ± 0.0	336.3 ± 7.8	1251.9 ± 143.1	0.3 ± 0.0	84.0 ± 0.0
adult	ABB	Inf	RNA	84.2 ± 0.2	11.0 ± 0.0	244.3 ± 5.2	36.3 ± 0.1	3.1 ± 0.0	89.0 ± 0.1
adult	ABB	Liu	C45	86.0 ± 0.2	9.9 ± 0.1	1364.4 ± 12.8	5.3 ± 0.1	0.2 ± 0.0	87.0 ± 0.0
adult	ABB	Liu	kNN	83.0 ± 0.2	9.9 ± 0.1	1377.6 ± 14.6	0.1 ± 0.0	251.3 ± 1.6	90.0 ± 0.0
adult	ABB	Liu	Nba.	83.0 ± 0.3	9.9 ± 0.1	2616.0 ± 34.6	2263.6 ± 258.6	0.6 ± 0.0	84.0 ± 0.0

continúa en la página siguiente

viene de la página anterior

Datos	Búsq.	Med.	Apr.	%AC ± se	nCar ± se	tSC ± se	tAprd. ± se	tTest ± se	Entr. ± se
adult	ABB	Liu	RNA	85.0 ± 0.2	9.9 ± 0.1	1354.1 ± 13.4	28.2 ± 0.1	2.7 ± 0.0	88.0 ± 0.0
adult	ABB	RFS	C45	85.2 ± 0.5	10.3 ± 1.1	111150 ± 56300	3.5 ± 0.9	0.3 ± 0.0	86.0 ± 0.6
adult	ABB	RFS	kNN	82.2 ± 0.2	10.3 ± 1.1	111150 ± 56300	0.1 ± 0.0	310.6 ± 8.2	86.8 ± 0.8
adult	ABB	RFS	Nba.	81.1 ± 0.4	10.3 ± 1.1	111150 ± 56300	474.8 ± 263.1	0.3 ± 0.0	81.2 ± 0.4
adult	ABB	RFS	RNA	84.0 ± 0.3	10.3 ± 1.1	111150 ± 56300	1158.3 ± 71.7	5.5 ± 0.4	87.1 ± 0.6
adult	ABB	RSC	C45	86.0 ± 0.2	11.0 ± 0.0	304.1 ± 1.3	5.4 ± 0.1	0.3 ± 0.0	87.0 ± 0.0
adult	ABB	RSC	kNN	83.0 ± 0.2	11.0 ± 0.0	310.0 ± 1.5	0.1 ± 0.0	257.0 ± 1.2	90.0 ± 0.0
adult	ABB	RSC	Nba.	83.0 ± 0.2	11.0 ± 0.0	320.0 ± 2.2	1252.2 ± 142.9	0.3 ± 0.0	84.0 ± 0.0
adult	ABB	RSC	RNA	84.2 ± 0.2	11.0 ± 0.0	228.2 ± 0.6	36.4 ± 0.1	3.1 ± 0.0	89.0 ± 0.1
adult	BB	IEP	C45	85.0 ± 0.2	9.0 ± 0.0	2308.1 ± 139.2	6.9 ± 0.2	0.5 ± 0.0	86.7 ± 0.1
adult	BB	IEP	kNN	82.6 ± 0.2	9.0 ± 0.0	1446.9 ± 26.9	0.1 ± 0.0	258.7 ± 1.7	89.4 ± 0.0
adult	BB	IEP	Nba.	83.3 ± 0.2	9.0 ± 0.0	2721.0 ± 41.8	2246.7 ± 259.1	0.5 ± 0.0	83.3 ± 0.0
adult	BB	IEP	RNA	85.3 ± 0.3	9.0 ± 0.0	1467.8 ± 26.4	27.8 ± 0.0	2.6 ± 0.0	86.3 ± 0.0
adult	BB	Inf	C45	85.0 ± 0.2	9.0 ± 0.0	2386.4 ± 172.6	6.1 ± 0.3	0.3 ± 0.0	86.7 ± 0.1
adult	BB	Inf	kNN	82.6 ± 0.2	9.0 ± 0.0	1673.4 ± 14.0	0.1 ± 0.0	256.4 ± 1.1	89.4 ± 0.0
adult	BB	Inf	Nba.	83.3 ± 0.2	9.0 ± 0.0	3000.5 ± 146.2	2189.9 ± 286.6	0.6 ± 0.0	83.3 ± 0.0
adult	BB	Inf	RNA	85.3 ± 0.3	9.0 ± 0.0	1711.4 ± 16.8	27.9 ± 0.1	2.7 ± 0.0	86.3 ± 0.0
adult	BB	Liu	C45	85.1 ± 0.2	9.0 ± 0.0	2496.0 ± 142.1	6.5 ± 0.3	0.4 ± 0.0	86.7 ± 0.1
adult	BB	Liu	kNN	82.6 ± 0.2	9.0 ± 0.0	1548.4 ± 11.9	0.1 ± 0.0	256.5 ± 1.8	89.5 ± 0.0
adult	BB	Liu	Nba.	83.3 ± 0.2	9.0 ± 0.0	2790.2 ± 144.2	2170.4 ± 241.6	0.6 ± 0.0	83.4 ± 0.1
adult	BB	Liu	RNA	85.4 ± 0.2	9.0 ± 0.0	1566.2 ± 12.6	27.7 ± 0.2	2.6 ± 0.0	86.3 ± 0.0
adult	BB	RFS	C45	83.9 ± 0.5	9.0 ± 0.0	24409.5 ± 3122.1	2.3 ± 0.1	0.3 ± 0.0	84.8 ± 0.3
adult	BB	RFS	kNN	81.8 ± 0.4	9.0 ± 0.0	24409.5 ± 3122.1	0.1 ± 0.0	303.6 ± 2.7	87.7 ± 0.3
adult	BB	RFS	Nba.	81.1 ± 0.4	9.0 ± 0.0	67161.4 ± 10472.8	907.4 ± 334.7	0.3 ± 0.0	81.2 ± 0.4
adult	BB	RFS	RNA	83.3 ± 0.4	9.0 ± 0.0	24409.5 ± 3122.1	1040.4 ± 68.3	4.9 ± 0.3	86.0 ± 0.3
adult	BB	RSC	C45	85.0 ± 0.2	9.0 ± 0.0	2340.5 ± 160.6	6.7 ± 0.3	0.4 ± 0.0	86.7 ± 0.1
adult	BB	RSC	kNN	82.6 ± 0.2	9.0 ± 0.0	1468.2 ± 13.1	0.1 ± 0.0	256.7 ± 1.4	89.4 ± 0.0
adult	BB	RSC	Nba.	83.3 ± 0.2	9.0 ± 0.0	2794.8 ± 21.6	2204.1 ± 247.8	0.5 ± 0.0	83.3 ± 0.0
adult	BB	RSC	RNA	85.3 ± 0.3	9.0 ± 0.0	1503.0 ± 12.0	27.8 ± 0.0	2.6 ± 0.0	86.3 ± 0.0
adult	Ex	IEP	C45	85.8 ± 0.1	13.0 ± 0.0	26375.4 ± 27.6	3.5 ± 0.1	0.6 ± 0.0	87.4 ± 0.0
adult	Ex	IEP	kNN	82.7 ± 0.1	13.0 ± 0.0	26375.4 ± 27.6	0.1 ± 0.0	429.3 ± 0.6	89.4 ± 0.0
adult	Ex	IEP	Nba.	81.0 ± 0.3	13.0 ± 0.0	26375.4 ± 27.6	919.8 ± 136.1	0.5 ± 0.0	81.1 ± 0.0
adult	Ex	IEP	RNA	84.2 ± 0.2	13.0 ± 0.0	26375.4 ± 27.6	940.4 ± 0.1	12.3 ± 0.0	89.3 ± 0.1
adult	Ex	Inf	C45	85.9 ± 0.2	13.5 ± 0.3	27604.2 ± 103.4	4.2 ± 0.4	0.6 ± 0.0	87.5 ± 0.1
adult	Ex	Inf	kNN	82.7 ± 0.1	13.5 ± 0.3	27604.2 ± 103.4	0.1 ± 0.0	434.4 ± 2.7	89.4 ± 0.0
adult	Ex	Inf	Nba.	81.5 ± 0.3	13.5 ± 0.3	27604.2 ± 103.4	947.3 ± 134.9	0.5 ± 0.0	81.5 ± 0.3
adult	Ex	Inf	RNA	84.0 ± 0.1	13.5 ± 0.3	27604.2 ± 103.4	946.2 ± 3.5	12.3 ± 0.0	89.2 ± 0.2
adult	Ex	Liu	C45	85.8 ± 0.1	13.0 ± 0.0	25728.6 ± 180.2	3.5 ± 0.0	0.6 ± 0.0	87.4 ± 0.0
adult	Ex	Liu	kNN	82.7 ± 0.1	13.0 ± 0.0	25728.6 ± 180.2	0.1 ± 0.0	431.1 ± 1.9	89.4 ± 0.0
adult	Ex	Liu	Nba.	81.0 ± 0.3	13.0 ± 0.0	25372.3 ± 26.3	907.1 ± 136.0	0.5 ± 0.0	81.1 ± 0.0
adult	Ex	Liu	RNA	84.2 ± 0.2	13.0 ± 0.0	25728.6 ± 180.2	941.4 ± 0.6	12.3 ± 0.1	89.3 ± 0.1
adult	Ex	MDL	C45	75.9 ± 0.0	1.0 ± 0.0	234735 ± 6798	5.1 ± 2.9	0.5 ± 0.1	75.9 ± 0.0
adult	Ex	MDL	kNN	72.7 ± 0.1	1.0 ± 0.0	234735 ± 6798	0.1 ± 0.0	362.4 ± 5.5	81.5 ± 0.0
adult	Ex	MDL	Nba.	75.9 ± 0.0	1.0 ± 0.0	234735 ± 6798	966.0 ± 137.7	0.4 ± 0.0	75.9 ± 0.0
adult	Ex	MDL	RNA	75.9 ± 0.0	1.0 ± 0.0	234735 ± 6798	135.3 ± 0.3	1.5 ± 0.0	75.9 ± 0.0
adult	Ex	RFS	C45	85.2 ± 0.2	11.0 ± 0.0	2015127 ± 78261	2.4 ± 0.1	0.2 ± 0.0	86.1 ± 0.0
adult	Ex	RFS	kNN	82.2 ± 0.2	11.0 ± 0.0	2015127 ± 78261	0.1 ± 0.0	188.7 ± 0.7	89.0 ± 0.0
adult	Ex	RFS	Nba.	81.7 ± 0.1	11.0 ± 0.0	2015127 ± 78261	1040.6 ± 117.8	0.2 ± 0.0	81.8 ± 0.0
adult	Ex	RFS	RNA	84.2 ± 0.2	11.0 ± 0.0	2015127 ± 78261	956.2 ± 0.2	4.3 ± 0.0	87.1 ± 0.0
adult	Ex	RSC	C45	85.8 ± 0.1	13.0 ± 0.0	25850.5 ± 79.7	3.5 ± 0.1	0.6 ± 0.0	87.4 ± 0.0
adult	Ex	RSC	kNN	82.7 ± 0.1	13.0 ± 0.0	25850.5 ± 79.7	0.1 ± 0.0	433.4 ± 3.8	89.4 ± 0.0
adult	Ex	RSC	Nba.	81.0 ± 0.3	13.0 ± 0.0	26375.4 ± 27.6	965.9 ± 139.6	0.6 ± 0.0	81.1 ± 0.0
adult	Ex	RSC	RNA	84.2 ± 0.2	13.0 ± 0.0	25850.5 ± 79.7	940.2 ± 0.2	12.3 ± 0.0	89.3 ± 0.1
adult	Ex	SU	C45	80.9 ± 0.1	2.0 ± 0.0	28190.8 ± 102.3	0.2 ± 0.0	0.6 ± 0.0	80.9 ± 0.0
adult	Ex	SU	kNN	77.2 ± 2.2	2.0 ± 0.0	28190.8 ± 102.3	0.1 ± 0.0	348.6 ± 2.8	77.2 ± 2.2
adult	Ex	SU	Nba.	79.2 ± 0.1	2.0 ± 0.0	28190.8 ± 102.3	0.1 ± 0.0	0.4 ± 0.0	79.2 ± 0.0
adult	Ex	SU	RNA	79.6 ± 0.0	2.0 ± 0.0	28190.8 ± 102.3	179.0 ± 0.0	2.3 ± 0.0	79.6 ± 0.1
adult	Focus2		C45	86.2 ± 0.2	14.0 ± 0.0	8710.1 ± 726.7	7.8 ± 0.1	0.2 ± 0.0	87.9 ± 0.1
adult	Focus2		kNN	82.7 ± 0.2	14.0 ± 0.0	8779.7 ± 815.3	0.1 ± 0.0	272.4 ± 1.8	89.5 ± 0.0
adult	Focus2		Nba.	81.9 ± 0.1	14.0 ± 0.0	8913.6 ± 764.8	1775.3 ± 199.8	0.3 ± 0.0	82.0 ± 0.0
adult	Focus2		RNA	84.0 ± 0.3	14.0 ± 0.0	8842.3 ± 799.6	40.5 ± 0.1	2.9 ± 0.0	89.4 ± 0.1
adult	GA	IEP	C45	85.5 ± 0.4	11.1 ± 0.4	1248.4 ± 24.4	4.7 ± 0.3	0.3 ± 0.0	86.9 ± 0.3
adult	GA	IEP	kNN	82.7 ± 0.2	12.1 ± 0.3	1441.3 ± 20.0	0.1 ± 0.0	341.2 ± 2.2	89.5 ± 0.0
adult	GA	IEP	Nba.	81.8 ± 0.3	11.8 ± 0.4	811.3 ± 13.2	1125.9 ± 138.1	0.3 ± 0.0	81.9 ± 0.3
adult	GA	IEP	RNA	84.1 ± 0.2	11.6 ± 0.3	776.5 ± 8.9	918.4 ± 52.7	4.2 ± 0.2	87.8 ± 0.2
adult	GA	Inf	C45	85.6 ± 0.3	11.7 ± 0.4	1375.9 ± 20.6	5.5 ± 0.4	0.3 ± 0.0	87.2 ± 0.2
adult	GA	Inf	kNN	82.7 ± 0.2	11.8 ± 0.2	1497.1 ± 18.7	0.1 ± 0.0	333.1 ± 2.6	89.3 ± 0.0
adult	GA	Inf	Nba.	82.5 ± 0.3	11.7 ± 0.4	853.8 ± 8.9	1764.6 ± 290.9	0.3 ± 0.0	82.3 ± 0.3
adult	GA	Inf	RNA	84.1 ± 0.2	12.0 ± 0.3	853.8 ± 8.9	979.1 ± 51.1	4.5 ± 0.2	87.9 ± 0.2
adult	GA	Liu	C45	86.0 ± 0.2	12.2 ± 0.4	1291.2 ± 26.3	5.8 ± 0.3	0.3 ± 0.0	87.7 ± 0.1
adult	GA	Liu	kNN	82.9 ± 0.3	11.4 ± 0.2	1367.2 ± 15.2	0.1 ± 0.0	329.8 ± 2.2	89.3 ± 0.0
adult	GA	Liu	Nba.	81.9 ± 0.4	12.1 ± 0.3	800.6 ± 10.3	1114.0 ± 128.3	0.3 ± 0.1	81.9 ± 0.3
adult	GA	Liu	RNA	84.4 ± 0.4	12.7 ± 0.4	810.8 ± 11.9	963.3 ± 63.9	4.4 ± 0.3	88.2 ± 0.2
adult	GA	MDL	C45	77.8 ± 0.7	3.4 ± 0.5	9491.2 ± 38.6	5.2 ± 2.3	0.3 ± 0.0	78.0 ± 0.7
adult	GA	MDL	kNN	75.7 ± 1.0	3.2 ± 0.7	9795.2 ± 41.5	0.1 ± 0.0	292.0 ± 6.9	81.8 ± 1.4
adult	GA	MDL	Nba.	77.0 ± 0.5	3.4 ± 0.5	5216.1 ± 11.9	1134.1 ± 221.4	0.2 ± 0.0	77.2 ± 0.5
adult	GA	MDL	RNA	78.4 ± 0.7	3.1 ± 0.3	5216.1 ± 11.9	504.0 ± 56.2	2.2 ± 0.2	78.5 ± 0.7

