

Universidad de Granada



Departamento de Ciencias de la Computación e Inteligencia Artificial

Tesis Doctoral
Selección de ejemplos para clasificación: un enfoque basado
en la caracterización de los datos de entrada

Enrique José Leyva Miranda

GRANADA, DICIEMBRE DE 2011

Editor: Editorial de la Universidad de Granada
Autor: Enrique José Leyva Miranda
D.L.: GR 1180-2012
ISBN: 978-84-695-1193-0

Universidad de Granada



Selección de ejemplos para clasificación: un enfoque basado en la caracterización de los datos de entrada

Memoria que presenta
Enrique José Leyva Miranda
para optar al grado de Doctor en Informática

Directores:
Antonio González Muñoz
Francisco Gabriel Raúl Pérez Rodríguez

Departamento de Ciencias de la Computación
e Inteligencia Artificial

GRANADA, DICIEMBRE DE 2011

La memoria titulada “Selección de ejemplos para clasificación: un enfoque basado en la caracterización de los datos de entrada”, que presenta Don Enrique José Leyva Miranda para optar al grado de Doctor, ha sido realizada en el Departamento de Ciencias de la Computación e Inteligencia Artificial de la Universidad de Granada bajo la dirección de los Doctores Don Antonio González Muñoz y Don Francisco Gabriel Raúl Pérez Rodríguez.

Granada, Noviembre de 2011.

El Doctorando

Los Directores

Enrique José Leyva miranda

Antonio González Muñoz

Francisco Gabriel Raúl Pérez Rodríguez

Dedicatoria

A mi familia, en especial a mi esposa e hijos, a los que he tenido que imponerles tantos meses de separación.

A los directores de mi tesis, quienes a pesar de tantas cosas en contra, siempre me apoyaron y confiaron en mí.

Agradecimientos

Muchas personas han influido en que este proyecto llegara a feliz término. En primer lugar tengo mucho que agradecer a mi esposa Yanet, pues sin su apoyo y comprensión no hubiera podido dedicar tanto tiempo a mi doctorado. Tiempo que necesariamente he tenido que quitarle a ella y a mis niños. Asimismo agradezco a otros miembros de mi familia, en especial a mi madre y mi abuela, y de mi familia política a mi suegro y mi cuñado; por las tantas ocasiones en que me sustituyeron durante mis largas ausencias ayudando a mi esposa durante esos dos embarazos que prácticamente ha pasado sin mí, y luego en el cuidado y la educación de mis hijos.

Agradezco a Raúl y Antonio, quienes mas que directores de tesis han sido amigos que me han dado todo el apoyo que he necesitado. A los profesores Verdegay, Rosa, Mariño y todos los que en Granada u Holguín han trabajado muy duro para que este programa de doctorado saliera adelante. A mis amigos cubanos con los que he coincidido en Granada y han hecho más llevadera la distancia de mi tierra, en especial a Yoel.

A todos ellos y a los que no menciono porque intentando hacer exhaustiva la lista sólo conseguiría ser injusto con los que involuntariamente omitiese, muchas gracias.

Índice general

1. Introducción	1
1.1. Objetivos y Aportaciones	3
1.2. Estructura de la Memoria	4
2. Selección de Ejemplos para Clasificación	5
2.1. El Problema de la Selección de Ejemplos	5
2.2. Enfoques Fundamentales Para la Selección de Ejemplos	6
2.2.1. Conservación de las Fronteras	7
2.2.1.1. CNN	8
2.2.1.2. MCS	8
2.2.1.3. FCNN	9
2.2.1.4. EOP y POP	11
2.2.2. Filtros de Ruido	14
2.2.2.1. ENN	14
2.2.2.2. RENN y All-kNN	14
2.2.2.3. HPFF	15
2.2.3. Condensación con Filtrado Previo del Ruido	15
2.2.3.1. DROP3	16
2.2.3.2. ICF	17
2.2.3.3. Cpruner	18
2.2.3.4. CCIS	20
2.2.4. Condensación con Filtrado de Ruido Embebido	21
2.2.4.1. IB3	22

2.2.4.2.	NRMCS	23
2.2.5.	Generación de Prototipos	25
2.2.5.1.	PNN	25
2.2.5.2.	HYB	26
2.2.5.3.	ISPA	28
2.2.6.	Enfoques Meta-heurísticos	29
2.2.6.1.	Explore	29
2.2.6.2.	Búsqueda Tabú	30
2.2.6.3.	Algoritmos genéticos	31
2.2.6.4.	SSMA	32
2.2.6.5.	CoCoIS	32
2.2.6.6.	Evolución Diferencial	33
2.2.7.	Enfoques Basados en Agrupamiento	34
2.2.7.1.	NSC	34
2.2.7.2.	SGP	35
2.2.7.3.	PSC	36
2.3.	El problema de la escalabilidad. Enfoques fundamentales	37
2.3.1.	Particionamiento recursivo	37
2.3.2.	Estratificación	38
2.3.3.	Selección democrática	38
2.3.4.	Consideraciones generales acerca de las propuestas para escalar métodos de selección de ejemplos	39
2.4.	Conclusiones Parciales	39
3.	Una Propuesta de Atributos para la Caracterización de Bases de Datos	41
3.1.	Meta-aprendizaje y medidas de complejidad de los datos	41
3.1.1.	Medidas del solapamiento de los valores de los atributos entre clases diferentes	42
3.1.2.	Medidas de la separabilidad de las clases	43
3.1.3.	Medidas de geometría y densidad de regiones	44
3.1.4.	Medidas estadísticas	46
3.2.	Nuevo sistema de medidas de complejidad	46

3.2.1.	Medidas basadas en conjuntos locales	47
3.2.1.1.	Cardinalidad promedio de los conjuntos locales	47
3.2.1.2.	Número de clusters	50
3.2.1.3.	Puntos invasores	53
3.2.2.	Medidas estadísticas	54
3.2.2.1.	Proporción de atributos continuos	54
3.2.2.2.	Número de clases	55
3.2.2.3.	Desbalance	55
3.2.3.	Ruido	56
3.3.	Optimización de las medidas mediante métodos de búsqueda rápida de vecinos más cercanos	58
3.4.	Conclusiones Parciales	66
4.	Aprendizaje de Meta-Modelos para Seleccionar el Método Adecuado	67
4.1.	Arquitectura de un sistema de selección de ejemplos basado en conocimiento	69
4.2.	Caracterización de datos y obtención de la base de datos ISP	71
4.3.	Meta-aprendizaje	75
4.4.	Resultados	78
4.4.1.	Diseño experimental	79
4.4.2.	Desempeño del sistema ante el clasificador KNN	80
4.4.2.1.	Reducción y precisión ante distintos valores del parámetro b	80
4.4.2.2.	Comparación con los métodos individuales empleados	81
4.4.3.	Desempeño del sistema ante otros clasificadores	87
4.5.	Conclusiones Parciales	90
5.	Reutilizando en la Selección las Estructuras de Datos Creadas en la Etapa de Caracterización	91
5.1.	Nuevos métodos de selección de ejemplos basados en conjuntos locales y clusters	91
5.1.1.	Selección de núcleos	92
5.1.2.	Selección de puntos fronterizos	93
5.1.3.	Estudio del costo computacional	96
5.2.	Meta-aprendizaje. Aprendiendo meta-modelos específicos para cada clasificador	99

5.3.	Resultados	105
5.3.1.	Reducción y precisión ante distintos valores del parámetro b	105
5.3.2.	Comparación con los métodos individuales empleados	107
5.3.3.	Comparación con métodos convencionales	110
5.3.4.	Comparación con métodos basados en meta-heurísticas	116
5.4.	Conclusiones Parciales	125
6.	Conclusiones y Trabajos Futuros	126
A.	Detalles de los experimentos del capítulo 4	128
A.1.	Desempeño del sistema ante KNN	128
A.2.	Desempeño de los métodos individuales ante KNN	133
A.3.	Comparación del desempeño del sistema (con KNN) con los de los métodos individuales que emplea. Detalles de los test de Friedman y Holm	136
A.4.	Desempeño del sistema ante C4.5, LDA y PART	148
B.	Detalles de los experimentos del capítulo 5	178
B.1.	Desempeño del sistema	178
B.2.	Desempeño de los métodos individuales	195
B.3.	Desempeño de los métodos convencionales	201
B.4.	Desempeño de los métodos basados en meta-heurísticas	207
B.5.	Comparación del desempeño del sistema con los de los métodos convencionales. Detalles de los test de Friedman y Holm	214
B.6.	Comparación del desempeño del sistema con los de los algoritmos genéticos. Detalles de los test de Friedman y Holm	230

Índice de figuras

2.1. Algoritmo CNN.	9
2.2. Algoritmo MCS.	10
2.3. Algoritmo FCNN.	11
2.4. Ejemplo de los resultados de aplicar POP en un espacio bidimensional.	12
2.5. Algoritmo POP.	13
2.6. Algoritmo ENN.	14
2.7. Algoritmo DROP3.	17
2.8. Algoritmo ICF.	18
2.9. Algoritmo Cpruner.	19
2.10. Algoritmo CC.	21
2.11. Algoritmo THIN.	22
2.12. Algoritmo IB3.	23
2.13. Algoritmo NRMCS.	24
2.14. Algoritmo PNN.	27
2.15. Algoritmo PSC.	36
2.16. Algoritmo ARP.	37
3.1. Ejemplo de un conjunto local en un espacio bidimensional.	48
3.2. Ejemplo de cómo una frontera angosta provoca LSC bajos.	48
3.3. Ejemplo de cómo una frontera extensa provoca que el promedio de LSC sea bajo.	49
3.4. Dos problemas con tipos diferentes de fronteras.	50
3.5. Dos problemas con diferentes estructuras de clases.	51
3.6. Algoritmo de agrupamiento supervisado basado en conjuntos locales.	52

3.7.	Ejemplo del uso de conjuntos locales para detectar clusters.	52
3.8.	Ejemplo de un punto invasor.	54
3.9.	Efecto del desbalance sobre estrategias de selección basadas en fronteras y centroides.	57
3.10.	Procedimiento para la búsqueda de los k vecinos más cercanos a un ejemplo usando un árbol- vp	61
3.11.	Procedimiento para la búsqueda del conjunto local asociado a un ejemplo usando un árbol- vp	62
4.1.	Modelo que se propone para el uso de meta-aprendizaje en la selección de ejemplos.	68
4.2.	Arquitectura de un sistema de selección de ejemplos basado en conocimiento. . . .	70
4.3.	Esquema del proceso de construcción de la base de datos ISP.	71
4.4.	Desempeño de varios algoritmos de aprendizaje sobre la base de datos ISP empleando validación cruzada con 10 particiones. Promedio general por algoritmo de la raíz del error cuadrático medio en los conjuntos de entrenamiento y prueba.	76
4.5.	Reducción de ambas versiones del sistema a distintos valores del parámetro b	80
4.6.	Precisión de ambas versiones del sistema con KNN a distintos valores del parámetro b .	81
4.7.	Niveles de reducción alcanzados por ambas versiones del sistema a distintos valores del parámetro b . Comparación con las reducciones de los métodos individuales que estas emplean.	82
4.8.	Niveles de precisión alcanzados por KNN empleando los resultados de ambas versiones del sistema a distintos valores del parámetro b . Comparación con las precisiones correspondientes a los métodos individuales que estas emplean.	82
4.9.	Niveles de balance alcanzados por ambas versiones del sistema (con KNN como clasificador) a distintos valores del parámetro b . Comparación con los balances correspondientes a los métodos individuales que estas emplean.	83
4.10.	Niveles de precisión alcanzados por varios clasificadores empleando los resultados de ambas versiones del sistema con $b = 0,1$. Comparación con las precisiones correspondientes a los métodos individuales que estas emplean.	88
4.11.	Niveles de precisión alcanzados por varios clasificadores empleando los resultados de ambas versiones del sistema con $b = 0,2$. Comparación con las precisiones correspondientes a los métodos individuales que estas emplean.	88
4.12.	Niveles de balance alcanzados por ambas versiones del sistema (con varios clasificadores) para $b = 0,1$. Comparación con los balances correspondientes a los métodos individuales que estas emplean.	89

4.13. Niveles de balance alcanzados por ambas versiones del sistema (con varios clasificadores) para $b = 0,2$. Comparación con los balances correspondientes a los métodos individuales que estas emplean.	89
5.1. Resultados de someter la base de datos Banana a ENN y distintas estrategias de selección de núcleos.	94
5.2. Procedimiento para la selección de ejemplos fronterizos.	95
5.3. Base de datos Banana procesada por ENN y por el método de selección de puntos fronterizos.	95
5.4. Promedio de los tiempos de ejecución (en segundos) sobre 40 bases de datos de tres de los métodos que reutilizan las estructuras de datos creadas durante el cómputo de las medidas y de los cuatro métodos empleados en el capítulo 4 que no las aprovechan.	98
5.5. Tiempos de ejecución (en segundos) sobre 40 bases de datos de tres de los métodos que reutilizan las estructuras de datos creadas durante el cómputo de las medidas y de los cuatro métodos empleados en el capítulo 4 que no las aprovechan.	98
5.6. Arquitectura de un sistema KBIS con selección del clasificador.	100
5.7. Desempeño de varios algoritmos de aprendizaje sobre la base de datos ISP2 empleando validación cruzada con 10 particiones. Promedio general por algoritmo de la raíz del error cuadrático medio en los conjuntos de entrenamiento y prueba.	102
5.8. Error cuadrático medio ajustado de varios algoritmos de aprendizaje sobre las bases de datos ISP e ISP2 (restringida a KNN) empleando validación cruzada con 10 particiones.	104
5.9. Reducción de ambas versiones del sistema a distintos valores del parámetro b y varios clasificadores.	106
5.10. Precisión de ambas versiones del sistema a distintos valores del parámetro b y varios clasificadores.	106
5.11. Reducción de ambas versiones del sistema y los métodos individuales empleados por estas a distintos valores del parámetro b y varios clasificadores.	107
5.12. Precisión de ambas versiones del sistema y los métodos individuales empleados por estas a distintos valores del parámetro b y varios clasificadores.	108
5.13. Balance de ambas versiones del sistema y los métodos individuales empleados por estas a distintos valores del parámetro b y varios clasificadores.	109
5.14. Balance de ambas versiones del sistema y varios métodos convencionales a distintos valores del parámetro b y varios clasificadores.	110
5.15. Balance de ambas versiones del sistema y varios métodos convencionales con varios clasificadores para los valores del parámetro b comprendidos entre 0 y 0,5.	112

5.16. Balance de ambas versiones del sistema y varios métodos convencionales a distintos valores del parámetro b y varios clasificadores.	113
5.17. Balance de ambas versiones del sistema y varios métodos basados en meta-heurísticas a distintos valores del parámetro b y varios clasificadores.	118
5.18. Balance de ambas versiones del sistema y varios métodos basados en meta-heurísticas con varios clasificadores para los valores del parámetro b comprendidos entre 0 y 0,5.	119
5.19. Balance de ambas versiones del sistema y varios métodos basados en meta-heurísticas a distintos valores del parámetro b y varios clasificadores.	120
5.20. Tiempos de ejecución promedio de ambas versiones del sistema y varios métodos basados en meta-heurísticas a distintos valores del parámetro b y varios clasificadores.	124

Índice de tablas

3.1. Tiempos de ejecución (en segundos) del cómputo de las medidas. Variantes exhaustiva y basada en árbol-vp.	64
3.2. Tiempos de ejecución (en segundos) del cómputo de las medidas propuestas en este trabajo y las propuestas por Ho y Basu.	65
4.1. Bases de datos empleadas para la construcción de la base de datos ISP. Las columnas Atrib, Cont, Ent y Nom, muestran las cantidades de atributos, y de atributos continuos, enteros y nominales respectivamente.	73
4.2. Métodos individuales de selección de ejemplos incluidos en el estudio.	74
4.3. Desempeño de varios algoritmos de aprendizaje sobre la base de datos ISP empleando validación cruzada con 10 particiones. Raíz del error cuadrático medio en los conjuntos de entrenamiento y prueba.	77
4.4. Desempeño de M5 y Mogul_TSK sobre la base de datos ISP. Raíz del error cuadrático medio en el conjunto de entrenamiento.	78
4.5. Ejemplo de tabla resumen de los tests de Holm de comparaciones múltiples entre varios métodos.	79
4.6. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en reducción con $b = 0,1$. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.	84
4.7. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en reducción con $b = 0,2$. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.	84
4.8. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,1$ y el clasificador <i>KNN</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.	85
4.9. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,2$ y el clasificador <i>KNN</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.	85

4.10. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,1$ y el clasificador <i>KNN</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	86
4.11. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,2$ y el clasificador <i>KNN</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	86
5.1. Desempeño de varios algoritmos de aprendizaje sobre la base de datos ISP2 empleando validación cruzada con 10 particiones. Raíz del error cuadrático medio en los conjuntos de entrenamiento y prueba.	101
5.2. Desempeño de varios algoritmos de aprendizaje sobre las bases de datos ISP e ISP2 (restringida a <i>KNN</i>) empleando validación cruzada con 10 particiones.	103
5.3. Desempeño de <i>M5</i> y <i>Mogul_TSK</i> sobre la base de datos ISP2. Raíz del error cuadrático medio en el conjunto de entrenamiento.	104
5.4. Test de Holm. Resumen por clasificador.	114
5.5. Test de Holm. Resumen por valor de b	115
5.6. Test de Holm. Resumen general.	116
5.7. Test de Holm. Resumen por clasificador.	121
5.8. Test de Holm. Resumen por valor de b	122
5.9. Test de Holm. Resumen general.	122
A.1. Reducciones logradas por la versión crisp del sistema a distintos valores del parámetro b	129
A.2. Reducciones logradas por la versión difusa del sistema a distintos valores del parámetro b	130
A.3. Precisiones logradas (con <i>KNN</i>) por la versión crisp del sistema a distintos valores del parámetro b	131
A.4. Precisiones logradas (con <i>KNN</i>) por la versión difusa del sistema a distintos valores del parámetro b	132
A.5. Reducciones logradas por los métodos individuales que el sistema emplea.	134
A.6. Precisiones logradas (con <i>KNN</i>) por los métodos individuales que el sistema emplea.	135
A.7. Comparación de las reducciones obtenidas por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	136
A.8. Comparación de las reducciones obtenidas por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	137

A.9. Comparación de las reducciones obtenidas por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	138
A.10. Comparación de las reducciones obtenidas por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	139
A.11. Comparación de las precisiones obtenidas (con KNN) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	140
A.12. Comparación de las precisiones obtenidas (con KNN) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	141
A.13. Comparación de las precisiones obtenidas (con KNN) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	142
A.14. Comparación de las precisiones obtenidas (con KNN) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	143
A.15. Comparación de los balances obtenidos (con KNN) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	144
A.16. Comparación de los balances obtenidos (con KNN) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	145
A.17. Comparación de los balances obtenidos (con KNN) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	146
A.18. Comparación de los balances obtenidos (con KNN) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	147
A.19. Comparación de las precisiones obtenidas (con C4.5) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	148
A.20. Comparación de las precisiones obtenidas (con C4.5) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	149
A.21. Comparación de las precisiones obtenidas (con C4.5) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	150
A.22. Comparación de las precisiones obtenidas (con C4.5) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	151
A.23. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,1$ y el clasificador C4,5. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	152
A.24. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,2$ y el clasificador C4,5. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	152
A.25. Comparación de los balances obtenidos (con C4.5) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	153

A.26. Comparación de los balances obtenidos (con C4.5) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	154
A.27. Comparación de los balances obtenidos (con C4.5) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	155
A.28. Comparación de los balances obtenidos (con C4.5) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	156
A.29. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,1$ y el clasificador C4,5. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	157
A.30. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,2$ y el clasificador C4,5. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	157
A.31. Comparación de las precisiones obtenidas (con LDA) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	158
A.32. Comparación de las precisiones obtenidas (con LDA) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	159
A.33. Comparación de las precisiones obtenidas (con LDA) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	160
A.34. Comparación de las precisiones obtenidas (con LDA) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	161
A.35. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,1$ y el clasificador LDA. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	162
A.36. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,2$ y el clasificador LDA. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	162
A.37. Comparación de los balances obtenidos (con LDA) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	163
A.38. Comparación de los balances obtenidos (con LDA) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	164
A.39. Comparación de los balances obtenidos (con LDA) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	165
A.40. Comparación de los balances obtenidos (con LDA) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	166
A.41. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,1$ y el clasificador LDA. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	167

A.42. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos aplicados a los resultados en balance con $b = 0,2$ y el clasificador <i>LDA</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	167
A.43. Comparación de las precisiones obtenidas (con PART) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	168
A.44. Comparación de las precisiones obtenidas (con PART) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	169
A.45. Comparación de las precisiones obtenidas (con PART) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	170
A.46. Comparación de las precisiones obtenidas (con PART) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	171
A.47. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos aplicados a los resultados en precisión con $b = 0,1$ y el clasificador <i>PART</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.	172
A.48. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos aplicados a los resultados en precisión con $b = 0,2$ y el clasificador <i>PART</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.	172
A.49. Comparación de los balances obtenidos (con PART) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.	173
A.50. Comparación de los balances obtenidos (con PART) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.	174
A.51. Comparación de los balances obtenidos (con PART) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.	175
A.52. Comparación de los balances obtenidos (con PART) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.	176
A.53. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos aplicados a los resultados en balance con $b = 0,1$ y el clasificador <i>PART</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	177
A.54. Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos aplicados a los resultados en balance con $b = 0,2$ y el clasificador <i>PART</i> . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea. .	177
B.1. Reducciones logradas (para C4.5) por la versión crisp del sistema a distintos valores del parámetro b	179
B.2. Reducciones logradas (para C4.5) por la versión difusa del sistema a distintos valores del parámetro b	180

B.3. Precisiones logradas (con C4.5) por la versión crisp del sistema a distintos valores del parámetro b	181
B.4. Precisiones logradas (con C4.5) por la versión difusa del sistema a distintos valores del parámetro b	182
B.5. Reducciones logradas (para KNN) por la versión crisp del sistema a distintos valores del parámetro b	183
B.6. Reducciones logradas (para KNN) por la versión difusa del sistema a distintos valores del parámetro b	184
B.7. Precisiones logradas (con KNN) por la versión crisp del sistema a distintos valores del parámetro b	185
B.8. Precisiones logradas (con KNN) por la versión difusa del sistema a distintos valores del parámetro b	186
B.9. Reducciones logradas (para LDA) por la versión crisp del sistema a distintos valores del parámetro b	187
B.10. Reducciones logradas (para LDA) por la versión difusa del sistema a distintos valores del parámetro b	188
B.11. Precisiones logradas (con LDA) por la versión crisp del sistema a distintos valores del parámetro b	189
B.12. Precisiones logradas (con LDA) por la versión difusa del sistema a distintos valores del parámetro b	190
B.13. Reducciones logradas (para PART) por la versión crisp del sistema a distintos valores del parámetro b	191
B.14. Reducciones logradas (para PART) por la versión difusa del sistema a distintos valores del parámetro b	192
B.15. Precisiones logradas (con PART) por la versión crisp del sistema a distintos valores del parámetro b	193
B.16. Precisiones logradas (con PART) por la versión difusa del sistema a distintos valores del parámetro b	194
B.17. Reducciones logradas por los métodos individuales que el sistema emplea.	196
B.18. Precisiones logradas (con C4.5) por los métodos individuales que el sistema emplea.	197
B.19. Precisiones logradas (con KNN) por los métodos individuales que el sistema emplea.	198
B.20. Precisiones logradas (con LDA) por los métodos individuales que el sistema emplea.	199
B.21. Precisiones logradas (con PART) por los métodos individuales que el sistema emplea.	200
B.22. Reducciones logradas por los métodos convencionales.	202

B.23. Precisiones logradas (con C4.5) por los métodos convencionales.	203
B.24. Precisiones logradas (con KNN) por los métodos convencionales.	204
B.25. Precisiones logradas (con LDA) por los métodos convencionales.	205
B.26. Precisiones logradas (con PART) por los métodos convencionales.	206
B.27. Reducciones logradas por Explore, SSMA y ZhangTS.	208
B.28. Precisiones logradas (con C4.5) por Explore, SSMA y ZhangTS.	209
B.29. Precisiones logradas (con KNN) por Explore, SSMA y ZhangTS.	210
B.30. Precisiones logradas (con LDA) por Explore, SSMA y ZhangTS.	211
B.31. Precisiones logradas (con PART) por Explore, SSMA y ZhangTS.	212
B.32. Promedio de los balances logrados por los algoritmos genéticos a distintos valores del parámetro b y varios clasificadores.	213
B.33. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 1$ y el clasificador C4.5. Tests de Friedman y Holm.	214
B.34. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 1$ y el clasificador C4.5. Tests de Friedman y Holm.	214
B.35. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 1$ y el clasificador KNN. Tests de Friedman y Holm.	215
B.36. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 1$ y el clasificador KNN. Tests de Friedman y Holm.	215
B.37. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 1$ y el clasificador LDA. Tests de Friedman y Holm.	216
B.38. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 1$ y el clasificador LDA. Tests de Friedman y Holm.	216
B.39. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 1$ y el clasificador PART. Tests de Friedman y Holm.	217
B.40. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 1$ y el clasificador PART. Tests de Friedman y Holm.	217

B.41. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,2$ y el clasificador C4.5. Tests de Friedman y Holm.	218
B.42. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,2$ y el clasificador C4.5. Tests de Friedman y Holm.	218
B.43. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,2$ y el clasificador KNN. Tests de Friedman y Holm.	219
B.44. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,2$ y el clasificador KNN. Tests de Friedman y Holm.	219
B.45. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,2$ y el clasificador LDA. Tests de Friedman y Holm.	220
B.46. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,2$ y el clasificador LDA. Tests de Friedman y Holm.	220
B.47. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,2$ y el clasificador PART. Tests de Friedman y Holm.	221
B.48. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,2$ y el clasificador PART. Tests de Friedman y Holm.	221
B.49. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,3$ y el clasificador C4.5. Tests de Friedman y Holm.	222
B.50. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,3$ y el clasificador C4.5. Tests de Friedman y Holm.	222
B.51. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,3$ y el clasificador KNN. Tests de Friedman y Holm.	223
B.52. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,3$ y el clasificador KNN. Tests de Friedman y Holm.	223

B.53. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,3$ y el clasificador LDA. Tests de Friedman y Holm.	224
B.54. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,3$ y el clasificador LDA. Tests de Friedman y Holm.	224
B.55. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,3$ y el clasificador PART. Tests de Friedman y Holm.	225
B.56. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,3$ y el clasificador PART. Tests de Friedman y Holm.	225
B.57. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,4$ y el clasificador C4.5. Tests de Friedman y Holm.	226
B.58. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,4$ y el clasificador C4.5. Tests de Friedman y Holm.	226
B.59. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,4$ y el clasificador KNN. Tests de Friedman y Holm.	227
B.60. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,4$ y el clasificador KNN. Tests de Friedman y Holm.	227
B.61. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,4$ y el clasificador LDA. Tests de Friedman y Holm.	228
B.62. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,4$ y el clasificador LDA. Tests de Friedman y Holm.	228
B.63. Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,4$ y el clasificador PART. Tests de Friedman y Holm.	229
B.64. Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,4$ y el clasificador PART. Tests de Friedman y Holm.	229
B.65. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,1$ y el clasificador C4.5. Tests de Friedman y Holm. . .	230

B.66. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 1$ y el clasificador C4.5. Tests de Friedman y Holm. . .	230
B.67. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 1$ y el clasificador KNN. Tests de Friedman y Holm. . .	231
B.68. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 1$ y el clasificador KNN. Tests de Friedman y Holm. . .	231
B.69. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 1$ y el clasificador LDA. Tests de Friedman y Holm. . .	232
B.70. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 1$ y el clasificador LDA. Tests de Friedman y Holm. . .	232
B.71. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 1$ y el clasificador PART. Tests de Friedman y Holm. . .	233
B.72. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 1$ y el clasificador PART. Tests de Friedman y Holm. . .	233
B.73. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 2$ y el clasificador C4.5. Tests de Friedman y Holm. . .	234
B.74. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 2$ y el clasificador C4.5. Tests de Friedman y Holm. . .	234
B.75. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 2$ y el clasificador KNN. Tests de Friedman y Holm. . .	235
B.76. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 2$ y el clasificador KNN. Tests de Friedman y Holm. . .	235
B.77. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 2$ y el clasificador LDA. Tests de Friedman y Holm. . .	236
B.78. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 2$ y el clasificador LDA. Tests de Friedman y Holm. . .	236
B.79. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 2$ y el clasificador PART. Tests de Friedman y Holm. . .	237
B.80. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 2$ y el clasificador PART. Tests de Friedman y Holm. . .	237
B.81. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 3$ y el clasificador C4.5. Tests de Friedman y Holm. . .	238
B.82. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 3$ y el clasificador C4.5. Tests de Friedman y Holm. . .	238
B.83. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 3$ y el clasificador KNN. Tests de Friedman y Holm. . .	239

B.84. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 3$ y el clasificador KNN. Tests de Friedman y Holm. . .	239
B.85. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 3$ y el clasificador LDA. Tests de Friedman y Holm. . .	240
B.86. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 3$ y el clasificador LDA. Tests de Friedman y Holm. . .	240
B.87. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 3$ y el clasificador PART. Tests de Friedman y Holm. . .	241
B.88. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 3$ y el clasificador PART. Tests de Friedman y Holm. . .	241
B.89. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 4$ y el clasificador C4.5. Tests de Friedman y Holm. . .	242
B.90. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 4$ y el clasificador C4.5. Tests de Friedman y Holm. . .	242
B.91. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 4$ y el clasificador KNN. Tests de Friedman y Holm. . .	243
B.92. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 4$ y el clasificador KNN. Tests de Friedman y Holm. . .	243
B.93. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 4$ y el clasificador LDA. Tests de Friedman y Holm. . .	244
B.94. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 4$ y el clasificador LDA. Tests de Friedman y Holm. . .	244
B.95. Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 4$ y el clasificador PART. Tests de Friedman y Holm. . .	245
B.96. Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 4$ y el clasificador PART. Tests de Friedman y Holm. . .	245

Capítulo 1

Introducción

A medida que se desarrolla la capacidad de procesamiento y almacenamiento de datos, se hace cada vez más común encontrar grandes bases de datos en los más disímiles entornos. Negocios de todo tipo, centros científicos, instituciones de salud y militares entre otras, almacenan enormes cantidades de datos con el fin de extraer información útil para su desempeño.