continúa en la página siguiente

viene de la página anterior

Datos	Búsq.	Med.	Apr.	%AC ± se	nCar ± se	tSC ± se	tAprd. ± se	tTest ± se	Entr. ± se
adult	GA	RFS	C45	84.2 ± 0.3	9.9 ± 0.4	90186.0 ± 1044.1	3.4 ± 0.4	0.3 ± 0.0	85.3 ± 0.4
adult	GA	RFS	kNN	82.4 ± 0.2	10.8 ± 0.5	50252.2 ± 6734.7	0.1 ± 0.0	217.4 ± 13.1	88.7 ± 0.4
adult	GA	RFS	Nba.	81.4 ± 0.4	9.9 ± 0.4	1745084 ± 1704042	788.2 ± 172.5	0.2 ± 0.0	81.3 ± 0.4
adult	GA	RFS	RNA	84.1 ± 0.4	10.4 ± 0.4	1745084 ± 1704042	866.1 ± 65.5	4.0 ± 0.3	87.1 ± 0.3
adult	GA	RSC	C45	85.5 ± 0.4	12.3 ± 0.4	1263.8 ± 16.5	6.3 ± 0.4	0.3 ± 0.0	87.2 ± 0.3
adult	GA	RSC	kNN	82.6 ± 0.2	12.2 ± 0.4	1434.4 ± 22.3	0.1 ± 0.0	346.3 ± 2.7	89.4 ± 0.1
adult	GA	RSC	Nba.	81.6 ± 0.3	12.3 ± 0.4	767.6 ± 15.1	1893.2 ± 215.5	0.3 ± 0.0	81.8 ± 0.3
adult	GA	RSC	RNA	84.9 ± 0.3	11.3 ± 0.5	767.6 ± 15.1	813.4 ± 72.1	3.8 ± 0.3	87.5 ± 0.2
adult	GA	SU	C45	85.3 ± 0.4	9.5 ± 1.1	1299.8 ± 81.5	4.2 ± 0.8	0.3 ± 0.0	86.4 ± 0.4
adult	GA	SU	kNN	82.6 ± 0.3	8.1 ± 1.0	1241.5 ± 90.7	0.1 ± 0.0	318.4 ± 8.3	86.4 ± 1.0
adult	GA	SU	Nba.	81.8 ± 0.8	9.5 ± 1.1	780.8 ± 47.3	1353.9 ± 342.2	0.3 ± 0.0	81.8 ± 0.7
adult	GA	SU	RNA	83.3 ± 0.6	9.6 ± 1.2	780.8 ± 47.3	821.7 ± 86.6	3.8 ± 0.4	85.9 ± 1.0
adult	LVF	IEP	C45	78.7 ± 0.2	3.0 ± 0.0	2471.7 ± 70.5	1.2 ± 0.1	0.3 ± 0.0	78.9 ± 0.2
adult	LVF	IEP	kNN	76.0 ± 0.3	3.0 ± 0.0	1886.1 ± 64.0	0.1 ± 0.0	244.0 ± 4.7	85.0 ± 0.5
adult	LVF	IEP	Nba.	77.6 ± 0.2	3.0 ± 0.0	2820.8 ± 93.2	1765.2 ± 313.5	0.3 ± 0.0	77.8 ± 0.1
adult	LVF	IEP	RNA	78.5 ± 0.4	3.0 ± 0.0	1773.9 ± 30.1	11.2 ± 1.4	1.1 ± 0.1	78.7 ± 0.2
adult	LVF	Inf	C45	85.9 ± 0.2	12.6 ± 0.4	3192.6 ± 184.1	8.3 ± 0.8	0.2 ± 0.0	87.6 ± 0.1
adult	LVF	Inf	kNN	82.7 ± 0.2	13.3 ± 0.2	2988.5 ± 107.3	0.1 ± 0.0	297.2 ± 8.5	89.5 ± 0.0
adult	LVF	Inf	Nba.	82.7 ± 0.3	12.8 ± 0.3	3556.3 ± 205.6	1763.8 ± 216.8	0.4 ± 0.0	82.7 ± 0.2
adult	LVF	Inf	RNA	84.0 ± 0.2	12.7 ± 0.3	3794.6 ± 188.7	43.1 ± 2.9	3.9 ± 0.3	88.9 ± 0.1
adult	LVF	Liu	C45	85.8 ± 0.3	10.8 ± 0.2	3013.2 ± 130.2	7.3 ± 0.4	0.2 ± 0.0	87.1 ± 0.1
adult	LVF	Liu	kNN	82.7 ± 0.2	10.8 ± 0.2	2840.6 ± 75.2	0.1 ± 0.0	293.6 ± 14.7	89.5 ± 0.0
adult	LVF	Liu	Nba.	82.6 ± 0.4	10.9 ± 0.3	4180.4 ± 207.2	2071.6 ± 229.8	0.5 ± 0.0	82.5 ± 0.3
adult	LVF	Liu	RNA	84.1 ± 0.3	10.7 ± 0.2	2733.5 ± 111.0	38.6 ± 2.0	3.0 ± 0.2	88.4 ± 0.1
adult	LVF	RFS	C45	83.5 ± 0.4	6.5 ± 0.6	68891.8 ± 2169.4	1.5 ± 0.3	0.2 ± 0.0	84.2 ± 0.4
adult	LVF	RFS	kNN	81.7 ± 0.3	6.5 ± 0.6	68891.8 ± 2169.4	0.1 ± 0.0	296.7 ± 5.0	85.9 ± 0.8
adult	LVF	RFS	Nba.	81.3 ± 0.4	6.5 ± 0.6	68891.8 ± 2169.4	384.3 ± 276.6	0.3 ± 0.0	81.5 ± 0.3
adult	LVF	RFS	RNA	83.2 ± 0.3	6.5 ± 0.6	68891.8 ± 2169.4	856.8 ± 90.8	3.9 ± 0.4	84.7 ± 0.4
adult	LVF	RSC	C45	85.9 ± 0.2	12.6 ± 0.3	3435.1 ± 184.6	8.6 ± 0.6	0.3 ± 0.0	87.6 ± 0.1
adult	LVF	RSC	kNN	82.8 ± 0.2	12.3 ± 0.3	3383.3 ± 244.7	0.1 ± 0.0	337.7 ± 22.6	89.5 ± 0.0
adult	LVF	RSC	Nba.	82.5 ± 0.2	12.4 ± 0.3	3067.5 ± 100.8	1659.3 ± 212.4	0.3 ± 0.0	82.7 ± 0.2
adult	LVF	RSC	RNA	84.0 ± 0.3	12.9 ± 0.3	3953.5 ± 228.7	49.7 ± 3.9	4.3 ± 0.3	89.0 ± 0.1
adult	LVW	MDL	C45	77.9 ± 0.8	2.2 ± 0.4	1588.7 ± 133.8	2.3 ± 1.5	0.2 ± 0.0	77.8 ± 0.7
adult	LVW	MDL	kNN	75.7 ± 0.9	1.2 ± 0.1	1546.0 ± 201.6	0.1 ± 0.0	310.3 ± 7.3	76.7 ± 0.9
adult	LVW	MDL	Nba.	77.0 ± 0.6	1.5 ± 0.6	1195.1 ± 85.2	385.9 ± 207.4	0.2 ± 0.0	77.0 ± 0.6
adult	LVW	MDL	RNA	77.8 ± 0.4	2.7 ± 0.4	2735.2 ± 214.9	12.3 ± 1.9	2.3 ± 0.2	77.8 ± 0.4
adult	LVW	SU	C45	84.2 ± 0.8	7.3 ± 1.2	397.1 ± 38.5	6.8 ± 1.8	0.4 ± 0.0	84.9 ± 1.0
adult	LVW	SU	kNN	82.1 ± 1.1	5.1 ± 1.0	277.7 ± 41.3	0.1 ± 0.0	330.1 ± 6.7	83.8 ± 1.4
adult	LVW	SU	Nba.	81.0 ± 1.2	5.4 ± 1.2	222.9 ± 25.1	240.7 ± 167.4	0.2 ± 0.0	81.0 ± 1.2
adult	LVW	SU	RNA	83.0 ± 0.6	4.4 ± 0.5	537.2 ± 54.7	24.0 ± 4.5	3.5 ± 0.5	83.3 ± 0.7
adult	NoFS		C45	86.2 ± 0.2	14.0 ± 0.0	0.1 ± 0.0	10.8 ± 0.6	0.4 ± 0.1	87.9 ± 0.1
adult	NoFS		kNN	82.7 ± 0.2	14.0 ± 0.0	0.1 ± 0.0	0.1 ± 0.0	304.4 ± 12.1	89.5 ± 0.0
adult	NoFS		Nba.	81.9 ± 0.1	14.0 ± 0.0	0.1 ± 0.0	1526.7 ± 223.6	0.3 ± 0.0	82.0 ± 0.0
adult	NoFS		RNA	84.0 ± 0.3	14.0 ± 0.0	0.1 ± 0.0	52.6 ± 0.1	4.6 ± 0.0	89.4 ± 0.1
adult	SA	IEP	C45	85.7 ± 0.2	12.0 ± 0.4	1772.2 ± 15.0	5.2 ± 0.4	0.2 ± 0.0	87.2 ± 0.2
adult	SA	IEP	kNN	82.8 ± 0.2	12.0 ± 0.4	1943.7 ± 10.1	0.1 ± 0.0	310.5 ± 3.1	89.5 ± 0.0
adult	SA	IEP	Nba.	82.8 ± 0.2	12.0 ± 0.4	1982.1 ± 7.5	1705.6 ± 199.1	0.3 ± 0.0	82.8 ± 0.2
adult	SA	IEP	RNA	84.6 ± 0.2	12.0 ± 0.4	1866.0 ± 6.5	43.1 ± 2.1	6.4 ± 0.3	88.5 ± 0.1
adult	SA	Inf	C45	86.0 ± 0.2	13.4 ± 0.2	2213.0 ± 8.8	5.9 ± 0.3	0.2 ± 0.0	87.6 ± 0.1
adult	SA	Inf	kNN	82.7 ± 0.2	13.4 ± 0.2	1340.0 ± 6.0	0.1 ± 0.0	198.5 ± 1.4	89.5 ± 0.0
adult	SA	Inf	Nba.	81.4 ± 0.2	13.4 ± 0.2	2478.3 ± 17.1	1729.4 ± 201.3	0.4 ± 0.0	81.5 ± 0.1
adult	SA	Inf	RNA	84.2 ± 0.2	13.4 ± 0.2	2398.9 ± 25.1	50.4 ± 0.4	7.3 ± 0.1	88.8 ± 0.1
adult	SA	Liu	C45	85.9 ± 0.2	12.7 ± 0.3	1965.8 ± 15.8	5.7 ± 0.4	0.2 ± 0.0	87.5 ± 0.1
adult	SA	Liu	kNN	82.7 ± 0.2	12.7 ± 0.3	2050.9 ± 12.9	0.1 ± 0.0	316.0 ± 2.1	89.5 ± 0.0
adult	SA	Liu	Nba.	82.4 ± 0.2	12.7 ± 0.3	2075.7 ± 20.0	1695.6 ± 192.6	0.4 ± 0.0	82.4 ± 0.2
adult	SA	Liu	RNA	84.4 ± 0.2	12.7 ± 0.3	1921.9 ± 24.9	44.4 ± 0.5	6.5 ± 0.1	88.8 ± 0.1
adult	SA	MDL	C45	75.9 ± 0.0	1.0 ± 0.0	17133.8 ± 81.3	9.7 ± 2.1	0.2 ± 0.0	75.9 ± 0.0
adult	SA	MDL	kNN	72.7 ± 0.1	1.0 ± 0.0	9172.8 ± 25.7	0.1 ± 0.0	153.0 ± 1.1	81.5 ± 0.0
adult	SA	MDL	Nba.	75.9 ± 0.0	1.0 ± 0.0	16629.8 ± 39.9	1699.2 ± 190.8	0.3 ± 0.0	75.9 ± 0.0
adult	SA	MDL	RNA	75.9 ± 0.0	1.0 ± 0.0	16667.4 ± 151.3	2.3 ± 0.0	1.0 ± 0.0	75.9 ± 0.0
adult	SA	RFS	C45	84.6 ± 0.4	11.0 ± 0.4	1972904 ± 1707998	3.4 ± 0.3	0.2 ± 0.0	86.0 ± 0.4
adult	SA	RFS	kNN	82.3 ± 0.3	11.0 ± 0.4	1972904 ± 1707998	0.0 ± 0.0	161.9 ± 2.2	88.9 ± 0.1
adult	SA	RFS	Nba.	81.5 ± 0.5	10.9 ± 0.4	1971963 ± 1704453	1049.9 ± 345.1	0.3 ± 0.0	81.6 ± 0.5
adult	SA	RFS	RNA	83.3 ± 0.3	11.0 ± 0.4	1972904 ± 1707998	1137.0 ± 59.4	4.6 ± 0.2	87.4 ± 0.3
adult	SA	RSC	C45	86.0 ± 0.2	13.0 ± 0.0	2069.4 ± 6.1	5.1 ± 0.1	0.2 ± 0.0	87.3 ± 0.0
adult	SA	RSC	kNN	82.7 ± 0.2	13.0 ± 0.0	1548.5 ± 144.8	0.1 ± 0.0	219.7 ± 16.6	89.5 ± 0.0
adult	SA	RSC	Nba.	81.0 ± 0.2	13.0 ± 0.0	2301.4 ± 17.9	1703.2 ± 190.2	0.3 ± 0.0	81.1 ± 0.0
adult	SA	RSC	RNA	84.2 ± 0.2	13.0 ± 0.0	2167.0 ± 8.3	47.3 ± 0.6	6.9 ± 0.1	88.7 ± 0.1
adult	SA	SU	C45	82.1 ± 0.5	3.0 ± 0.3	1991.8 ± 14.2	0.4 ± 0.1	0.3 ± 0.0	82.2 ± 0.5
adult	SA	SU	kNN	79.1 ± 1.0	3.0 ± 0.3	1151.8 ± 8.2	0.0 ± 0.0	150.8 ± 1.7	79.4 ± 1.0
adult	SA	SU	Nba.	79.7 ± 1.0	3.0 ± 0.3	2067.6 ± 15.4	0.2 ± 0.0	0.3 ± 0.0	79.8 ± 1.0
adult	SA	SU	RNA	81.0 ± 0.5	3.0 ± 0.3	2113.9 ± 14.5	10.0 ± 2.6	2.0 ± 0.3	81.1 ± 0.5
adult	SFS	IEP	C45	86.0 ± 0.2	12.8 ± 0.1	230.3 ± 18.2	7.9 ± 0.5	0.3 ± 0.0	87.3 ± 0.0
adult	SFS	IEP	kNN	82.7 ± 0.2	12.8 ± 0.1	294.0 ± 9.6	0.1 ± 0.0	459.3 ± 10.5	89.5 ± 0.0
adult	SFS	IEP	Nba.	81.2 ± 0.1	12.8 ± 0.1	168.8 ± 1.1	1211.2 ± 139.2	0.3 ± 0.0	81.3 ± 0.2
adult	SFS	IEP	RNA	83.8 ± 0.3	12.8 ± 0.1	171.3 ± 1.0	51.7 ± 0.2	4.6 ± 0.0	89.4 ± 0.1
adult	SFS	Inf	C45	86.1 ± 0.2	13.4 ± 0.2	265.2 ± 14.9	9.8 ± 0.7	0.4 ± 0.1	87.5 ± 0.1