Este contexto ha propiciado el desarrollo de un conjunto de técnicas comprendidas en lo que se conoce como Descubrimiento de Conocimiento en Bases de Datos (KDD, del inglés “Knowledge Discovery in Databases”) que en [39] se define como “*el proceso no trivial de identificación de patrones válidos, novedosos, potencialmente útiles y comprensibles en los datos*”. El KDD como todo proceso está integrado por varios pasos o etapas: selección de los datos, preprocesamiento, minería de datos, interpretación y evaluación [73].

Dentro de este proceso, la etapa de Minería de Datos es la encargada de obtener modelos capaces de predecir resultados o revelar relaciones y patrones no triviales presentes en los datos. Entre los modelos más comunes están los de clasificación, encargados de asignar una clase o categoría a un nuevo registro (ejemplo) basándose en la clasificación de los ejemplos contenidos en los datos. El proceso de obtención de estos modelos de clasificación (o simplemente clasificadores) a partir de ejemplos previamente clasificados se conoce como aprendizaje inductivo.

Este es un campo en el que se ha venido trabajando desde hace mucho tiempo, incluso antes de que se acuñara el término KDD. Existe un número bastante amplio de algoritmos de aprendizaje capaces de construir clasificadores, así como varias familias de clasificadores que se diferencian entre sí por la forma de representar el conocimiento aprendido. Entre las familias de clasificadores más conocidas están: los árboles de decisión [82], las reglas [43], las máquinas de vectores de soporte [31], las redes bayesianas [57] y el vecino más cercano [30], este último emplea como modelo a la propia base de datos o un subconjunto de la misma.

Una de las limitantes que surge al intentar aplicar la mayoría de los algoritmos de aprendizaje a problemas de KDD es que no son capaces de procesar en un tiempo razonable bases de datos del tamaño de las que suelen encontrarse en estos problemas. En algunos casos, incluso unos pocos miles de ejemplos pueden constituir una barrera infranqueable. Ante esta situación se presentan dos

alternativas fundamentales para la comunidad científica que trabaja en el tema. La primera es el diseño de nuevos algoritmos de aprendizaje o modificaciones de los existentes con una mayor escalabilidad. La segunda, acerca de la que trata esta memoria, es el desarrollo de técnicas de reducción de datos capaces de suministrar a los algoritmos de aprendizaje conjuntos de datos más pequeños pero con suficiente información como para que estos aprendan correctamente.

Este tipo de técnicas se incluyen en la etapa de selección de los datos dentro del KDD; y cuando su objetivo es reducir el número de ejemplos, al problema que enfrentan se le conoce como Selección de Ejemplos [73]. Este problema ha sido abordado por numerosos autores desde épocas tan tempranas como la década del 60 del siglo pasado [53]. Algunas propuestas han tenido más éxito en sus propósitos que otras, aunque hasta ahora ninguna ha demostrado ser indiscutiblemente superior a las restantes [19, 73]. De hecho, no parece probable que ningún enfoque particular pueda constituir una solución generalmente buena, es decir, que su desempeño esté al nivel de los mejores en cualquier conjunto de datos e independientemente del clasificador que se entrene con los ejemplos seleccionados. Al respecto, el teorema conocido como “No Free Lunch” (NFL) [112] plantea que en general no es posible encontrar un algoritmo que sea el mejor para todos los problemas de un dominio dado.

Independientemente de lo anterior, es posible encontrar estrategias que tengan poca dependencia de las especificidades del problema y con esto aumentar la probabilidad de lograr un buen desempeño en entornos variados. Como en muchos otros campos, en la selección de ejemplos los enfoques meta-heurísticos han sido aplicados con bastante éxito. Estos constituyen un ejemplo del tipo de estrategias a las que nos referimos. Gracias a su capacidad para explorar el espacio de soluciones, han demostrado excelentes resultados en cuanto a la reducción del número de ejemplos y la calidad de los modelos que de estos se obtienen. En particular, los algoritmos genéticos han sido la meta-heurística más empleada con estos propósitos [68, 91, 24, 27, 45, 48].

Sin embargo, el buen comportamiento de las soluciones basadas en meta-heurísticas lleva aparejado un alto costo computacional que las pone en desventaja con respecto a otros enfoques a la hora de su aplicación práctica. Un costo elevado resulta un problema bastante considerable en el campo de la selección de ejemplos, porque precisamente estos métodos están diseñados para procesar bases de datos significativamente grandes. Además, si el costo de la selección es mayor que el del aprendizaje, no se justifica el riesgo de aprender modelos con menor calidad que se deriva de la reducción en el número de ejemplos.

Una alternativa que puede lograr similares resultados a los enfoques meta-heurísticos, pero con un costo computacional muy inferior, es el meta-aprendizaje [17]. Este, a diferencia de los anteriores, ha sido poco abordado hasta el momento en el campo de la selección de ejemplos. Básicamente, consiste en aprender modelos que sirvan para elegir entre varias estrategias de selección de ejemplos, aquella que mejor se adecúe a cada nuevo problema que se presente.

Los primeros pasos en la dirección de aplicar el meta-aprendizaje a la selección de ejemplos, aunque incipientes, ya han sido dados por algunos autores. En [100] y [99] se presenta el meta-aprendizaje como un marco general, aplicable a disímiles tareas de la minería de datos entre las que se incluye la selección de ejemplos. En [80] se usan varias medidas de complejidad de los datos para contrastar los resultados de dos métodos de selección de ejemplos por separado y una combinación de ambos

ante dominios de diferentes complejidades. En [44] se emplean las mismas medidas para predecir cuando es efectivo el uso de un método evolutivo de selección de ejemplos.

Estos trabajos ilustran las enormes posibilidades que ofrece el meta-aprendizaje para la selección de ejemplos. Sin embargo, hay muchos retos que enfrentar para obtener un sistema que explote estas posibilidades. En primer lugar, hasta donde conocemos no se han propuesto modelos teóricos que guíen la aplicación de este tipo de técnicas a esta tarea. La definición de los atributos para el meta-aprendizaje aplicado a la selección de ejemplos, es otro campo abierto que presenta dificultades tanto teóricas como prácticas. Finalmente, la ausencia de conjuntos de meta-ejemplos es otro de los principales problemas a resolver en este campo poco explorado. Este último aspecto, aunque es el de menor complejidad teórica, implica un inmenso volumen de trabajo experimental. Téngase en cuenta que cada meta-ejemplo debe describir el desempeño de un método de selección de ejemplos ante una base de datos y un clasificador, y se tendrá una idea del número de ejecuciones necesarias para construir un conjunto de meta-ejemplos con un tamaño dado.

1.1. Objetivos y Aportaciones

Basándonos en lo anterior, hemos establecido como objetivo del presente trabajo desarrollar un modelo teórico para la aplicación del meta-aprendizaje al problema de la selección de ejemplos, que posibilite obtener métodos de selección de ejemplos cuyos desempeños sean similares a los métodos basados en meta-heurísticas pero su costo computacional sea sensiblemente inferior.

Como resultado del trabajo desarrollado y del cumplimiento del objetivo propuesto, se obtuvieron aportaciones teóricas y prácticas. Entre la primeras, las fundamentales son:

- Un sistema de medidas de complejidad de los datos, diseñado específicamente para su empleo como conjunto de atributos en el meta-aprendizaje para la selección de ejemplos.
- Un modelo teórico para la aplicación del meta-aprendizaje al problema de la selección de ejemplos.
- Una arquitectura para el desarrollo de sistemas para la selección de ejemplos basada en conocimiento.

En cuanto a las aportaciones prácticas se destacan las siguientes:

- Dos instanciaciones de la arquitectura anterior, que constituyen nuevas alternativas para la selección de ejemplos.
- Un conjunto de meta-ejemplos para el problema de la selección de ejemplos.
- Un amplio estudio experimental que incluye más de 20 métodos de selección de ejemplos, 40 bases de datos y cuatro clasificadores.

1.2. Estructura de la Memoria

Esta memoria consta de seis capítulos. Estos son: la presente introducción, cuatro capítulos de contenido y las conclusiones.

En el capítulo 2 describimos formalmente el problema de la selección de ejemplos para clasificación, y analizamos una muestra amplia y representativa de las propuestas dentro de este campo. Esta abarca más de cuatro décadas de aportes e incluye representantes de distintas tendencias dentro de la extensa familia de métodos desarrollados para resolver este problema.

En el capítulo 3 comentamos algunos trabajos sobre meta-aprendizaje que resultan relevantes para el campo de la selección de ejemplos y presentamos un conjunto de medidas que constituyen nuestra propuesta de atributos para la caracterización de bases de datos.

Por su parte el capítulo 4 presenta el modelo que proponemos para el uso del meta-aprendizaje en la selección de ejemplos, así como una implementación del mismo. En ella, empleamos como métodos candidatos siete métodos del estado del arte. Estos se seleccionaron por ser propuestas bien conocidas, ser computacionalmente eficientes y constituir una muestra diversa en cuanto a sus estrategias de selección. Por medio del sistema desarrollado ponemos a prueba la efectividad de las medidas propuestas sometiendo este a un extenso estudio experimental donde demuestra tener mejor desempeño que cualquiera de los métodos candidatos por separado.

Finalmente, en el capítulo 5 proponemos varios métodos de selección de ejemplos concebidos especialmente para reducir el costo computacional de un sistema basado en el modelo propuesto. Al emplear estos en la instanciación de la arquitectura presentada en el capítulo anterior, se explota una de las potencialidades que presenta el sistema de medidas propuesto en esta memoria; la reutilización de gran parte de las estructuras de datos creadas durante su cómputo. Con estos métodos como candidatos, se gana en eficiencia y además se reduce el margen de error al estimar los desempeños. El sistema desarrollado es comparado experimentalmente en un extenso estudio que incluye a varios métodos convencionales del estado del arte y varios métodos basados en meta-heurísticas. Todos estos son evaluados con cuatro clasificadores pertenecientes a diferentes familias. Por medio de este estudio, se demuestra que el objetivo propuesto para el presente trabajo ha sido cumplido.

Capítulo 2

Selección de Ejemplos para Clasificación

En este capítulo describimos el problema de la selección de ejemplos, y presentamos una muestra amplia y representativa de las propuestas dentro de este campo. Con ella hemos intentado abarcar más de cuatro décadas de aportes que se iniciaron en la década de los 60 del siglo pasado y llegan hasta el presente. Hemos tratado además de incluir representantes de distintas tendencias dentro de la amplia familia de métodos desarrollados para resolver este problema.

2.1. El Problema de la Selección de Ejemplos

La selección de ejemplos puede verse como un proceso de transformación de un conjunto T en otro S , definido en el mismo universo pero de menor cardinalidad. En esta memoria, nos centraremos en problemas donde el universo U es un espacio vectorial de M dimensiones. Cada elemento de T o de S es entonces un punto en ese espacio y por tanto un vector de M componentes, a estos puntos se les llama ejemplos y a cada dimensión atributo.

En [73] se define la selección de ejemplos como la búsqueda del menor subconjunto S de un conjunto original de ejemplos T , tal que S permita obtener los mismos resultados que T a un algoritmo de minería de datos L . Para los propósitos de este trabajo, L será en específico un algoritmo de aprendizaje inductivo [79].

En general, no todos los métodos de selección de ejemplos buscan un subconjunto de T , existen también los enfoques que generan nuevos ejemplos (siempre en U) [92]. En este trabajo se entiende por selección de ejemplos al problema consistente en:

Dado un conjunto de ejemplos T^1 definidos sobre un universo de posibles ejemplos U ($T \subseteq U$), y un algoritmo de aprendizaje L , buscar el menor conjunto S ($S \subset U$ y $|S| < |T|$) que permita a L obtener un clasificador tan o más preciso que el que puede obtener a partir de T .

¹Nos referiremos indistintamente al mismo concepto como conjunto de ejemplos o base de datos

Si la técnica empleada no incluye la posibilidad de generar nuevos ejemplos entonces la búsqueda se limita a $S \subseteq T$. En cualquier caso, para el problema de la selección de ejemplos no existe una solución con un costo computacional polinomial con respecto a $|T|$ [52, 118], por lo que todas las propuestas existentes están basadas en heurísticas y ninguna garantiza la minimalidad de S ni la maximalidad de la precisión.

En el caso ideal, un método de selección de ejemplos debería ser independiente del modelo [73]. Esto significa que los resultados logrados con S sean igualmente buenos para cualquier algoritmo de aprendizaje (o minería de datos) que los emplee. Sin embargo hasta la fecha la independencia del modelo es otro objetivo que está lejos de alcanzarse. En la práctica, la mayoría de las propuestas existentes han sido concebidas para mejorar el desempeño de un clasificador específico, siendo este generalmente KNN [30]. No obstante, varios autores han empleado métodos originalmente creados para KNN en la selección de ejemplos para otros clasificadores [32, 102, 51, 24, 27]. Estos intentos han producido en algunos casos buenos resultados, pero nunca una real independencia del modelo.

Otra aspiración aparentemente no alcanzable es diseñar un método de selección de ejemplos que obtenga buenos resultados en cualquier dominio. Entre estos, la naturaleza de los datos puede variar con la presencia o no de ruido o información redundante, así como con la distribución de las clases que puede ser más o menos homogénea, y cada vez parece más claro que la naturaleza de los datos es determinante en el comportamiento de cualquier algoritmo de este tipo [19].

2.2. Enfoques Fundamentales Para la Selección de Ejemplos

El problema de la selección de ejemplos ha sido abordado por numerosos autores con diversos enfoques desde épocas tan tempranas como la década del 60 del siglo pasado [53]. Algunos han tenido más éxito en sus propósitos que otros, aunque hasta ahora ninguno ha demostrado ser indiscutiblemente superior a los restantes [19, 73]. Por la abundancia de propuestas existentes, estudiar todos los algoritmos publicados es una tarea difícilmente concretable. Sin embargo, como parte de este trabajo ha sido necesario analizar una muestra bastante amplia de los algoritmos más relevantes en este campo. Este epígrafe presenta las características fundamentales de algunos de los algoritmos de selección de ejemplos estudiados. Para facilitar su análisis, estos han sido agrupados en categorías que, a nuestro juicio, ilustran las tendencias más importantes en este campo. Estas no siempre son mutuamente exclusivas, lo que causa que determinados algoritmos puedan corresponderse con más de una categoría. Tales casos se han descrito solamente en la categoría que consideramos que mejor los caracteriza.

En lo adelante, para evitar repeticiones innecesarias, usaremos la siguiente notación:

- T denota al conjunto de ejemplos inicial o de entrada.
- S denota al conjunto de ejemplos resultante o de salida.
- N denota la cardinalidad de T , es decir, el número de ejemplos de entrada.
- M denota la dimensionalidad de T , es decir, el número de atributos.

- $dist(x, y)$ denota la distancia entre los ejemplos x e y . A menos que se especifique lo contrario, se asumirá el uso de la distancia euclídea.
- $nn(e, H)$, siendo e un ejemplo y H un conjunto de ejemplos, denota al ejemplo cuya distancia a e sea la menor entre todos los ejemplos de H .

$$e' = nn(e, H) \Leftrightarrow \forall (x \in H) dist(x, e) \geq dist(e', e)$$

- KNN denota el método de clasificación de los vecinos más cercanos [30], siendo K el número de vecinos considerados y pudiendo ser reemplazado por un valor concreto, por ejemplo 1NN.
- $KNN(e, H)$, siendo e un ejemplo y H un conjunto de ejemplos, denota la clasificación que KNN asigna a e tomando H como base de ejemplos. al igual que en el caso anterior K puede ser reemplazado por un valor concreto.
- $class(e)$ denota la clase del ejemplo e .
- Nos referiremos como enemigo más cercano de un ejemplo e en el conjunto H (o $ne(e, H)$), al ejemplo de clase diferente a la de e cuya distancia a este es la menor entre los ejemplos de H que cumplen tal condición.

$$e' = ne(e, H) \Leftrightarrow \forall (x \in H)[class(x) \neq class(e) \Rightarrow dist(x, e) \geq dist(e', e)]$$

- Usaremos la expresión “clasificado por H ” (siendo H un conjunto de ejemplos) para referirnos a “clasificado por 1NN usando a H como base de conocimiento”.
- Usaremos la expresión “precisión de A ” (siendo A un método de selección de ejemplos) para referirnos a la precisión lograda por un clasificador luego de ser entrenado empleando los resultados de A . A menos que se especifique lo contrario, este clasificador será 1NN.
- $randomSelect(H)$, siendo H un conjunto de ejemplos, denota un ejemplo elegido aleatoriamente entre sus miembros.

2.2.1. Conservación de las Fronteras

Para muchos clasificadores, incluido KNN, los ejemplos ubicados en las fronteras de clases tienden a ser determinantes en la correcta clasificación de nuevos ejemplos. En cambio, aquellos ubicados en zonas centrales, suelen ser de poca relevancia. Teniendo esto en cuenta, muchos autores han apostado por el desarrollo de métodos de selección de ejemplos que conserven las fronteras de clases.

Un modo bastante recurrido de preservar estas fronteras es mediante la búsqueda de un subconjunto consistente. Dado un clasificador L , se dice que S es consistente con respecto a T , si al entrenar a L empleando a S , el modelo aprendido clasifica correctamente todos los ejemplos de T . La búsqueda de un S consistente con T fue uno de los primeros enfoques en el campo de la selección de ejemplos aunque no es el único modo de preservar las fronteras.

El principal inconveniente de esta estrategia radica en una alta sensibilidad al ruido en el conjunto de entrenamiento. La causa de este comportamiento es que cada ejemplo ruidoso es interpretado como frontera de un cluster unitario y por tanto se preserva. A continuación se describen cuatro representantes de esta familia, tres de ellos basados en la búsqueda de consistencia.

2.2.1.1. CNN

Probablemente la primera propuesta para la selección de ejemplos fue CNN (Condensed Nearest Neighbor) [53]. Este método sigue un enfoque incremental, lo que significa que inicialmente S está vacío o contiene un número inicial mínimo de ejemplos y se van agregando iterativamente nuevos ejemplos que satisfagan determinada condición sobre S .

CNN busca que cada miembro de T sea correctamente clasificado por S . Para lograrlo, parte de una definición inicial de S que contiene un ejemplo de cada clase c en T y va analizando cada ejemplo $e \in T$. Si e no está correctamente clasificado por S , se agrega a este conjunto para garantizar su correcta clasificación, de lo contrario no se adiciona. Este proceso se repite hasta que no se agreguen más ejemplos a S . El algoritmo es muy simple como puede apreciarse en la figura 2.1.

Este método tiende a retener los ejemplos cercanos a las fronteras de las clases, eliminando una gran porción de las regiones interiores. Esta propia característica determina su principal debilidad, una alta sensibilidad al ruido. La condición de retención interpreta los ejemplos ruidosos como fronterizos y por tanto los conserva. El resultado es que la proporción de ruido en S suele ser mucho mayor que en T y en consecuencia la precisión en la clasificación de nuevos ejemplos se ve afectada. Este es un caso típico de sobre-especialización.

A favor de este algoritmo puede decirse que en bases de ejemplos con poco ruido logra un alto nivel de reducción y un desempeño aceptable en la posterior clasificación. Este método además presenta uno de los enfoques más simples e intuitivos en su campo, mientras su costo computacional es relativamente bajo. Por estas razones, a pesar de su antigüedad, sigue siendo un patrón de comparación en estudios contemporáneos [19, 24, 51, 80, 27, 44].

2.2.1.2. MCS

CNN logra que S sea consistente con T , pero no garantiza que sea el menor subconjunto que cumple tal condición. Esto ha motivado que otros autores hayan desarrollado propuestas posteriores con la intención de mejorar estos resultados garantizando la minimalidad de S . Tal es el caso de Dasarathy, quien propuso MCS (Minimal Consistent Subset) [33].

MCS se basa en el concepto de NUN (nearest unlike neighbor), equivalente al de enemigo más cercano. Para garantizar la consistencia de la base de ejemplos, basta con garantizar la permanencia en ésta de al menos un ejemplo para cada ejemplo e , tal que la distancia entre ambos sea menor que la distancia entre e y su NUN .

Este algoritmo (figura 2.2) se inicia buscando en el conjunto consistente (CS , inicialmente T) el NUN de cada ejemplo de T , cada ejemplo “aporta un voto” a cada uno de los que estén más cercanos

```

function CNN( $T$ )
begin
   $S = \{\}$ 
  foreach( $c \in \text{classes}(T)$ )
     $S = S \cup \text{randomSelect}(\{x \in T \mid \text{class}(x) = c\})$ 
  do
     $DS = \{\}$ 
    foreach( $e \in T$ )
      if( $1NN(e, S) \neq \text{class}(e)$ ) // condición de retención
         $S = S \cup \{e\}$ 
         $DS = DS \cup \{e\}$ 
         $T = T - \{e\}$ 
    while( $DS \neq \{\}$ )
  return  $S$ 
end

```

Figura 2.1: Algoritmo CNN.

a él que su NUN . Luego de esta etapa CS es reiniciado, el ejemplo más votado se le adiciona y se anulan todos los votos aportados por los ejemplos que votaron por él, pues ya su correcta clasificación está garantizada. La selección y el consecuente reajuste de la votación, se repiten hasta que no queden ejemplos con votos, lo que indica que se ha alcanzado la consistencia. Una vez lograda la consistencia, se repite el proceso buscando los $NUNs$ en el nuevo CS y el algoritmo itera de este modo hasta que no sea posible reducir más el número de ejemplos en este conjunto, con lo que se asume que ha sido encontrado el mínimo conjunto consistente.

Aunque no caben dudas de la capacidad de MCS para garantizar la consistencia de S , sí ha sido demostrado [115] que contrario a lo que empíricamente planteaba su autor, este algoritmo no garantiza la minimalidad del conjunto resultante. Experimentos publicados [115] reflejan además que aunque MCS logra mejores niveles de reducción de la base de ejemplos que CNN, la precisión que alcanza es menor. A esto se suma que tanto el costo computacional como la complejidad de este algoritmo son superiores a los de CNN y padece una sensibilidad al ruido similar a la de este.

2.2.1.3. FCNN

FCNN [9] es uno de los aportes más recientes que sigue el enfoque de búsqueda de la consistencia. A diferencia de la propuesta comentada previamente, con este no se pretende alcanzar la minimalidad de S , sino un método rápido a la vez que independiente del orden de presentación de los ejemplos.

Los autores proponen 4 variantes de FCNN, la mas simple se puede apreciar en la figura 2.3. El método comienza seleccionando el centroide de cada clase y luego itera hasta alcanzar la consistencia seleccionando representantes de los ejemplos mal clasificados por S . Las diferencias fundamentales que presenta respecto a CNN radican en la inicialización de S , el uso de celdas de

```

function MCS(T)
begin
  CS = T
  do
    S = CS
    foreach(e ∈ T) e.votes = 0
    //calcular NUNs y asignar votos
    foreach(e ∈ T)
      e.nun = NearestUnlikeNeighbor(e, CS)
      e.vote_receivers = {}
      foreach(x ∈ T)
        if(dist(e, x) < dist(e, e.nun))
          e.vote_receivers = e.vote_receivers ∪ {x}
          x.votes ++
    Crear conjunto de candidatos con miembros de CS y ejemplos que no creen inconsistencias
    PCS = {}
    foreach(e ∈ T)
      if(e ∈ CS ∨ ¬∃(x ∈ CS)[class(x) <> class(e) ∧ Dist(x, x.NUN) > Dist(x, e)] )
        PCS = PCS ∪ {e}
    //Construir nuevo CS añadiendo ejemplos más votados hasta lograr consistencia
    CS = {}
    do
      max = max_voted(PCS)
      CS = S ∪ {max}
      //localizar ejemplos cuya correcta clasificación es garantizada por max
      foreach(e ∈ T)
        if(dist(e, max) < dist(e, e.nun))
          //anular votos producidos por e
          foreach(x ∈ e.vote_receivers) x.votes --
    while(max_voted(PCS).votes > 0)
  while(|CS| < |S|)
  return S
end

```

Figura 2.2: Algoritmo MCS.

```

function FCNN( $T$ )
begin
   $S = \{\}$ 
  //Seleccionar el centroide de cada clase
   $DS = Centroids(T)$ 
  while( $DS \neq \{\}$ )
     $S = S \cup DS$ 
     $DS = \{\}$ 
    foreach( $e \in S$ )
       $Vor = \{q \in T \mid e = nn(q, S)\}$ 
       $Voren = \{q \in Vor \mid class(e) \neq class(q)\}$ 
       $DS = DS \cup \{nn(e, Voren)\}$ 
  return  $S$ 
end

```

Figura 2.3: Algoritmo FCNN.

Voronoi² para identificar ejemplos mal clasificados y en que S no se actualiza sino hasta haber sido procesados todos sus ejemplos.

La inicialización de S con los centroides, posibilita la independencia del resultado con respecto al orden de presentación de los ejemplos. Por su parte, el uso de las celdas de Voronoi y la estrategia de actualización de S , permiten implementar optimizaciones basadas en la conservación de listas de vecinos, así como en desigualdades triangulares que reducen el tiempo de procesamiento requerido.

Las diferencias entre las 4 variantes propuestas por los autores, son el número de ejemplos añadidos a S en cada iteración y el criterio para seleccionar entre los ejemplos mal clasificados los que se añaden en cada iteración. Como puede apreciarse, el algoritmo de la figura 2.3 añade hasta un nuevo ejemplo por cada miembro e de S y para esto selecciona el más cercano entre los miembros de la celda de Voronoi de e que tienen un clase diferente a este.

Al ser un método bastante reciente, no abundan los trabajos donde se compare con otras propuestas. Sin embargo, es bastante lógico suponer que mas allá de las mejoras en velocidad e independencia del orden de presentación, no debe diferir significativamente de CNN en sus resultados.