continúa en la página siguiente

viene de la página anterior

Datos	Búsq.	Med.	Apr.	%AC \pm se	nCar \pm se	tSC \pm se	tAprd. \pm se	tTest \pm se	Entr. \pm se
adult	SFS	Inf	kNN	82.7 \pm 0.2	13.4 \pm 0.2	201.0 \pm 5.0	0.1 \pm 0.0	353.7 \pm 32.9	89.5 \pm 0.0
adult	SFS	Inf	Nba.	81.3 \pm 0.2	13.4 \pm 0.2	192.2 \pm 12.7	1457.1 \pm 157.7	0.4 \pm 0.0	81.4 \pm 0.2
adult	SFS	Inf	RNA	83.7 \pm 0.3	13.4 \pm 0.2	174.5 \pm 0.5	52.1 \pm 0.1	4.6 \pm 0.0	89.3 \pm 0.1
adult	SFS	Liu	C45	86.0 \pm 0.2	13.0 \pm 0.0	203.1 \pm 7.0	7.4 \pm 0.3	0.3 \pm 0.0	87.3 \pm 0.0
adult	SFS	Liu	kNN	82.7 \pm 0.2	13.0 \pm 0.0	176.3 \pm 2.6	0.1 \pm 0.0	288.6 \pm 4.1	89.5 \pm 0.0
adult	SFS	Liu	Nba.	81.0 \pm 0.2	13.0 \pm 0.0	161.2 \pm 1.0	1205.8 \pm 138.1	0.3 \pm 0.0	81.1 \pm 0.0
adult	SFS	Liu	RNA	83.7 \pm 0.3	13.0 \pm 0.0	163.3 \pm 0.5	51.9 \pm 0.1	4.6 \pm 0.0	89.3 \pm 0.1
adult	SFS	MDL	C45	76.5 \pm 0.1	2.9 \pm 0.1	597.1 \pm 32.4	1.3 \pm 0.1	0.3 \pm 0.0	76.5 \pm 0.1
adult	SFS	MDL	kNN	72.8 \pm 0.2	2.9 \pm 0.1	466.2 \pm 7.0	0.1 \pm 0.0	243.3 \pm 9.7	83.9 \pm 0.0
adult	SFS	MDL	Nba.	76.3 \pm 0.1	2.9 \pm 0.1	604.3 \pm 21.9	1712.1 \pm 210.6	0.3 \pm 0.0	76.3 \pm 0.0
adult	SFS	MDL	RNA	76.6 \pm 0.1	2.9 \pm 0.1	475.0 \pm 1.3	7.0 \pm 0.4	1.1 \pm 0.0	76.5 \pm 0.1
adult	SFS	RFS	C45	84.7 \pm 0.5	11.0 \pm 0.6	13804.2 \pm 299.4	5.0 \pm 0.5	0.2 \pm 0.0	86.1 \pm 0.5
adult	SFS	RFS	kNN	82.4 \pm 0.3	11.0 \pm 0.6	17172.0 \pm 451.7	0.1 \pm 0.0	316.9 \pm 3.9	89.0 \pm 0.1
adult	SFS	RFS	Nba.	81.2 \pm 0.4	10.4 \pm 0.4	17172.0 \pm 451.7	527.1 \pm 290.6	0.3 \pm 0.0	81.5 \pm 0.4
adult	SFS	RFS	RNA	83.7 \pm 0.3	11.0 \pm 0.6	17172.0 \pm 451.7	1382.8 \pm 64.7	6.4 \pm 0.3	87.5 \pm 0.4
adult	SFS	RSC	C45	86.0 \pm 0.2	13.0 \pm 0.0	235.0 \pm 14.1	8.3 \pm 0.4	0.4 \pm 0.0	87.3 \pm 0.0
adult	SFS	RSC	kNN	82.7 \pm 0.2	13.0 \pm 0.0	205.2 \pm 9.5	0.1 \pm 0.0	316.9 \pm 17.5	89.5 \pm 0.0
adult	SFS	RSC	Nba.	81.0 \pm 0.2	13.0 \pm 0.0	160.0 \pm 0.8	1230.3 \pm 146.2	0.3 \pm 0.0	81.1 \pm 0.0
adult	SFS	RSC	RNA	83.7 \pm 0.3	13.0 \pm 0.0	163.0 \pm 0.3	51.8 \pm 0.1	4.6 \pm 0.0	89.3 \pm 0.1
adult	SFS	SU	C45	80.9 \pm 0.1	2.0 \pm 0.0	272.9 \pm 11.0	0.7 \pm 0.0	0.4 \pm 0.0	81.0 \pm 0.0
adult	SFS	SU	kNN	78.6 \pm 1.2	2.0 \pm 0.0	169.9 \pm 2.1	0.1 \pm 0.0	224.2 \pm 3.0	78.7 \pm 1.2
adult	SFS	SU	Nba.	79.2 \pm 0.1	2.0 \pm 0.0	236.7 \pm 12.4	0.1 \pm 0.0	0.4 \pm 0.0	79.2 \pm 0.0
adult	SFS	SU	RNA	79.9 \pm 0.1	2.0 \pm 0.0	171.1 \pm 0.1	6.2 \pm 0.0	0.9 \pm 0.0	79.9 \pm 0.1
adult	SBS	IEP	C45	86.0 \pm 0.2	13.0 \pm 0.0	232.2 \pm 0.4	5.1 \pm 0.1	0.3 \pm 0.0	87.3 \pm 0.0
adult	SBS	IEP	kNN	82.7 \pm 0.2	13.0 \pm 0.0	253.6 \pm 0.2	0.1 \pm 0.0	371.0 \pm 6.2	89.5 \pm 0.0
adult	SBS	IEP	Nba.	81.0 \pm 0.2	13.0 \pm 0.0	251.4 \pm 0.9	1952.0 \pm 231.6	0.4 \pm 0.0	81.1 \pm 0.0
adult	SBS	IEP	RNA	84.9 \pm 0.2	13.0 \pm 0.0	253.7 \pm 0.1	684.3 \pm 0.2	6.0 \pm 0.0	87.7 \pm 0.1
adult	SBS	Inf	C45	86.0 \pm 0.2	13.3 \pm 0.1	247.4 \pm 0.3	5.8 \pm 0.2	0.3 \pm 0.0	87.6 \pm 0.1
adult	SBS	Inf	kNN	82.7 \pm 0.2	13.3 \pm 0.1	270.1 \pm 0.2	0.1 \pm 0.0	370.6 \pm 3.8	89.5 \pm 0.0
adult	SBS	Inf	Nba.	81.3 \pm 0.2	13.3 \pm 0.1	270.1 \pm 0.6	1911.8 \pm 225.7	0.4 \pm 0.0	81.4 \pm 0.1
adult	SBS	Inf	RNA	84.8 \pm 0.3	13.3 \pm 0.1	268.3 \pm 0.2	686.8 \pm 1.3	6.0 \pm 0.0	87.7 \pm 0.1
adult	SBS	Liu	C45	86.0 \pm 0.2	13.0 \pm 0.0	227.0 \pm 0.8	5.1 \pm 0.1	0.3 \pm 0.0	87.3 \pm 0.0
adult	SBS	Liu	kNN	82.7 \pm 0.2	13.0 \pm 0.0	249.6 \pm 0.2	0.1 \pm 0.0	366.7 \pm 4.1	89.5 \pm 0.0
adult	SBS	Liu	Nba.	81.0 \pm 0.2	13.0 \pm 0.0	242.3 \pm 1.6	1855.6 \pm 205.9	0.4 \pm 0.0	81.1 \pm 0.0
adult	SBS	Liu	RNA	84.9 \pm 0.2	13.0 \pm 0.0	251.0 \pm 0.4	684.2 \pm 0.1	6.0 \pm 0.0	87.7 \pm 0.1
adult	SBS	MDL	C45	75.9 \pm 0.0	1.0 \pm 0.0	1940.6 \pm 2.1	9.9 \pm 2.1	0.3 \pm 0.0	75.9 \pm 0.0
adult	SBS	MDL	kNN	72.7 \pm 0.1	1.0 \pm 0.0	1996.5 \pm 1.7	0.1 \pm 0.0	292.0 \pm 0.3	81.5 \pm 0.0
adult	SBS	MDL	Nba.	75.9 \pm 0.0	1.0 \pm 0.0	2006.7 \pm 2.9	1914.1 \pm 218.4	0.3 \pm 0.0	75.9 \pm 0.0
adult	SBS	MDL	RNA	75.9 \pm 0.0	1.0 \pm 0.0	1988.9 \pm 1.4	1088.7 \pm 0.0	0.9 \pm 0.0	75.9 \pm 0.0
adult	SBS	RFS	C45	85.6 \pm 0.3	12.5 \pm 0.3	14408.9 \pm 49.5	5.6 \pm 0.4	0.2 \pm 0.0	87.2 \pm 0.3
adult	SBS	RFS	kNN	82.4 \pm 0.2	12.5 \pm 0.3	18470.5 \pm 46.4	0.1 \pm 0.0	333.3 \pm 2.6	89.2 \pm 0.2
adult	SBS	RFS	Nba.	81.6 \pm 0.2	12.4 \pm 0.3	18470.5 \pm 46.4	1607.6 \pm 340.3	0.4 \pm 0.0	81.7 \pm 0.2
adult	SBS	RFS	RNA	84.4 \pm 0.3	12.5 \pm 0.3	18470.5 \pm 46.4	1450.0 \pm 50.9	6.8 \pm 0.2	88.2 \pm 0.2
adult	SBS	RSC	C45	86.0 \pm 0.2	13.0 \pm 0.0	226.4 \pm 0.3	5.3 \pm 0.1	0.3 \pm 0.0	87.3 \pm 0.0
adult	SBS	RSC	kNN	82.7 \pm 0.2	13.0 \pm 0.0	247.7 \pm 0.3	0.1 \pm 0.0	368.7 \pm 3.9	89.5 \pm 0.0
adult	SBS	RSC	Nba.	81.0 \pm 0.2	13.0 \pm 0.0	247.1 \pm 0.8	1917.8 \pm 218.4	0.3 \pm 0.0	81.1 \pm 0.0
adult	SBS	RSC	RNA	84.9 \pm 0.2	13.0 \pm 0.0	245.5 \pm 0.3	684.4 \pm 0.2	6.0 \pm 0.0	87.7 \pm 0.1
adult	SBS	SU	C45	75.9 \pm 0.0	1.0 \pm 0.0	254.2 \pm 0.4	0.0 \pm 0.0	0.3 \pm 0.0	75.9 \pm 0.0
adult	SBS	SU	kNN	73.8 \pm 1.2	1.0 \pm 0.0	275.6 \pm 0.7	0.1 \pm 0.0	289.5 \pm 0.7	73.9 \pm 1.2
adult	SBS	SU	Nba.	75.9 \pm 0.0	1.0 \pm 0.0	273.0 \pm 0.2	0.0 \pm 0.0	0.3 \pm 0.0	75.9 \pm 0.0
adult	SBS	SU	RNA	75.9 \pm 0.0	1.0 \pm 0.0	274.6 \pm 0.2	138.1 \pm 0.0	1.1 \pm 0.0	75.9 \pm 0.0
adult	TS	IEP	C45	85.3 \pm 0.2	9.5 \pm 0.3	1507.4 \pm 25.0	4.1 \pm 0.1	0.2 \pm 0.0	86.8 \pm 0.1
adult	TS	IEP	kNN	82.7 \pm 0.2	8.7 \pm 0.3	2584.3 \pm 61.4	0.1 \pm 0.0	334.2 \pm 2.9	89.4 \pm 0.1
adult	TS	IEP	Nba.	83.1 \pm 0.3	8.7 \pm 0.3	2584.3 \pm 61.4	1929.3 \pm 214.4	0.4 \pm 0.0	83.2 \pm 0.2
adult	TS	IEP	RNA	84.8 \pm 0.3	8.7 \pm 0.3	2584.3 \pm 61.4	458.9 \pm 23.6	3.2 \pm 0.2	86.8 \pm 0.2
adult	TS	Inf	C45	85.0 \pm 0.3	8.9 \pm 0.4	1501.1 \pm 36.2	4.1 \pm 0.1	0.2 \pm 0.0	86.6 \pm 0.1
adult	TS	Inf	kNN	82.7 \pm 0.3	9.2 \pm 0.4	2743.1 \pm 60.1	0.1 \pm 0.0	207.0 \pm 1.7	89.5 \pm 0.0
adult	TS	Inf	Nba.	83.3 \pm 0.3	9.2 \pm 0.4	2743.1 \pm 60.1	1915.6 \pm 215.3	0.4 \pm 0.0	83.4 \pm 0.1
adult	TS	Inf	RNA	84.8 \pm 0.2	9.2 \pm 0.4	2743.1 \pm 60.1	394.5 \pm 30.4	2.6 \pm 0.2	87.1 \pm 0.2
adult	TS	Liu	C45	85.3 \pm 0.2	9.6 \pm 0.3	1483.0 \pm 13.8	4.2 \pm 0.1	0.2 \pm 0.0	86.9 \pm 0.1
adult	TS	Liu	kNN	82.6 \pm 0.2	8.7 \pm 0.3	2390.7 \pm 68.2	0.1 \pm 0.0	333.0 \pm 3.2	89.4 \pm 0.0
adult	TS	Liu	Nba.	83.2 \pm 0.3	8.7 \pm 0.3	2390.7 \pm 68.2	1893.3 \pm 219.9	0.4 \pm 0.0	83.3 \pm 0.1
adult	TS	Liu	RNA	84.9 \pm 0.3	8.7 \pm 0.3	2390.7 \pm 68.2	495.6 \pm 6.8	3.5 \pm 0.1	86.9 \pm 0.2
adult	TS	MDL	C45	79.2 \pm 0.3	5.0 \pm 0.3	22114.3 \pm 40.2	533.5 \pm 41.4	3.4 \pm 0.2	79.9 \pm 0.1
adult	TS	MDL	kNN	76.7 \pm 0.2	5.0 \pm 0.3	22114.3 \pm 40.2	0.1 \pm 0.0	207.5 \pm 3.8	85.1 \pm 0.0
adult	TS	MDL	Nba.	77.5 \pm 0.2	5.0 \pm 0.3	22114.3 \pm 40.2	1982.5 \pm 238.5	0.4 \pm 0.0	77.6 \pm 0.0
adult	TS	MDL	RNA	78.3 \pm 0.2	5.0 \pm 0.3	22114.3 \pm 40.2	533.5 \pm 41.4	3.4 \pm 0.2	79.9 \pm 0.1
adult	TS	RFS	C45	83.5 \pm 0.1	9.2 \pm 0.4	1938197.0 \pm 1699663.8	648.8 \pm 43.3	4.3 \pm 0.3	85.9 \pm 0.2
adult	TS	RFS	kNN	81.8 \pm 0.3	9.2 \pm 0.4	1938197.0 \pm 1699663.8	0.1 \pm 0.0	210.3 \pm 1.5	88.1 \pm 0.4
adult	TS	RFS	Nba.	81.2 \pm 0.4	9.2 \pm 0.4	1938197.0 \pm 1699663.8	625.0 \pm 295.7	0.3 \pm 0.0	81.3 \pm 0.3
adult	TS	RFS	RNA	83.4 \pm 0.2	9.2 \pm 0.4	1938197.0 \pm 1699663.8	648.8 \pm 43.3	4.3 \pm 0.3	85.9 \pm 0.2
adult	TS	RSC	C45	85.0 \pm 0.4	9.0 \pm 0.3	1455.5 \pm 28.9	4.1 \pm 0.1	0.2 \pm 0.0	86.7 \pm 0.2
adult	TS	RSC	kNN	82.6 \pm 0.2	9.8 \pm 0.4	2570.9 \pm 49.8	0.1 \pm 0.0	249.4 \pm 21.3	89.5 \pm 0.0
adult	TS	RSC	Nba.	83.1 \pm 0.3	9.8 \pm 0.4	2570.9 \pm 49.8	1910.9 \pm 219.5	0.4 \pm 0.0	83.2 \pm 0.2
adult	TS	RSC	RNA	84.8 \pm 0.2	9.8 \pm 0.4	2570.9 \pm 49.8	442.7 \pm 37.7	3.0 \pm 0.2	87.4 \pm 0.1
adult	TS	SU	C45	83.6 \pm 0.5	4.9 \pm 0.5	1427.1 \pm 29.9	1.5 \pm 0.3	0.2 \pm 0.0	83.7 \pm 0.5
adult	TS	SU	kNN	80.2 \pm 1.4	5.6 \pm 1.0	2620.2 \pm 80.4	0.1 \pm 0.0	193.2 \pm 5.0	81.6 \pm 1.8