2.2.1.4. EOP y POP

En [1] se presenta el algoritmo EOP (Editing by Ordered Projections), y en [93] POP (Patterns by Ordered Projections), una versión mejorada del primero. Estos algoritmos, al igual que los anteriores, preservan las fronteras de clases; pero a diferencia de estos no garantizan la consistencia de S con respecto a T .

Ambos métodos buscan en el espacio M -dimensional en que se encuentran los ejemplos de T , hiper-rectángulos donde todos los ejemplos contenidos tengan la misma clasificación. Un ejemplo

²En la celda de Voronoi de un ejemplo e están todos los ejemplos de T que tienen a e como vecino más cercano en S .

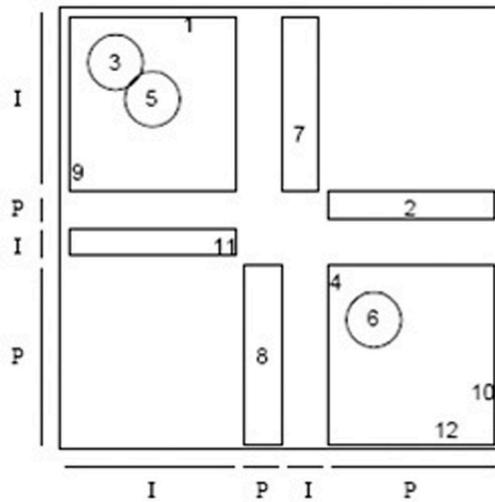


Figura 2.4: Ejemplo de los resultados de aplicar POP en un espacio bidimensional.

se conserva sólo si es indispensable para delimitar en al menos una de las dimensiones uno de los hiper-rectángulos identificados. La heurística que se sigue identifica en las proyecciones ordenadas de los ejemplos sobre los ejes que representan los valores posibles de cada atributo, segmentos en los que todos los ejemplos contenidos tengan la misma clasificación. En cada segmento el ejemplo del extremo izquierdo y el del derecho se catalogan como bordes, y para cada ejemplo se define su debilidad como el número de ejes en los cuales no está catalogado como borde. Finalmente se eliminan todos los ejemplos cuya debilidad sea igual a M .

La figura 2.4 (tomada de [93]), ilustra el proceder de POP. En la gráfica aparece representado un conjunto de 12 ejemplos en un espacio bidimensional. Existen dos clases (P e I) y a lo largo de los ejes están identificados los segmentos, etiquetado cada uno con su clase correspondiente. Aparecen también delimitados los seis rectángulos que identificaría POP para esta base de ejemplos, en cuyo interior están circulados los ejemplos a eliminar (3, 5 y 6).

Las principales mejoras que presenta POP con respecto a EOP, consisten en la adaptación para el trabajo con atributos discretos y la refinación del ordenamiento de las proyecciones para lograr un mejor desempeño en casos de múltiples ejemplos con diferentes clasificaciones y valores coincidentes para un atributo.

El costo computacional de POP está en el orden de $O(MN \log N)$, que resulta muy bueno, aún en comparación con otros métodos considerados escalables como CNN, ENN [110] e IB3 [3]. Sin embargo los niveles de reducción que produce son insuficientes si se comparan con la mayoría de los métodos del estado del arte. Por otra parte, aunque no se han publicado experimentos sobre el desempeño de POP en presencia de altos niveles de ruido, es razonable pensar que debe deteriorarse sensiblemente. Esto se debe a que muchos ejemplos ruidosos serán considerados como bordes que delimitan segmentos unitarios y serán retenidos en la base de ejemplos, aumentando su proporción y con ello el impacto que tienen en la clasificación de nuevos ejemplos.

```

function POP( $T$ )
begin
  foreach( $e \in T$ )
     $e.weakness = 0$ 
    //procesamiento para atributos continuos
    foreach( $a \in ContinuousAttributes(T)$ )
       $QuickSort(projection(T, a))$ 
      //refinación del ordenamiento
       $ReSort(projection(T, a))$ 
      foreach( $e \in T$ )
        if ( $\neg border(e)$ )
           $e.weakness = e.weakness + 1$ 
    //procesamiento para atributos discretos
    foreach( $a \in DiscreteAttributes(T)$ )
      foreach( $v \in Values(a)$ )
         $V = \{e \in T \mid e.a = v\}$ 
         $V_{min} = \{e \mid e \in V \wedge e.weakness = Min_{e \in V}(e.weakness)\}$ 
        foreach( $e \in V - V_{min}$ )
           $e.weakness = e.weakness + 1$ 
    //aplicación del criterio de retención
     $S = \{\}$ 
    foreach( $e \in T$ )
      if( $e.weakness < M$ )
         $S = S \cup \{e\}$ 
    return  $S$ 
end

```

Figura 2.5: Algoritmo POP.

```

function ENN(T)
begin
  S = {}
  foreach(e ∈ T)
    if(3NN(e, T - {e}) = class(e))// condición de inclusión
      S = S ∪ {e}
  return S
end

```

Figura 2.6: Algoritmo ENN.

2.2.2. Filtros de Ruido

Si bien los enfoques que persiguen la consistencia de S con T , son por naturaleza muy sensibles al ruido, no tardaron en aparecer propuestas que se caracterizaron por eliminar precisamente aquellos ejemplos ubicados en regiones donde su clase no sea la predominante. En contraposición a las anteriores, estas no logran reducciones significativas en el número de ejemplos, pero tienen un desempeño muy bueno como filtros de ruido y por lo general su aplicación da lugar a mejoras en la precisión. Las propuestas de este tipo que describimos a continuación incluyen: el primer y más empleado enfoque de esta familia, dos modificaciones de este y una propuesta más reciente inspirada en los filtros de frecuencias empleados en áreas como el procesamiento de señales e imágenes.

2.2.2.1. ENN

La primera y más difundida de estas propuestas fue ENN (Edited Nearest Neighbor) [110]. Este método retiene en S aquellos ejemplos que son correctamente clasificados por sus tres vecinos más cercanos en T , tal como se define a continuación:

$$S = \{e \mid 3NN(e, T - \{e\}) = class(e)\}$$

Como CNN, ENN tiene la ventaja de su simplicidad y escalabilidad. Pero en contraste con aquel, sus niveles de reducción de la base de ejemplos son muy bajos. Lo que es consecuencia lógica de su propiedad de eliminar solamente los ejemplos cercanos a las fronteras de clases o ruidosos. Estas características lo han hecho atractivo para incorporarlo como una etapa inicial de eliminación de ruido en algoritmos más recientes [19, 111].

2.2.2.2. RENN y All-kNN

En [104] se proponen dos variantes de ENN. La primera consiste en aplicar este método iterativamente hasta que todos los ejemplos sean correctamente clasificados por 3NN. Esta se conoce como repeated ENN o RENN. La otra modificación es conocida como All-kNN, en ella ENN se repite para todos los valores de k por debajo de cierto límite.

Ambas alternativas han sido experimentadas en trabajos como [111, 24, 51, 45], observándose niveles de reducción de la base de ejemplos ligeramente mayores que los logrados por ENN y precisiones prácticamente iguales, resultados que a nuestro juicio, no compensan el costo computacional extra que el procesamiento iterativo implica.

2.2.2.3. HPFF

Un filtro de ruido que no se basa en ENN es presentado en [69]. En este caso se trata de HPFF (High Pass Filter), inspirado en los filtros de frecuencias empleados en áreas como el procesamiento de señales e imágenes.

Este método está concebido para trabajar con bases de datos constituidas por vectores con n componentes binarios ($T \subseteq \{0, 1\}^n$), cada uno de los cuales pertenece o no a la clase objetivo. T se divide en dos subconjuntos disjuntos: P y N . El primero se integra por los ejemplos que pertenecen a la clase objetivo y el segundo está constituido por aquellos que no cumplen esta condición. A partir de P y N se define un conjunto de patrones S' de la siguiente manera:

$$S' = \{\bar{a} \oplus \bar{b} \mid \bar{a} \in P, \bar{b} \in N\}$$

Donde:

$$\bar{a} \oplus \bar{b} = \langle a_1 \text{ XOR } b_1, \dots, a_n \text{ XOR } b_n \rangle$$

Cada patrón en S' representa diferencias entre ejemplos de P y de N . Varios pares de ejemplos pueden dar lugar al mismo patrón, por lo que a cada $s \in S'$ se asocia un entero que representa la frecuencia de aparición de s al calcularse S' (número de pares $\langle \bar{a}, \bar{b} \rangle$ que dan lugar a s). El supuesto sobre el que trabaja HPFF es que los ejemplos ruidosos en T , generan patrones en S' con baja frecuencia de aparición. Por lo tanto, a partir de los parámetros t y k , elimina aquellos ejemplos que generan al menos t patrones con frecuencia menor o igual a k .

Consideramos que este enfoque constituye una alternativa interesante para la selección de ejemplos de aprendizaje. Sin embargo, el autor no propone criterios para la elección de los valores adecuados para los parámetros t y k , su limitación al trabajo con clasificación binaria y atributos binarios restringe mucho su aplicabilidad y los resultados experimentales publicados [69] son escasos por lo que no permiten evaluar su desempeño real.

2.2.3. Condensación con Filtrado Previo del Ruido

Como hemos comentado, los métodos que buscan la consistencia de S respecto a T reducen significativamente el número de ejemplos en comparación con los filtros de ruido. Sin embargo, son extremadamente sensibles al ruido, ante cuya presencia ven seriamente afectado su desempeño. En cambio, los filtros de ruido aunque no logran reducciones significativas en el número de ejemplos, pueden incrementar la precisión de los clasificadores al eliminar los ejemplos ruidosos.

A continuación describimos cuatro métodos con cuyo enfoque se pretende aprovechar las ventajas de los dos enfoques anteriores. Este se caracteriza por la aplicación inicial de un filtro de ruido y la posterior eliminación de los ejemplos que resulten irrelevantes para el aprendizaje. Los métodos de esta familia tienden a retener ejemplos cercanos a una versión suavizada de las fronteras entre las clases, son mucho más tolerantes al ruido que los métodos que buscan la consistencia de S con T y logran mejores niveles de reducción que estos.

2.2.3.1. DROP3

Wilson y Martínez [111] proponen cinco algoritmos que denominan DROP1-5 (Decremental Reduction Optimization Procedure) con los objetivos de lograr tolerancia al ruido, alta precisión en la generalización, no sensibilidad al orden de presentación de los ejemplos y reducción significativa de la base de ejemplos. Entre estos, DROP3 es el que mejor resultados generales ha mostrado, siendo incluido frecuentemente como patrón de comparación en estudios posteriores por diversos autores [19, 117, 24, 76, 45, 77, 85, 44, 48].

Para cada ejemplo e , estos métodos mantienen una lista de vecinos cercanos y una de asociados. Esta última contiene los ejemplos que tienen a e como vecino cercano. Así, los algoritmos DROP se basan en la idea de que al eliminar un ejemplo e , los únicos ejemplos cuya clasificación pudiera verse afectada son sus asociados. Entonces, un buen indicador del impacto de la supresión se logra comparando esta clasificación en el caso de emplear e (*with*) y en el caso de no emplearlo (*without*). Si prescindiendo de e se clasifican correctamente al menos tantos de sus asociados como empleándolo, se elimina este ejemplo del conjunto.

Como puede apreciarse (figura 2.7), el primer paso en DROP3 es la ejecución de ENN sobre el conjunto T , de este modo se pretende eliminar el ruido que pudiera estar presente en los datos. A continuación se ordenan decrecientemente los ejemplos en S tomando como criterio la distancia con respecto a sus enemigos más cercanos (*sortDescNE(S)*). Con este ordenamiento inicial se logra la independencia del orden de presentación de los ejemplos y se busca analizar primero los más lejanos de las fronteras de las clases. Este criterio aumenta la probabilidad de retener ejemplos cercanos a las fronteras, que suelen ser los más significativos en la clasificación.

Luego se crean las listas de vecinos y asociados y para cada ejemplo se calculan los valores de *with* y *without*, con los que se determina la retención o supresión del mismo. En caso de ser eliminado, se actualizan las listas de vecinos de sus asociados. Nótese como en cambio no se actualizan las listas de asociados de sus vecinos cercanos. De este modo el efecto de eliminar cada ejemplo se analiza sobre la clasificación de sus asociados en el conjunto inicial ($S = ENN(T)$) en lugar de hacerlo sobre los correspondientes al actual S .

Los resultados de este método son buenos tanto en precisión como en reducción, lo que ha hecho que como comentábamos, haya sido incluido como patrón de comparación en estudios posteriores por diversos autores.

```

function DROP3(T)
begin
  S = ENN(T)
  sortDescNE(S)
  //Crear listas de vecinos y asociados
  foreach(e ∈ S)
    e.Neighbors = neighborhood(e, S, k + 1)
    foreach(n ∈ e.Neighbors)
      n.Asociates = n.Asociates ∪ {e}
  foreach(e ∈ S)
    with = {a | a ∈ e.Asociates ∧ kNN(a, a.N) = class(a)}
    without = {a | a ∈ e.Asociates ∧ kNN(a, a.Neighbors - {e}) = class(a)}
    //Eliminar si without ≥ with y actualizar vecindario de sus asociados
    if(without ≥ with)
      S = S - {e}
      foreach(a ∈ e.Asociates)
        a.Neighbors = neighborhood(a, S, k + 1)
        foreach(an ∈ a.Neighbors)
          an.Asociates = an.Asociates ∪ {a}
  return S
end

```

Figura 2.7: Algoritmo DROP3.

2.2.3.2. ICF

Brigthon y Mellish proponen en [19] ICF (Iterative Case Filtering), otro de los métodos de selección de ejemplos recurrentemente citados y empleados en comparaciones por los autores de esta área [24, 51, 76, 109, 77, 48].

ICF presenta unas cuantas similitudes con DROP3. Ambos comienzan con una pasada de un filtro de ruido basado en ENN. Los dos algoritmos determinan, para cada ejemplo, sus conjuntos de vecinos y asociados (llamados conjunto alcance y cubrimiento respectivamente en ICF) y emplean estos en la decisión de mantener o excluir cada ejemplo. Otra característica común es que en ambos casos la parte del conjunto de ejemplos que se retiene es una banda estrecha cercana a las fronteras de las clases.

Sin embargo existen diferencias muy importantes entre DROP3 y la propuesta de Brigthon y Mellish. Una de ellas es que ICF requiere varias iteraciones sobre la base de ejemplos, el resto están relacionadas con la manera en que este define y emplea los conjuntos alcance y cubrimiento.

En primer lugar, el conjunto alcance para un ejemplo e no tiene un tamaño prefijado, sino que incluye a todos los ejemplos e' tales que $dist(e, e') < dist(e, ne(e, S))$, los que obviamente tienen la misma clasificación que e . En otras palabras, el conjunto alcance está contenido en una hiper-esfera cuyo radio es igual a $dist(e, ne(e, S))$. En segundo lugar, no es necesario mantener un listado de los miembros de cada conjunto sino que basta almacenar el número de ellos, valor que por demás se actualiza en cada iteración. Por último, el criterio que determina la exclusión de un ejemplo e es que

```

function ICF(T)
begin
  S = T
  Rem = {}
  foreach(e ∈ S)
    if(kNN(e, S) ≠ class(e))
      Rem = Rem ∪ {e}
  S = S − Rem
  do
    foreach(e ∈ S)
      //calcular conjunto alcance
      computeReachable(e)
      //calcular conjunto cubrimiento
      computeCoverage(e)
    Rem = {}
    foreach(e ∈ S)
      if(reachable(e) > coverage(e))
        Rem = Rem ∪ {e}
    S = S − Rem
  while (Rem ≠ {})
  return S
end

```

Figura 2.8: Algoritmo ICF.

tenga un conjunto alcance mayor que su cubrimiento, lo que significa que e puede ser correctamente clasificado por más ejemplos que los que él puede clasificar correctamente.

Según sus propios autores y las evidencias que aportan las pruebas experimentales publicadas por estos [19], DROP3 e ICF mantienen un desempeño promedio muy similar en 30 bases de datos del repositorio UCI [11]. A favor del último puede decirse que requiere menos memoria para estructuras de datos auxiliares por no mantener las listas de vecinos y asociados. Pudiera pensarse que esto se logra a costa de una menor eficiencia computacional por su carácter iterativo y la necesidad de recalcular los conjuntos alcance y cubrimiento para cada ejemplo en cada iteración, sin embargo los experimentos sugieren que no existe una dependencia entre el número de ejemplos y la cantidad de iteraciones necesarias, sino que la velocidad de convergencia está determinada más bien por la estructura de las clases y suelen ser muy pocas las iteraciones requeridas.

2.2.3.3. Cpruner

En [117] se describe Cpruner, método por medio del que se intenta una vez más eliminar del conjunto de ejemplos tanto los ejemplos ruidosos como los redundantes. En este caso los autores refieren emplear criterios “más cuidadosos” que los de métodos precedentes como IB3 [3] y DROP3, para evitar eliminar ejemplos cuya exclusión pueda provocar una pérdida de precisión en la clasificación.

Asociados a Cpruner se definen para cada ejemplo los conjuntos k -alcance y k -cubrimiento. El

```

function Cpruner( $T, k$ )
begin
   $S = T$ 
  foreach( $e \in S$ )
    computeReachability( $e, k$ )
    computeCoverage( $e, k$ )
  //eliminar ruido
  foreach( $e \in S$ )
    if(noisy( $e$ ))
       $S = S - \{e\}$ 
      foreach(  $c \in coverage(e, k)$ )
        computeReachability( $c, k$ )
      foreach(  $r \in reachability(e, k)$ )
        Coverage( $r, k$ ) = coverage( $r, k$ ) -  $\{e\}$ 
  //ordenar según lejanía de las fronteras de clases
  Sort( $S$ )
  //eliminar redundancias
  foreach(  $e \in S$ )
    if(superfluous( $e$ )  $\wedge$   $\neg$ critical( $e$ ))
       $S = S - \{e\}$ 
      foreach( $c \in coverage(e, k)$ )
        computeReachability( $c, k$ )
  return  $S$ 
end

```

Figura 2.9: Algoritmo Cpruner.

conjunto *k-alcance* de un ejemplo está formado por sus *k* vecinos más cercanos, mientras el conjunto *k-cubrimiento* se compone por aquellos que lo tienen como uno de sus *k* vecinos más cercanos y además comparten su misma clase.

Estos conjuntos se emplean para determinar si un ejemplo es superfluo, crítico o ruidoso. Se considera un ejemplo como superfluo si es clasificado correctamente por su conjunto *k-alcance*. Un ejemplo es crítico si existe otro en su *k-cubrimiento* que no es superfluo o si su supresión provoca que esto ocurra. Mientras se consideran ruidosos los ejemplos no superfluos cuyos correspondientes *k-alcances* tienen menos ejemplos que sus *k-cubrimientos*.

Las propiedades anteriores son empleadas para determinar la exclusión o no de cada ejemplo, eliminándose aquellos que sean ruidosos, o sean superfluos y no críticos. Adicionalmente se ordenan de manera decreciente los ejemplos para su procesamiento, atendiendo en primer lugar al número ejemplos de su propia clase que existen en su *k-alcance*, y en segundo lugar a la distancia con respecto a su enemigo más cercano. Con esto se busca aumentar la probabilidad de retener los ejemplos cercanos a las fronteras de las clases, mientras se aumenta la de eliminar ejemplos ubicados en las zonas centrales de estas.

Según los resultados experimentales publicados [117], la precisión alcanzada por Cpruner es algo superior a la de IB3 y DROP3. Esto se logra al costo de una menor reducción de la base de ejemplos,

aunque los valores registrados pueden considerarse aceptables. Lamentablemente los autores no especifican el valor del parámetro k .

A pesar de ser escasamente referenciado en la literatura especializada, hemos comprobado experimentalmente [21] que en efecto, este método mejora ligeramente no sólo los resultados de IB3 y DROP3, sino también los de ICF.

2.2.3.4. CCIS

Marchiori propone en [77] un método basado en la representación de T como una red de éxitos y fracasos (*HMN*, de hit miss network). Esta representación, propuesta por la misma autora en un trabajo previo [76], consiste en un grafo donde los vértices son los ejemplos de T , mientras las aristas definen una relación de vecindad de la siguiente manera:

Sea $\{x, y\} \subseteq T$,
 $C_y = \{e \in T \mid \text{class}(e) = \text{class}(y)\}$
 si $y = \text{nn}(x, C_y - \{x\})$, entonces existe una arista con origen en x y destino en y

A su vez, se definen éxitos (hits) y fracasos (miss) como sigue:

Sea $\{x, y\} \subseteq T$, tal que exista una arista de x a y en el correspondiente *HMN*
 si $\text{class}(x) = \text{class}(y)$, entonces x es un éxito de y ($x \in \text{Hit}(y)$), de lo contrario es un fracaso ($x \in \text{Miss}(y)$)

El método, denominado CCIS (Class Conditional Instance Selection), en una primera etapa que elimina ruido y ejemplos muy cercanos a las fronteras de clases; mientras en una segunda etapa elimina puntos centrales irrelevantes para la clasificación. Sin embargo, a diferencia de la mayoría de los métodos que siguen este enfoque, no emplea en su primera etapa una estrategia inspirada en ENN, sino una más agresiva que además de ruido elimina otros ejemplos que no afectan la clasificación.

Esta estrategia, denominada CC (class conditional), se basa en el algoritmo de la figura 2.10, donde:

- El método *scoreCompute*(T) asigna a cada ejemplo $e \in T$ un valor $e.\text{score}$ que por simplicidad³ asumiremos como $\text{Hit}(e) - \text{Miss}(e)$
- $\epsilon^H = |\{e \mid e \in T \wedge 1\text{NN}(e, H - \{e\}) \neq \text{class}(e)\}|$

En esencia, lo que se hace es primeramente eliminar todos los ejemplos e con $\text{Hit}(e) < \text{Miss}(e)$, tomando esto como un indicador de que se trata de ruido. A continuación, partiendo de un núcleo

³La autora emplea una fórmula más sofisticada basada en la divergencia de Kullback-Leibler [72] que coincide con $\text{Hit}(e) - \text{Miss}(e)$ en el signo y la monotonía.

```

function CC(T)
begin
  scoreCompute(T)
  //Ordenar descendientemente por score
  scoreSort(T)
   $k_0 = \text{Max}(\text{classes}(T), \lceil \epsilon^T / 2 \rceil)$ 
  //Seleccionar los  $k_0$  primeros
   $S = \text{First}(k_0, T)$ 
  //Descartar los de  $\text{score} \leq 0$ 
   $\text{PreS} = T - \{e \mid e.\text{score} \leq 0\}$ 
  while( $\text{PreS} \neq \{\} \wedge \epsilon^S > \epsilon^T \wedge \epsilon^S > \epsilon^{S \cup \{\text{First}(\text{PreS})\}}$ )
     $S = S \cup \{\text{First}(\text{PreS})\}$ 
     $\text{PreS} = \text{PreS} - \{\text{First}(\text{PreS})\}$ 
return S
end

```

Figura 2.10: Algoritmo CC.

inicial que contiene los ejemplos con mayores valores de *score*, el algoritmo va añadiendo los restantes ejemplos en orden decreciente de su *score* mientras el error asociado a *S* disminuya y se mantenga superior al de *T*.

CC es aplicado independientemente a un número de pares de clases igual a $\text{classes}(T)$, los pares se seleccionan asociando a cada clase la que le sea más similar de acuerdo a los subgrafos que ambas inducen en *HMN*. La unión de estos subconjuntos obtenidos independientemente, es el resultado que se entrega a la segunda etapa.

Esta segunda etapa, denominada THIN, es la que se ilustra en el algoritmo de la figura 2.11, donde $\text{Miss}^H(e)$ y $\text{Hit}^H(e)$ son equivalentes a $\text{Miss}(e)$ y $\text{Hit}(e)$ respectivamente, pero calculados a partir del *HMN* correspondiente a *H*.

THIN inicialmente añade a *S* los ejemplos con $\text{Miss}(\cdot) > 0$ por ser cercanos a las fronteras de clases. Lo que continúa es un bucle donde se añaden nuevos ejemplos a *S* si cumplen la misma condición en el conjunto de los no añadidos y en la iteración anterior no eran puntos aislados ($\text{Miss}^{S_{\text{prev}}}(e) + \text{Hit}^{S_{\text{prev}}}(e) > 0$), esto siempre que el error asociado al nuevo *S* sea mejor que el anterior, de lo contrario se detiene el proceso.

La autora compara el desempeño de CCIS sobre 22 bases de datos con los de ICF y DROP3, mostrando resultados ligeramente superiores a ambos en precisión, similares al primero en reducción y ligeramente mejores que el segundo en este parámetro. No se encontraron trabajos de otros autores donde se incluya a este método.

2.2.4. Condensación con Filtrado de Ruido Embebido

Aunque la aplicación de un filtro de ruido como etapa inicial en la selección de ejemplos es un enfoque simple y efectivo, no es esta la única alternativa que se ha empleado para incorporar filtros de

```

function THIN( $T$ )
begin
   $S = \{e \in T \mid Miss(e) > 0\}$ 
   $S_{prev} = T$ 
   $S_1 = T - S$ 
  do
     $S_T = \{e \in S_1 \mid Miss^{S_1}(e) > 0 \wedge Miss^{S_{prev}}(e) + Hit^{S_{prev}}(e) > 0\}$ 
     $go\_on = \epsilon^{S \cup S_T} < \epsilon^S$ 
    if( $go\_on$ )
       $S = S \cup S_T$ 
       $S_{prev} = S_1$ 
       $S_1 = T - S$ 
    while( $go\_on$ )
  return  $S$ 
end

```

Figura 2.11: Algoritmo THIN.

ruido a este tipo de métodos. A continuación presentamos dos propuestas que a lo largo del proceso de selección eliminan simultáneamente ejemplos ruidosos e irrelevantes. La primera de ellas data de hace dos décadas mientras la segunda fue publicada recientemente. En cuanto a su desempeño, estas son muy similares a las anteriormente vistas. Se caracterizan por retener los ejemplos cercanos a las fronteras entre las clases, presentar una buena tolerancia al ruido y lograr altos niveles de reducción.

2.2.4.1. IB3

Este algoritmo se presenta en [3] como parte de una trilogía donde la primera propuesta (IB1) es en esencia una versión del propio 1NN y la segunda (IB2) es muy similar a CNN. IB3 como novedades incorpora un criterio de utilización para determinar cuales de los ejemplos preseleccionados deben ser empleados en la clasificación de nuevos ejemplos y un criterio de eliminación para desechar ejemplos preseleccionados con muy mal desempeño en la clasificación. Con estas estrategias se pretende atenuar la sensibilidad al ruido que sufren enfoques como CNN o IB2.

En la figura 2.12 puede apreciarse este algoritmo. Al igual que en CNN, los ejemplos correctamente clasificados son descartados automáticamente. Sin embargo la diferencia fundamental es que los no descartados pasan a un estado de “preseleccionado”, dependiendo su posterior uso o eliminación de los criterios antes mencionados.

El criterio de utilización es implementado por medio de la función *acceptable()*, que retorna el conjunto de ejemplos preseleccionados que han tenido un porcentaje de aciertos en la clasificación de nuevos ejemplos hasta el momento significativamente mejor que el porcentaje observado para su clase. Para determinar la pertenencia a este conjunto, se lleva un contador de la cantidad de nuevos ejemplos bien clasificados y mal clasificados por cada ejemplo almacenado. Como puede apreciarse, el algoritmo da preferencia al uso de estos ejemplos para la clasificación y sólo se acude al resto cuando no existe ninguno acceptable.

```

function IB3( $T$ )
begin
   $S = \{\}$ 
  foreach( $x \in T$ )
    if( acceptable()  $\neq \{\}$ )
       $best = y \mid y \in \textit{acceptable}() \wedge \textit{dist}(x, y) = \textit{Min}_{z \in \textit{acceptable}()}(\textit{dist}(x, z))$ 
    else
       $best = \textit{randomSelect}(S)$ 
    if (class( $x$ )  $\neq$  class( $best$ ))
       $S = S \cup \{x\}$ 
    foreach(  $y \in S$ )
      if( $\textit{dist}(x, y) \leq \textit{dist}(x, best)$ )
        updateRecord( $y, x$ )
      if(significantlyPoor( $y$ ))
         $S = S - \{y\}$ 
  return  $S$ 
end

```

Figura 2.12: Algoritmo IB3.

Por su parte la función *updateRecord*(y, x) se encarga de comparar las categorías de ambos ejemplos, si son iguales incrementa el contador de ejemplos bien clasificados por y , de lo contrario incrementa el de ejemplos mal clasificados. Una vez actualizado el contador correspondiente se verifica el criterio de eliminación, si el desempeño de y en la clasificación de nuevos ejemplos hasta el momento es significativamente peor (*significantlyPoor*(y)) que el observado para su clase, este es eliminado del conjunto de ejemplos almacenados.

En resumen, IB3 emplea en la clasificación de nuevos ejemplos, aquellos ejemplos cuya precisión en la clasificación es significativamente mayor que la frecuencia observada para su clase y elimina los que tienen una precisión significativamente menor. Para determinar si utilizar un ejemplo, eliminarlo o posponer la decisión, los autores emplean una prueba de proporciones con una confianza de 90 % para la utilización y un 75 % para la eliminación.

Con este método los autores logran una aceptable tolerancia al ruido. En su favor puede decirse además que presenta uno de los niveles de escalabilidad más altos entre los métodos de selección de ejemplos más reconocidos [27]. IB3 ha sido y es aún, uno de los métodos de selección de ejemplos más ampliamente citados y empleados en comparaciones por los autores de esta área [111, 117, 24, 45, 48].

2.2.4.2. NRMCS

En [109] se presenta NRMCS, un método inspirado en MCS [33] pero que a diferencia de este, no busca un S consistente con T . Este método incorpora un filtro de ruido tratando de hacer frente así a la principal limitación de los enfoques que buscan la consistencia.

Entre la características distintivas de NRMCS destacan que su filtro de ruido es menos propenso a

```

function NRMCS( $T$ )
begin
   $NS = T$ 
  do
     $S = NS$ 
    //Cómputo de votos
    foreach( $e \in T$ )
       $e.voters = \{\}$ 
    foreach( $e \in T$ )
       $e.votes = \{x \mid x \in T \wedge class(x) = class(e) \wedge dist(x, e) \leq dist(NUN(e, S), e)\}$ 
      foreach( $v \in e.votes$ )
         $v.voters = v.voters \cup \{e\}$ 
    //Incremento consistente
     $Consistent = \{x \in T - S \mid \forall(y \in T)[class(x) = class(y) \vee dist(x, y) \geq dist(NUN(y, S), y)]\}$ 
     $C = S \cup Consistent$ 
    //Filtro de ruido
     $Noise = \{x \in C \mid x.votes = \{x\} \wedge x.voters = \{x\}\}$ 
     $C = C - Noise$ 
     $NS = \{\}$ 
    //Construcción de nuevo conjunto
    while( $|C| > 0$ )
       $best = e \mid e \in C \wedge |e.voters| = Max_{x \in C}(|x.voters|)$ 
       $NS = NS \cup \{best\}$ 
       $C = C - best.voters$ 
      foreach( $e \in T$ )
         $e.voters = e.voters - best.voters$ 
    while( $|NS| < |S|$ )
  return  $S$ 
end

```

Figura 2.13: Algoritmo NRMCS.

eliminar ejemplos cercanos a las fronteras de clases que el clásico enfoque basado en ENN y que este se aplica dentro del proceso iterativo, lo que lo diferencia de la mayoría de los enfoques, donde el filtrado del ruido suele ser una etapa previa e independiente.

Al igual que MCS, este método basa su criterio de selección en el concepto de NUN , equivalente al de enemigo más cercano. Como puede apreciarse en la figura 2.13, este es un algoritmo iterativo donde cada iteración consta de cuatro etapas y que se repite mientras el último conjunto seleccionado sea menor que el previo.

La primera etapa dentro de cada iteración consiste en el cómputo de votos. Los votos asociados a cada ejemplo e cuantifican el número de ejemplos de la clase de e , para los que este se encuentra más cercano que su correspondiente NUN en el conjunto seleccionado en la iteración anterior (inicialmente T).

En la segunda etapa, se crea un conjunto de candidatos uniendo al conjunto seleccionado en la

iteración anterior los ejemplos que no crean inconsistencias, para lo que se emplea como criterio que no estén mas cerca a ningún miembro de otra clase que el NUN de este en S .

La siguiente etapa es el filtro de ruido, que elimina los candidatos que sólo reciben y otorgan votos a sí mismos. Esto es un indicador de que están rodeados de ejemplos de clases diferentes.

Finalmente se construye una nueva selección a partir de los candidatos restantes. Para esto, en un bucle que se repite hasta que no queden candidatos por procesar se selecciona al ejemplo más votado y se descartan sus votantes así como los votos otorgados por estos a terceros.

En su artículo, los autores comparan el desempeño de este método con los de MCS e ICF en 8 bases de datos y los resultados son prometedores. Aunque es un enfoque muy interesante, no se encontraron otros trabajos donde se evalúe el desempeño de NRMCS.

2.2.5. Generación de Prototipos

Tal como definimos en la sección 2.1, los ejemplos contenidos en S no tienen necesariamente que formar parte de T . Existe una familia de métodos conocidos indistintamente como “constructivos”, “creativos” o de “generación de prototipos”, que crean nuevos ejemplos en lugar de simplemente seleccionar algunos de los de T . En [105] se presenta un abarcador estudio sobre este tipo de métodos, que incluye una taxonomía y un estudio experimental donde se compara el desempeño de varios representantes de esta familia.

La principal ventaja de la generación de prototipos es que posibilita obtener soluciones inalcanzables por medio de la selección pura, pues no siempre los ejemplos más representativos están contenidos en T . Entre sus limitaciones está que, para algunos dominios como los relacionados con la medicina, construir ejemplos no reales puede resultar una práctica que conduzca a resultados carentes de sentido. Por otro lado, el aumento del espacio de búsqueda que conlleva la generación de prototipos, da como resultado generalmente métodos con un alto costo computacional.

A continuación se describen tres representantes de esta familia basados en diferentes enfoques. El primero de ellos es la primer propuesta de generación de prototipos publicada en la literatura especializada, el segundo es una propuesta que se basa en el conocido LVQ3 [64] y logra un buen equilibrio entre el costo computacional y la calidad de los resultados, mientras el último es una propuesta muy reciente que emplea una matriz de disimilaridad y un conocido algoritmo de serialización.

2.2.5.1. PNN

Chang presentó en [29] un enfoque que en lugar de seleccionar los ejemplos en el sentido estricto de la palabra, los combina para obtener “prototipos” que los representen. El algoritmo PNN (Prototypes for Nearest Neighbor, figura 2.14) busca iterativamente los dos ejemplos más próximos entre sí tales que uno esté en S mientras el otro no haya sido procesado aún. Si sus clases son diferentes, adiciona el no procesado a S . Si coinciden, a partir de ellos obtiene uno nuevo que es el promedio ponderado de sus respectivos atributos. En este caso, si la sustitución del ejemplo tomado de S por el nuevo

ejemplo no incrementa el error de S , esta se lleva a cabo. De lo contrario el ejemplo hasta ahora no procesado se adiciona a S .

Este procedimiento a su vez se repite iterativamente retomando en cada iteración a S como conjunto no procesado y se detiene cuando en una iteración no ocurran nuevas transformaciones.

Aunque PNN es referenciado por muchos autores, no abundan estudios experimentales que lo incluyan. No obstante, los resultados que se muestran en [59], [60] y [105] no sugieren que sea uno de los enfoques más efectivos de esta familia. Adicionalmente, su costo computacional de $O(n^3)$ lo limita al trabajo con bases de ejemplos relativamente pequeñas como se verifica en [105].

2.2.5.2. HYB

En [60] Kim y Oommen proponen una técnica (HYB) consistente en procesar la salida de algún método de selección de ejemplos usando uno de los métodos LVQ (Learning Vector Quantization) [64].

LVQ es una conocida familia de algoritmos de aprendizaje competitivo. Básicamente, estos métodos parten de un número inicial fijo de ejemplos (code-books) que actualizan en un proceso iterativo, desplazándolos hacia posiciones más propicias para la clasificación. En cada iteración, un nuevo ejemplo del conjunto de entrenamiento se toma como entrada, el método LVQ1 actualiza el code-book más cercano, mientras en el caso de LVQ2 y LVQ3 se actualizan dos code-books. Este último es el más elaborado y efectivo de estos métodos, y el escogido por Kim y Oommen para implementar su propuesta.

En LVQ3 los dos code-books (m_i y m_j) más cercanos al ejemplo de entrada e , se actualizan si e está contenido en una región llamada ventana. Condición que se verifica a partir de la siguiente expresión:

$$\text{Min} \left(\frac{d_i}{d_j}, \frac{d_j}{d_i} \right) > \left(\frac{1-w}{1+w} \right)$$

donde w es un parámetro conocido como ancho de ventana, $d_i = \text{dist}(e, m_i)$ y $d_j = \text{dist}(e, m_j)$.

En caso de cumplirse la condición anterior, si $\text{class}(e) = \text{class}(m_j)$ y $\text{class}(e) \neq \text{class}(m_i)$, los code-books se actualizan de la siguiente manera:

$$m_i = m_i - \alpha[e - m_i]$$

$$m_j = m_j + \alpha[e - m_j]$$

donde α es un parámetro conocido como factor de aprendizaje.

En cambio, si $\text{class}(e) = \text{class}(m_j) = \text{class}(m_i)$, la actualización es la siguiente:

$$m_i = m_i + \varepsilon\alpha[e - m_i]$$

$$m_j = m_j + \varepsilon\alpha[e - m_j]$$

donde ε es un parámetro conocido como factor de aprendizaje relativo.

```

function PNN(T)
begin
  Tnp = T
  foreach( e ∈ Tnp)
    We = 1
  do
    merge = false
    e = randomSelect(Tnp)
    S = {e}
    Tnp = Tnp - {e}
    while(Tnp ≠ {})
      //buscar los dos ejemplos más próximos entre sí tales que uno esté en S y el otro en Tnp
      {s, t} = selectClosest(S, Tnp)
      if(class(s) ≠ class(t))
        S = S ∪ {t}
      else
        Wst = Ws + Wt
        st = (s * Ws + t * Wt) / Wst //promedio ponderado
        //Porcentaje de error al clasificar los ejemplos de T con 1NN a partir de S
        es = 1NNError(T, S)
        //Porcentaje de error al clasificar los ejemplos de T con 1NN a partir de S ∪ {st} - {s}
        est = 1NNError(T, S ∪ {st} - {s})
        if(est > es)
          S = S ∪ {t}
        else
          S = S ∪ {st} - {s}
          merge = true
      Tnp = Tnp - {t}
    Tnp = S
  while(merge)
return S
end

```

Figura 2.14: Algoritmo PNN.

Finalmente, si ambos code-books son de clase distinta a la de e ninguno es actualizado.

Como se aprecia, son varios los parámetros involucrados en LVQ3, de hecho los valores óptimos para estos no son los mismos en todas las bases de datos. Para sacar el mejor partido a este método, HYB aprende esos valores óptimos de la siguiente manera. Primeramente divide T en dos conjuntos, uno llamado “colocación” y otro “optimización”. A partir del primero se entrenan los code-books (obtenidos a partir de algún método de selección de ejemplos) con diferentes valores de los parámetros de LVQ3 y se prueba la calidad de estos clasificando los ejemplos del conjunto “optimización”. Finalmente se toman los valores que mejor precisión arrojaron y se entrenan los code-books con T usando esos valores.

Aunque no abundan los trabajos que incluyen este método, en [105] puede apreciarse que ofrece un buen equilibrio entre costo computacional y calidad de los resultados, lo que lo hace especialmente interesante dentro de esta familia caracterizada por los altos costos computacionales.

2.2.5.3. ISPA

ISPA (Instance Seriation for Prototype Abstraction) [84], es una propuesta muy reciente en la que sus autores aplican la serialización al campo de la selección de ejemplos. Esta es una técnica aplicada con éxito en diversos campos, que consiste en el ordenamiento de los datos atendiendo a alguna función de disimilaridad con el objetivo de revelar secuencias estructurales subyacentes.

Como muchos de los algoritmos propuestos en los últimos tiempos, ISPA cuenta con una etapa de eliminación de ruido consistente en la aplicación de ENN. El siguiente paso es crear una matriz de disimilaridad empleando la distancia euclídea como función de disimilaridad. Esta matriz, de orden N^2 , contiene la distancia entre cada par de ejemplos en T' ; donde T' es el conjunto resultante de la aplicación de ENN a T . A continuación se aplica el algoritmo VAT [15] de serialización. Del par de ejemplos con mayor disimilaridad, se toma uno cualquiera como elemento inicial. Así, en un proceso iterativo se reordenan los ejemplos atendiendo a su similaridad con respecto al último ejemplo procesado.

Este reordenamiento da como resultado que los ejemplos consecutivos tiendan a ser muy cercanos en el espacio vectorial. ISPA aprovecha esta característica para mezclar los ejemplos adyacentes de la misma clase mediante un promedio ponderado donde los más cercanos a la frontera tienen asociados los mayores pesos. Así, si x_i y x_{i+1} son ejemplos consecutivos de una misma clase, ambos son sustituidos por un nuevo ejemplo z_j generado por medio de la siguiente ecuación:

$$z_j = \frac{w_i x_i + w_{i+1} x_{i+1}}{w_i + w_{i+1}}$$

donde w_i y w_{i+1} son los pesos asociados a x_i y x_{i+1} respectivamente y se calculan del siguiente modo:

$$w_i = \exp\left(\frac{-r_i^2}{2\sigma}\right)$$

donde $r_i = \text{dist}(x_i, ne(x_i))$ y σ es un parámetro de “suavizado”.

Según los experimentos publicados por sus autores [84], ISPA ofrece buenos resultados. Estos se caracterizan por una reducción bastante alta y una precisión comparable a la de los otras propuestas recientes en este campo. Sin embargo, consideramos que sus demandas de memoria en el orden de N^2 lo hacen poco escalable y por tanto limitan seriamente su empleo en la práctica.

2.2.6. Enfoques Meta-heurísticos

Las meta-heurísticas son técnicas de optimización que exploran el espacio de soluciones guiándose por alguna medida de calidad con la que evalúan los estados por los que transitan. Al igual que las heurísticas no garantizan encontrar el óptimo, lo que se compensa con su capacidad para trabajar en espacios de soluciones muy grandes donde la búsqueda exhaustiva es impracticable. A diferencia de las heurísticas, en las meta-heurísticas se asume muy poco o nada acerca del problema. El número de meta-heurísticas existentes es bastante grande y continúa en crecimiento. Entre las más conocidas se encuentran la búsqueda local, el enfriamiento simulado, la búsqueda tabú y los algoritmos genéticos, por sólo citar algunos. En [101] se puede encontrar un amplio compendio de diversas técnicas meta-heurísticas precedido de un análisis de las bases conceptuales comunes a estas.

Varios enfoques meta-heurísticos han sido aplicados con bastante éxito en la selección de ejemplos. Gracias a su capacidad para explorar el espacio de soluciones, han demostrado excelentes resultados en cuanto a la reducción del número de ejemplos y la calidad de los modelos que de estos se obtienen. En este sentido sobresalen los enfoques evolutivos, acerca de cuya aplicación al campo de la selección de ejemplos puede encontrarse un abarcador estudio en [36]. En general, la principal limitación de las meta-heurísticas en este campo es un alto costo computacional que los pone en desventaja con respecto a otros enfoques a la hora de su aplicación práctica. A continuación se presentan varios representantes de esta familia.

2.2.6.1. Explore

Explore [23] es un método basado en una de las meta-heurísticas más simples, el ascenso de colinas, la que es implementada de un modo bastante peculiar que incluye elementos como múltiples etapas y mutaciones. Este método busca un subconjunto de ejemplos que minimice el valor de la siguiente función de costo:

$$COST(|S|, |T|, x) = F(|S|, |T|) + |S| * \log_2(C) + F(x, |T| - |S|) + x * \log_2(C - 1)$$

Donde x es el número de ejemplos procesados que son mal clasificados por S , C es el número de clases en S , mientras $F(m, n)$ es el costo asociado a la retención de m ejemplos de entre n posibles y se define de la siguiente manera:

$$F(m, n) = \log^* \left(\sum_{j=0, m}^n \left[\frac{n!}{(j!(n-j)!)} \right] \right)$$

Siendo $\log^*(x)$ la suma de los términos positivos de la serie:

$$\log_2(x), \log_2(\log_2(x)), \log_2(\log_2(\log_2(x))), \dots$$

El algoritmo transita por tres fases consecutivas: crecimiento, depuración y mutación. En la fase de crecimiento, cada ejemplo e en T se agrega a S sólo si el valor de $COST$ asociado a $S \cup \{e\}$ y T , es menor que el asociado a S y T . La depuración consiste en eliminar aquellos ejemplos de S , cuya exclusión reduce el valor de $COST$. Por su parte la mutación es una fase donde se llevan a cabo 1000 iteraciones. Cada una consiste en la adición de un ejemplo a S , la exclusión de un ejemplo de S o el intercambio de un ejemplo de S por uno de $T - S$. En todos los casos el cambio es deshecho si conduce a un incremento del valor de $COST$.

Aunque es un método poco empleado en estudios comparativos, los resultados de Explore en reducción y precisión han sido realmente buenos en los escasos trabajos que lo incluyen [51, 45, 111].

2.2.6.2. Búsqueda Tabú

La búsqueda tabú [49] es una conocida meta-heurística cuyo principio consiste en llevar a cabo una búsqueda local en un entorno donde las vecindades se modifican de acuerdo a la evolución de la propia búsqueda. Para lograr esto, se memoriza un conjunto de estados (lista tabú) a los que no se puede acceder. Los estados que componen la lista tabú son, en la versión más simple del método, los últimos visitados. Sin embargo pueden seguirse criterios más específicos en función del problema concreto, en estos casos cada estado visitado adiciona una condición a la lista y se consideran tabú todos los estados que la cumplan.

En algunos casos, es deseable que se ignore la condición tabú de determinado estado, por ejemplo porque este sea mejor que el mejor candidato visitado. Con este fin se definen condiciones conocidas como criterios de aspiración, las que de cumplirse por algún estado tabú, permitirían que el método lo visite a pesar de su condición.

Como norma, la lista tabú funciona como una cola y memoriza un número máximo de estados (o condiciones). Esto hace que al llenarse, cada nuevo estado que ingresa, permite la salida del más antiguo. La búsqueda concluye cuando se satisfaga determinado criterio de parada, como puede ser alcanzar un máximo de iteraciones sin mejora, o de iteraciones totales.

Algunos autores han empleado la búsqueda tabú para la selección de ejemplos. Cerverón y Ferri [28] proponen un esquema en el que la vecindad de la solución actual son aquellas soluciones que pueden obtenerse con la adición o supresión de un ejemplo de esta. La lista tabú que emplean contiene los últimos ejemplos adicionados o suprimidos de S . De este modo cualquier solución vecina de la solución actual se considera tabú si para acceder a ella hay que añadir o suprimir uno de estos ejemplos. Como criterio de aspiración se emplea la superación del mejor candidato y para evaluar la bondad de las soluciones candidatas emplean la siguiente función:

$$fitness(s) = acc(S) - \alpha|S|$$

donde $acc(S)$ es el porcentaje de ejemplos de T correctamente clasificados por la solución actual y α es un parámetro que toma valores positivos y permite establecer el peso que se le pretenda dar a la reducción.

Para obtener el conjunto S inicial, estos autores parten de una selección aleatoria de ejemplos de cada clase que es incrementada por medio de una búsqueda tabú donde se ha deshabilitado el operador de eliminación de ejemplos.

En otra propuesta, Zhang y Sun [115] emplean una búsqueda tabú para encontrar el menor subconjunto cuyo error sea inferior a determinado umbral establecido por un parámetro. En correspondencia con la restricción establecida, la vecindad de la solución actual depende de si esta sobrepasa el umbral de error admitido. En este caso está constituida por aquellas soluciones que pueden obtenerse con la adición de un ejemplo y que tengan un error menor que la actual. Si por el contrario, el error de esta no sobrepasa el umbral, integran la vecindad aquellas soluciones que pueden obtenerse con la supresión de un ejemplo.

Otra diferencia de esta propuesta con respecto a la anterior, es que su lista tabú contiene los últimos subconjuntos candidatos. Como similitudes, tenemos que el conjunto inicial se obtiene a partir de una selección aleatoria o bien se parte de un conjunto vacío, y como criterio de aspiración se emplea la superación del mejor candidato.

En ambos casos descritos, los resultados publicados por sus autores son buenos en reducción y precisión aunque los experimentos son escasos y no se encontraron estudios publicados por otros autores que los incluyan. Como aspecto negativo, el costo en tiempo de ambas propuestas es bastante alto en comparación con métodos convencionales.

2.2.6.3. Algoritmos genéticos

Los algoritmos genéticos [97] agrupan una familia de meta-heurísticas inspiradas en la teoría de la evolución de las especies. En ellos, un conjunto (población) de soluciones candidatas (individuos) evolucionan a lo largo de sucesivas iteraciones (generaciones) por medio de operadores genéticos. Los operadores más comunes son la mutación (variaciones aleatorias), la selección (elección de individuos) y el cruce (obtención de nuevos individuos mezclando otros). El grado de ajuste o bondad de la solución representada por cada individuo se mide por medio de una función de ajuste, y la probabilidad que tendrá cada individuo de sobrevivir o participar en la creación de nuevos individuos es proporcional al valor que dicha función le otorgue.

En [24] se estudia el desempeño en selección de ejemplos de cuatro tipos de algoritmos genéticos; GGA (generacional), SGA (estacionario), CHC (búsqueda adaptativa) y PBIL (aprendizaje incremental basado en poblaciones). Los que mostraron resultados que pueden considerarse buenos tanto en reducción como en precisión.

El enfoque seguido consiste en que cada individuo represente un subconjunto de T . Para esto, se emplean vectores donde cada componente indica la presencia o no de un elemento de T en la solución candidata. Como función de ajuste, se empleó una combinación lineal de la precisión y la reducción asociadas al subconjunto que cada individuo representa. Mientras la reducción se puede

calcular directamente, la precisión debe ser estimada, para lo que se usa el porcentaje de aciertos de 1NN en T empleando S como base de conocimiento.

La limitación fundamental de esta clase de métodos es que requieren llevar a cabo un volumen muy grande de cálculos y necesitan además grandes cantidades de memoria para almacenar las poblaciones. Esto provoca que sean muy costosos en términos de tiempo computacional cuando se comparan con enfoques tradicionales. Adicionalmente, cuando el tamaño de los individuos es muy grande, los algoritmos genéticos experimentan problemas para converger a buenas soluciones.

2.2.6.4. SSMA

SSMA [45] es otra propuesta evolutiva. En este caso se trata específicamente de un algoritmo memético [67], donde se combina la computación evolutiva con búsquedas locales para optimizar los nuevos individuos en cada población. Este tipo de algoritmos suelen implementarse empleando una estrategia evolutiva estacionaria, que es aquella donde en cada población sólo cambian a los sumo uno o dos individuos con respecto a la población anterior.

La codificación de los individuos y la función de ajuste empleadas en SSMA son las mismas propuestas en [24] y que fueron descritas con anterioridad en este epígrafe. Para la selección se emplea el método del torneo (binario en este caso), el cruce intercambia aleatoriamente la mitad de los bits entre los padres y se introduce una mutación aleatoria también. Por su parte la estrategia de reemplazo consiste en sustituir los dos peores individuos de la población si los nuevos representan mejores soluciones.

Para evitar los cálculos masivos que implica la evaluación de la función de ajuste, en la búsqueda local este método emplea lo que los autores denominan “evaluación parcial”. Esta se sustenta en que dicha búsqueda se lleva a cabo eliminando en cada paso un ejemplo de la solución candidata a optimizar, de modo que sólo los ejemplos de T que tenían al eliminado como vecino más cercano en S verán alterada su clasificación. Así, almacenando el vecino de cada ejemplo se puede determinar eficientemente cuales son los afectados con el cambio y si se ha visto perjudicada o beneficiada su clasificación.

En los resultados experimentales publicados por los autores, SSMA aventaja a varios métodos clásicos y evolutivos en precisión y reducción. Además mejora significativamente los tiempos de ejecución de los otros enfoques evolutivos, aunque sigue siendo mucho más costoso que los métodos clásicos. No se encontraron estudios de otros autores donde se incluya este método

2.2.6.5. CoCoIS

CoCoIS⁴[48] es otra propuesta evolutiva. Se trata de un método basado en la coevolución cooperativa, una estrategia donde varias soluciones parciales de un problema complejo evolucionan simultáneamente de un modo tal que pueden cooperar para que el sistema llegue a una solución

⁴Los autores llaman a este método CCIS, pero para diferenciarlo del presentado previamente con el mismo nombre nos referiremos a él como CoCoIS (Cooperative Coevolutionary Instance Selection)

global.

En esta propuesta, T se divide en varios subconjuntos disjuntos (estratos) y para cada uno de ellos hay una población de individuos que evolucionará en busca de la mejor solución parcial. La representación de cada solución parcial candidata es la propuesta en [24]. Sin embargo la función de ajuste incluye, además de los convencionales criterios de reducción y precisión, uno nuevo que mide el efecto de la solución parcial en la general.

La manera en que estas soluciones parciales cooperan se basa en el mantenimiento de una población de soluciones generales que evolucionan simultáneamente a las anteriores. Cada una de estas representa una combinación de las soluciones parciales (una por cada estrato) y a su vez es evaluada por una función de ajuste, que en este caso si es la clásica que combina reducción y precisión.

La presentación de CoCoIS en [48] se complementa con un extenso estudio experimental donde su desempeño en reducción es superior al de reconocidos métodos del estado del arte mientras su precisión es similar a las mejores de estos. Aunque los autores plantean que es más eficiente computacionalmente que propuestas previas basadas en técnicas evolutivas, algunas pruebas que hemos desarrollado con la implementación disponible en la herramienta KEEL [5] evidencian tiempos de ejecución prohibitivos. Sin embargo es justo reconocer que las mayores potencialidades de este método en cuanto a eficiencia computacional radican en la facilidad de adaptarlo para su ejecución en arquitecturas distribuidas.

2.2.6.6. Evolución Diferencial

La evolución diferencial [87] es otra técnica evolutiva empleada en la selección de ejemplos. En este tipo de algoritmos los individuos son vectores de componentes reales. En cada generación se crea por cada individuo (vector objetivo) un vector “mutante” que a su vez es combinado con el vector objetivo para obtener un vector candidato. Finalmente, pasa a la siguiente generación el que tenga el mejor valor asociado a la función ajuste.

En [106] se propone el empleo de la evolución diferencial para la selección de ejemplos. Específicamente, los autores presentan un esquema de generación de prototipos. En este, al igual que en los otros enfoques evolutivos previamente comentados, cada individuo codifica una solución candidata. Pero a diferencia de estos, no se emplea la representación propuesta en [24]. En este caso cada individuo codifica r ejemplos, siendo r un parámetro definido por el usuario que determinará el número de ejemplos en la solución. Otra diferencia fundamental es que los individuos son matrices de r filas y $M + 1$ columnas, donde cada fila contiene los M atributos de un ejemplo y su clase.

La propuesta se acompaña por un estudio experimental donde se incluyen varias estrategias de mutación y cruce. Los resultados de este estudio, cuyos autores catalogan de preliminar, sugieren la pertinencia de esta técnica para la generación de prototipos.

2.2.7. Enfoques Basados en Agrupamiento

El agrupamiento, a pesar de ser una técnica más propia del aprendizaje no supervisado que del supervisado, ha sido con frecuencia empleado en la selección de ejemplos para este último. Este enfoque está estrechamente relacionado con el de generación de prototipos, aunque no puede afirmarse que esté contenido en aquel. Generalmente consiste en retener los centroides de cada cluster y se caracteriza por altas reducciones, aunque como se podrá comprobar mediante los tres representantes de esta familia que describimos a continuación, no siempre es así.