continúa en la página siguiente

viene de la página anterior

Datos	Búsq.	Med.	Apr.	%AC ± se	nCar ± se	tSC ± se	tAprd. ± se	tTest ± se	Entr. ± se
adult	TS	SU	Nba.	81.0 ± 0.4	5.6 ± 1.0	2620.2 ± 80.4	376.0 ± 267.1	0.4 ± 0.0	81.0 ± 0.4
adult	TS	SU	RNA	81.9 ± 0.5	5.6 ± 1.0	2620.2 ± 80.4	288.6 ± 60.0	2.0 ± 0.4	82.5 ± 0.9
anneal	GA	IEP	C45	90.3 ± 1.1	23.1 ± 1.2	185.9 ± 6.4	0.1 ± 0.0	0.0 ± 0.0	93.2 ± 0.8
anneal	GA	IEP	kNN	86.4 ± 2.4	22.1 ± 0.4	61.3 ± 1.0	0.0 ± 0.0	0.2 ± 0.0	87.4 ± 2.2
anneal	GA	IEP	Nba.	91.5 ± 1.5	22.7 ± 0.9	179.0 ± 3.6	0.1 ± 0.0	0.0 ± 0.0	91.1 ± 1.0
anneal	GA	IEP	RNA	97.4 ± 0.9	21.3 ± 0.8	58.9 ± 1.8	7.7 ± 0.1	0.1 ± 0.0	98.6 ± 0.5
anneal	GA	Inf	C45	90.8 ± 1.4	24.8 ± 0.9	202.4 ± 4.2	0.1 ± 0.0	0.0 ± 0.0	93.2 ± 1.2
anneal	GA	Inf	kNN	84.5 ± 1.9	22.3 ± 0.9	62.9 ± 1.6	0.0 ± 0.0	0.2 ± 0.0	85.9 ± 2.3
anneal	GA	Inf	Nba.	90.1 ± 1.3	23.7 ± 0.8	198.0 ± 4.3	0.1 ± 0.0	0.0 ± 0.0	90.7 ± 0.8
anneal	GA	Inf	RNA	98.8 ± 0.5	20.9 ± 0.7	59.1 ± 1.5	7.7 ± 0.2	0.1 ± 0.0	99.4 ± 0.1
anneal	GA	Liu	C45	90.3 ± 1.3	22.2 ± 0.9	181.5 ± 5.0	0.1 ± 0.0	0.0 ± 0.0	91.5 ± 0.8
anneal	GA	Liu	kNN	85.5 ± 2.1	21.5 ± 1.1	58.2 ± 1.7	0.0 ± 0.0	0.2 ± 0.0	87.1 ± 2.1
anneal	GA	Liu	Nba.	91.5 ± 0.4	22.8 ± 0.9	165.5 ± 5.7	0.0 ± 0.0	0.0 ± 0.0	91.8 ± 0.6
anneal	GA	Liu	RNA	99.3 ± 0.2	21.9 ± 0.9	58.3 ± 1.5	7.9 ± 0.1	0.1 ± 0.0	99.3 ± 0.2
anneal	GA	MDL	C45	91.3 ± 0.8	33.7 ± 0.6	1946.8 ± 17.7	0.2 ± 0.0	0.0 ± 0.0	93.1 ± 0.6
anneal	GA	MDL	kNN	81.7 ± 2.4	31.0 ± 1.0	680.8 ± 3.8	0.0 ± 0.0	0.3 ± 0.0	83.0 ± 2.4
anneal	GA	MDL	Nba.	92.3 ± 0.8	32.7 ± 0.5	1789.6 ± 13.8	0.1 ± 0.0	0.0 ± 0.0	92.8 ± 0.3
anneal	GA	MDL	RNA	96.7 ± 0.9	31.6 ± 0.5	603.5 ± 1.3	9.1 ± 0.2	0.1 ± 0.0	98.2 ± 0.5
anneal	GA	RFS	C45	91.7 ± 1.1	21.0 ± 0.9	4715.4 ± 51.9	0.1 ± 0.0	0.0 ± 0.0	93.9 ± 0.5
anneal	GA	RFS	kNN	85.2 ± 1.9	21.3 ± 1.1	4752.4 ± 58.1	0.0 ± 0.0	0.3 ± 0.0	85.7 ± 1.8
anneal	GA	RFS	Nba.	89.9 ± 1.1	22.9 ± 1.1	26028.5 ± 409.7	0.1 ± 0.0	0.0 ± 0.0	91.6 ± 0.6
anneal	GA	RFS	RNA	94.8 ± 1.6	23.5 ± 0.7	4069.4 ± 7.8	8.1 ± 0.2	0.1 ± 0.0	96.8 ± 0.9
anneal	GA	RSC	C45	91.1 ± 1.3	23.5 ± 0.8	183.8 ± 3.9	0.1 ± 0.0	0.0 ± 0.0	93.2 ± 0.8
anneal	GA	RSC	kNN	85.2 ± 1.5	24.2 ± 1.5	63.2 ± 2.2	0.0 ± 0.0	0.3 ± 0.0	85.5 ± 1.6
anneal	GA	RSC	Nba.	90.4 ± 1.1	23.1 ± 0.9	184.7 ± 4.5	0.1 ± 0.0	0.0 ± 0.0	90.9 ± 0.7
anneal	GA	RSC	RNA	98.9 ± 0.5	21.5 ± 0.9	58.0 ± 1.8	7.7 ± 0.2	0.1 ± 0.0	99.3 ± 0.1
anneal	GA	SU	C45	81.8 ± 0.6	14.5 ± 0.8	135.8 ± 4.3	0.0 ± 0.0	0.0 ± 0.0	82.0 ± 0.4
anneal	GA	SU	kNN	87.4 ± 2.5	14.3 ± 0.6	47.8 ± 1.3	0.0 ± 0.0	0.2 ± 0.0	87.6 ± 2.3
anneal	GA	SU	Nba.	85.4 ± 1.3	13.9 ± 0.8	141.3 ± 4.3	0.0 ± 0.0	0.0 ± 0.0	84.9 ± 1.0
anneal	GA	SU	RNA	97.2 ± 1.2	15.2 ± 0.7	46.7 ± 1.6	5.9 ± 0.2	0.1 ± 0.0	97.3 ± 0.9
anneal	GA	Wra	C45	92.2 ± 0.9	21.2 ± 0.8	105.2 ± 5.5	0.0 ± 0.0	0.0 ± 0.0	94.6 ± 0.3
anneal	GA	Wra	kNN	97.4 ± 0.5	20.2 ± 0.7	691.3 ± 12.5	0.0 ± 0.0	0.2 ± 0.0	98.2 ± 0.2
anneal	GA	Wra	Nba.	93.8 ± 0.8	24.2 ± 0.7	372.3 ± 17.8	0.1 ± 0.0	0.0 ± 0.0	94.0 ± 0.5
anneal	GA	Wra	RNA	98.9 ± 0.5	21.3 ± 0.8	12690.9 ± 807.3	7.4 ± 0.5	0.1 ± 0.0	99.6 ± 0.1
anneal	LVF	IEP	C45	82.0 ± 1.4	4.2 ± 0.2	64.0 ± 0.4	0.0 ± 0.0	0.0 ± 0.0	82.0 ± 1.1
anneal	LVF	IEP	kNN	80.0 ± 1.9	4.2 ± 0.1	65.1 ± 0.6	0.0 ± 0.0	0.1 ± 0.0	81.0 ± 2.1
anneal	LVF	IEP	Nba.	78.0 ± 0.4	3.9 ± 0.2	65.0 ± 1.1	0.0 ± 0.0	0.0 ± 0.0	78.0 ± 0.5
anneal	LVF	IEP	RNA	88.0 ± 1.4	3.9 ± 0.2	63.4 ± 0.6	0.2 ± 0.0	0.0 ± 0.0	89.0 ± 1.2
anneal	LVF	Inf	C45	89.0 ± 1.2	15.1 ± 0.9	106.6 ± 3.1	0.0 ± 0.0	0.0 ± 0.0	91.0 ± 1.1
anneal	LVF	Inf	kNN	85.0 ± 2.3	13.4 ± 1.1	103.4 ± 3.3	0.0 ± 0.0	0.2 ± 0.0	85.0 ± 2.6
anneal	LVF	Inf	Nba.	88.0 ± 1.4	14.5 ± 0.8	102.7 ± 2.1	0.0 ± 0.0	0.0 ± 0.0	88.0 ± 0.8
anneal	LVF	Inf	RNA	98.0 ± 0.7	13.9 ± 0.8	105.0 ± 3.3	0.4 ± 0.0	0.0 ± 0.0	99.0 ± 0.2
anneal	LVF	Liu	C45	88.0 ± 1.7	11.7 ± 0.5	93.9 ± 1.3	0.0 ± 0.0	0.0 ± 0.0	88.0 ± 1.4
anneal	LVF	Liu	kNN	84.0 ± 2.6	10.6 ± 0.5	91.0 ± 1.2	0.0 ± 0.0	0.2 ± 0.0	84.0 ± 2.6
anneal	LVF	Liu	Nba.	86.0 ± 1.8	11.0 ± 0.6	91.2 ± 1.9	0.0 ± 0.0	0.0 ± 0.0	86.0 ± 1.9
anneal	LVF	Liu	RNA	98.0 ± 0.5	11.0 ± 1.0	91.8 ± 2.5	0.4 ± 0.0	0.0 ± 0.0	99.0 ± 0.1
anneal	LVF	RFS	C45	87.9 ± 0.8	18.0 ± 1.3	39222.8 ± 2013.4	0.0 ± 0.0	0.0 ± 0.0	91.0 ± 1.2
anneal	LVF	RFS	kNN	86.2 ± 2.2	17.1 ± 1.8	22794.1 ± 2339.5	0.0 ± 0.0	0.2 ± 0.0	86.5 ± 2.4
anneal	LVF	RFS	Nba.	91.4 ± 0.9	20.4 ± 1.3	32941.6 ± 2635.4	0.0 ± 0.0	0.0 ± 0.0	90.9 ± 1.0
anneal	LVF	RFS	RNA	94.3 ± 1.2	19.5 ± 2.5	39222.8 ± 0.0	3.3 ± 0.2	0.0 ± 0.0	95.3 ± 0.9
anneal	LVF	RSC	C45	89.0 ± 1.5	14.6 ± 0.9	103.5 ± 3.1	0.0 ± 0.0	0.0 ± 0.0	90.0 ± 1.4
anneal	LVF	RSC	kNN	85.0 ± 2.6	15.5 ± 1.0	105.8 ± 2.6	0.0 ± 0.0	0.2 ± 0.0	87.0 ± 2.7
anneal	LVF	RSC	Nba.	89.0 ± 1.6	14.6 ± 1.1	103.7 ± 3.7	0.0 ± 0.0	0.0 ± 0.0	88.0 ± 0.9
anneal	LVF	RSC	RNA	98.0 ± 0.7	17.5 ± 0.9	110.2 ± 3.3	0.5 ± 0.0	0.1 ± 0.0	99.0 ± 0.2
anneal	LVW	MDL	C45	91.4 ± 1.5	27.7 ± 0.3	102.0 ± 10.9	0.1 ± 0.0	0.0 ± 0.0	92.7 ± 0.4
anneal	LVW	MDL	kNN	81.3 ± 2.1	26.5 ± 0.3	86.4 ± 2.6	0.0 ± 0.0	0.2 ± 0.0	82.5 ± 2.3
anneal	LVW	MDL	Nba.	90.0 ± 1.0	27.2 ± 0.3	107.0 ± 10.7	0.0 ± 0.0	0.0 ± 0.0	91.3 ± 0.8
anneal	LVW	MDL	RNA	97.2 ± 0.9	26.4 ± 0.4	96.2 ± 6.4	0.5 ± 0.0	0.1 ± 0.0	97.3 ± 0.9
anneal	LVW	SU	C45	82.4 ± 1.2	16.9 ± 1.0	12.1 ± 1.1	0.0 ± 0.0	0.0 ± 0.0	82.8 ± 1.4
anneal	LVW	SU	kNN	82.2 ± 3.8	16.0 ± 0.7	13.0 ± 1.1	0.0 ± 0.0	0.1 ± 0.0	81.6 ± 3.9
anneal	LVW	SU	Nba.	86.1 ± 1.1	16.4 ± 0.9	11.8 ± 1.1	0.0 ± 0.0	0.0 ± 0.0	86.9 ± 0.7
anneal	LVW	SU	RNA	98.3 ± 0.3	18.1 ± 0.7	13.3 ± 1.5	0.4 ± 0.0	0.0 ± 0.0	98.1 ± 0.3
anneal	LVW	Wra	C45	92.0 ± 1.0	21.1 ± 0.8	46.9 ± 4.4	0.0 ± 0.0	0.0 ± 0.0	94.3 ± 0.2
anneal	LVW	Wra	kNN	97.4 ± 0.3	18.8 ± 0.5	196.7 ± 18.6	0.0 ± 0.0	0.2 ± 0.0	98.4 ± 0.1
anneal	LVW	Wra	Nba.	93.2 ± 0.7	20.7 ± 0.7	20.7 ± 3.0	0.0 ± 0.0	0.0 ± 0.0	94.0 ± 0.2
anneal	LVW	Wra	RNA	99.0 ± 0.3	22.2 ± 1.1	443.6 ± 53.6	0.5 ± 0.0	0.1 ± 0.0	99.6 ± 0.1
anneal	NoFS		C45	92.3 ± 0.6	38.0 ± 0.0	0.0 ± 0.0	0.1 ± 0.0	0.0 ± 0.0	94.5 ± 0.2
anneal	NoFS		kNN	83.0 ± 1.0	38.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.3 ± 0.0	83.9 ± 0.2
anneal	NoFS		Nba.	92.9 ± 0.8	38.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	93.2 ± 0.1
anneal	NoFS		RNA	99.1 ± 0.3	38.0 ± 0.0	0.0 ± 0.0	1.0 ± 0.0	0.1 ± 0.0	99.8 ± 0.0
anneal	SA	IEP	C45	89.0 ± 1.2	20.2 ± 0.7	114.7 ± 0.9	0.0 ± 0.0	0.0 ± 0.0	91.6 ± 1.1
anneal	SA	IEP	kNN	89.1 ± 2.4	18.5 ± 0.8	116.6 ± 1.2	0.0 ± 0.0	0.2 ± 0.0	90.7 ± 2.6
anneal	SA	IEP	Nba.	88.8 ± 1.5	21.2 ± 0.8	113.3 ± 1.1	0.0 ± 0.0	0.0 ± 0.0	89.4 ± 1.3
anneal	SA	IEP	RNA	99.0 ± 0.4	19.8 ± 1.1	115.0 ± 0.7	0.8 ± 0.0	0.1 ± 0.0	99.6 ± 0.1
anneal	SA	Inf	C45	91.0 ± 0.9	16.6 ± 0.8	123.5 ± 1.0	0.0 ± 0.0	0.0 ± 0.0	92.7 ± 0.7
anneal	SA	Inf	kNN	86.4 ± 2.2	17.7 ± 0.5	124.7 ± 1.4	0.0 ± 0.0	0.2 ± 0.0	86.7 ± 2.2
anneal	SA	Inf	Nba.	89.1 ± 1.1	18.1 ± 0.9	122.6 ± 0.7	0.0 ± 0.0	0.0 ± 0.0	89.5 ± 0.6

continúa en la página siguiente

viene de la página anterior

Datos	Búsq.	Med.	Apr.	%AC \pm se	nCar \pm se	tSC \pm se	tAprd. \pm se	tTest \pm se	Entr. \pm se
anneal	SA	Inf	RNA	98.8 \pm 0.5	17.6 \pm 0.7	123.7 \pm 0.7	0.7 \pm 0.0	0.1 \pm 0.0	99.6 \pm 0.1
anneal	SA	Liu	C45	88.4 \pm 1.5	19.1 \pm 0.8	116.0 \pm 1.1	0.0 \pm 0.0	0.0 \pm 0.0	91.8 \pm 1.0
anneal	SA	Liu	kNN	88.1 \pm 2.6	19.1 \pm 0.6	119.0 \pm 1.1	0.0 \pm 0.0	0.3 \pm 0.0	88.9 \pm 2.5
anneal	SA	Liu	Nba.	87.1 \pm 2.0	19.9 \pm 1.2	111.8 \pm 1.5	0.0 \pm 0.0	0.0 \pm 0.0	89.3 \pm 1.1
anneal	SA	Liu	RNA	99.3 \pm 0.3	19.6 \pm 0.7	116.3 \pm 0.7	0.8 \pm 0.0	0.1 \pm 0.0	99.6 \pm 0.1
anneal	SA	MDL	C45	92.1 \pm 0.6	38.0 \pm 0.0	1321.3 \pm 1.8	0.1 \pm 0.0	0.0 \pm 0.0	94.4 \pm 0.3
anneal	SA	MDL	kNN	83.0 \pm 1.0	38.0 \pm 0.0	1323.7 \pm 1.3	0.0 \pm 0.0	0.3 \pm 0.0	83.9 \pm 0.2
anneal	SA	MDL	Nba.	92.9 \pm 0.8	38.0 \pm 0.0	1328.6 \pm 1.8	0.1 \pm 0.0	0.0 \pm 0.0	93.2 \pm 0.1
anneal	SA	MDL	RNA	99.1 \pm 0.3	38.0 \pm 0.0	1307.0 \pm 13.0	1.3 \pm 0.0	0.1 \pm 0.0	99.8 \pm 0.0
anneal	SA	RFS	C45	90.7 \pm 1.3	22.8 \pm 0.8	21184.7 \pm 115.8	0.1 \pm 0.0	0.0 \pm 0.0	93.8 \pm 0.6
anneal	SA	RFS	kNN	86.5 \pm 1.8	22.4 \pm 0.5	21054.3 \pm 13.8	0.0 \pm 0.0	0.3 \pm 0.0	86.7 \pm 1.6
anneal	SA	RFS	Nba.	91.0 \pm 1.1	21.8 \pm 1.1	21243.1 \pm 60.1	0.0 \pm 0.0	0.0 \pm 0.0	92.0 \pm 0.7
anneal	SA	RFS	RNA	94.7 \pm 1.2	22.1 \pm 0.7	20175.4 \pm 163.9	0.7 \pm 0.0	0.1 \pm 0.0	96.2 \pm 0.7
anneal	SA	RSC	C45	89.4 \pm 1.2	16.9 \pm 0.8	117.2 \pm 0.9	0.0 \pm 0.0	0.0 \pm 0.0	91.3 \pm 1.1
anneal	SA	RSC	kNN	84.1 \pm 2.0	18.4 \pm 0.7	120.5 \pm 1.0	0.0 \pm 0.0	0.2 \pm 0.0	84.8 \pm 1.9
anneal	SA	RSC	Nba.	86.2 \pm 1.6	18.0 \pm 0.6	115.2 \pm 0.7	0.0 \pm 0.0	0.0 \pm 0.0	86.6 \pm 1.3
anneal	SA	RSC	RNA	99.0 \pm 0.4	16.9 \pm 0.5	118.4 \pm 0.8	0.7 \pm 0.0	0.1 \pm 0.0	99.5 \pm 0.1
anneal	SA	SU	C45	81.7 \pm 0.8	13.6 \pm 0.6	112.5 \pm 0.9	0.0 \pm 0.0	0.0 \pm 0.0	82.4 \pm 0.7
anneal	SA	SU	kNN	85.6 \pm 2.3	13.6 \pm 0.4	115.6 \pm 1.6	0.0 \pm 0.0	0.2 \pm 0.0	85.9 \pm 2.4
anneal	SA	SU	Nba.	84.5 \pm 0.9	14.9 \pm 0.9	112.5 \pm 0.8	0.0 \pm 0.0	0.0 \pm 0.0	84.9 \pm 1.0
anneal	SA	SU	RNA	95.7 \pm 1.4	15.2 \pm 0.7	112.8 \pm 0.9	0.5 \pm 0.0	0.1 \pm 0.0	97.2 \pm 0.7
anneal	SA	Wra	C45	91.1 \pm 0.9	22.2 \pm 0.6	249.5 \pm 2.8	0.1 \pm 0.0	0.0 \pm 0.0	94.7 \pm 0.3
anneal	SA	Wra	kNN	98.3 \pm 0.6	19.2 \pm 1.0	1592.2 \pm 12.9	0.0 \pm 0.0	0.2 \pm 0.0	98.9 \pm 0.1
anneal	SA	Wra	Nba.	93.9 \pm 0.6	24.1 \pm 0.8	195.9 \pm 3.1	0.0 \pm 0.0	0.0 \pm 0.0	94.7 \pm 0.3
anneal	SA	Wra	RNA	99.1 \pm 0.4	22.4 \pm 0.8	935.9 \pm 15.8	0.6 \pm 0.0	0.1 \pm 0.0	99.7 \pm 0.0
anneal	SFS	IEP	C45	92.5 \pm 0.8	9.1 \pm 0.3	11.0 \pm 0.4	0.0 \pm 0.0	0.0 \pm 0.0	93.9 \pm 0.2
anneal	SFS	IEP	kNN	82.5 \pm 1.0	9.1 \pm 0.3	10.9 \pm 0.3	0.0 \pm 0.0	0.1 \pm 0.0	83.7 \pm 0.6
anneal	SFS	IEP	Nba.	85.6 \pm 1.0	9.1 \pm 0.3	10.9 \pm 0.3	0.0 \pm 0.0	0.0 \pm 0.0	86.4 \pm 0.5
anneal	SFS	IEP	RNA	99.0 \pm 0.4	9.1 \pm 0.3	11.0 \pm 0.4	0.3 \pm 0.0	0.0 \pm 0.0	99.3 \pm 0.1
anneal	SFS	Inf	C45	91.0 \pm 0.8	30.5 \pm 1.4	4.5 \pm 0.2	0.1 \pm 0.0	0.0 \pm 0.0	92.0 \pm 0.5
anneal	SFS	Inf	kNN	84.2 \pm 0.8	30.5 \pm 1.4	4.5 \pm 0.2	0.0 \pm 0.0	0.2 \pm 0.0	85.3 \pm 0.3
anneal	SFS	Inf	Nba.	90.9 \pm 0.9	30.5 \pm 1.4	4.5 \pm 0.2	0.0 \pm 0.0	0.0 \pm 0.0	91.5 \pm 0.4
anneal	SFS	Inf	RNA	99.2 \pm 0.3	30.5 \pm 1.4	4.5 \pm 0.2	0.7 \pm 0.0	0.1 \pm 0.0	99.9 \pm 0.0
anneal	SFS	Liu	C45	87.2 \pm 1.5	9.2 \pm 0.1	10.1 \pm 0.3	0.0 \pm 0.0	0.0 \pm 0.0	89.0 \pm 1.0
anneal	SFS	Liu	kNN	89.8 \pm 1.6	9.2 \pm 0.1	10.1 \pm 0.3	0.0 \pm 0.0	0.1 \pm 0.0	90.4 \pm 2.1
anneal	SFS	Liu	Nba.	88.6 \pm 0.5	9.2 \pm 0.1	9.8 \pm 0.3	0.0 \pm 0.0	0.0 \pm 0.0	89.0 \pm 0.3
anneal	SFS	Liu	RNA	99.1 \pm 0.4	9.2 \pm 0.1	10.2 \pm 0.3	0.3 \pm 0.0	0.0 \pm 0.0	99.5 \pm 0.1
anneal	SFS	MDL	C45	92.0 \pm 0.6	38.0 \pm 0.0	31.1 \pm 0.0	0.1 \pm 0.0	0.0 \pm 0.0	94.3 \pm 0.2
anneal	SFS	MDL	kNN	83.0 \pm 1.0	38.0 \pm 0.0	31.2 \pm 0.1	0.0 \pm 0.0	0.2 \pm 0.0	83.9 \pm 0.2
anneal	SFS	MDL	Nba.	92.9 \pm 0.8	38.0 \pm 0.0	31.0 \pm 0.1	0.0 \pm 0.0	0.0 \pm 0.0	93.2 \pm 0.1
anneal	SFS	MDL	RNA	99.1 \pm 0.3	38.0 \pm 0.0	31.2 \pm 0.1	0.9 \pm 0.0	0.1 \pm 0.0	99.8 \pm 0.0
anneal	SFS	RFS	C45	91.9 \pm 1.1	16.9 \pm 1.1	8242.2 \pm 1349.6	0.0 \pm 0.0	0.0 \pm 0.0	93.9 \pm 0.3
anneal	SFS	RFS	kNN	85.4 \pm 2.7	12.9 \pm 1.4	4201.3 \pm 522.6	0.0 \pm 0.0	0.2 \pm 0.0	86.6 \pm 2.5
anneal	SFS	RFS	Nba.	90.5 \pm 1.1	21.3 \pm 1.6	11929.9 \pm 552.2	0.0 \pm 0.0	0.0 \pm 0.0	91.6 \pm 1.0
anneal	SFS	RFS	RNA	90.5 \pm 2.0	10.0 \pm 0.9	4201.3 \pm 522.2	2.6 \pm 0.1	0.0 \pm 0.0	91.9 \pm 1.4
anneal	SFS	RSC	C45	89.3 \pm 1.5	8.2 \pm 0.1	9.0 \pm 0.1	0.0 \pm 0.0	0.0 \pm 0.0	91.0 \pm 0.9
anneal	SFS	RSC	kNN	81.4 \pm 0.8	8.2 \pm 0.1	9.0 \pm 0.1	0.0 \pm 0.0	0.1 \pm 0.0	82.2 \pm 0.6
anneal	SFS	RSC	Nba.	83.3 \pm 1.1	8.2 \pm 0.1	8.9 \pm 0.1	0.0 \pm 0.0	0.0 \pm 0.0	84.4 \pm 0.5
anneal	SFS	RSC	RNA	99.1 \pm 0.4	8.2 \pm 0.1	8.9 \pm 0.1	0.3 \pm 0.0	0.0 \pm 0.0	99.3 \pm 0.1
anneal	SFS	SU	C45	80.2 \pm 0.3	6.0 \pm 0.5	45.6 \pm 0.1	0.0 \pm 0.0	0.0 \pm 0.0	80.2 \pm 0.0
anneal	SFS	SU	kNN	22.9 \pm 1.6	6.0 \pm 0.5	47.1 \pm 0.2	0.0 \pm 0.0	0.1 \pm 0.0	21.7 \pm 0.9
anneal	SFS	SU	Nba.	88.6 \pm 0.5	6.0 \pm 0.5	45.3 \pm 0.3	0.0 \pm 0.0	0.0 \pm 0.0	89.4 \pm 0.2
anneal	SFS	SU	RNA	89.0 \pm 0.5	6.0 \pm 0.5	48.1 \pm 0.2	0.2 \pm 0.0	0.0 \pm 0.0	89.6 \pm 0.2
anneal	SFS	Wra	C45	92.4 \pm 0.8	28.0 \pm 1.8	199.7 \pm 10.0	0.1 \pm 0.0	0.0 \pm 0.0	94.8 \pm 0.3
anneal	SFS	Wra	kNN	86.4 \pm 1.9	13.8 \pm 2.7	1320.7 \pm 7.0	0.0 \pm 0.0	0.3 \pm 0.0	87.3 \pm 2.1
anneal	SFS	Wra	Nba.	95.0 \pm 0.6	31.8 \pm 2.4	116.7 \pm 9.2	0.1 \pm 0.0	0.0 \pm 0.0	95.0 \pm 0.2
anneal	SFS	Wra	RNA	83.6 \pm 0.2	1.0 \pm 0.0	1980.2 \pm 5.7	0.1 \pm 0.0	0.0 \pm 0.0	83.6 \pm 0.0
anneal	SBS	IEP	C45	90.5 \pm 0.6	10.4 \pm 0.2	122.6 \pm 0.4	0.0 \pm 0.0	0.0 \pm 0.0	93.6 \pm 0.3
anneal	SBS	IEP	kNN	83.0 \pm 0.6	10.4 \pm 0.2	122.3 \pm 0.2	0.0 \pm 0.0	0.3 \pm 0.0	86.1 \pm 0.9
anneal	SBS	IEP	Nba.	86.5 \pm 0.4	10.4 \pm 0.2	118.1 \pm 0.2	0.1 \pm 0.0	0.0 \pm 0.0	87.1 \pm 0.2
anneal	SBS	IEP	RNA	97.1 \pm 0.7	10.4 \pm 0.5	75.6 \pm 0.2	0.4 \pm 0.0	0.0 \pm 0.0	98.8 \pm 0.2
anneal	SBS	Inf	C45	90.5 \pm 0.6	10.4 \pm 0.2	124.1 \pm 0.3	0.0 \pm 0.0	0.0 \pm 0.0	93.6 \pm 0.3
anneal	SBS	Inf	kNN	83.0 \pm 0.6	10.4 \pm 0.2	125.0 \pm 0.4	0.0 \pm 0.0	0.3 \pm 0.0	86.1 \pm 0.9
anneal	SBS	Inf	Nba.	86.5 \pm 0.4	10.4 \pm 0.2	120.9 \pm 0.2	0.1 \pm 0.0	0.0 \pm 0.0	87.1 \pm 0.2
anneal	SBS	Inf	RNA	97.1 \pm 0.7	10.4 \pm 0.5	106.2 \pm 7.3	0.5 \pm 0.0	0.1 \pm 0.0	98.8 \pm 0.2
anneal	SBS	Liu	C45	90.5 \pm 0.6	10.4 \pm 0.2	122.3 \pm 0.3	0.0 \pm 0.0	0.0 \pm 0.0	93.6 \pm 0.3
anneal	SBS	Liu	kNN	83.0 \pm 0.6	10.4 \pm 0.2	123.0 \pm 0.2	0.0 \pm 0.0	0.3 \pm 0.0	86.1 \pm 0.9
anneal	SBS	Liu	Nba.	86.5 \pm 0.4	10.4 \pm 0.2	116.9 \pm 0.3	0.1 \pm 0.0	0.0 \pm 0.0	87.1 \pm 0.2
anneal	SBS	Liu	RNA	97.1 \pm 0.7	10.4 \pm 0.5	101.7 \pm 7.0	0.5 \pm 0.0	0.0 \pm 0.0	98.8 \pm 0.2
anneal	SBS	MDL	C45	92.6 \pm 0.2	38.0 \pm 0.0	892.4 \pm 10.7	0.2 \pm 0.0	0.0 \pm 0.0	94.1 \pm 0.1
anneal	SBS	MDL	kNN	83.0 \pm 1.0	38.0 \pm 0.0	870.6 \pm 55.0	0.0 \pm 0.0	0.3 \pm 0.0	83.9 \pm 0.2
anneal	SBS	MDL	Nba.	92.9 \pm 0.3	38.0 \pm 0.0	952.6 \pm 3.0	0.1 \pm 0.0	0.0 \pm 0.0	93.2 \pm 0.0
anneal	SBS	MDL	RNA	99.1 \pm 0.3	38.0 \pm 0.0	816.0 \pm 73.8	1.1 \pm 0.1	0.1 \pm 0.0	99.8 \pm 0.0
anneal	SBS	RFS	C45	93.0 \pm 0.8	29.6 \pm 2.3	12658.2 \pm 1387.6	0.1 \pm 0.0	0.0 \pm 0.0	94.8 \pm 0.3
anneal	SBS	RFS	kNN	84.6 \pm 1.3	31.6 \pm 1.3	12658.2 \pm 0.0	0.0 \pm 0.0	0.2 \pm 0.0	86.1 \pm 0.8
anneal	SBS	RFS	Nba.	92.7 \pm 0.8	31.3 \pm 1.6	13722.8 \pm 1593.5	0.1 \pm 0.0	0.0 \pm 0.0	93.4 \pm 0.3
anneal	SBS	RFS	RNA	98.9 \pm 0.3	31.6 \pm 1.3	12658.2 \pm 0.0	4.2 \pm 0.1	0.1 \pm 0.0	99.6 \pm 0.1