El primero de los métodos que describimos se basa en el agrupamiento independiente de las clases, retiene los centroides y por medio de un parámetro puede controlarse el nivel de reducción que produce. El segundo agrupa simultáneamente todos los ejemplos aunque garantiza que cada cluster corresponda a solo una clase, retiene los centroides y logra altas reducciones. Mientras el último, aplica un agrupamiento convencional donde cada cluster puede contener ejemplos de diversas clases, retiene ejemplos centrales y fronterizos, se caracteriza por ser muy rápido y no logra grandes reducciones.

2.2.7.1. NSC

NSC [108] es un método basado en el algoritmo de agrupamiento MVC [107]. Este último busca minimizar el error cuadrático de cada ejemplo con respecto a la media del cluster que lo contiene. Error que se determina mediante la siguiente ecuación:

$$\frac{\sum_{i=1}^M H(C_i)}{N}$$

donde M es el número de clusters, $C_1..C_M$ los clusters y $H(A)$ es la homogeneidad del cluster A definida como:

$$H(A) = \sum_{x \in A} |x - \mu(A)|^2$$

donde $\mu(A)$ es la media en A .

Para evitar que cada ejemplo esté contenido en un cluster unitario, se impone la siguiente restricción:

$$\forall C_i, C_j, i \neq j : \text{Var}(C_i \cup C_j) \geq \sigma_{max}^2$$

donde $\text{Var}(A)$ es la varianza en A , mientras σ_{max}^2 es un parámetro definido por el usuario.

NSC consiste en aplicar MVC a cada clase por separado, para finalmente conservar en S los centroides de cada cluster. Por medio del parámetro σ_{max}^2 se puede controlar el número de ejemplos en S , siendo los casos extremos: $\sigma_{max}^2 = 0$ donde $S = T$ y $\sigma_{max}^2 = \infty$ donde en S sólo habrá un ejemplo por clase.

Los autores comparan este método con MCS, DROP3 y la búsqueda tabú, reportando buenos resultados. No se encontraron otros resultados experimentales que incluyan este método. Como una debilidad del mismo, consideramos que el agrupamiento por separado de las clases, puede dar lugar a clusters de diferentes clases solapados en el espacio.

2.2.7.2. SGP

SGP [38] es otro método basado en agrupamiento que selecciona el centroide de cada cluster como miembro de S . A diferencia de NSC, en este caso el agrupamiento no se aplica a cada clase por separado.

La estrategia de agrupamiento comienza formando para cada clase un cluster que contiene todos sus ejemplos. Luego en un proceso iterativo se aplican las siguientes reglas para dividir clusters e intercambiar ejemplos:

- Si para todos los ejemplos en un cluster, el centroide más cercano es el del propio cluster, este se mantiene inalterado.
- Si para todos los ejemplos en un cluster, el centroide más cercano es de una clase diferente a la del cluster, este se divide separando los ejemplos por un hiperplano que pasa por el centroide del cluster y que es perpendicular al primer componente principal del mismo.
- Si para algún ejemplo de un cluster, el centroide más cercano es de otro cluster de la misma clase, el ejemplo es movido a ese cluster.
- Si para algunos ejemplos de un cluster, el centroide más cercano es de otro cluster de diferente clase, estos pasan a formar un nuevo cluster.

Estas modificaciones se repiten hasta que no ocurran más cambios en los clusters.

Adicionalmente, se aplica una etapa de mezcla, en la que se combinan dos clusters de la misma clase si el centroide de cada uno de ellos es el segundo más cercano a los ejemplos del otro. Para finalizar, en una etapa de poda, se eliminan los clusters cuyos ejemplos (todos) tienen como segundo centroide más cercano uno de la misma clase.

Para evitar el sobreajuste, los autores introducen dos umbrales. El primero permite eliminar clusters muy pequeños respecto al mayor de todos. Mientras el segundo, permite mantener cierta cantidad de ejemplos mal clasificados en los clusters.

Los resultados experimentales publicados por sus autores son escasos. Sin embargo, en [105], *SGP* aparece como parte de un estudio comparativo entre 24 métodos sobre 59 bases de datos. En este estudio, *SGP* muestra altas reducciones, aunque sus resultados en precisión no están entre los mejores y en bases de datos grandes es el peor de los métodos comparados.

```

function PSC( $T, ClsNumb$ )
begin
   $S = \{\}$ 
   $clusters = clusteringMethod(T, ClsNumb)$ 
  foreach( $cluster \in clusters$ )
    if( $\forall(\{x, y\} \subseteq cluster)[class(x) = class(y)]$ )
       $mean = center(cluster)$ 
       $S = S \cup \{nn(mean, cluster)\}$ 
    else
      foreach( $e \in cluster$ )
         $C_e = \{x \in cluster \mid class(x) = class(e)\}$ 
         $C_m = \{e \in cluster \mid \forall(x \in cluster)|C_e| \geq |C_x|\}$ 
        foreach( $e \in cluster - C_m$ )
           $n_m^e = nn(e, C_m)$ 
           $n_e^m = nn(n_m^e, C_e)$ 
           $S = S \cup \{n_m^e, n_e^m\}$ 
  return  $S$ 
end

```

Figura 2.15: Algoritmo PSC.

2.2.7.3. PSC

En [85] se propone PSC, un método basado en agrupamiento que selecciona tanto puntos centrales como fronterizos y cuya mayor ventaja es ser muy rápido en comparación con la mayoría de los métodos más conocidos en este campo.

La estrategia de selección de PSC se basa en dividir T en varios clusters empleando cualquier método convencional de agrupamiento ⁵. Estos clusters son catalogados en homogéneos (sólo contienen ejemplos de una clase) y no homogéneos, recibiendo cada tipo un tratamiento distinto. En los primeros sólo se selecciona un ejemplo, que será el más cercano al centro del cluster. En los segundos se seleccionan ejemplos fronterizos de la siguiente manera: siendo C_m el conjunto de los ejemplos pertenecientes a la clase mayoritaria en el cluster, para cada ejemplo e tal que $e \notin C_m$, se busca el vecino más cercano en C_m ($n_m^e = nn(e, C_m)$) y se agrega a S , a la vez que se agrega el vecino más cercano de este en la clase de e ($nn(n_m^e, \{x \mid class(x) = class(e)\})$).

En los experimentos publicados por los autores, la reducción que produce PSC es superada claramente por la de DROP3, mientras la precisión obtenida por 1NN con la selección hecha por el primero también es inferior a la correspondiente al segundo, aunque con otros clasificadores la ventaja es a favor del PSC. En este comportamiento debe influir la presencia de puntos centrales que hacen que $|S|$ sea mayor y no aportan nada en el desempeño de 1NN pero pueden ser una ventaja en otros modelos de aprendizaje/clasificación.

⁵Concretamente los autores emplean el conocido k-medias

```

function ARP( $T, k, j, PRS$ )
begin
  if( $|T| \leq k$ )
    return  $PRS(T)$ 
   $TTemp = \{\}$ 
  //dividir  $T$  en  $j$  conjuntos disjuntos
   $Partition = GetPartition(T, j)$ 
  for( $subT \in Partition$ )
     $TTemp = TTemp \cup ARP(subT, k, j, PRS)$ 
  return  $ARP(TTemp, k, j, PRS)$ 
end

```

Figura 2.16: Algoritmo ARP.

2.3. El problema de la escalabilidad. Enfoques fundamentales

La selección de ejemplos es en sí misma una estrategia que permite que bases de datos originalmente grandes sean accesibles para algoritmos de aprendizaje que de otra manera no podrían procesarlas. Sin embargo el ritmo al que crece el tamaño medio de las bases de datos es tal, que hoy en día es frecuente encontrar problemas reales donde ninguno de los algoritmos antes descritos es capaz de manejar el número de ejemplos involucrados.

Aunque algunas de las propuestas más recientes como PSC y CoCoIS llevan implícita la posibilidad de una implementación distribuida, no siempre es factible disponer del hardware que esto implica. Por demás, se trata de dos alternativas concebidas como parte de métodos de selección específicos, por lo que su adaptación a otras estrategias de selección no es directa.

En este epígrafe presentamos algunas propuestas que enfocan el problema de la escalabilidad en la selección de ejemplos desde una perspectiva más general, lo que les permite ser aplicadas con cualquier método de los anteriores.

2.3.1. Particionamiento recursivo

En [61] se propone un método adaptativo recursivo al que sus autores denominan ARP (adaptive recursive partitioning). En esencia este algoritmo ataca el problema de la falta de escalabilidad que presenta la mayoría de sus predecesores empleando una estrategia “divide y vencerás”. ARP no emplea ningún criterio de orden o agrupamiento para generar la partición de la base de ejemplo, sino que lo hace de manera aleatoria.

El algoritmo se detalla en la figura 2.16, como puede apreciarse, más que un método de selección de ejemplos ARP es un esquema para escalar otros métodos. El que en realidad hace la selección es el que se recibe como parámetro PRS (prototype reduction scheme) y que puede ser en principio cualquier método. La función de ARP se limita a particionar recursivamente la base de ejemplos hasta obtener subconjuntos de un tamaño máximo k que sean “manejables” para PRS y construir a partir de los resultados parciales, el resultado general. Evidentemente, la selección de k dependerá

del costo computacional del método concreto de selección y del hardware disponible.

Los autores reportan drásticas reducciones del tiempo de ejecución de algunos algoritmos al ser combinados con ARP. Esto se logra manteniendo los niveles de precisión en valores muy similares a los obtenidos de manera independiente. Resultados que sugieren que esta es una herramienta nada despreciable si de procesar grandes bases de datos se trata. Sin embargo, no se encontraron estudios posteriores que la involucren.

2.3.2. Estratificación

En [25] también se aborda el problema de la escalabilidad en la selección de ejemplos. El enfoque que proponen consiste en la estratificación de la base de datos, dividiéndola en t subconjuntos (estratos) con igual número de ejemplos e igual distribución de clases. Cada estrato D_i es procesado por un método de selección de ejemplos para extraer una solución parcial ($DS_i \subseteq D_i$). La solución general por su parte, se obtiene uniendo todos los conjuntos DS_i ($S = \bigcup_{0 < i < t} DS_i$).

El estudio que presentan los autores involucra varios métodos de selección de ejemplos clásicos y uno genético. En él se muestra que el tiempo de ejecución de todos los métodos de selección de ejemplos involucrados se reduce significativamente. Sin embargo en cuanto a la reducción y la precisión los resultados no son homogéneos, el algoritmo genético en general respondió mejor a la estratificación que los clásicos. Estos resultados se confirman en [26], donde se compara el desempeño de los mismos métodos pero esta vez empleando a C4.5 [90] como clasificador. En [35] también se pone a prueba la eficacia de la estratificación para escalar a SSMA en este caso, aquí una vez más el enfoque evolutivo mostró mejores resultados que los clásicos.

2.3.3. Selección democrática

La selección democrática de ejemplos [46], es una reciente propuesta para escalar este tipo de métodos. En ella, la base de datos se divide en varios subconjuntos disjuntos y cada uno es procesado por un método de selección de ejemplos. Este proceso se repite un número predeterminado de veces (rondas) pero con diferentes particiones de T . En cada ronda a cada ejemplo eliminado se le asigna un voto, al final son eliminados los ejemplos cuyo número de votos supere determinado umbral.

Para construir las particiones, los autores proponen generar diferentes vectores usando el método Grand Tour [10], proyectar los ejemplos en los vectores, ordenarlos y dividirlos de modo que cada subconjunto sea una secuencia de ejemplos adyacentes en dicha proyección. Con esto buscan que haya cierto criterio de localidad en los subconjuntos, tomando como premisa que esto favorece el trabajo del método de selección de ejemplos.

Finalmente, para determinar el umbral de votos requeridos para eliminar un ejemplo, se propone un método que estima el efecto de cada posible valor (entre 1 y el número de rondas) en la reducción y precisión del conjunto resultante. Esta estimación se lleva a cabo sobre una muestra aleatoria de T .

Los autores documentan un extenso estudio del impacto de esta estrategia en cinco métodos de selección de ejemplos sobre 30 bases de datos. Los tiempos de ejecución se reducen significativamente

excepto en algunas bases de datos muy pequeñas. Por otra parte la reducción y la precisión muestran comportamientos favorables aunque muy poco homogéneos tal como ocurre con la estrategia de estratificación.

2.3.4. Consideraciones generales acerca de las propuestas para escalar métodos de selección de ejemplos

La eficiencia de las propuestas anteriores es incuestionable, y el sólo hecho de permitir procesar bases de datos que de otro modo serían intratables es suficiente para que se les tenga en cuenta. No obstante, presentan algunas debilidades que consideramos importante señalar.

En primer lugar, la dependencia de parámetros para cuya elección no existen criterios claros. En este sentido ARP parece ser la menos afectada, pues el tamaño “manejable” se establece a partir de criterios bastante objetivos y el número de divisiones no parece tener mayor incidencia en los resultados. En cambio, sobre esta estrategia asecha una amenaza muy particular, que PRS no sea capaz de reducir a T a menos de k ejemplos, en ese caso caería en una recursividad infinita.

Otro aspecto al que no dan respuesta ni ARP ni la estratificación es que la retención de muchos ejemplos puede depender de la presencia de otros que no coinciden con ellos en el mismo subconjunto. En cambio cuando ocurre lo contrario, o sea que la exclusión de muchos ejemplos dependa de otros con los que no coincidan inicialmente, la estratificación tampoco tiene manera de manejarlo pero ARP conserva oportunidades de eliminarlos posteriormente si estos se mezclan en otro nivel de la recursividad. La estrategia democrática puede lidiar mejor con estas situaciones, pero esto está condicionado a que el número de rondas sea suficiente.

Finalmente, consideramos que el objetivo de cualquier estrategia para escalar métodos de selección de ejemplos debe ser reducir sensiblemente el costo computacional de estos, alterando lo menos posible su desempeño en reducción y precisión. El primer criterio se satisface por cada una de las alternativas descritas, sin embargo ninguna de ellas logra un impacto nulo o estadísticamente poco significativo en los resultados de los métodos de selección de ejemplos.

2.4. Conclusiones Parciales

La recopilación de métodos de selección de ejemplos presentada no pretende ser exhaustiva aunque sí representativa. En ella se evidencia que durante décadas han sido abundantes las propuestas en este campo.

Algo llamativo es que es frecuente encontrar resultados publicados aparentemente contradictorios. Al cambiarse las bases de datos, o incluso la partición de estas, métodos que parecían superar a otros se ven superados por aquellos. Para ilustrar dicho fenómeno hemos escogido los dos métodos que probablemente sean los más referenciados en este campo, IB3 y DROP3. Si observamos 8 estudios documentados en la última década donde coinciden ambos métodos⁶, podemos comprobar que cada

⁶Los estudios referidos están contenidos en [48, 117, 111, 51] y en [45, 24] (dos en cada uno de estos últimos), en

uno de ellos tiene mejores resultados en precisión promedio que el otro en la mitad de los casos.

Esto confirma, a nuestro juicio, algo que en los últimos años varios autores vienen planteando. Nos referimos a que las características del dominio son determinantes en el éxito del método empleado y que un sólo criterio de selección no es suficiente para obtener buenos resultados en cualquier entorno.

Una alternativa para enfrentar esto parece venir de la mano de los enfoques meta-heurísticos, los que implícitamente y gracias su capacidad para explorar el espacio de soluciones, logran aportar soluciones adaptadas a las características de cada dominio concreto. Esto sin embargo, implica un costo computacional prohibitivo que los pone en desventaja con respecto a otros enfoques a la hora de su aplicación práctica.

Una segunda alternativa, en la que basamos nuestro trabajo, es emplear conocimiento acerca del dominio para elegir la estrategia de selección más adecuada entre varias opciones. En esta dirección se mueven muchos de los planteamientos a que nos referíamos antes.

Brighton y Mellish [19] plantean que una estructura de clases homogénea es determinante para el éxito de algoritmos que retienen puntos fronterizos, mientras que ante dominios no homogéneos resulta ventajoso retener prototipos⁷. Wilson y Martínez [111] presentan una comparación de los resultados de varios métodos ante la presencia de ruido en los datos de entrada y concluyen que deben identificarse aún otros factores que determinan el éxito de cada algoritmo para emplear el adecuado en cada contexto. Kim y Oommen [59] comparan varios métodos ante datos artificiales y de la vida real, concluyendo que ninguno es superior a los restantes uniformemente sobre todos los dominios, plantean además que pueden ser determinantes en estos resultados los patrones de agrupamiento, los outliers de la distribución de clases, el número de ejemplos y la proximidad entre las clases. Reinartz [92] presenta como una dirección de trabajo la definición de heurísticas para elegir el método más apropiado en cada caso. Mollineda y otros [80] redefinen para problemas multiclases un conjunto de medidas de la complejidad de los datos previamente presentadas en [55] para problemas binarios, en el mismo trabajo usan estas medidas para contrastar los resultados de *CNN*, *ENN* y una combinación de ambos ante dominios de diferentes complejidades. Luego son retomadas en [44] para predecir cuando es efectivo el uso de un método evolutivo de selección de ejemplos y en [74] para predecir el desempeño de los clasificadores *C4.5* [90] y *PART* [40] en bases de datos desbalanceadas al ser estas preprocesadas por diferentes métodos diseñados para este tipo de problemas.

Todos estos autores coinciden en que es necesario caracterizar de algún modo el dominio y aplicar el método mas adecuado a este. Las medidas para esta caracterización y las técnicas para determinar el método apropiado continúan siendo un problema abierto.

todos los casos se trata de experimentos que involucran varias bases de datos

⁷En este caso, los autores usan el término prototipo para referirse a ejemplos típicos o centrales, aquellos que son más parecidos a sus vecinos. Sin embargo es común encontrar en este campo el mismo término para referirse en general a cualquier ejemplo seleccionado, de hecho muchos autores usan indistintamente “selección de prototipos”, “selección de instancias” y “selección de ejemplos”

Capítulo 3

Una Propuesta de Atributos para la Caracterización de Bases de Datos

Como comentábamos antes, el éxito de una estrategia de selección de ejemplos al procesar una base de datos está muy influenciado por las características de esta última. Esto hace que los métodos que emplean los mismos criterios de selección en todos los problemas, tengan desempeños que varían significativamente en función del dominio en que se apliquen.

En este trabajo, presentamos una alternativa para obtener desempeños estables en una amplia gama de problemas. Esta consiste en emplear conocimiento acerca del dominio para elegir la estrategia de selección más adecuada entre varias opciones. Este conocimiento permitiría estimar el desempeño de cada estrategia candidata y así aplicar la de mejor estimación.

Una manera efectiva de obtener el conocimiento necesario es emplear algún método de aprendizaje automático, lo que se enmarca en el campo del meta-aprendizaje, que se encarga según Sotoca y otros [100] del “*aprendizaje de meta-conocimiento acerca del conocimiento aprendido*”. Esto nos lleva a la necesidad contar con un conjunto de meta-datos que describa el desempeño de distintos métodos de selección en distintos dominios. Lo que a su vez hace necesario poder caracterizar estos últimos de alguna manera.

En este capítulo comentamos algunos trabajos sobre meta-aprendizaje que resultan relevantes para el campo de la selección de ejemplos y presentamos un conjunto de medidas que constituyen nuestra propuesta de atributos para la caracterización de bases de datos. Algunos resultados preliminares relacionados con estas medidas han sido publicados en [20, 21, 22].

3.1. Meta-aprendizaje y medidas de complejidad de los datos

La variedad de métodos de minería de datos que se puede encontrar en la literatura especializada y la ausencia de criterios determinantes que permitan seleccionar uno u otro para un problema concreto, han motivado el interés por desarrollar técnicas de meta-aprendizaje capaces de predecir

el desempeño de un método ante una base de datos, basándose en las características de esta última [55, 96, 103, 13, 14, 8, 17, 75].

Aunque mayormente ha sido enfocado hacia el estudio del desempeño de los clasificadores, algunos autores [100, 99] han presentado el meta-aprendizaje como un marco más general, aplicable a disímiles tareas de la minería de datos entre las que se incluye la selección de ejemplos. Probablemente, entre los trabajos relacionados con el meta-aprendizaje, el que más ha influido en este último campo ha sido el de Ho y Basu [55]. En este se presentan varias medidas de complejidad de los datos para problemas de clasificación binaria, las que posteriormente han sido empleadas en tareas de selección de ejemplos y ampliadas a problemas multiclase por otros autores [80, 44, 74]. En otro trabajo muy relacionado con éste, Sotoca y otros [100] presentan un sistema de medidas de complejidad para el meta-aprendizaje, donde retoman las propuestas de Ho y Basu e incorporan nuevas medidas al conjunto.

Las medidas que forman parte de este sistema se pueden agrupar en cuatro categorías:

- Medidas del solapamiento de los valores de los atributos entre clases diferentes.
- Medidas de la separabilidad de las clases.
- Medidas de geometría y densidad.
- Medidas estadísticas.

A continuación describimos brevemente cada una de ellas.

3.1.1. Medidas del solapamiento de los valores de los atributos entre clases diferentes

Máximo discriminante de Fisher ($F1$). Esta medida se define como:

$$F1 = \text{Max}_{a \in A}(f_a) \quad (3.1)$$

donde A es el conjunto de atributos de la base de datos y f_a es el discriminante de Fisher para el atributo a , que se define a su vez como:

$$f_a = \frac{(\mu_1 - \mu_2)^2}{\sigma_1^2 + \sigma_2^2}$$

siendo μ_1, μ_2, σ_1 y σ_2 las medias y varianzas del atributo a limitado a los ejemplos de las clases 1 y 2 respectivamente.

Posteriormente, en [80] se propone la siguiente generalización de $F1$ para problemas multiclase:

$$F1 = \frac{\sum_{i=1}^C n_i \cdot \text{dist}(m, m_i)}{\sum_{i=1}^C \sum_{j=1}^{n_i} \text{dist}(x_j^i, m_i)} \quad (3.2)$$

donde C denota la cantidad de clases del problema, n_i el número de ejemplos de la clase i , m es un punto tal que en cada atributo toma el valor promedio de los valores de todos los ejemplos, m_i es el equivalente a m restringido a la clase i , y x_j^i representa el ejemplo j de la clase i .

Volumen de la región solapada ($F2$). Esta medida es el producto de las porciones del rango de cada atributo donde se solapan las dos clases, lo que se expresa por medio de la siguiente ecuación:

$$F2 = \prod_{i=1}^{|A|} \left(\frac{\text{Min}(\text{Max}(f_i, c_1), \text{Max}(f_i, c_2)) - \text{Max}(\text{Min}(f_i, c_1), \text{Min}(f_i, c_2))}{\text{Max}(\text{Max}(f_i, c_1), \text{Max}(f_i, c_2)) - \text{Min}(\text{Min}(f_i, c_1), \text{Min}(f_i, c_2))} \right) \quad (3.3)$$

donde $\text{Max}(f_i, c_j)$ y $\text{Min}(f_i, c_j)$ son el máximo y mínimo valor que toma el atributo f_i para la clase c_j respectivamente. Aunque en la expresión no queda explícito, de los comentarios de las autoras se deduce que el numerador de la ecuación 3.3 se fuerza a tomar valor 0 cuando la resta es negativa, lo que correspondería a ausencia de solapamiento.

Aunque $F2$ es originalmente definida para clasificación binaria, luego es extendida a problemas multiclase en [80] por medio de la siguiente expresión:

$$F2 = \sum_{1 \leq i \leq j \leq C} \prod_{k=1}^{|A|} \left(\frac{\text{Min}(\text{Max}(f_k, c_i), \text{Max}(f_k, c_j)) - \text{Max}(\text{Min}(f_k, c_i), \text{Min}(f_k, c_j))}{\text{Max}(\text{Max}(f_k, c_i), \text{Max}(f_k, c_j)) - \text{Min}(\text{Min}(f_k, c_i), \text{Min}(f_k, c_j))} \right) \quad (3.4)$$

que no es más que la suma de los volúmenes de las regiones solapadas correspondientes a todos los pares de clases.

Eficiencia de atributos ($F3$). La eficiencia individual de un atributo para clasificación binaria, es según Ho y Basu la porción de ejemplos que pueden ser clasificados empleando sólo ese atributo. Considerando todos los atributos, $F3$ se define como el máximo valor que toma la eficiencia individual.

Al generalizar esta medida a problemas multiclase, en [80] se invierte el análisis, teniéndose en cuenta los ejemplos que están en al menos una región donde se solapan al menos dos clases en algún atributo.

3.1.2. Medidas de la separabilidad de las clases

Separabilidad lineal ($L1$ y $L2$). Estas dos medidas se basan en la formulación para programación lineal propuesta por Smith [98]. En esta se busca el hiperplano que minimize la suma de las distancias entre él y los ejemplos mal clasificados si se toma este como frontera de decisión, solucionando el siguiente problema de optimización:

$$\begin{aligned} \text{objetivo : } & \text{minimizar } a^t t \\ \text{restricciones : } & Z^t w + t \geq b \\ & t \geq 0 \end{aligned}$$

donde a y b son vectores constantes arbitrarios (a los que Ho y Basu proponen asignar valor 1), w es un vector de pesos, t es un vector de errores y Z es una matriz donde cada columna z se obtiene a partir de un ejemplo x de la siguiente manera:

$$z = \begin{cases} +x, & \text{si } class(x) = c_1 \\ -x, & \text{si } class(x) = c_2 \end{cases}$$

A partir de la solución del problema anterior, se define como $L1$ el valor de la función objetivo (0 indica totalmente separable) normalizado respecto al número de ejemplos y la longitud de la diagonal del menor hiperrectángulo que contiene a todos estos. Se define además como $L2$ el porcentaje de error en T del clasificador obtenido. En el caso de estas medidas no se ha encontrado ninguna propuesta de generalización a problemas multiclase.

Puntos frontera ($N1$). Esta medida se calcula a partir de un árbol de expansión mínimo correspondiente a un grafo donde los vértices son los ejemplos y a cada par de estos los conecta una arista de peso igual a la distancia entre ellos. $N1$ es la porción de ejemplos que en este árbol son adyacentes a un ejemplo de otra clase. Aunque originalmente se propone en un marco donde se abordan sólo problemas binarios, es directamente aplicable a cualquier número de clases.

Dispersión ($N2$). Para estimar cuan relativamente cerca están unas clases de otras, Ho y Basu proponen la razón entre la distancia promedio al vecino más cercano de igual clase y la distancia promedio al enemigo más cercano:

$$N2 = \frac{\sum_{e \in T} (dist(e, nn(e, C_e)))}{\sum_{e \in T} (dist(e, ne(e, T)))} \quad (3.5)$$

donde C_e es el conjunto de ejemplos de la clase de e . Al igual que la anterior medida, esta es directamente aplicable a cualquier número de clases.

Error de 1NN ($N3$). Esta es la última medida de separabilidad de las clases y consiste en el porcentaje error de 1NN en T excluyendo cada ejemplo del conjunto para su propia clasificación (estrategia “leave-one-out”). Puede comprobarse fácilmente que es también aplicable a múltiples clases.

3.1.3. Medidas de geometría y densidad de regiones

Todas las medidas de esta categoría excepto $L3$ son aplicables a problemas multiclase, por lo que no lo especificaremos para cada una por separado.

No linealidad ($L3$ y $N4$). La medida de no linealidad fue originalmente propuesta en [56] y consiste en el porcentaje de error de un clasificador sobre un conjunto de prueba formado por ejemplos obtenidos por interpolación lineal (con coeficientes aleatorios) de pares aleatorios de ejemplos de una misma clase. Ho y Basu proponen como $L3$ la implementación de esta medida empleando el clasificador obtenido al calcular $L1$, y como $N4$ su equivalente empleando 1NN.

Como comentábamos antes, $L3$ sólo es aplicable a problemas binarios, esto está determinado por su definición a partir de un clasificador binario.

Número de adherencias (T1). Para definir esta medida, Ho y Basu se basan en el concepto de adherencia (ad) [70], que se define a través de la siguiente recurrencia:

$$ad(X) = \begin{cases} \{\}, & \text{si } X = \{\} \\ \{x\} \cup \Gamma(x), & \text{si } X = \{x\} \\ \bigcup_{x \in X} ad(\{x\}), & \text{si } |X| > 1 \end{cases} \quad (3.6)$$

donde $\Gamma(x) = \{y \mid yRx\}$, siendo R una relación binaria, que en este caso concreto las autoras establecen como $xRy \Leftrightarrow dist(x, y) < \epsilon$ fijando $\epsilon = 0,55\delta$, con δ igual a la distancia entre los dos ejemplos más cercanos de clases diferentes.

Por medio de la composición se pueden obtener adherencias de diversos órdenes:

$$ad^0(X) = X, ad^1(X) = ad(X), \dots, ad^m(X) = ad(ad^{m-1}(X))$$

Para cada ejemplo $e \in T$, Ho y Basu buscan la adherencia $ad^k(\{e\})$, tal que incluya sólo ejemplos de la misma clase de e y $ad^{k+1}(\{e\})$ incluya algún ejemplo de otra clase. Posteriormente, desechan todas las adherencias totalmente contenidas en otra y el número de las restantes normalizado respecto a N es el valor de $T1$

Ejemplos por dimensión (T2). Esta medida se usa como un indicador de la densidad de ejemplos en relación con el espacio donde estos están definidos y se calcula como:

$$T2 = \frac{N}{|A|} \quad (3.7)$$

Densidad (D1). Esta es una de las nuevas propuestas que aparecen en [100]. Al igual que la anterior es un indicador de la densidad de ejemplos en relación con el espacio donde estos están definidos. Pero en este caso en el denominador lo que aparece es el volumen donde los ejemplos están contenidos. Este volumen se define como el producto de las longitudes de los rangos de valores que toman los atributos en el conjunto de ejemplos.

Volumen promedio de las vecindades locales (D2). Usando el mismo criterio de volumen, pero restringido a los k vecinos más cercanos de un ejemplo, en [100] se define el volumen de la vecindad de un ejemplo e como:

$$V_e = \prod_{h=1}^{|A|} (Max(f_h, N_k(e)) - Min(f_h, N_k(e)))$$

donde $Max(f_h, N_k(e))$ y $Min(f_h, N_k(e))$ son el máximo y mínimo valor que toma el atributo f_h en los k vecinos más cercanos a e .

A partir de esta definición, se establece como $D2$ el valor promedio de V_e -en el conjunto de ejemplos:

$$D2 = \frac{1}{N} \sum_{e \in T} (V_e) \quad (3.8)$$

Densidad en región de solapamiento (D3). Esta es una medida que se propone en [100] para

estimar para cada clase el número de ejemplos que están ubicados en regiones de otras clases. Para esto se emplea la misma estrategia que el método ENN usa para determinar la eliminación de un ejemplo. Si $kNN(e, T - \{e\}) \neq class(e)$, se cuenta a e como ubicado en una región de otra clase.

3.1.4. Medidas estadísticas

Finalmente, Sotoca y otros [100] sugieren varias medidas estadísticas que pueden ser de utilidad para el meta-aprendizaje: el número de ejemplos, de atributos y clases, el coeficiente de correlación entre atributos, la curtosis de atributos, la entropía de las clases, entre otras. Como ejemplo del uso de estas citan el proyecto STATLOG [63], donde varios clasificadores fueron estudiados para tratar de predecir su aplicabilidad basándose en una caracterización de los datos que empleaba este tipo de medidas.

3.2. Nuevo sistema de medidas de complejidad

El anterior sistema de medidas, presenta una amplia gama de opciones para caracterizar las bases de datos en el campo de la minería de datos en general. Entre ellas no todas tienen la misma relevancia, como lo demuestran los resultados publicados en [55] y [80]. Al respecto consideramos que esta relevancia estará determinada en última instancia por el tipo de problema que se aborde. A esto se añade que el uso indiscriminado de todas las medidas daría lugar a ejemplos con más de 15 atributos (medidas), lo que puede resultar excesivo en muchos contextos.

Un aspecto que no se puede ignorar al aplicar los principios del meta-aprendizaje, es que la caracterización de los datos implica un costo adicional al del procedimiento de minería a que estos se pretendan someter. Para la selección de ejemplos en específico, mantener este costo en el mínimo posible es un factor crucial, pues se trata de métodos de preprocesamiento que precisamente tienen entre sus objetivos una reducción del tiempo necesario para el aprendizaje.

El único trabajo que encontramos relacionado con el costo computacional de la caracterización de los datos es el de Kim y Oommen [62], los que proponen un esquema basado en aplicar algún método de selección de ejemplos y luego calcular las medidas sobre el subconjunto seleccionado por este. Consideramos que tal procedimiento no es aconsejable cuando el objetivo de la caracterización es precisamente determinar cual es el método de selección de ejemplos más adecuado, pues su aplicación implicaría un problema equivalente al que se pretende resolver, elegir el método para seleccionar los ejemplos para la caracterización.

Otro enfoque para atenuar el efecto del costo adicional que implica la caracterización, puede ser aprovechar las estructuras de datos generadas en esta etapa en el proceso posterior de selección de ejemplos. Si con esto se logra que la selección sea más eficiente, entonces el costo total se verá disminuido.

En este epígrafe proponemos un sistema de medidas para la caracterización de bases de datos para la selección de ejemplos. En él retomamos y adaptamos varias de las propuestas comentadas anteriormente. Hemos pretendido obtener con este sistema una descripción intuitiva y compacta del

problema, basada en conceptos relacionados con el aprendizaje y la selección de ejemplos. Hemos buscado además obtener en su cálculo no sólo información cuantitativa, sino además estructuras de datos potencialmente útiles en el propio proceso de selección de ejemplos.

Estas medidas las hemos organizado en tres categorías:

- Medidas basadas en conjuntos locales. Con estas medidas pretendemos aprovechar el concepto de “conjunto local” presentado en [19] para caracterizar los datos, algo que sus autores plantean como una de sus potencialidades. A partir del mismo hemos desarrollado tres medidas que describen desde distintas perspectivas a los datos, pero tienen la ventaja de aprovechar operaciones de cómputo y estructuras de datos comunes.
- Medidas estadísticas. En esta categoría proponemos tres medidas que se corresponden con propiedades de los datos cuyo efecto sobre las técnicas de minería de datos es ampliamente conocido, estas además que se caracterizan por ser simples y poco costosas en su cálculo.
- Ruido. Esta categoría sólo contiene una medida, el ruido es ampliamente reconocido como uno de los factores determinantes del desempeño tanto de clasificadores como de métodos de selección de ejemplos.

A continuación se describe en detalles cada una de estas medidas y la motivación para incluirla en el sistema.

3.2.1. Medidas basadas en conjuntos locales

El conjunto local asociado a un ejemplo e ($LS(e)$ de Local Set), es el conjunto de ejemplos cuya distancia a e es menor que la distancia de este a su enemigo más cercano.

$$LS(e) = \{x \mid dist(e, x) < dist(e, ne(e))\}$$

Dado un ejemplo e , llamaremos *radio* de su conjunto local a la distancia entre este y su enemigo más cercano ($LSR(e) = dist(e, nn(e))$), mientras la *cardinalidad* será el número de ejemplos en el conjunto local ($LSC(e) = |LS(e)|$).

La figura 3.1 muestra seis ejemplos en un espacio bidimensional, el conjunto local del ejemplo etiquetado como A tiene una cardinalidad igual a 3 y está delimitado por una circunferencia de radio r igual a la distancia entre A y B , que es su enemigo más cercano.

3.2.1.1. Cardinalidad promedio de los conjuntos locales

Por su definición, la cardinalidad del conjunto local de un ejemplo indica cuán cerca está de una frontera de decisión y cuán angosta es esta. Mientras más cerca esté de la frontera y más angosta sea esta, menor será el valor de LSC ; por lo que es lógico pensar entonces que mientras mayor longitud

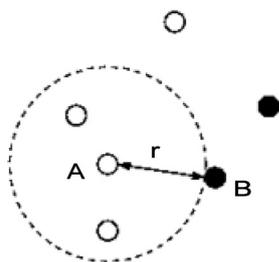


Figura 3.1: Ejemplo de un conjunto local en un espacio bidimensional.

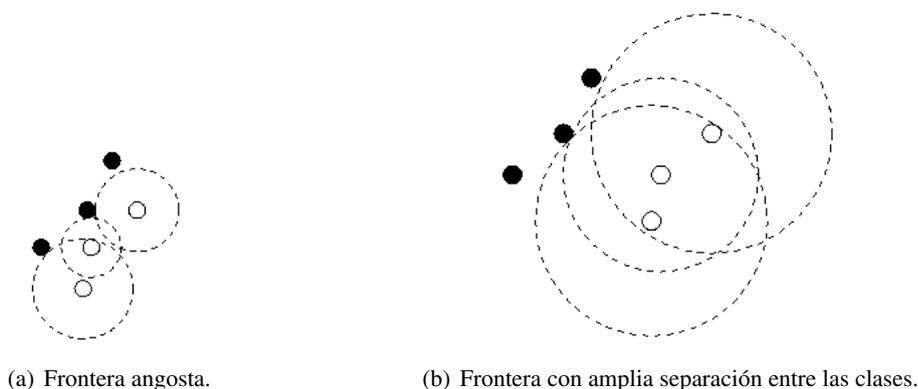


Figura 3.2: Ejemplo de cómo una frontera angosta provoca LSC bajos.

tengan las fronteras en proporción al área de las clases y más angostas sean, menor será el valor promedio de LSC .

En la figura 3.2 se aprecia un ejemplo de cómo una frontera angosta incide en que los valores de los LSC tiendan a ser menores que cuando el área entre las clases es amplia. Tanto la clase negra como la blanca tienen la misma distribución interna en 3.2(a) y en 3.2(b). En ambos casos, todos los ejemplos pueden considerarse fronterizos, pero en el segundo la separación entre ambas clases es mayor. Como puede observarse, en el primer caso todos los ejemplos blancos tienen LSC igual a 1, mientras en el segundo todos los LS abarcan a los tres ejemplos.

Por su parte la figura 3.3 ilustra cómo influye la longitud de las fronteras en la cardinalidad promedio de los LS. Tanto en 3.3(a) como en 3.3(b) aparecen dos clases con seis ejemplos cada una e iguales distribuciones intraclases. Sin embargo, la frontera de decisión en el primer caso está delimitada por tres ejemplos de cada clase y en el segundo por uno. Se han señalado con circunferencias discontinuas los límites de los conjuntos locales de tres ejemplos de la clase negra: A, que representa a los ejemplos más alejados de la frontera; B, que representa el nivel intermedio y C, que representa a los ejemplos fronterizos.

En el caso donde la frontera tiene mayor longitud, sólo el LS de A tiene una cardinalidad de 6,

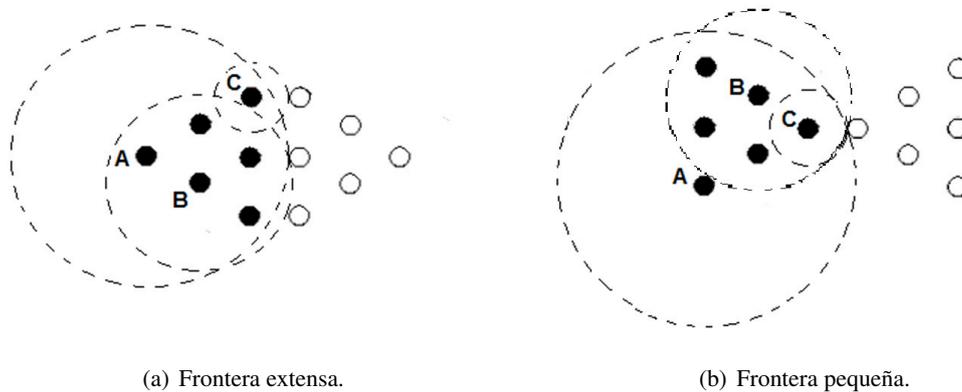


Figura 3.3: Ejemplo de cómo una frontera extensa provoca que el promedio de LSC sea bajo.

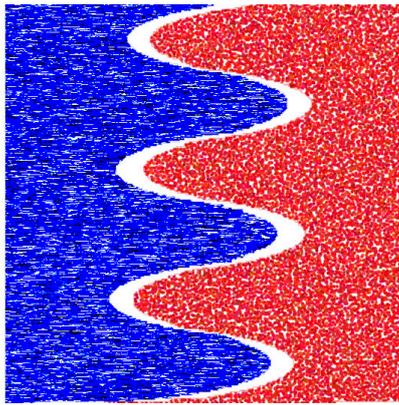
la correspondiente al LS de B es 5 y como puede apreciarse hay otro ejemplo en igual situación, mientras tres ejemplos (C incluido) tienen LS unitarios. Esto da una cardinalidad promedio de 3,17. Sin embargo en el caso donde la frontera está delimitada por un ejemplo de cada clase, tres de los ejemplos (incluyendo a A) tienen LS con cardinalidad igual a 6, dos (incluyendo a B) tienen LS con cardinalidad 5 y sólo C tiene un LS unitario, lo que da lugar a una cardinalidad promedio de 4,83.

La utilidad de la cardinalidad promedio de los LS para caracterizar las fronteras de decisión, radica en que estas pueden ser un factor determinante en el éxito de un método de selección de ejemplos u otro. En problemas donde las fronteras son muy irregulares y el área que separa las clases muy angosta como el de la figura 3.4(a), los métodos que seleccionan sólo ejemplos centrales pueden dejar fuera la información necesaria para clasificar a una buena parte de los ejemplos. En contraste, cuando las fronteras son regulares y el área entre las clases es amplia como en la figura 3.4(b), conservar los ejemplos fronterizos puede ir en detrimento de la reducción, sin aportar nada a la precisión.

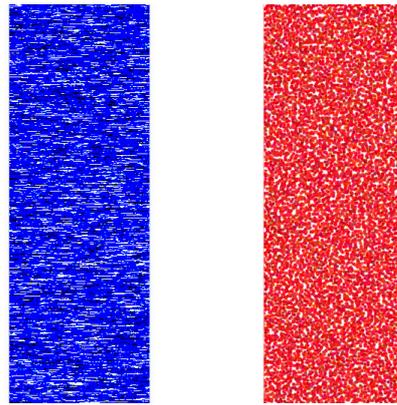
Basándonos en los elementos anteriores, consideramos que el valor promedio de LSC es relevante como atributo para caracterizar las bases de datos para la selección de ejemplos. Aunque es preciso señalar, que la anterior interpretación del valor de LSC puede verse seriamente afectada por el nivel de ruido en la base de datos. Un ejemplo mal etiquetado haría que sus vecinos tuvieran valores de LSC similares a los de ejemplos fronterizos. Para evitar esta anomalía, el cómputo de los LS lo llevamos a cabo luego de haber eliminado el ruido empleando el método ENN.

Finalmente, aunque hemos hablado de LSC mayores o menores, es importante tener en cuenta que estas son apreciaciones subjetivas cuyo significado varía de una base de datos a otra y lo que necesitamos es una medida que permita hacer comparaciones entre bases de datos diferentes. Al respecto, creemos que el factor que mayor influencia tiene en la relevancia del valor de LSC es el número de ejemplos, por lo que adoptamos como medida la siguiente:

$$LSC_{Avg} = \frac{1}{N^2} \sum_{\{e \in T\}} (LSC(e))$$



(a) Frontera irregular y angosta.



(b) Frontera regular y amplia separación entre las clases.

Figura 3.4: Dos problemas con tipos diferentes de fronteras.

Donde T y N corresponden al conjunto de ejemplos y su cardinalidad luego de haber sido eliminado el ruido, mientras el exponente de N se debe a la normalización del promedio con respecto al número de ejemplos.

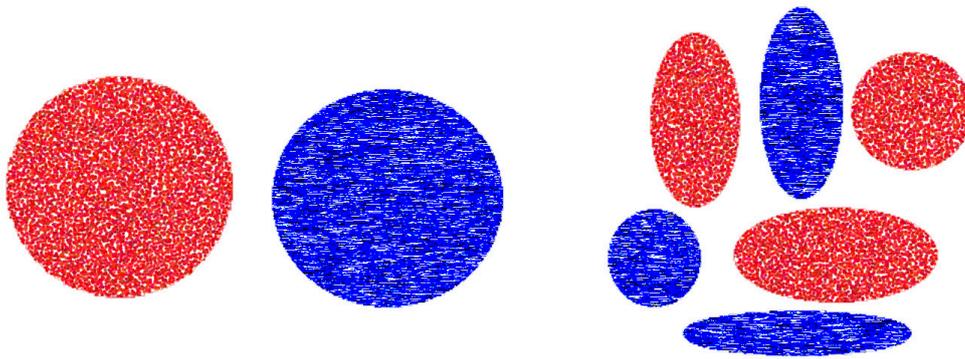
Entre las medidas de Ho y Basu, puede decirse que $N1$, $N2$ y $N3$ son las que más relación guardan con $LSCAvg$. Cada una de ellas mide, desde una perspectiva diferente, el grado de complejidad de las fronteras de decisión. Sin embargo, una diferencia muy significativa entre las tres primeras y $LSCAvg$, es que sólo esta última se calcula descartando el ruido, por lo que consideramos que es menos sensible a ser distorsionada por el efecto del mismo.

3.2.1.2. Número de clusters

Los conjuntos locales no sólo ofrecen información relevante para la caracterización de las bases de datos por medio de su cardinalidad, sino que pueden ser muy útiles para obtener otras estructuras como los clusters.

El número de clusters en que se agrupan los ejemplos de igual clase, puede dar una idea del patrón estructural de estas. Dadas dos bases de datos con igual número de ejemplos, aquella con menos clusters debe tener una estructura de clases más homogénea ya que cada cluster agrupa a más ejemplos. Para ilustrar esta idea observemos la figura 3.5. En 3.5(a) aparece un problema con una estructura de clases muy homogénea, en tal caso cada clase es cubierta por un único cluster. Por el contrario en 3.5(b) aparece un problema con una estructura de clases poco homogénea donde al menos tres clusters son necesarios para cubrir cada clase.

Existen numerosos métodos para la detección de clusters, sin embargo los enfoques tradicionales están concebidos para problemas de aprendizaje no supervisado, por lo que no aprovechan la información sobre la clasificación de los ejemplos aún cuando esta esté disponible como ocurre en los problemas de aprendizaje supervisado. Esto provoca que si se emplean en estos últimos pueden asignar ejemplos de diferentes clases a un mismo cluster. El agrupamiento supervisado, es un campo



(a) Estructura de clases muy homogénea.

(b) Estructura de clases poco homogénea.

Figura 3.5: Dos problemas con diferentes estructuras de clases.

emergente que trata precisamente la detección de clusters en entornos con algún grado de supervisión, sin embargo por ser relativamente joven son escasas las propuestas en la literatura especializada, la mayoría de ellas usan datos etiquetados o algún tipo de retroalimentación del usuario para ayudar al agrupamiento de datos no etiquetados. En [7] se discute acerca de algunos de los trabajos en este campo y se presenta “k-medias supervisado”, una propuesta interesante basada en el popular método que le da nombre.

Aunque “k-medias supervisado” muestra un buen desempeño según los resultados publicados por sus autores, tiene dos características que lo hacen inapropiado para el contexto que nos ocupa. Este método requiere que el número de clusters se fije de antemano y además no garantiza que cada cluster contenga estrictamente ejemplos de una sola clase.

Por otra parte, entre los métodos de selección de ejemplos estudiados, SGP presenta una estrategia de agrupamiento que puede considerarse supervisada. Este garantiza además que cada cluster contenga ejemplos de una sola clase. Sin embargo, su desempeño depende de dos parámetros y los resultados experimentales [105] sugieren que provoca una reducción demasiado agresiva en el número de ejemplos (centroides).

Considerando que los métodos existentes no se adecuaban a nuestros objetivos, decidimos aprovechar la característica que presentan los conjuntos locales, de contener ejemplos de una misma clase y agrupados espacialmente, para desarrollar nuestra propia propuesta de agrupamiento supervisado que aparece en la figura 3.6.

El método se inicia con un filtro de ruido, para eliminar la distorsión que este provoca en los conjuntos locales y que comentamos con anterioridad.

El segundo paso es ordenar los ejemplos descendientemente según la cardinalidad de sus conjuntos locales. El objetivo es procesar primeramente los ejemplos con mayores LSC y favorecer así su selección como núcleos de clusters. Consideramos que estos, al ser los que tienen en sus vecindades la mayor cantidad de ejemplos de sus mismas clases, son los mejores candidatos para este papel. Alternativamente pudieran ordenarse los ejemplos por su LSR , pero esta decisión favorecería la

```

function LSclustering( $T$ )
begin
   $T = ENN(T)$ 
  //Ordenar los ejemplos por su  $LSC$  descendientemente
   $sortDescLSC(T)$ 
   $Clusters = \{\}$ 
  foreach( $e \in T$ )
    if( $\exists(c \in Clusters)[e \in LS(c.core)]$ )
       $c.members = c.members \cup \{e\}$ 
    else
       $newC.members = \{e\}$ 
       $newC.core = e$ 
       $Clusters = Clusters \cup \{newC\}$ 
  return  $Clusters$ 
end

```

Figura 3.6: Algoritmo de agrupamiento supervisado basado en conjuntos locales.

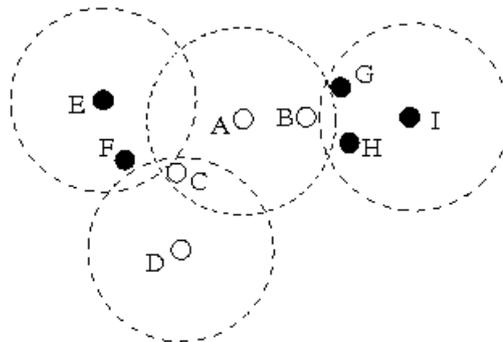


Figura 3.7: Ejemplo del uso de conjuntos locales para detectar clusters.

obtención de clusters extensos en área en lugar de densos en número de ejemplos, siendo esta última la característica que consideramos más relevante.

Una vez ordenados los ejemplos, en el bucle se determina para cada uno de ellos si se incluye en un cluster previamente obtenido o da lugar a uno nuevo. La inclusión tiene lugar cuando el ejemplo es miembro del conjunto local del núcleo de alguno de los clusters detectados.

Por la forma en que han sido elegidos, los núcleos de clusters no son necesariamente sus medoides. Esto podría resolverse con un post-procesamiento, pero hemos considerado que la aproximación que ofrecen es suficiente para nuestro objetivo por lo que el costo extra sería injustificado.

A través de la figura 3.7 podemos ilustrar como funciona este método. Como puede apreciarse, aunque hay 6 ejemplos, sólo 4 conjuntos locales están señalados, estos corresponden a los clusters detectados por nuestra propuesta. Los primeros núcleos identificados son A e I en cualquier orden, ya que ambos tienen conjuntos locales con cardinalidad igual a 3, que es el máximo en el conjunto

de ejemplos. D y E darán lugar al tercer y cuarto núcleos, ambos con $LSC = 2$. Los restantes ejemplos no constituyen núcleos porque están contenidos en conjuntos locales de ejemplos previamente seleccionados como tales, pasando a formar parte de los clusters asociados a ellos. Nótese que C pertenece a los conjuntos locales de A y D , pero el ordenamiento inicial hace que forme parte del cluster asociado al primero, que tiene un LSC mayor.

Para emplear el número de clusters como medida, al igual que hacemos con LSC_{Avg} , lo normalizamos con respecto al número de ejemplos una vez eliminado el ruido.

El tipo de clusters que obtenemos con el método propuesto, guarda estrecha relación con las adherencias que se emplean por Ho y Basu para definir $T1$. En ambos casos se trata de estructuras construidas a partir de hiperesferas centradas en un ejemplo y acotadas por el ejemplo más cercano a este entre los miembros de otra clase. Ambas medidas además, coinciden en ofrecer una descripción interior de las clases en contraste con la caracterización de las fronteras que ofrecen otras como $N1$, $N2$, $N3$ y LSC_{Avg} .

Optamos por el uso de los conjuntos locales como concepto base en lugar de las adherencias porque consideramos que el primero es más intuitivo y simple en su definición. Adicionalmente, el método que proponemos para la construcción de los clusters es más directo que el que se deriva de la composición de adherencias; además, como parte de un sistema de medidas, se integra perfectamente con LSC_{Avg} al compartir ambos el filtrado de ruido y la construcción de los conjuntos locales. Esto último es coherente con la intención de obtener un sistema eficiente.

3.2.1.3. Puntos invasores

Entre los clusters que identifica nuestra propuesta, existen algunos que sólo contienen un ejemplo. En ocasiones esto se debe a que el conjunto local de su núcleo comparte sus ejemplos con los de los núcleos de clusters vecinos y estos son de mayor cardinalidad, tal es el caso del cluster con núcleo D en la figura 3.6. Pero si además del cluster, el conjunto local de su núcleo también es unitario (o sea, sólo contiene al propio núcleo), estamos en presencia de lo que hemos denominado “punto invasor”. Este nombre se debe a que suelen ser ejemplos fronterizos que sin llegar a ser ruido¹, están de cierto modo adentrados en el área de la clase adyacente. El número de tales puntos normalizado respecto al número de ejemplos es otra de las medidas que proponemos como atributo para la caracterización de bases de datos. Esta medida puede catalogarse como una medida de la complejidad de las fronteras de decisión, lo que la relaciona con $N1$, $N2$ y $N3$.

La figura 3.8 ilustra lo anterior. En ella puede apreciarse como un ejemplo de la clase negra está algo adentrado en la región de la clase blanca. Este ejemplo no es descartado como ruido, porque dos de sus tres vecinos más cercanos (los otros ejemplos con el LS señalado) son de su propia clase. Sin embargo, nótese que ni su LS contiene a ninguno de sus vecinos, ni el está contenido en alguno de los de estos.

¹Nótese que el ruido ha sido previamente eliminado

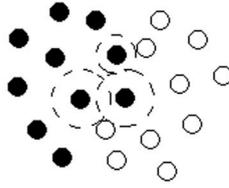


Figura 3.8: Ejemplo de un punto invasor.

3.2.2. Medidas estadísticas

3.2.2.1. Proporción de atributos continuos

Muchos métodos de minería de datos y en particular de selección de ejemplos, han sido originalmente concebidos para procesar solamente atributos continuos. Por ejemplo, entre los métodos analizados en el capítulo 2: PNN construye nuevos prototipos a partir del promedio ponderado de los atributos de dos ejemplos cercanos entre sí, operación esta que no es aplicable a datos nominales; HYB se basa en LVQ3, que a su vez ajusta los code-books mediante operaciones de producto y suma no aplicables a datos nominales; y POP es una versión mejorada de EOP, que no era capaz de manejar atributos nominales.

Generalmente, estos métodos se pueden adaptar para asimilar datos discretos. En algunos casos, como ocurre en POP, es posible incorporar un tratamiento diferenciado para este tipo de atributos. En otros, como pudieran ser LVQ3 y PNN, siempre es posible limitarse a modificar los valores continuos sin alterar los discretos. Sin embargo, estas adaptaciones por lo regular no garantizan que el desempeño ante atributos nominales sea equivalente al desempeño ante atributos continuos.

Analicemos el caso de POP para ilustrar este fenómeno. Este método busca por cada atributo aquellos ejemplos que sirvan para discriminar entre una clase y otra. Con este fin, para cada atributo continuo A_c , ordena los ejemplos de acuerdo al valor de A_c que posean, e identifica segmentos donde todos los ejemplos sean de una misma clase, marcando como relevantes aquellos ubicados en los extremos de cada uno de estos segmentos. Evidentemente los atributos nominales no tienen orden, lo que de entrada elimina la posibilidad de buscar segmentos y obliga, de acuerdo a la solución que aportan sus autores, a que el método marque como relevante un ejemplo por cada valor de cada atributo nominal. Si la cantidad de atributos nominales o de valores de uno de ellos es muy alta, esto pudiera llevar a retener un número exagerado de ejemplos.