continúa en la página siguiente

viene de la página anterior

Datos	Búsq.	Med.	Apr.	%AC \pm se	nCar \pm se	tSC \pm se	tAprd. \pm se	tTest \pm se	Entr. \pm se
anneal	SBS	RSC	C45	90.5 \pm 0.6	10.4 \pm 0.2	123.0 \pm 0.3	0.0 \pm 0.0	0.0 \pm 0.0	93.6 \pm 0.3
anneal	SBS	RSC	kNN	83.0 \pm 0.6	10.4 \pm 0.2	122.6 \pm 0.3	0.0 \pm 0.0	0.3 \pm 0.0	86.1 \pm 0.9
anneal	SBS	RSC	Nba.	86.5 \pm 0.4	10.4 \pm 0.2	117.6 \pm 0.2	0.1 \pm 0.0	0.0 \pm 0.0	87.1 \pm 0.2
anneal	SBS	RSC	RNA	97.1 \pm 0.7	10.4 \pm 0.5	75.2 \pm 0.1	0.4 \pm 0.0	0.0 \pm 0.0	98.8 \pm 0.2
anneal	SBS	SU	C45	81.2 \pm 0.2	12.2 \pm 0.3	111.2 \pm 0.8	0.0 \pm 0.0	0.0 \pm 0.0	81.8 \pm 0.0
anneal	SBS	SU	kNN	93.3 \pm 0.6	12.2 \pm 0.3	112.0 \pm 0.3	0.0 \pm 0.0	0.3 \pm 0.0	93.4 \pm 0.6
anneal	SBS	SU	Nba.	84.8 \pm 0.4	12.2 \pm 0.3	109.0 \pm 0.2	0.0 \pm 0.0	0.0 \pm 0.0	84.9 \pm 0.0
anneal	SBS	SU	RNA	97.9 \pm 0.4	13.0 \pm 0.7	116.3 \pm 2.6	0.5 \pm 0.0	0.1 \pm 0.0	98.5 \pm 0.1
anneal	SBS	Wra	C45	91.8 \pm 0.9	26.6 \pm 2.4	246.0 \pm 11.7	0.1 \pm 0.0	0.0 \pm 0.0	95.4 \pm 0.3
anneal	SBS	Wra	kNN	98.7 \pm 0.4	7.5 \pm 0.3	1609.1 \pm 88.1	0.0 \pm 0.0	0.2 \pm 0.0	98.9 \pm 0.1
anneal	SBS	Wra	Nba.	94.7 \pm 0.6	13.8 \pm 0.4	221.1 \pm 7.1	0.0 \pm 0.0	0.0 \pm 0.0	95.4 \pm 0.1
anneal	SBS	Wra	RNA	99.0 \pm 0.4	32.0 \pm 0.0	2911.3 \pm 9.7	1.0 \pm 0.0	0.1 \pm 0.0	99.2 \pm 0.0
anneal	TS	IEP	C45	91.5 \pm 0.9	21.8 \pm 0.8	53.8 \pm 1.6	0.1 \pm 0.0	0.0 \pm 0.0	93.7 \pm 0.4
anneal	TS	IEP	kNN	86.6 \pm 2.5	18.9 \pm 0.7	52.4 \pm 1.3	0.0 \pm 0.0	0.2 \pm 0.0	86.2 \pm 2.4
anneal	TS	IEP	Nba.	90.5 \pm 1.0	20.8 \pm 0.9	49.4 \pm 1.8	0.0 \pm 0.0	0.0 \pm 0.0	90.8 \pm 0.7
anneal	TS	IEP	RNA	98.7 \pm 0.5	19.7 \pm 1.0	46.2 \pm 1.6	9.1 \pm 0.2	0.1 \pm 0.0	99.5 \pm 0.1
anneal	TS	Inf	C45	89.1 \pm 1.6	19.2 \pm 0.8	51.1 \pm 1.2	0.1 \pm 0.0	0.0 \pm 0.0	92.7 \pm 0.5
anneal	TS	Inf	kNN	88.0 \pm 2.1	19.4 \pm 0.8	53.2 \pm 1.7	0.0 \pm 0.0	0.2 \pm 0.0	89.1 \pm 2.6
anneal	TS	Inf	Nba.	88.7 \pm 1.6	18.7 \pm 0.9	51.2 \pm 1.5	0.0 \pm 0.0	0.0 \pm 0.0	89.2 \pm 0.6
anneal	TS	Inf	RNA	98.0 \pm 0.5	17.6 \pm 1.1	47.2 \pm 1.1	8.6 \pm 0.2	0.1 \pm 0.0	99.3 \pm 0.1
anneal	TS	Liu	C45	91.9 \pm 0.7	19.2 \pm 0.8	49.6 \pm 1.5	0.0 \pm 0.0	0.0 \pm 0.0	93.8 \pm 0.7
anneal	TS	Liu	kNN	89.1 \pm 2.5	18.5 \pm 0.6	49.3 \pm 0.8	0.0 \pm 0.0	0.2 \pm 0.0	89.8 \pm 2.6
anneal	TS	Liu	Nba.	91.5 \pm 0.9	20.2 \pm 0.8	48.3 \pm 1.3	0.0 \pm 0.0	0.0 \pm 0.0	91.5 \pm 0.8
anneal	TS	Liu	RNA	98.8 \pm 0.4	19.1 \pm 1.0	45.3 \pm 1.6	9.1 \pm 0.2	0.1 \pm 0.0	99.6 \pm 0.1
anneal	TS	MDL	C45	82.5 \pm 1.7	21.3 \pm 0.6	603.2 \pm 6.2	0.1 \pm 0.0	0.0 \pm 0.0	86.2 \pm 2.2
anneal	TS	MDL	kNN	77.4 \pm 6.8	20.0 \pm 0.8	600.0 \pm 7.8	0.0 \pm 0.0	0.2 \pm 0.0	79.6 \pm 6.7
anneal	TS	MDL	Nba.	86.9 \pm 1.6	21.2 \pm 0.7	527.9 \pm 1.2	0.0 \pm 0.0	0.0 \pm 0.0	86.8 \pm 1.2
anneal	TS	MDL	RNA	92.2 \pm 1.8	21.8 \pm 0.6	527.3 \pm 1.1	8.7 \pm 0.2	0.1 \pm 0.0	93.5 \pm 1.6
anneal	TS	RFS	C45	90.3 \pm 1.4	18.7 \pm 1.1	4137.2 \pm 59.9	0.1 \pm 0.0	0.0 \pm 0.0	92.2 \pm 0.7
anneal	TS	RFS	kNN	83.7 \pm 3.7	19.1 \pm 1.1	4029.6 \pm 78.9	0.0 \pm 0.0	0.2 \pm 0.0	85.0 \pm 3.8
anneal	TS	RFS	Nba.	89.0 \pm 1.4	20.5 \pm 0.9	3454.8 \pm 3.4	0.0 \pm 0.0	0.0 \pm 0.0	90.2 \pm 0.8
anneal	TS	RFS	RNA	95.1 \pm 1.4	20.5 \pm 1.2	3452.8 \pm 4.6	9.2 \pm 0.3	0.1 \pm 0.0	96.7 \pm 1.1
anneal	TS	RSC	C45	90.1 \pm 1.3	18.9 \pm 1.1	48.3 \pm 1.8	0.0 \pm 0.0	0.0 \pm 0.0	93.1 \pm 0.6
anneal	TS	RSC	kNN	84.3 \pm 2.9	20.1 \pm 1.2	53.4 \pm 1.5	0.0 \pm 0.0	0.3 \pm 0.0	86.8 \pm 2.9
anneal	TS	RSC	Nba.	90.2 \pm 1.6	18.8 \pm 0.7	47.2 \pm 0.9	0.0 \pm 0.0	0.0 \pm 0.0	90.8 \pm 0.5
anneal	TS	RSC	RNA	98.9 \pm 0.4	19.9 \pm 1.1	44.8 \pm 1.8	9.4 \pm 0.3	0.1 \pm 0.0	99.5 \pm 0.1
anneal	TS	SU	C45	81.5 \pm 0.6	16.9 \pm 1.0	48.3 \pm 1.4	0.0 \pm 0.0	0.0 \pm 0.0	81.9 \pm 0.3
anneal	TS	SU	kNN	85.9 \pm 7.0	16.4 \pm 1.3	51.5 \pm 1.4	0.0 \pm 0.0	0.2 \pm 0.0	86.4 \pm 7.4
anneal	TS	SU	Nba.	85.2 \pm 1.1	19.0 \pm 0.7	48.6 \pm 1.0	0.0 \pm 0.0	0.0 \pm 0.0	85.2 \pm 0.6
anneal	TS	SU	RNA	98.0 \pm 0.6	19.1 \pm 0.6	48.6 \pm 1.2	7.6 \pm 0.2	0.1 \pm 0.0	98.9 \pm 0.1
anneal	TS	Wra	C45	91.2 \pm 0.9	20.2 \pm 0.9	113.6 \pm 6.5	0.1 \pm 0.0	0.0 \pm 0.0	94.2 \pm 0.5
anneal	TS	Wra	kNN	98.2 \pm 0.5	18.7 \pm 0.7	685.2 \pm 21.9	0.0 \pm 0.0	0.2 \pm 0.0	98.7 \pm 0.1
anneal	TS	Wra	Nba.	93.4 \pm 0.6	20.2 \pm 0.7	56.5 \pm 3.9	0.0 \pm 0.0	0.0 \pm 0.0	94.6 \pm 0.3
anneal	TS	Wra	RNA	99.3 \pm 0.3	19.5 \pm 1.2	10785.1 \pm 407.4	5.7 \pm 0.3	0.1 \pm 0.0	99.6 \pm 0.1
audiology	GA	IEP	C45	74.4 \pm 4.0	39.9 \pm 1.2	27.9 \pm 0.7	0.0 \pm 0.0	0.0 \pm 0.0	82.5 \pm 1.4
audiology	GA	IEP	kNN	69.9 \pm 2.7	39.6 \pm 1.2	29.2 \pm 0.8	0.0 \pm 0.0	0.0 \pm 0.0	83.1 \pm 0.5
.
.
.

Tabla A.1: Muestra de los resultados detallados de la batería de experimentos descrita en el capítulo 4

A.2. Otras tablas y figuras

A continuación, se exponen otras tablas y figuras, algunas mencionadas en el trabajo, y otras expuestas aquí para mostrar algunos resultados con mayor detalle.

<i>Conjunto de datos</i>	<i>NBayes</i> %Ac. Rank.	<i>C4.5</i> %Ac. Rank.	<i>k-NN</i> %Ac. Rank.	<i>RNA</i> %Ac. Rank.
SAheart	69.89 (3.0)	71.64 (2.0)	66.86 (4.0)	73.58 (1.0)
adult	81.90 (4.0)	86.22 (1.0)	82.73 (3.0)	84.01 (2.0)
anneal	92.87 (2.0)	92.32 (3.0)	82.97 (4.0)	99.11 (1.0)
audiology	69.92 (4.0)	79.66 (2.0)	70.75 (3.0)	80.57 (1.0)
balance-scale	91.84 (2.0)	61.77 (4.0)	76.79 (3.0)	99.36 (1.0)
breast-cancer	74.42 (2.0)	74.82 (1.0)	71.34 (3.0)	68.79 (4.0)
bupa	66.39 (3.0)	69.90 (2.0)	62.87 (4.0)	73.64 (1.0)
car	85.70 (4.0)	92.77 (2.0)	88.25 (3.0)	99.88 (1.0)
credit	85.22 (2.0)	84.35 (3.0)	86.52 (1.0)	83.19 (4.0)
echocardiogram	86.98 (2.5)	86.98 (2.5)	85.44 (4.0)	89.34 (1.0)
horse-colic	83.41 (2.0)	85.31 (1.0)	65.78 (4.0)	83.39 (3.0)
house-votes84	90.32 (4.0)	96.09 (1.0)	93.10 (3.0)	94.73 (2.0)
ionosphere	88.91 (3.0)	89.18 (2.0)	85.75 (4.0)	92.30 (1.0)
iris	94.00 (4.0)	95.33 (3.0)	96.00 (2.0)	97.33 (1.0)
labor-neg	63.33 (4.0)	77.00 (2.0)	73.67 (3.0)	96.00 (1.0)
led24-1200	75.75 (1.0)	73.50 (2.0)	59.50 (4.0)	62.67 (3.0)
lenses	73.33 (3.0)	85.00 (1.0)	66.67 (4.0)	76.67 (2.0)
lung-cancer	56.67 (1.0)	52.50 (2.0)	47.50 (3.0)	45.83 (4.0)
lymphography	47.76 (1.0)	46.57 (2.0)	37.24 (4.0)	42.57 (3.0)
mushrooms	99.74 (4.0)	100.00 (2.0)	100.00 (2.0)	100.00 (2.0)
parity3+3	55.00 (4.0)	100.00 (2.0)	100.00 (2.0)	100.00 (2.0)
pima	76.18 (1.0)	73.06 (4.0)	74.23 (3.0)	75.92 (2.0)
post-operative	66.67 (3.0)	70.00 (1.0)	68.89 (2.0)	58.89 (4.0)
primary-tumor	49.25 (1.0)	43.10 (3.0)	40.42 (4.0)	44.25 (2.0)
promoters	86.82 (1.0)	80.36 (3.0)	86.64 (2.0)	79.36 (4.0)
shuttle-landing	93.31 (4.0)	96.05 (2.0)	94.86 (3.0)	98.83 (1.0)
soybean	91.20 (1.0)	83.67 (4.0)	84.70 (3.0)	89.88 (2.0)
splice	95.61 (1.0)	94.42 (2.0)	81.94 (4.0)	93.79 (3.0)
tic-tac-toe	69.84 (4.0)	86.75 (3.0)	91.65 (2.0)	97.70 (1.0)
vehicle	45.76 (4.0)	71.76 (2.0)	71.05 (3.0)	83.34 (1.0)
vowel	68.99 (4.0)	80.00 (3.0)	96.46 (1.0)	88.79 (2.0)
wdbc	95.08 (3.0)	93.15 (4.0)	97.01 (2.0)	97.19 (1.0)
wine	99.44 (1.0)	93.86 (4.0)	96.08 (3.0)	98.30 (2.0)
yeast	16.44 (4.0)	55.18 (3.0)	57.75 (2.0)	60.31 (1.0)
yeast-class-RPR	100.00 (1.0)	95.73 (4.0)	96.75 (3.0)	99.44 (2.0)
zoo	90.00 (4.0)	91.09 (3.0)	95.09 (1.0)	92.00 (2.0)
Media	77.17 (2.7)	80.81 (2.4)	78.70 (2.9)	83.36 (2.0)

Tabla A.2: Rendimiento de los métodos de clasificación sin selección de características en cada conjunto de datos

<i>Método de selección</i>	<i>% Acierto</i>		<i>Nº características</i>	
	Media	Ranking medio	Media	Ranking medio
n5-info	76.51	13.74	4.91	16.07
n5-gain	73.85	14.19	4.91	16.07
n5-gini	76.82	13.76	4.91	16.07
n5-reli	77.17	15.56	4.91	16.07
n5-rele	75.15	15.00	4.91	16.07
p05-info	77.36	12.06	11.29	19.59
p05-gain	77.15	11.82	11.29	19.59
p05-gini	77.54	11.59	11.29	19.59
p05-reli	78.22	12.00	11.29	19.59
p05-rele	76.37	13.91	11.29	19.59
t01-info	75.67	15.66	8.47	14.81
t01-gain	73.87	15.51	6.57	11.96
t01-gini	66.73	21.57	2.31	5.51
t01-reli	79.15	11.17	22.32	27.96
t01-rele	73.31	14.89	13.16	19.37
pm05-info	75.88	14.62	4.88	12.82
pm05-gain	73.11	16.04	5.25	12.01
pm05-gini	74.65	15.49	4.56	11.79
pm05-reli	76.97	14.29	5.49	14.56
pm05-rele	73.26	15.99	6.14	14.22
d01-info	77.88	13.32	18.08	21.13
d01-gain	76.79	12.62	17.54	21.96
d01-gini	79.09	11.28	22.29	27.66
d01-reli	79.07	11.38	22.09	27.57
d01-rele	77.31	14.18	11.71	17.57
s1-info	66.13	22.78	1.32	4.35
s1-gain	65.79	22.74	1.50	4.68
s1-gini	66.40	22.74	1.34	4.51
s1-reli	66.45	22.61	1.59	6.57
s1-rele	66.45	22.48	1.61	5.66

Tabla A.3: Comparación de los métodos de selección de características que usan valoración individual de las características

<i>Conjunto</i>	<i>NoFS</i>	<i>%Ac.</i>	<i>SBS-Wra</i>	<i>%Ac.</i>	<i>Diferencia</i>
SAheart		73.580		70.100	3.480
anneal		99.110		99.000	0.110
audiology		80.570		77.450	3.120
balance-scale		99.360		99.370	-0.010
breast-cancer		68.790		66.710	2.080
bupa		73.640		72.170	1.470
car		99.880		98.790	1.090
credit		83.190		84.930	-1.740
echocardiogram		89.340		89.290	0.050
horse-colic		83.390		84.800	-1.410
house-votes84		94.730		95.870	-1.140
ionosphere		92.300		90.020	2.280
iris		97.330		95.330	2.000
labor-neg		96.000		92.670	3.330
led24-1200		62.670		74.330	-11.660
lenses		76.670		85.000	-8.330
lung-cancer		45.830		40.830	5.000
lymphography		42.570		42.430	0.140
mushrooms		100.000		100.000	0.000
parity3+3		100.000		100.000	0.000
pima		75.920		77.870	-1.950
post-operative		58.890		63.330	-4.440
primary-tumor		44.250		43.640	0.610
promoters		79.360		90.730	-11.370
shuttle-landing		98.830		98.820	0.010
soybean		89.880		89.240	0.640
splice		93.790		94.480	-0.690
tic-tac-toe		97.700		97.700	0.000
vehicle		83.340		80.740	2.600
vowel		88.790		78.080	10.710
wdbc		97.190		97.370	-0.180
wine		98.300		97.780	0.520
yeast		60.310		60.040	0.270
yeast-class-RPR		99.440		100.000	-0.560
zoo		92.000		97.000	-5.000
Medias:		83.340		83.600	-0.260

Tabla A.4: Comparación de porcentaje de acierto del mejor método en global y no realizar la selección de características

Apéndice B

Software desarrollado

En este Apéndice ofrecemos algunos datos acerca del software desarrollado durante la realización de este trabajo de investigación.