En el caso de que se adapte un método ignorando los atributos nominales, a medida que la porción de tales atributos sea mayor, menos validez tendrá el efecto del método en cuestión. Por ejemplo, LVQ3 no ajustaría ningún code-book.

En general, es razonable incluso suponer que en cualquier método basado en distancias pudiera

presentarse algún grado de sensibilidad a la presencia de atributos discretos, ya que estas generalmente demandan algún tipo de adaptación para asimilarlos. Es entonces lógico suponer también que la proporción de atributos que sean continuos, puede ser una característica relevante para seleccionar el método de selección de ejemplos apropiado, por tanto la incluimos entre las medidas que proponemos.

3.2.2.2. Número de clases

El número de clases que se presenten en un problema, es un aspecto relevante para la minería de datos. Muchos algoritmos han sido concebidos para trabajar exclusivamente con problemas binarios. De hecho, las medidas de Ho y Basu originalmente estaban dirigidas a este tipo de problemas. Existen estrategias para aplicar métodos binarios a problemas multi clases, entre ellas se destacan las estrategias “uno contra uno” y “uno contra todos” que descomponen el problema multi clase en múltiples problemas binarios, pero tienen un alto costo computacional derivado de las múltiples aplicaciones del método en cuestión.

En el campo de la selección de ejemplos, el problema del número de clases no ha sido abordado con profundidad. Generalmente las propuestas se presentan como aptas para el trabajo con múltiples clases pero no encontramos estudios sobre el efecto del número de estas en sus resultados. En [71], aparece una propuesta explícitamente diseñada para el trabajo con múltiples clases que emplea una estrategia del tipo “uno contra uno”. Esta sin embargo, está orientada principalmente hacia la mejora en la calidad del aprendizaje y no tiene en cuenta aspectos relacionados con la eficiencia como la reducción del número de ejemplos y el costo en tiempo de ejecución.

Aún a falta de trabajos que lo confirmen, es de suponer que el número de clases sea relevante para la selección de ejemplos. Un problema multi clases, es por naturaleza más complejo que uno binario, el número de fronteras de decisión en los primeros suele ser mucho mayor que en los segundos y la precisión de los clasificadores suele ser mucho menor en ellos. Atendiendo a estos criterios, consideramos oportuna su inclusión como medida de caracterización de los datos. La que por demás, no implica un aumento en el costo de la caracterización.

3.2.2.3. Desbalance

El desbalance típicamente se asocia a bases de datos donde los ejemplos se agrupan en dos clases y el número de ejemplos de una de ellas (mayoritaria o negativa) es muy superior al de la otra (minoritaria o positiva) [47]. Esta distribución implica un problema para la mayoría de los clasificadores ya que tienden a ignorar la clase minoritaria por su poca representatividad para el porcentaje de aciertos.

Como se explica en [47], cuando el desbalance es muy grande el criterio de evaluación convencional basado en el porcentaje de aciertos (o errores) no constituye un modelo válido. De hecho, estos problemas requieren métodos de evaluación que favorezcan los aciertos en la clase positiva o de otro modo esta es anulada. En el mismo artículo se llega a la conclusión de que los métodos generales de selección de ejemplos no son adecuados en estos contextos, donde existen alternativas mucho más

efectivas como el bajo muestreo.

Por la necesidad de un tratamiento muy específico, no incluimos entre nuestros objetivos el procesamiento de bases de datos con un alto grado de desbalance. Sin embargo, consideramos que el grado de desbalance en problemas generales puede ser relevante para la selección de ejemplos y decidimos incluirlo entre los atributos que proponemos. Para ilustrar el fundamento de esta decisión, analizaremos una base de datos y dos estrategias hipotéticas.

La figura 3.9 muestra dos bases de datos con dos clases cada una, la primera perfectamente balanceada (3.9(a)) y la segunda desbalanceada (3.9(b)). A continuación aparecen las regiones aproximadas que se seleccionaría siguiendo una estrategia de selección de puntos fronterizos en cada una de ellas (3.9(c) y 3.9(d)). Como puede apreciarse, en ambos casos el nivel de reducción sería aproximadamente igual y la precisión no se vería afectada pues se conservan las fronteras de decisión originales. Sin embargo si se selecciona el centroide de cada cluster (3.9(e) y 3.9(f)), en ambos casos se obtienen mejores reducciones que con la estrategia anterior, pero en la base de datos desbalanceada, la frontera de decisión (en línea de puntos) se desplaza hacia la derecha con respecto a la frontera original (3.9(b) y 3.9(d)), dando lugar a que ejemplos de la clase azul sean clasificados como rojos.

Generalmente se emplea como medida de desbalance la razón entre el número de ejemplos de la clase mayoritaria y la minoritaria. Sin embargo, como nos interesa procesar bases de datos con cualquier número de clases esta medida no nos resulta útil. En su lugar decidimos emplear la desviación estándar del número de ejemplos por clase, normalizada respecto al total de ejemplos tal como aparece en la ecuación 3.9.

$$Imbalance = \frac{1}{N} \sqrt{\frac{\sum_{c \in classes(T)} (N_c - \mu)^2}{|classes(T)|}} \quad (3.9)$$

donde:

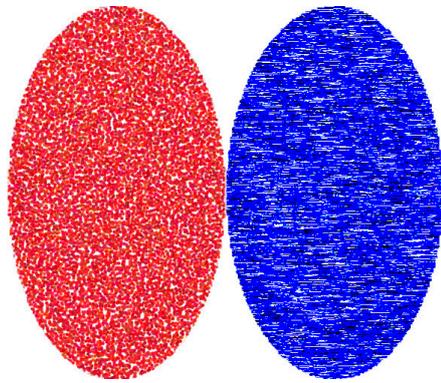
$$N_c = |\{x \mid class(x) = c\}|$$

$$\mu = \frac{\sum_{c' \in classes(T)} N_{c'}}{|classes(T)|}$$

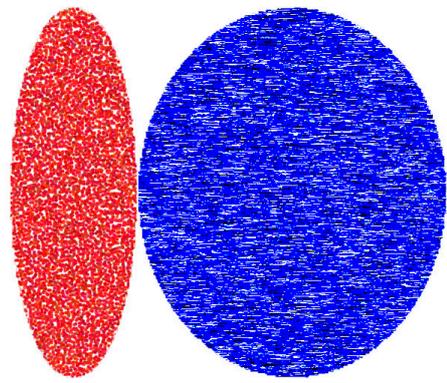
3.2.3. Ruido

La presencia de ruido en los datos es uno de los factores que afectan significativamente a los clasificadores. En [83] se presenta un estudio del desempeño de cuatro clasificadores de reconocido prestigio ante datos con diferentes tipos y niveles de ruido; en este puede apreciarse que en menor o mayor grado, todos se ven afectados por el mismo. En la selección de ejemplos es particularmente importante prestar atención a este hecho, pues una reducción que no tenga en cuenta el ruido presente en los datos, puede llevar a un incremento de su representatividad, como ocurre en el caso de CNN.

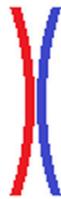
Para evitar esto, muchos de los métodos de selección de ejemplos usan alguna estrategia de eliminación del ruido. Algunos, como DROP3 e ICF, emplean un preprocesamiento basado en ENN. Otros introducen estrategias particulares más sofisticadas como la prueba de proporciones de IB3, el



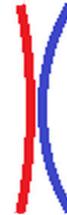
(a) Base de datos balanceada.



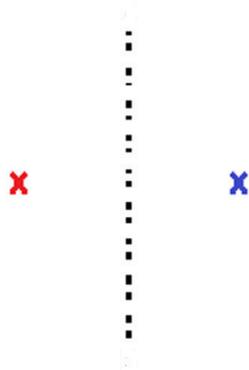
(b) Base de datos desbalanceada.



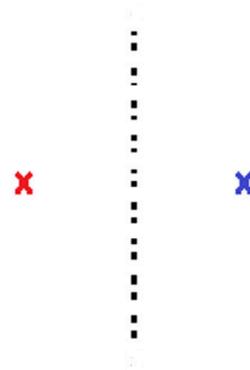
(c) Resultado de retener ejemplos fronterizos en la base de datos balanceada.



(d) Resultado de retener ejemplos fronterizos en la base de datos desbalanceada.



(e) Resultado de retener el centroide del cluster en la base de datos balanceada.



(f) Resultado de retener el centroide del cluster en la base de datos desbalanceada.

Figura 3.9: Efecto del desbalance sobre estrategias de selección basadas en fronteras y centroides.

criterio basado en cubrimiento y alcance de *CPruner*, la selección condicionada por clase de CCIS o el conteo de votos de NRMCS.

Creemos que el nivel de ruido en una base de datos debe ser uno de los atributos a considerar para elegir un método de selección de ejemplos que la procese. ENN es un método simple y muy conocido, cuyo desempeño como filtro de ruido ha sido previamente documentado [111, 19]. Este método incluso es empleado por Sotoca y otros para estimar para cada clase el número de ejemplos que están ubicados en regiones de otras clases (medida *D3*); coherentemente con esto, hemos decidido emplear el porcentaje de reducción que ENN logra como una estimación del nivel de ruido presente en los datos.

3.3. Optimización de las medidas mediante métodos de búsqueda rápida de vecinos más cercanos

Como planteábamos al inicio del epígrafe anterior, mantener el costo de la caracterización de los datos en el mínimo posible es un factor crucial para aplicar el meta-aprendizaje a la selección de ejemplos. En este epígrafe describimos como adaptamos una conocida técnica de optimización de búsqueda del vecino más cercano para ganar eficiencia en el cálculo de las medidas propuestas. Adicionalmente, comparamos el costo de nuestra propuesta con el de las medidas de Ho y Basu descritas en el epígrafe 3.1.

El costo computacional de las medidas que proponemos está determinado fundamentalmente por la invocación a ENN y la construcción de los conjuntos locales. Ambas tareas se basan en la búsqueda de vecindades y tienen un costo de $O(n * NN)$ donde n es el número de ejemplos y NN el costo de buscar los miembros de la vecindad correspondiente a un ejemplo.

Cuando se emplea el método exhaustivo, la búsqueda de los miembros de la vecindad correspondiente a un ejemplo dado implica el cálculo de la distancia entre este y cada uno de los restantes ejemplos en la base de datos, con un costo $O(nm)$ siendo m la cantidad de atributos y asumiendo que el costo de calcular la distancia entre dos ejemplos es m . Esto hace que el costo de la caracterización de los datos empleando este método sea $O(n^2m)$.

Existen sin embargo alternativas desarrolladas dentro del campo conocido como “algoritmos rápidos de búsqueda del vecino más cercano” [81] que consiguen buscar los miembros de una vecindad en un tiempo inferior a $O(nm)$ y generalmente cercano a $O(m \log n)$. Para esto usan estructuras de datos (generalmente árboles) que le permiten acotar la búsqueda descartando subconjuntos de ejemplos que no cumplen determinada condición de cercanía evaluada con respecto al vecino más cercano encontrado hasta el momento.

Cabe señalar que la construcción de estas estructuras de datos implica un costo adicional que varía de un método a otro, pero que siempre es mayor o igual que el correspondiente a una búsqueda exhaustiva simple. Esto constituye un aspecto muy importante a tener en cuenta para valorar la factibilidad del empleo de estas técnicas. Por ejemplo, si se intentara emplear un método de búsqueda rápida para obtener mayor velocidad en el método de selección de ejemplos CNN, el resultado sería todo lo contrario. Esto se debe a que en CNN cada búsqueda del vecino más cercano parte de

un conjunto de ejemplos que puede ser diferente al de la búsqueda que la precede. De este modo, habría que reconstruir en cada momento la estructura de datos, lo que sumaría un tiempo mayor al ahorrado en la búsqueda en sí. De manera similar, métodos como DROP3, CPruner e IB3 difícilmente podrían verse beneficiados con la inclusión de estas técnicas de búsqueda rápida.

Un método que sí podría beneficiarse es ENN, pues realiza n búsquedas de vecinos cercanos sobre el mismo conjunto de ejemplos, lo que permite reutilizar n veces las estructuras de datos. La condición necesaria en este caso sería que el costo de la construcción de las estructuras de datos sea inferior a $O(n^2m)$. Del mismo modo, ICF también pudiera beneficiarse pues en cada iteración emplea el conjunto de ejemplos seleccionados para calcular los conjuntos *reachable* y *coverage* de todos sus miembros.

En el caso concreto del cómputo de las medidas que se pretende optimizar existen dos momentos en los que habría que construir las estructuras de datos. El primero sería previo a la ejecución de ENN en su función de filtro de ruido y partiría del conjunto de ejemplos inicial. En el segundo, se partiría del conjunto libre de ruido y los resultados serían empleados en la construcción de los conjuntos locales. En ambos casos, las estructuras de datos creadas para un conjunto de n ejemplos, se emplearían en n búsquedas de vecindad, por lo que su costo estaría compensado si es inferior a $O(n^2m)$.

Un aspecto que hace peculiar la búsqueda de una vecindad en el caso de la construcción de conjuntos locales, es que su tamaño no está determinado por una constante como suele ocurrir. Téngase en cuenta que la cardinalidad de cada conjunto local está determinada por el número de ejemplos más cercanos que el enemigo más cercano. Esto implica que al método de búsqueda rápida que se emplee haya que hacerle algunas adaptaciones para manejar esta particularidad.

El más conocido de estos métodos de búsqueda rápida es kd-Tree [41]. Se basa en un árbol binario donde a cada nodo se asocia un atributo y un valor que divide en dos al conjunto de ejemplos que le corresponde (en la raíz es el conjunto completo). Para realizar la búsqueda en el árbol, se explora primero la rama que contiene al objetivo y en una especie de back-tracking se visitan solo aquellas otras alternativas que no puedan descartarse por solaparse con la hipersfera con centro en el objetivo y radio igual a la distancia entre este y el vecino más cercano encontrado hasta el momento.

Aunque es un enfoque bastante simple y efectivo, kd-Tree presenta dos desventajas que lo hacen poco aconsejable para nuestro propósito. En primer lugar, es conocida su característica de degenerar con la dimensionalidad de los datos [81]. El segundo aspecto es a nuestro juicio más relevante aún, se trata de que este método no está preparado para trabajar con atributos nominales pues presupone que en cada atributo existe una relación de orden entre sus valores. En la práctica este orden se podría simular empleando un ordenamiento alfabético o similar para los atributos nominales, pero el poder discriminante de los nodos afectados sería muy cuestionable.

Como alternativa hemos optado por emplear el método vp-Tree [114]. Este método tiene mucha similitud con el anterior, pero presenta una ventaja muy importante, no presupone nada con respecto a los atributos. De hecho ni siquiera presupone que se trabaje sobre un espacio euclídeo sino que está diseñado para trabajar sobre espacios métricos en general.

Al igual que kd-Tree, vp-Tree emplea como estructura de datos un árbol binario donde a cada nodo se asocia un subconjunto de la base de ejemplos. En este caso, el criterio empleado para dividir recursivamente el conjunto original se basa en el uso de un elemento pivote o “vantage-point” (vp). En cada nodo se elige como vp un ejemplo perteneciente al correspondiente conjunto de ejemplos C , y se calcula la mediana μ de las distancias de vp al resto de los ejemplos en C . Todos los ejemplos que están a una distancia de vp menor que μ estarán contenidos en el hijo izquierdo, mientras el resto de los ejemplos estarán en el hijo derecho. Adicionalmente se almacenan en cada nodo las cotas inferior y superior de las distancias del vp a los ejemplos de cada hijo.

En principio el vp podría seleccionarse de forma totalmente aleatoria, pero los autores proponen un método basado en muestreo argumentando que ofrece mejores resultados. La técnica consiste en seleccionar una muestra aleatoria de ejemplos como candidatos a vp y para cada uno a su vez seleccionar otra muestra sobre la que calcula μ y la varianza de las distancias al vp. El candidato con mayor varianza asociada será entonces el elegido.

El procedimiento de búsqueda en el método vp-Tree es similar al de kd-Tree, aunque con sus propios criterios de poda asociados a sus criterios de partición. La figura 3.10 muestra este procedimiento adaptado a la búsqueda de k vecinos, mientras la figura 3.11 muestra la adaptación para buscar un conjunto local.

En ambos casos el procedimiento se inicia verificando si el ejemplo visitado ($node.vp$) debe ser incluido en la solución candidata y se actúa en consecuencia. Acto seguido, se recorre recursivamente el resto del árbol en un orden que dependerá de la distancia del objetivo con respecto al ejemplo visitado (x), de modo que se visite primero la rama que contendría al objetivo si este estuviera en el árbol. Para recorrer una rama, se verifica antes si es posible encontrar una solución mejor a la actual, de lo contrario se poda. Este criterio de poda está guiado por un radio de búsqueda ($radius$) que inicialmente es ∞ y que se va actualizando con valores decrecientes ante cada actualización de la solución candidata.

En el caso de la búsqueda de los k vecinos más cercanos a q , la inclusión de $node.vp$ en la solución candidata ($KNNList$) puede deberse a dos circunstancias. La primera es que aún no haya k elementos en $KNNList$, ante lo que se agrega $node.vp$ a esta y en caso de alcanzarse la cantidad de k elementos, se actualiza $radius$ al valor de la mayor distancia entre q y los elementos de $KNNList$, es decir, la distancia al más lejano de los candidatos a vecinos más cercanos. La otra razón para incluir $node.vp$ en $KNNList$ es que, encontrándose esta última llena, x sea menor que $radius$. Esto significa que se ha encontrado un ejemplo que es más cercano a q que al menos uno de los candidatos encontrados previamente. En este caso no sólo se agrega $node.vp$ a $KNNList$ sino que se elimina de esta el ejemplo más distante de q y se actualiza $radius$.

Por otra parte, en la búsqueda del conjunto local asociado a q , la inclusión de $node.vp$ en la solución candidata (LS) siempre va a responder a que x sea menor que $radius$ puesto que en este caso no hay una cantidad predefinida de elementos a buscar. Otra peculiaridad es que ninguna inclusión modifica a $radius$, este se actualiza sólo cuando se cumple la condición anterior pero la clase de q es diferente a la de $node.vp$. Ante esta situación no se incluye este último en LS sino que se actualiza $radius$ al valor de x y se eliminan de LS todos los ejemplos que se encuentren a una distancia de q mayor que el nuevo valor de $radius$. Como puede apreciarse, $radius$ en este caso no corresponde a la distancia

```

procedure search_KNN_vp_tree(node, KNNList, q, k, radius)
begin
  x = dist(node.vp, q)
  if (|KNNList| < k)//aún hay menos de k candidatos
    KNNList = KNNList ∪ {node.vp}
    if |KNNList| = k//acaba de llenarse la lista, acotar radio de búsqueda
      radius = Maxe∈KNNList{dist(q, e)}
  else//ya existen k candidatos
    if(x < radius)
      //sustituir el candidato mas lejano
      KNNList = KNNList − First{e ∈ KNNList | dist(q, e) = radius}
      KNNList = KNNList ∪ {node.vp}
      //actualizar radio de búsqueda
      radius = Maxe∈KNNList{dist(q, e)}
  //búsqueda recursiva
  middle = (node.LSupBound + node.RInfBound)/2
  if(x < middle)
    if(node.LInfBound − radius < x < node.LSupBound + radius)
      search_KNN_vp_tree(node.Left, KNNList, q, k, radius)
    if(node.RInfBound − radius < x < node.RSupBound + radius)
      search_KNN_vp_tree(node.Right, KNNList, q, k, radius)
  else
    if(node.RInfBound − radius < x < node.RSupBound + radius)
      search_KNN_vp_tree(node.Right, KNNList, q, k, radius)
    if(node.LInfBound − radius < x < node.LSupBound + radius)
      search_KNN_vp_tree(node.Left, KNNList, q, k, radius)
end

```

Figura 3.10: Procedimiento para la búsqueda de los k vecinos más cercanos ($KNNList$) a un ejemplo (q) usando un árbol- vp . La variable $node$ es un nodo de un árbol- vp , $KNNList$ contiene la lista de candidatos a vecinos (inicialmente vacía), q es el ejemplo objetivo, k el número de vecinos a buscar y $radius$ la distancia entre q y el miembro de $KNNList$ mas lejano (inicialmente ∞). Por su parte $node$ contiene los campos: vp (pivote), $LInfBound$ (cota inferior de las distancias del vp a los ejemplos del subárbol izquierdo), $LSupBound$ (cota superior), $RInfBound$ (cota inferior, subárbol derecho), $RSupBound$ (cota superior, subárbol derecho), $Left$ (subárbol izquierdo) y $Right$ (subárbol derecho).

```

procedure search_LS_vp_tree(node, LS, q, radius)
begin
   $x = \text{dist}(\text{node.vp}, q)$ 
  if ( $x < \text{radius}$ )
    if( $\text{node.vp.class} = q.\text{class}$ )//agregar a LS
       $LS = LS \cup \{\text{node.vp}\}$ 
    else//acotar radio de búsqueda y descartar candidatos
       $\text{radius} = x$ 
       $LS = LS - \{e | \text{dist}(q, e) > \text{radius}\}$ 
    //búsqueda recursiva
     $\text{middle} = (\text{node.LSupBound} + \text{node.RInfBound})/2;$ 
    if( $\text{dist}(\text{node.vp}, q) < \text{middle}$ )
      if( $\text{node.LInfBound} - \text{radius} < x < \text{node.LSupBound} + \text{radius}$ )
        search_LS_vp_tree(node.Left, LS, q, radius)
      if( $\text{node.RInfBound} - \text{radius} < x < \text{node.RSupBound} + \text{radius}$ )
        search_LS_vp_tree(node.Right, LS, q, radius)
    else
      if( $\text{node.RInfBound} - \text{radius} < x < \text{node.RSupBound} + \text{radius}$ )
        search_LS_vp_tree(node.Right, LS, q, radius)
      if( $\text{node.LInfBound} - \text{radius} < x < \text{node.LSupBound} + \text{radius}$ )
        search_LS_vp_tree(node.Left, LS, q, radius)
end

```

Figura 3.11: Procedimiento para la búsqueda del conjunto local (LS) asociado a un ejemplo (q) usando un árbol- vp . Ver figura anterior para significado de variables.

entre q y ninguno de los elementos de la solución candidata, sino que se corresponde a la distancia entre este y el enemigo más cercano encontrado hasta el momento.

Para comprobar el efecto de la inclusión de este método en el costo computacional del cómputo de las medidas, se compararon sus tiempos de ejecución sobre 40 bases de datos del repositorio KEEL [6] usando la búsqueda exhaustiva y usando los algoritmos antes descritos. En ambos casos el tiempo computado fue el tiempo total necesario para obtener los valores de las medidas por lo que se incluye el costo de la construcción del árbol- vp .

La tabla 3.1 muestra los resultados obtenidos. Como puede apreciarse en la última columna, la versión que usa los métodos de búsqueda basados en el árbol- vp es más rápida que la que emplea la búsqueda exhaustiva en 33 de las 40 bases de datos, llegando a ser en promedio 2,2198 veces más veloz.

Los resultados anteriores evidencian que la versión optimizada es realmente menos costosa que la exhaustiva. Sin embargo, es importante verificar cómo es el costo computacional de las medidas que proponemos en comparación con el de otras propuestas anteriores. Con este propósito, decidimos tomar como patrón de comparación las medidas propuestas por Ho y Basu, que como hemos planteado, son a nuestro juicio las que más influencia han tenido en el campo de la selección de ejemplos.

Para esta comparación, en el caso de nuestra propuesta empleamos la versión optimizada descrita con anterioridad. Para las medidas de Ho y Basu empleamos la implementación disponible en la plataforma KEEL [5]. En las bases de datos con más de 2 clases, las medidas F1 y F2 se calcularon considerando solo dos de las clases, mientras L1, L2 y L3 no se calcularon. Para el cómputo de las medidas propuestas en este trabajo, así como para el resto de las medidas propuestas por Ho y Basu siempre se consideró la base de datos íntegra.

La tabla 3.2 muestra los tiempos de ejecución en cada una de las bases de datos empleadas con anterioridad. Puede apreciarse como en 29 de ellas nuestra propuesta es más rápida que la de Ho y Basu, contra solo 11 casos en los que esta última es mejor. En promedio, el cálculo de las medidas de Ho y Basu consume 25,6822 veces el tiempo de que consume el cálculo de las propuestas en esta memoria.

La diferencia observada es más relevante si tenemos en cuenta que 23 de las bases de datos tienen más de 2 clases, por lo que en realidad sólo han sido parcialmente computados los valores para las medidas de Ho y Basu. Ya hemos comentado que es posible adaptar algunas de estas medidas al trabajo con bases de datos de más de dos clases; sin embargo, esto incrementa su complejidad computacional, con lo que la ventaja de nuestra propuesta se acrecienta.

Finalmente, aunque ya hemos comprobado que son menos costosas que las propuestas por Ho y Basu, debemos recordar que estas medidas no sólo aportan su valor, sino que en su cálculo se obtienen estructuras de datos potencialmente útiles para la propia selección de ejemplos, que implican otra reserva de eficiencia tal como veremos en el capítulo 5.

Tabla 3.1: Tiempos de ejecución (en segundos) del cómputo de las medidas. Variantes exhaustiva y basada en árbol-*vp*.

Base de datos	Exhaustiva	Árbol-<i>vp</i>	Exhaustiva/Árbol-<i>vp</i>
appendicitis	0,0961	0,0810	1,1864
automobile	0,3390	0,2369	1,4310
balance	0,5231	0,3065	1,7067
bands	1,5478	1,0632	1,4558
breast	0,1411	0,1469	0,9605
car	2,9474	0,6725	4,3828
cleveland	0,3330	0,2938	1,1334
contraceptive	2,7794	0,6063	4,5842
dermatology	1,3668	0,9649	1,4165
ecoli	0,3146	0,2648	1,1881
flare	1,2694	0,4655	2,7270
glass	0,2316	0,1755	1,3197
haberman	0,1479	0,1808	0,8180
heart	0,3214	0,3028	1,0614
hepatitis	0,1886	0,2274	0,8294
horse-colic	0,5319	0,5548	0,9587
housevotes	0,5052	0,3680	1,3728
ionosphere	1,5778	0,7919	1,9924
iris	0,1081	0,0981	1,1019
led7digit	0,3451	0,2657	1,2988
lymphography	0,1376	0,1264	1,0886
mammographic	1,1453	0,4509	2,5400
monk-2	0,4233	0,3330	1,2712
movement_libras	3,7336	1,6556	2,2551
newthyroid	0,1507	0,1914	0,7874
pima	1,3575	0,7188	1,8886
saheart	0,5472	0,4156	1,3167
segment	51,4531	4,4952	11,4462
sonar	0,7788	0,6714	1,1600
spectfheart	0,6744	0,6550	1,0296
splice	120,2849	20,5872	5,8427
tic-tac-toe	1,1335	0,4179	2,7124
vehicle	2,9932	0,8721	3,4322
vowel	5,2229	0,8089	6,4568
wdbc	3,7011	1,7283	2,1415
wine	0,2325	0,2428	0,9576
winequality-red	6,0847	1,8367	3,3128
wisconsin	1,2507	0,6814	1,8355
yeast	3,3293	0,9651	3,4497
zoo	0,0508	0,0539	0,9425
Promedio	5,5075	1,1494	2,2198

Tabla 3.2: Tiempos de ejecución (en segundos) del cómputo de las medidas propuestas en este trabajo y las propuestas por Ho y Basu.