No se corresponden con métricas precisas definidas en algún modelo de desarrollo de software, si no más bien son datos simples cuyo único objetivo es proporcionar al lector una visión de las dimensiones del software creado y del esfuerzo de desarrollo.

B.1. Metodología de desarrollo

Para la ejecución eficaz de todo proceso de ingeniería es preciso aplicar una metodología adecuada. Sólo así se podrá garantizar la calidad del producto final, el uso efectivo de los recursos utilizados y la reducción de los costes de mantenimiento.

El desarrollo de software realizado durante la elaboración de la tesis doctoral, tiene unas peculiaridades que limita el tipo de metodología del software que podría usarse. Se precisa la obtención rápida de software funcional. En muchos casos se trata de software que no se ejecutará en muchas ocasiones, sólo unas pocas con el cometido de corroborar o refutar una hipótesis. Por ello es conveniente un ciclo de vida similar al basado en prototipos pero con algunas modificaciones.

Sin embargo, tras un diseño experimental adecuado, el número de experimentos a ejecutar puede ser elevado. En este caso, sí vale la pena hacer un esfuerzo extra

para intentar obtener código eficiente. Además, interesa un lenguaje que permita reutilizar gran parte del código ya disponible para diversas funciones: gestión de ficheros de datos, implementaciones fiables de clasificadores, evaluación, ...

Un lenguaje ideal para el prototipado es `python`. Permite el desarrollo rápido, imponiendo un estilo de codificación que produzca código claro, legible. Además, existe un nutrido conjunto de módulos para diversas tareas y permite enlazar fácilmente código desarrollado en `C++`. Por estas razones, ambos han sido los lenguajes de programación empleados.

B.2. Dimensión

Aunque no sean métricas precisas definidas por metodología de ingeniería del software, ofrecemos algunos datos estadísticos para que el lector se pueda hacer un esfuerzo de desarrollo realizado:

- Número de líneas de código fuente: 11000
- Número de clases: 96
- Número de módulos (ficheros de código fuente): 50
- Número de horas-programación y prueba: 4000

B.3. Herramientas

Para los trabajos realizados durante el desarrollo de este trabajo de investigación se ha utilizado Software Libre. Esto demuestra la madurez y oferta de este tipo de software para todas las tareas requeridas en nuestra investigación.

Los principales paquetes y grupos de software empleados han sido:

- Linux (<http://www.kernel.org>). Como sistema operativo gestionando todos los ordenadores empleados tanto equipos de sobremesa como portátiles así como el cluster de ordenadores sobre el que se ha ejecutado la carga

computacional del desarrollo experimental. Las dos distribuciones empleadas han sido Fedora Core 5 (<http://fedora.redhat.com>) y Ubuntu (<http://www.ubuntu-es.org>).

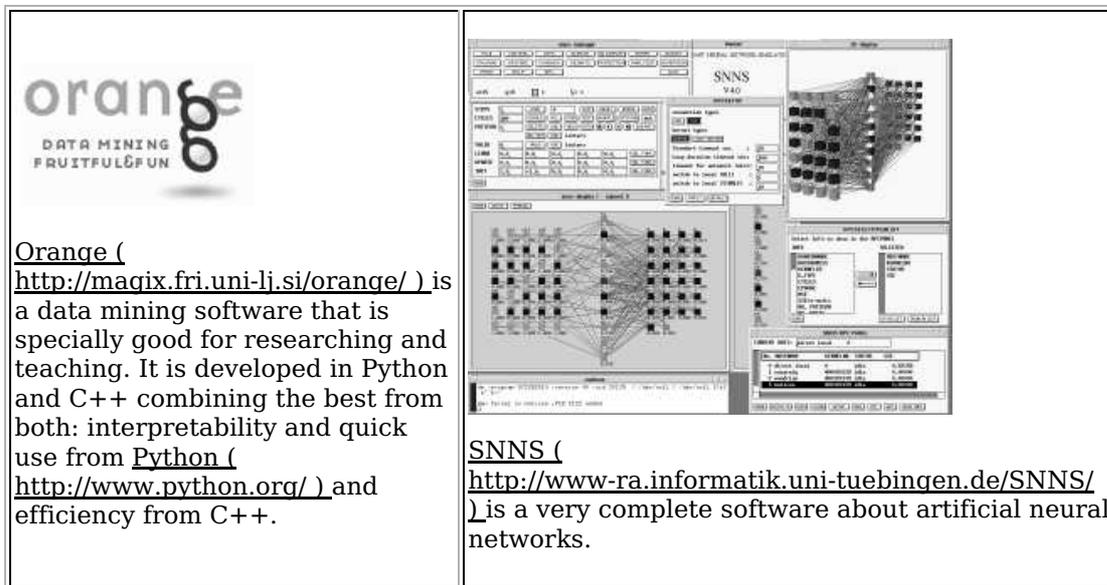
- OSCAR (<http://oscar.openclustergroup.org>) Software para la configuración y administración del cluster.
- Python (<http://www.python.org>) como lenguaje de programación fundamental. Empleado para interfaz de usuarios, guionado, prototipado rápido, ejecución y análisis de experimentos. Completado por los numerosos módulos disponibles: numpy, matplotlib, etc.
- GNU g++ compiler (<http://gcc.gnu.org>).
- R, (<http://www.r-project.org>) Lenguaje de procesamiento estadístico.
- Orange (<http://www.ailab.si/orange>) Plataforma de desarrollo y explotación de software para minería de datos. Incluye un interfaz gráfico de uso cómodo y una biblioteca extensa y eficaz para abordar múltiples tareas en el ámbito de la minería de datos.
- SNNS (<http://www-ra.informatik.uni-tuebingen.de/SNNS/>) Entorno software de desarrollo, simulación y explotación de Redes Neuronales Artificiales.

B.4. OrangeSNNS

Dado que el entorno Orange no incluía soporte para redes neuronales artificiales, fue necesario incorporarlo. Para ello, desarrollamos un modulo que permite el uso de las redes neuronales de SNNS dentro de Orange. Este módulo ha sido publicado como software libre, y nos consta que está siendo usado por otros investigadores. Está disponible en <http://ax5.com/antonio/orangesnns>. En las figuras B.1, B.2 y B.3, se expone la descripción publicada del módulo desarrollado.

Orange SNNS module

Interface module to use artificial neural networks from SNNS software as learning algorithms in Orange.



OrangeSNNS.py allows using SNNS to create, train and simulate neural networks as learners inside Orange. It is now in the process of development, but near its final state.

Download

There are two versions available.

- [OrangeSNNS.py \(orangeSNNS.py/file view\)](#) (version 1.09):
 - Supports feed-forward, fully-connected multi-layer perceptrons with [sigmoid](http://en.wikipedia.org/wiki/Sigmoid_function) (http://en.wikipedia.org/wiki/Sigmoid_function) activation function.
 - Is simpler than 0.99, as neural networks are evaluated in Python code.
 - Does not leak memory.
 - Diagram explaining [how it works \(diagramOrangeSNNS_110.png/image view\)](#).
- [OrangeSNNS.py \(orangeSNNS_099.py/file view\)](#) (version 0.99):
 - Is more easily generalizable to all types of neural networks supported by SNNS, as the network code is generated by SNNS.
 - Is supposed to be faster, as evaluation of network is .C code instead of Python.
 - Is an example of how to dynamically create, compile and use .C code from Python. Notice that it [leaks memory by importing and unimporting modules](http://mail.python.org/pipermail/python-list/2006-june/347047.html) (<http://mail.python.org/pipermail/python-list/2006-june/347047.html>). (I still don't know if it is a Python or Linux bug, or if I am not doing something necessary to free memory. Let me know if you have some news.) (Note: in the code, it is commented out a workaround that avoids memory leaking, but then, older networks may be used sometimes getting unreliable results)
 - You can see [its class and function definitions from help\(orangeSNNS\) \(doc\)](#).
 - Diagram explaining [how it works \(diagramOrangeSNNS/image view\)](#).

Use examples

- [exampleOrangeSNNS.py \(exampleOrangeSNNS.py\)](#) (uses [bupa.tab](#) (

Figura B.1: Página principal de presentación de OrangeSNNS (1/2)

<http://www.ailab.si/orange/doc/datasets/bupa.tab>) creates and trains a neural network with default parameters. The network is used to classify the training set showing the predicted class for each example.

```
import orange, orangeSNNS
data = orange.ExampleTable("bupa.tab")
learner = orangeSNNS.SNNSLearner()
classifier = learner(data)
for example in data:
    print example,
    print "->", classifier(example)
```

- [exampleOrangeSNNS2.py](#) ([exampleOrangeSNNS2.py](#)) (uses [bupa.tab](#) (<http://www.ailab.si/orange/doc/datasets/bupa.tab>) creates and trains a neural network:
 - With two hidden layers the first with 2 neurons and the second with 3
 - The training process will have 500 cycles on the training set.
 - If MSE=0 is achieved training stops.
 - The learning algorithm will be standard back propagation.
 - The learning parameter is 0.2

The network is then used to classify the training set showing the predicted class for each example.

```
import orange, orangeSNNS
# We set the path where SNNS binaries can be found, this
# is not necessary if they are in system path.
orangeSNNS.pathSNNS = "~/SNNSv4.2/tools/bin/i686-pc-linux-gnu/"
data = orange.ExampleTable("bupa.tab")
learner = orangeSNNS.SNNSLearner(name = 'SNNS neural network',
                                hiddenLayers = [2,3],
                                MSE = 0,
                                cycles = 500,
                                algorithm = "Std_Backpropagation",
                                learningParams = ["0.2"])
classifier = learner(data)
for example in data:
    print example,
    print "->", classifier(example)
```

Future plans

There will not be a 2.0 version, as this is just a quick solution. Instead, a completely integrated module with new code should be written, as SNNS is NOT free software and could not be adapted. More efforts on this module are worthless.

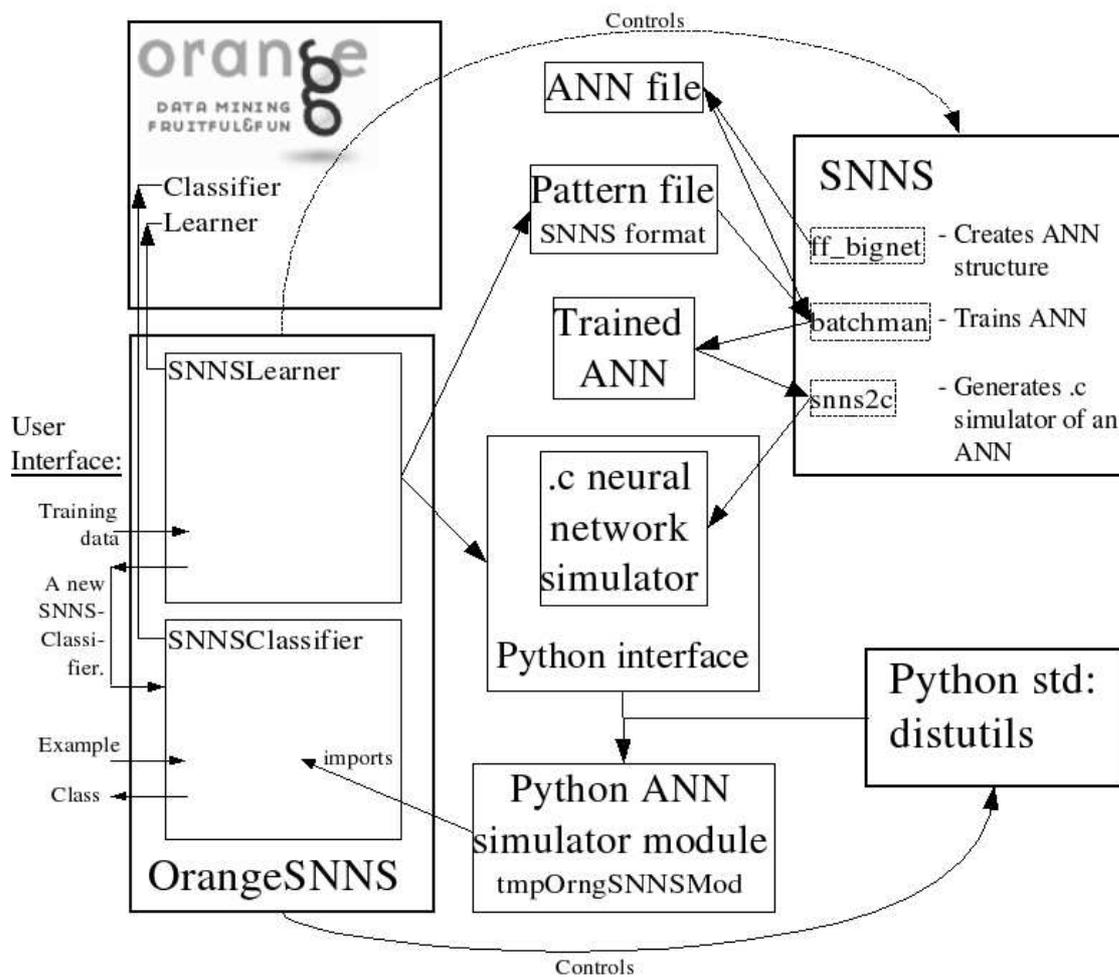
Probably a good choice to integrate neural networks in Orange is programming an interface to [FANN](http://leenissen.dk/fann/) (<http://leenissen.dk/fann/>).

There is a [summer of code 2006 project](http://wiki.python.org/moin/SummerOfCode) (<http://wiki.python.org/moin/SummerOfCode>) (Neural Nets in SciPy) that may be interesting having an eye on it.

Figura B.2: Página principal de presentación de OrangeSNNS (2/2)

Orange SNNS diagram

Diagram describing the internals of Orange SNNS.



Size : 110885

Figura B.3: Diagrama describiendo el funcionamiento de OrangeSNNS

Bibliografía

- [AD91] H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, volume 2, pages 547–552, San José, CA, 1991. AAAI Press.
- [AD94] H. Almuallim and T.G. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305, 1994.
- [AdLdC94] A. Martin Andres and J. de Luna del Castillo. *Bioestadística para las ciencias de la salud*. Ediciones Norma, Madrid (ES), 4^a edition, 1994.
- [BB97] G. Brassard and T. Bratley. *Fundamentos de Algoritmia*. Prentice Hall, 1997.
- [BD99] Jock A. Blackard and Denis J. Dean. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computer and electronics in agriculture*, 24(3):131–151, DiAvc 1999.
- [BHI⁺00] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, Alexander Kogan, Eddy Mayoraz, and Ilya Muchnik. An implementation of logical analysis of data. *IEEE Transactions on Knowledge Discovery and Data Engineering*, 12(2):292–306, 2000.
- [BK99] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.
- [BL97] Avrim L. Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271, 1997.

- [Boz02] Olcay Boz. Feature subset selection by using sorted feature relevance. In M. Arif Wani, Hamid R. Arabnia, Krzysztof J. Cios, Khalid Hafeez, and Graham Kendall, editors, *ICMLA*, pages 147–153. CSREA Press, 2002.
- [Bre98] Leo Breiman, editor. *Classification and regression trees*. Chapman & Hall, 1998.
- [Bun90] W.L. Buntine. Myths and legend in learning classification rules. In *Proceedings AAAI-90*, 1990.
- [Car93] Claire Cardie. Using decision trees to improve case-based learning. In Morgan Kaufmann, editor, *Proceedings of the Tenth International Conference on Machine Learning*, pages 25–32, 1993.
- [Cer85] V. Cerny. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985.
- [CGB96] Michal R. Chmielewski and Jerzy W. Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. *International Journal of Approximate Reasoning*, 15(4):319–331, November 1996.
- [CHHM01] O. Cordón, F. Herrera, F. Hoffmann, and L. Magdalena. *Genetic Fuzzy Systems. Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. World Scientific, 2001.
- [CQ99] Nidapan Chaikla and Yulu Qi. Genetic algorithms in feature selection. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 5, 1999.
- [CSCZ04] J.J. Castro-Schez, J. Castro, and J.M. Zurita. Fuzzy repertory table, a method for acquiring knowledge about input variables to machine learning algorithms. 2004.
- [CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information theory*. Wiley-Interscience, 1991.
- [Das97] Manoranjan Dash. Feature selection via set cover. In *IEEE Knowledge and Data Engineering Exchange Workshop*, 1997.
- [Dav91] Y. Davidor. *Genetic Algorithms and Robotics: a Heuristic Strategy for Optimization*. World Scientific, 1991.

- [Dem06] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [DL97] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent Data Analysis*, 1(1-4):131–156, 1997.
- [DL03] Manoranjan Dash and Huan Liu. Consistency-based search in feature selection. *Artificial Intelligence*, 151(1-2):155–176, 2003.
- [DR94] S. Davies and S. Russell. Np-completeness of searches for smallest possible feature sets. In AAAI Press, editor, *Proceedings of the 1994 AAAI Fall Symposium on Relevance*, pages 37–39, 1994.
- [DRS99] Justin C.W. Debusse and Victor J. Rayward-Smith. Feature subset selection within a simulated annealing data mining algorithm. *Journal of Intelligent Information Systems*, 9(1):57–81, 1999.
- [DSG93] K. A. DeJong, W. M. Spears, and D. F. Gordon. Using Genetic Algorithms for Concept Learning. *Machine Learning*, 13:166–188, 1993.
- [DZ04] J. Demsar and B. Zupan. Orange: From experimental machine learning to interactive data mining. (White paper) <http://www.ailab.si/orange>, 2004.
- [DZL06] J. Demsar, B. Zupan, and G. Leban. Datasets. <http://www.ailab.si/orange/datasets.asp>, 2006.
- [Gin06] Gini coefficient. http://en.wikipedia.org/wiki/Gini_index, Febrero 2006.
- [GK03] F. Glover and G. Kochenberger, editors. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.
- [Glo89] F. Glover. Tabu-Search Part I. *ORSA Journal on Computing*, 1(3):190–206, 1989.
- [Gol89] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, New York, 1989.
- [Gra06] Silicon Graphics. Datasets. <http://www.sgi.com/tech/mlc/db/>, 2006.
- [Hay99] Simon Haykin. *Neural Networks. A comprehensive foundation*. Prentice-Hall, 1999.

- [HB99] S. Hettich and S. D. Bay. The uci kdd archive. <http://kdd.ics.uci.edu/>, 1999.
- [Hol75] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.
- [HP02] Y. C. Ho and D. L. Pepyne. Simple explanation of the no-free-lunch theorem and its implications. *Journal of Optimization Theory and Applications*, 115(3):549–570, 2002.
- [HS78] Ellis Horowitz and Sartaj Sahni. *Fundamentals of Computer Algorithms*. Computer Science Press, 1978.
- [HSW89] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.
- [HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of statistical learning : data mining, inference, and prediction*. Springer-Verlag, New York, 2001.
- [Ign91] J.P. Ignizio. *An Introduction to Expert Systems*. McGraw-Hill, 1991. KA: chap 5.
- [Jan93] C. Z. Janikow. A Knowledge-Intensive Genetic Algorithm for Supervised Learning. *Machine Learning*, 13:189–228, 1993.
- [JKP94] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Proc. of the Eleventh International Conference on Machine Learning*, pages 121–129, New Brunswick, NJ, 1994. Morgan Kaufmann.
- [JZ97] Anil Jain and Douglas Zongker. Feature selection: Evaluation, application and small sample performance. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(2):153–158, Febrero 1997.
- [KJ97] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
- [KJV83] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [Koh94] Ron Kohavi. Feature subset selection as search with probabilistic estimates. In *AAAI Fall Symposium on Relevance*, pages 122–126, Noviembre 1994.

- [Kon94] Igor Kononenko. Estimating attributes: Analysis and extensions of RELIEF. In *European Conference on Machine Learning*, pages 171–182, 1994.
- [KPPS98] Jan Komorowski, Zdzislaw Pawlak, Lech Polkowski, and Andrezej Skowron. Rough sets: a tutorial, 1998.
- [KR92] Kenji Kira and Larry A. Rendell. A practical approach to feature selection. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 249–256, Aberdeen, Scotland, 1992. Morgan Kaufmann.
- [KS96] Daphne Koller and Mehran Sahami. Toward optimal feature selection. In *International Conference on Machine Learning*, pages 284–292, 1996.
- [KY95] George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic*. Prentice Hall, Englewood Cliffs, NJ 07632, 1995.
- [LA04] George V. Lashkia and Laurence Anthony. Relevant, irredundant feature selection and noisy example elimination. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 34(2):888–898, April 2004.
- [Lan94] Pat Langley. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 1–5, New Orleans, LA, USA, 1994. AAAI Press.
- [LHL05] S. Liu, H. Hou, and X. Li. Feature selection method based on genetic and simulated annealing algorithm. *Jisuanji Gongcheng Computer Engineering*, 31(16):157–159, 2005.
- [LHTD02] Huan Liu, Farhad Hussain, Chew Lim Tan, and Manoranjan Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 6:393–423, 2002.
- [LKM01] Mark Last, Abraham Kandel, and Oded Maimon. Information-theoretic algorithm for feature selection. *Pattern Recognition Letters*, 22(6/7):799–811, 2001.
- [LM98] Huan Liu and Hiroshi Motoda. Feature transformation and subset selection. *IEEE Intelligent Systems*, pages 26–28, April 1998.
- [LMD98] Huan Liu, Hiroshi Motoda, and Manoranjan Dash. A monotonic measure for optimal feature selection. In *European Conference on Machine Learning*, pages 101–106, 1998.

- [LS96a] Huan Liu and Rudy Setiono. Feature selection and classification - a probabilistic wrapper approach. 1996.
- [LS96b] Huan Liu and Rudy Setiono. A probabilistic approach to feature selection - a filter solution. In *International Conference on Machine Learning*, pages 319–327, 1996.
- [LTPV02] F.C.G. Lopez, M.G. Torres, J.A.M. Perez, and J.M.M. Vega. Feature selection using tabu search method. *Pattern Recognition*, 35:701–711, 2002.
- [LY05] Huan Liu and Lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(3), Marzo 2005.
- [MC04] A. Mittal and L.F. Cheong. Addressing the problems of bayesian network classification of video using high dimensional features. *IEEE Transactions on Knowledge and Data Engineering*, 16(2):230–244, 2004.
- [Mic96] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1996.
- [Mod93] Maciej Modrzejewski. Feature selection using rough sets theory. In *Proceedings of the European Conference on Machine Learning*, pages 213–216, 1993.
- [MZ06] Ronen Meiri and Jacob Zahavi. Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, 171(3):842–858, Junio 2006.
- [NF77] Patrenahalli M. Narendra and Keinosuke Fukunaga. A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers*, 26(9):917–922, Septiembre 1977.
- [Nis93] V. Nissen. Evolutionary Algorithms in Management Science: an overview and list of references. In *Papers on Economics and Evolution*, 1993.
- [OK96] I. H. Osman and J. P. Kelly, editors. *Meta-Heuristics: Theory & Applications*. Kluwer Academic Publishers, 1996.
- [oS95] University of Stuttgart. Stuttgart neural network simulator. <http://www-ra.informatik.uni-tuebingen.de/SNNS/>, 1995.

- [OSV92] A.L. Oliveira and A. Sangiovanni-Vicentelli. Constructive induction using a non-greedy strategy for feature selection. In *Proceedings of Ninth International Conference on Machine Learning*, pages 355–360, Aberdeen, Scotland, 1992. Morgan Kaufmann.
- [Pap94] Christos M. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [Paw91] Zdzislaw Pawlak. *Rough Sets, Theoretical aspects of reasoning about data*. Kluwer Academic Publishers, 1991.
- [Ped97] W. Pedrycz. *Fuzzy Evolutionary Computation*. Kluwer Academic, 1997.
- [PFTV98] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical recipes in C*. Cambridge University Press, 1998.
- [PN98] P. Pudil and J. Novovicova. Novel methods for subset selection with respect to problem knowledge. *IEEE Intelligent Systems & Their Applications*, 13(2):66–74, 1998.
- [PS98] L. Polkowski and A. Skowron, editors. *Rough Sets in Knowledge Discovery*. Heidelberg: Physica Verlag, 1998.
- [Qui86] John R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.
- [Qui93] John R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [Ris01] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 2001.
- [RS97] Marko Robnik-Sikonja. Core - a system that predicts continuous variables. In *Proceedings of ERK'97*, Portoroz, Slovenia, 1997.
- [RS98] Marko Robnik-Sikonja. Speeding up relief algorithms with k-d trees. In *Proceedings of the Electrotechnical and Computer Science Conference ERK'98*, Portoroz, Slovenia, 1998.
- [RSK97] Marko Robnik-Sikonja and Igor Kononenko. An adaptation of relief for attribute estimation in regression. In Morgan Kaufmann, editor, *Machine Learning: Proceedings of the Fourteenth International Conference*, pages 296–304, 1997.

- [RSK99] Marko Robnik-Sikonja and Igor Kononenko. Attribute dependencies, understandability and split selection in tree based models. In *Machine Learning, Proceedings of 16th International Conference on Machine Learning ICML'99*, Bled, Slovenia, 1999.
- [RSK03] Marko Robnik-Sikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53:23–69, 2003.
- [SABAA03] B. Samanta, K.R. Al-Balushi, and S.A. Al-Araimi. Artificial neural networks and support vector machines with genetic algorithm for bearing fault detection. *Engineering Applications of Artificial Intelligence*, 16(7-8):657–665, 2003.
- [Sch93] J.C. Schlimmer. Efficiently inducing determinations: A complete and efficient search algorithm that uses optimal pruning. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 284–290, New Brunswick, NJ, 1993. Morgan Kaufmann.
- [SDN90] J. Sheinvald, B. Dom, and W.Ñiblack. A modelling approach to feature selection. In *10th International Conference on Pattern Recognition*, volume i, pages 535–539, 1990.
- [SK93] Avi Silberschatz and Hank Korth. *Database systems concepts*. McGraw-Hill Inc, 2nd edition, 1993.
- [SK96] R. Sikonja and M. Kononenko. Non-myopic attribute estimation in regression, 1996.
- [SL95] Rudy Setiono and Huan Liu. Chi2: Feature selection and discretization of numeric attributes. In *Proceedings of 7th IEEE Int. Conference on Tools with Artificial Intelligence*, 1995.
- [SP04] Pert Somol and Pavel Pudil. Fast branch & bound algorithms for optimal feature selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):900–912, Julio 2004.
- [SP05] Riyaz Sikora and Selwyn Piramuthu. Efficient genetic algorithm based data mining using feature selection with hausdorff distance. *Information Technology and Management*, 6(4):315–331, Octubre 2005.
- [SPLC04] P. Scanlon, G. Potamianos, V. Libal, and S. M. Chu. Mutual information based visual feature selection for lipreading. In *Int. Conf. on Spoken Language Processing*, South Korea, 2004.

- [SWE92] J. D. Schaffer, D. Whitley, and L. J. Eshelman. Combinations of Genetic Algorithms and Neural Networks: A Survey of the State of the Art. *COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks*, 1992.
- [Sán00] José Jesús Castro Sánchez. *Diseño e implementación de una herramienta para la adquisición automática de conocimiento*. PhD thesis, E.T.S. de Ingeniería Informática, Universidad de Granada, 2000.
- [TMNRDCF⁺02] Jose Maria de la Torre-Martinez, M^a Angustias Navarro-Ruiz, Miguel Delgado-Calvo-Flores, Juan Luis Castro-Peña, and Antonio Arauzo Azofra. Aplicación de los modelos separate y regresión logística para determinar la relevancia de la información contable en el mercado de capitales: una comparación. In *X ENCUENTRO DE PROFESORES UNIVERSITARIOS DE CONTABILIDAD*, 2002.
- [Tor03] University of Toronto. Data for evaluating learning in valid experiments. <http://www.cs.utoronto.ca/delve/>, 2003.
- [TS02] Francis E. H. Tay and Lixiang Shen. A modified chi2 algorithm for discretization. *IEEE Trans. on Knowledge and Data Engineering*, 14(3):666–670, 2002.
- [VMOR99] S. Voss, S. Martello, I.H. Osman, and C. Roucairol, editors. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. Kluwer Academic Publishers, 1999.
- [WAM97] Dietrich Wettschereck, David W. Aha, and Takao Mohri. A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms. *Artificial Intelligence Review*, 11(1-5):273–314, 1997.
- [WM92] Li-Xin Wang and Jerry M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Trans. Systems, Man and Cybernetics*, 22(6):1414–1427, November/December 1992.
- [Wol90] D. Wolpert. A mathematical theory of generalization: parts i and ii. *Complex Systems*, 1990.
- [Xu88] Lei Xu. Some applications of simulated annealing to pattern recognition. In *Proceedings - International Conference on Pattern Recognition*, pages 1040–1042, 1988.

- [YL03] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC, 2003.
- [Zad65] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 3(8):338–353, 1965.
- [Zad75a] Lotfi A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. Part I. *Information Sciences*, 8:199–249, 1975.
- [Zad75b] Lotfi A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. Part II. *Information Sciences*, 8:301–357, 1975.
- [Zad75c] Lotfi A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. Part III. *Information Sciences*, 9:43–80, 1975.
- [Zad75d] Lotfi A. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30:407–428, 1975.
- [Zar99] Jerrold H. Zar. *Biostatistical Analysis*. Prentice-Hall, New Jersey (US), 4th edition, 1999.
- [ZDO01] Ning Zhong, Juzhen Dong, and Setsuo Ohsuga. Using rough sets with heuristics for feature selection. *Journal of Intelligent Information Systems*, 16(3):199–214, 2001.
- [ZG04] Fangming Zhu and Steven Guan. Feature selection for modular ga-based classification. *Applied Soft Computing*, 4(4):381–393, Septiembre 2004.
- [ÖUT06] Özge Uncu and I.B. Türksen. A novel feature selection approach: Combining feature wrappers and filters. *Information Sciences (IN PRESS)*, 2006.



UNIVERSIDAD DE GRANADA