Base de datos	Medidas de Ho y Basu (A)	Nuestra propuesta (B)	A/B
appendicitis	0,0846	0,0810	1,0444
automobile	0,6427	0,2369	2,7130
balance	0,2979	0,3065	0,9719
bands	108,2452	1,0632	101,8108
breast	0,2509	0,1469	1,7080
car	1,8796	0,6725	2,7949
cleveland	0,4621	0,2938	1,5728
contraceptive	1,8844	0,6063	3,1080
dermatology	1,5368	0,9649	1,5927
ecoli	0,4152	0,2648	1,5680
flare	1,8876	0,4655	4,0550
glass	0,0645	0,1755	0,3675
haberman	0,4606	0,1808	2,5476
heart	2,6138	0,3028	8,6321
hepatitis	4,4639	0,2274	19,6302
horse-colic	454,9070	0,5548	819,9477
housevotes	0,5041	0,3680	1,3698
ionosphere	0,7466	0,7919	0,9428
iris	0,0689	0,0981	0,7023
led7digit	0,7183	0,2657	2,7034
lymphography	0,2402	0,1264	1,9003
mammographic	1,2528	0,4509	2,7784
monk-2	0,2422	0,3330	0,7273
movement_libras	8,8614	1,6556	5,3524
newthyroid	0,1209	0,1914	0,6317
pima	0,6592	0,7188	0,9171
saheart	1,0992	0,4156	2,6449
segment	10,9732	4,4952	2,4411
sonar	0,6997	0,6714	1,0422
spectfheart	4,7542	0,6550	7,2583
splice	43,4626	20,5872	2,1111
tic-tac-toe	0,9862	0,4179	2,3599
vehicle	1,9438	0,8721	2,2289
vowel	3,3448	0,8089	4,1350
wdbc	1,2789	1,7283	0,7400
wine	0,1938	0,2428	0,7982
winequality-red	1,7448	1,8367	0,9500
wisconsin	0,6467	0,6814	0,9491
yeast	3,1128	0,9651	3,2254
zoo	0,2325	0,0539	4,3135
Promedio	16,6996	1,1494	25,6822

3.4. Conclusiones Parciales

En este capítulo hemos abordado los trabajos que, dentro del campo del meta-aprendizaje, han sido a nuestro juicio los más relevantes para la selección de ejemplos. Hemos propuesto un nuevo sistema de medidas para la caracterización de los datos con la intención de que sean útiles para aplicar los principios del meta-aprendizaje a la selección de ejemplos. Consideramos que en este propósito es vital la eficiencia computacional, la que hemos buscado por medio de la reutilización de cálculos, la elección de medidas simples y el empleo de un número relativamente pequeño de ellas.

Comprobamos además como las técnicas de búsqueda rápida de vecinos más cercanos favorecen la eficiencia en el cómputo de las medidas propuestas. Asimismo, mostramos experimentalmente que estas resultan mucho menos costosas que el conjunto de medidas propuestas por Ho y Basu, que hasta el momento han sido las de mayor influencia para el campo de la selección de ejemplos.

Tres de las medidas propuestas están basadas en el concepto de conjunto local. Las estructuras de datos que se crean para su cálculo, son potencialmente útiles para la propia selección de ejemplos como veremos en el capítulo 5; pero antes estudiaremos la utilidad de esta propuesta para elegir automáticamente uno entre varios de los métodos analizados en el capítulo 2.

Capítulo 4

Aprendizaje de Meta-Modelos para Seleccionar el Método Adecuado

En este capítulo pondremos a prueba la efectividad de las medidas propuestas para caracterizar bases de datos con vistas a la elección de un método de selección de ejemplos adecuado. Como antes hemos comentado, nuestra propuesta se basa en el aprendizaje de modelos capaces de predecir el desempeño de diferentes métodos de selección de ejemplos ante una base de datos. Al tratarse de aprendizaje acerca de una tarea cuya finalidad es a su vez el aprendizaje, podemos referirnos a este como meta-aprendizaje, y a los modelos obtenidos como meta-modelos.

En la figura 4.1 se ilustra el modelo que proponemos para el uso del meta-aprendizaje en la selección de ejemplos. El mismo lo concebimos como un proceso formado por dos fases, cada una de las cuales contiene a su vez dos etapas.

En la primera fase se aprenden los meta-modelos predictivos, cada uno de estos está asociado a un método de selección de ejemplos (en lo sucesivo nos referiremos a estos métodos como “individuales”) y es capaz de predecir el desempeño del mismo en una nueva base de datos. La entrada de esta fase la constituyen los resultados de un conjunto de experimentos, cada uno de los cuales consiste en suministrar una base de datos a un método individual de selección de ejemplos y a su vez la salida de este a un clasificador. En la etapa de caracterización, estos resultados experimentales se sintetizan en un conjunto de meta-datos, donde cada base de datos es representada por la tupla de valores asociados a un conjunto de medidas calculadas sobre ella. Es a partir de estos meta-datos, que empleando algún algoritmo de aprendizaje se extraen los meta-modelos en la etapa de meta-aprendizaje.

La segunda fase es la que contiene la selección de ejemplos como tal, específicamente en la etapa de selección de ejemplos basada en conocimiento (*KBIS*, de Knowledge Based Instance Selection). En esta, cada nuevo conjunto de ejemplos es transformado en un meta-ejemplo y suministrado a los meta-modelos para predecir el desempeño que tendría cada uno de los métodos individuales sobre el mismo. Entonces, al método para el que se obtenga la mejor predicción se le suministra el conjunto de ejemplos de entrada siendo el encargado de generar la salida (ejemplos seleccionados). A partir

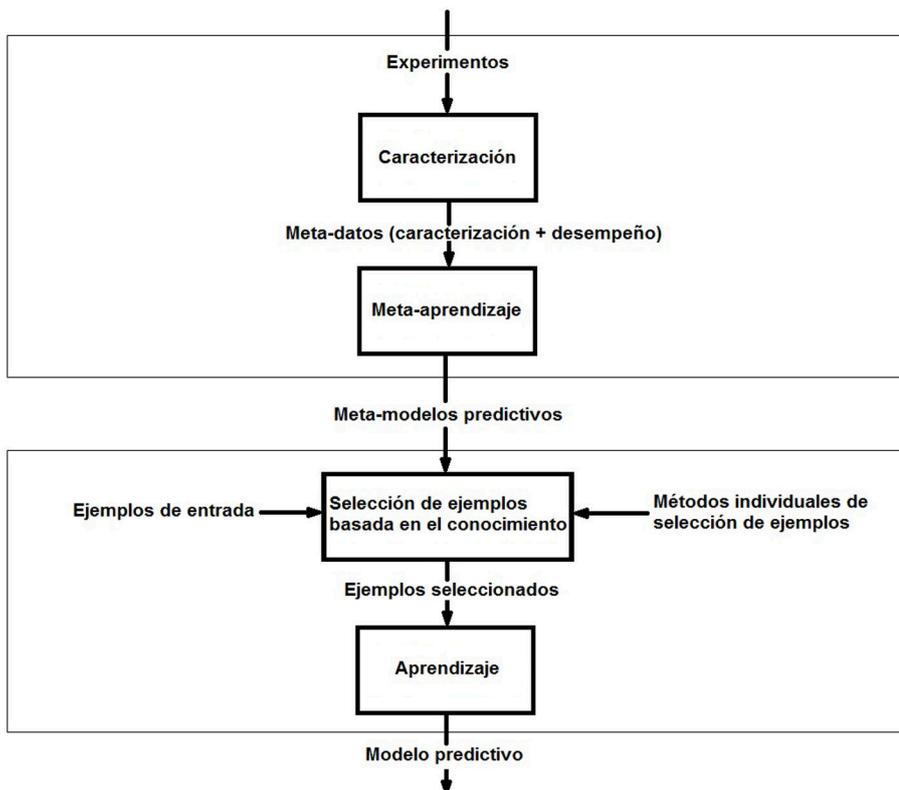


Figura 4.1: Modelo que se propone para el uso de meta-aprendizaje en la selección de ejemplos. Los rectángulos exteriores delimitan las fases y los interiores las etapas.

de aquí, lo que sigue es idéntico a cualquier proceso de aprendizaje con selección de ejemplos, esta salida es empleada para entrenar un clasificador y se obtiene un modelo predictivo.

Las dos fases anteriores gozan de bastante independencia entre sí. La única restricción inviolable en su interrelación, es que la necesidad de disponer de los meta-modelos predictivos obliga a que antes de llevar a cabo la segunda, la primera debe haber sido completada al menos una vez. Por lo demás, los meta-modelos pueden ser estáticos, lo que implicaría una única ejecución de la primera fase; o dinámicos, actualizándose periódicamente a partir de nuevas ejecuciones de la primera fase sobre nuevos resultados experimentales.

A continuación describiremos una implementación del modelo descrito. Comenzaremos por la arquitectura que proponemos para un sistema de KBIS pues esta, aunque no es la primera, condiciona la manera en que se desarrollan las etapas previas de caracterización y meta-aprendizaje, que describimos con posterioridad. Por su parte la etapa de aprendizaje no requiere comentarios adicionales, pues es el clásico aprendizaje automático.

4.1. Arquitectura de un sistema de selección de ejemplos basado en conocimiento

Un sistema KBIS debe ser capaz de predecir el desempeño de varios métodos individuales de selección de ejemplos sobre una base de datos y aplicar a ésta aquel para el que se obtenga la mejor predicción. Esto implica que una decisión importante en su diseño sea definir como se medirá el desempeño de los métodos individuales.

La selección de ejemplos como problema plantea dos objetivos fundamentales y generalmente contrapuestos: maximizar la reducción de la base de datos y maximizar la precisión alcanzada por un clasificador entrenado con los ejemplos seleccionados. Tal situación conlleva a que la medición del desempeño de un método de selección implique combinar de alguna manera ambos objetivos. Esto puede llevarse a cabo mediante una combinación lineal, como se ha hecho exitosamente en el caso de los enfoques evolutivos [24] donde se suele emplear tal tipo de combinación como función de ajuste.

Una primera aproximación pudiera ser entonces definir el desempeño como una combinación lineal de reducción y precisión, y aprender para cada método individual un meta-modelo capaz de predecir el valor de esta combinación dada una base de datos. Este enfoque sin embargo lleva implícita la decisión a priori de los pesos asociados a ambos objetivos, algo que no es para nada trivial. En contraposición, hemos optado por un sistema más flexible, que permita al usuario ponderar la prioridad que le confiere a la precisión y la reducción. De este modo, para una misma base de datos, el método elegido podría variar en función de sus intereses.

La arquitectura que proponemos (figura 4.2) se basa en el empleo de dos meta-modelos predictivos por cada método individual. Uno encargado de predecir el porcentaje de reducción y el otro el porcentaje de aciertos. Al presentarse una nueva base de datos al sistema, esta es procesada por una etapa de caracterización que es la encargada de obtener el meta-ejemplo que la describa. Este meta-ejemplo tiene como atributos las medidas propuestas en el capítulo 3, de modo que la etapa de caracterización que proponemos, lo que hace es calcular el valor de estas medidas para la base de datos de entrada.

Tomando el meta-ejemplo como entrada, por cada método individual ($Seli$), un meta-modelo predice la reducción que lograría en la base de datos de entrada (Ri), mientras otro predice la precisión (Pi). Con estas dos predicciones se mide el grado en que cada método satisface el objetivo del usuario (maximizar $b * Ri + (1 - b) * Pi$) y se aplica aquel que mayor grado obtiene ($Selmax$). Para combinar ambas predicciones se emplea el parámetro (b), que toma valores en el intervalo $[0,1]$ y que es usado en el criterio de elección del método para ponderar la prioridad que se le confiere a la precisión y la reducción. Fijando $b = 0$, el usuario indica que le confiere toda la prioridad a obtener la mayor precisión, mientras el valor 1 se la confiere a la reducción.

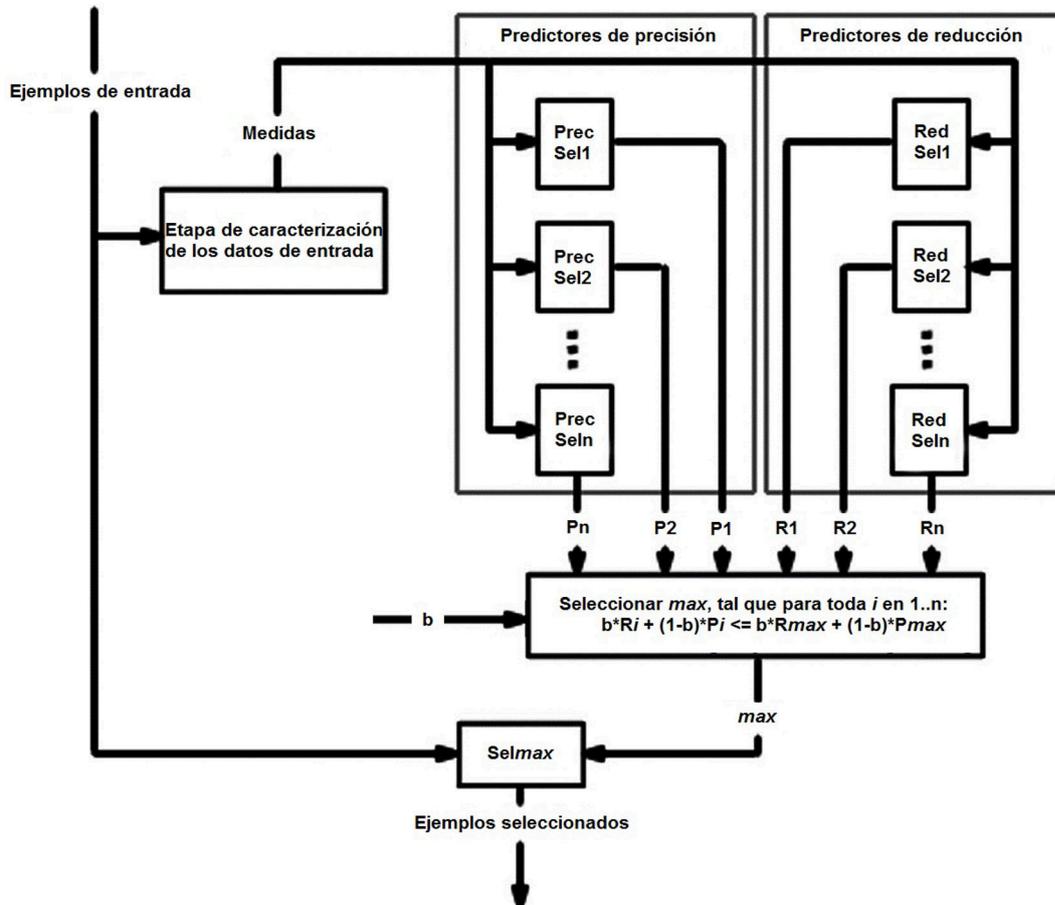


Figura 4.2: Arquitectura de un sistema de selección de ejemplos basado en conocimiento.

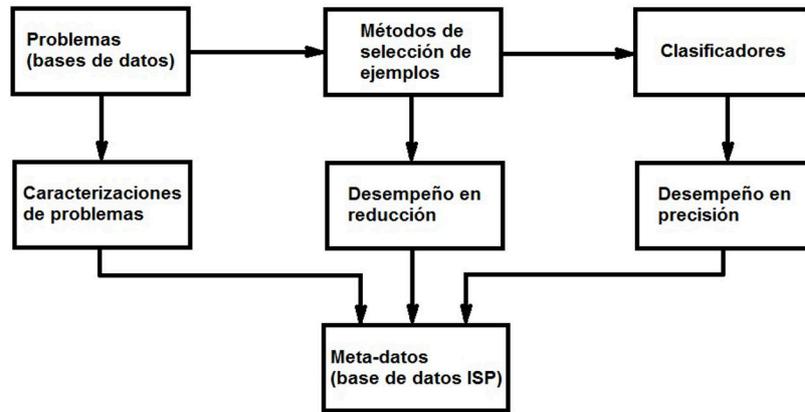


Figura 4.3: Esquema del proceso de construcción de la base de datos ISP.

4.2. Caracterización de datos y obtención de la base de datos ISP

Por supuesto, para que un sistema KBIS como el propuesto sea operativo, antes deben haberse aprendido los meta-modelos predictivos que usa¹. Esto nos remite a la primera fase del modelo descrito al inicio del capítulo (figura 4.1), a cuya etapa de caracterización dedicamos este epígrafe.

Esta etapa es la encargada de obtener a partir de los resultados de un conjunto de experimentos, el conjunto de meta-datos necesario para aprender los meta-modelos. A la base de datos que se obtiene como resultado de la etapa de caracterización la hemos llamado ISP (de Instance Selection Performance), en la figura 4.3 se muestra un esquema de su proceso de construcción.

El primer paso consistió en seleccionar un conjunto de bases de datos. Estas serían empleadas en experimentos que permitirían aprender acerca del desempeño de los diferentes métodos individuales de selección ante diversos dominios. El conjunto debería ser amplio y diverso, a la vez que debía estar formado por problemas reales en los que tuviera sentido el aprendizaje automático. Con esto se aumentarían las posibilidades de que los meta-modelos que posteriormente serían aprendidos, fueran válidos para nuevos problemas de este campo.

Una restricción adicional en la selección de las bases de datos fue que su tamaño no debía ser excesivamente grande, esto responde a una razón de índole más bien práctica. Cada una de ellas sería procesada por varios métodos de selección de ejemplos, lo que da lugar a un número bastante grande de ejecuciones y podría hacer inmanejable en un tiempo razonable el trabajo con bases de datos de muchos ejemplos.

Siguiendo estos criterios escogimos las bases de datos que se describen en la tabla 4.1, tomadas del

¹Restricción a la que antes nos referíamos al describir el modelo propuesto para el uso de meta-aprendizaje en la selección de ejemplos

repositorio KEEL [6]. Como puede apreciarse, existe entre ellas una amplia diversidad en el número de ejemplos, número y tipos de atributos, y número de clases. La cantidad (40) de bases de datos no es pequeña si atendemos al número de ellas que se suele emplear en la mayoría de los estudios experimentales en este campo. Sin embargo, consideramos que puede ser una cifra insuficiente de meta-ejemplos para extraer meta-modelos con la calidad necesaria. Esto nos enfrentó a algo que constituye un serio obstáculo en la actualidad para el meta-aprendizaje en sentido general, la escasa disponibilidad de bases de datos de problemas de clasificación reales.

Para ilustrar a lo que nos referimos, basta con analizar el conocido repositorio UCI [11], que contiene a nuestro juicio el más abarcador conjunto de bases de datos para la experimentación en el campo del aprendizaje y la minería de datos. Al consultar² el listado completo de bases de datos en este repositorio, nos encontramos que 133 de ellas corresponden a problemas de clasificación. De estas, 32 tienen 5000 o más ejemplos, lo que las hace demasiado grandes para nuestros objetivos por los motivos antes explicados. De modo que unas 101 bases de datos tendrían un tamaño manejable para el tipo de experimentos que nos ocupa. Sin embargo, entre estas, varias carecen de una representación matricial; lo que provoca que no las podamos emplear directamente pues trabajamos con métodos que asumen la representación vectorial de los ejemplos. Finalmente, podemos afirmar que el repositorio UCI sólo nos ofrece unas pocas decenas de bases de datos que se adecúan a nuestros objetivos. Si buscamos en otros repositorios, la heterogeneidad de las representaciones se multiplica y por demás, la mayoría están solapados en alto grado con UCI, por lo que aportan poco o nada al objetivo de aumentar el número de bases de datos útiles.

Atendiendo a esta situación, es que decidimos el empleo de las bases de datos que provee el repositorio KEEL, menos conocido que UCI, pero bastante abarcador, con mayor homogeneidad en su representación y la fortaleza indiscutible de formar parte de una plataforma [5] que integra además un conjunto bastante grande de algoritmos de minería de datos y cómodas herramientas para el diseño y ejecución de experimentos. La alternativa por la que optamos para aumentar el número de bases de datos fue emplear la clásica validación cruzada con 10 particiones, de modo que por cada una en realidad obtuvimos 10 meta-ejemplos, elevándose así hasta 400 el número total de estos en ISP.

Una vez seleccionadas las bases de datos, el proceso se ramifica en dos actividades independientes. Una de ellas es su caracterización, consistente en calcular los valores de las medidas propuestas en el capítulo 3 para cada una de ellas.

La otra actividad consistió en suministrar las bases de datos a los métodos individuales de selección de ejemplos; lo que implicó a su vez determinar, entre el amplio abanico de propuestas que existen en este campo, cuales incluiríamos en el estudio. Al hacer esta selección tratamos de incluir propuestas bien conocidas, que fueran eficientes computacionalmente y que constituyeran una muestra diversa en cuanto a sus estrategias de selección (tabla 4.2).

El criterio de eficiencia excluyó métodos como Explore y todos los enfoques evolutivos, con muy buenos desempeños en precisión y reducción, pero cuyas limitaciones en este sentido son conocidas como comentamos en el capítulo 2.

²Consulta hecha el 17 de diciembre de 2010

Tabla 4.1: Bases de datos empleadas para la construcción de la base de datos ISP. Las columnas Atrib, Cont, Ent y Nom, muestran las cantidades de atributos, y de atributos continuos, enteros y nominales respectivamente.

Nombre corto	Nombre	Ejemplos	Atrib	Cont	Ent	Nom	Clases
appendicitis	Appendicitis	106	7	7	0	0	2
automobile	Automobile	205	25	15	0	10	6
balance	Balance Scale	625	4	4	0	0	3
bands	Cylinder Bands	539	19	13	6	0	23
breast	Breast Cancer	286	9	0	0	9	2
car	Car Evaluation	1728	6	0	0	6	4
cleveland	Heart Disease (Cleveland)	303	13	13	0	0	5
contraceptive	Contraceptive Method Choice	1473	9	0	9	0	3
dermatology	Dermatology	366	34	0	34	0	6
ecoli	Ecoli	336	7	7	0	0	8
flare	Solar flare	1066	11	0	0	11	6
glass	Glass Identification	214	9	9	0	0	7
haberman	Haberman's survival	306	3	0	3	0	2
heart	Statlog (Heart)	270	13	1	12	0	2
hepatitis	Hepatitis	155	19	2	17	0	2
horse-colic	Horse Colic	368	23	7	1	15	2
housevotes	Congressional Voting Records	435	16	0	0	16	2
ionosphere	Ionosphere	351	33	32	1	0	2
iris	Iris plants	150	4	4	0	0	3
led7digit	LED Display	500	7	7	0	0	10
lymphography	Lymphography	148	18	0	3	15	4
mammographic	Mammographic Mass	961	5	0	5	0	2
monk-2	MONK's Problem 2	432	6	0	6	0	2
movement_libras	Libras Movement	360	90	90	0	0	15
newthyroid	Thyroid Disease (New Thyroid)	215	5	4	1	0	3
pima	Pima Indians Diabetes	768	8	8	0	0	2
saheart	South African Hearth	462	9	5	3	1	2
segment	Image Segmentation	2310	19	19	0	0	7
sonar	Sonar, Mines vs. Rocks	208	60	60	0	0	2
spectfheart	SPECTF Heart	267	44	0	44	0	2
splice	Splice-junction Gene Sequences	3190	60	0	0	60	3
tic-tac-toe	Tic-Tac-Toe Endgame	958	9	0	0	9	2
vehicle	Vehicle Silhouettes	846	18	0	18	0	4
vowel	Connectionist Bench (Vowel Recognition - Deterding Data)	990	13	10	3	0	11
wdbc	Breast Cancer Wisconsin (Diagnostic)	569	30	30	0	0	2
wine	Wine	178	13	13	0	0	3
winequality-red	Red Wine Quality	1599	11	11	0	0	11
wisconsin	Breast Cancer Wisconsin (Original)	699	9	0	9	0	2
yeast	Yeast	1484	8	8	0	0	10
zoo	Zoo	101	17	0	0	17	7

Tabla 4.2: Métodos individuales de selección de ejemplos incluidos en el estudio.

Método	Tipo de ejemplos que preserva	Estrategia de selección
CNN	Fronteras y ruido	Retener ejemplos mal clasificados por 1NN
coreHYB	Interiores típicos	Retener núcleos de clusters y ajustarlos con LVQ3
Cpruner	Fronteras sin ruido	Filtro de ruido + eliminar ejemplos bien clasificados por sus vecinos si no son decisivos en la correcta clasificación de otro ejemplo
DROP3	Fronteras sin ruido	Filtro de ruido + eliminar ejemplos que no son decisivos en la correcta clasificación de otro ejemplo
ENN	No ruidosos	Eliminar ejemplos mal clasificados por 3NN
IB3	Fronteras sin ruido	Retener ejemplos mal clasificados pero buenos clasificadores
ICF	Fronteras sin ruido	Filtro de ruido + eliminar ejemplos bien clasificados por más ejemplos que los que ellos clasifican correctamente

Decidimos incluir ENN, con muy buen desempeño en precisión, aunque muy pobre en reducción. Consideramos que el papel de este puede ser decisivo cuando el usuario pondere mucho más el primero de ambos objetivos que el segundo. Adicionalmente, tiene la ventaja de que al ser aplicado como filtro de ruido en el cálculo de las medidas, se puede determinar con certeza la reducción que provoca, y en caso de ser elegido no implica costo computacional adicional alguno.

Incluimos también a CNN, que es todo un clásico en el campo de la selección de ejemplos y con su enfoque muy simple, logra buenos resultados cuando el nivel de ruido en la base de datos es pequeño. Otro clásico (aunque un poco más reciente) incluido fue IB3, que presenta un excelente costo computacional, buen desempeño y una estrategia de selección muy particular. Otros métodos que forman parte de la selección son DROP3, ICF y Cpruner, todos tienen en común que sus estrategias se basan en la eliminación previa del ruido y la posterior retención de los ejemplos fronterizos; sin embargo cada uno lo consigue aplicando su propio enfoque particular y sus resultados difieren significativamente en muchas bases de datos.

Completa la lista HYB, aunque en este caso no empleamos para generar los code-books iniciales ningún método clásico, sino que tomamos los núcleos de los clusters construidos en la etapa de caracterización de los datos. Con esto pretendimos incluir un método que seleccionara (o mas bien construyera) ejemplos típicos con un enfoque más eficiente que el que aportan la mayoría de los métodos de esta categoría. Al igual que ENN, tiene la ventaja adicional de que se puede determinar con certeza la reducción que provoca. A esta combinación la hemos denominado *coreHYB*.

En todos los casos exceptuando el de coreHYB empleamos las implementaciones que provee la plataforma KEEL³ con los valores por defecto en todos sus parámetros. Para coreHYB fue necesario desarrollar una implementación propia⁴ con el fin de poder combinar HYB con el método de construcción de clusters presentado en el capítulo 3.

Para construir los meta-ejemplos, las bases de datos se partitionaron de acuerdo al esquema de la validación cruzada con diez particiones y cada cada conjunto de entrenamiento fue suministrado a cada método de selección de ejemplos. Llegado a este punto, ya se tienen dos componentes de ISP, un vector de caracterización por cada conjunto de entrenamiento y un valor de reducción para cada método en cada conjunto de entrenamiento. El próximo paso es procesar cada salida obtenida de los métodos de selección de ejemplos con algún clasificador para determinar la precisión que este logra sobre el correspondiente conjunto de prueba. Para este estudio escogimos KNN por su simplicidad y eficiencia computacional.

Finalmente, cada meta-ejemplo en ISP⁵ está asociado a un método de selección de ejemplos y contiene un atributo por cada medida (caracterización de un conjunto de entrenamiento), un atributo con el porcentaje de reducción logrado por el correspondiente método de selección de ejemplos en el conjunto que el ejemplo caracteriza, y un atributo con el porcentaje de aciertos obtenidos por KNN empleando la salida de este método de selección de ejemplos para clasificar al correspondiente conjunto de prueba.

4.3. Meta-aprendizaje

Luego de construirse la base de datos ISP, corresponde llevar a cabo la etapa de meta-aprendizaje, que es la encargada de obtener los meta-modelos empleados por el sistema KBIS para elegir el método individual adecuado para cada base de datos. La decisión más importante en esta etapa es qué algoritmo de aprendizaje emplear. No hay una respuesta obvia a la pregunta de qué método de aprendizaje es el más adecuado para un problema específico. Esto depende de varios factores incluyendo los intereses de los desarrolladores. De hecho, esta situación es la que da origen al campo del meta-aprendizaje [99].

Para decidir qué método emplear, seguimos el procedimiento de comparar el desempeño de varios de ellos por medio de la validación cruzada con 10 particiones. Algunos de los métodos analizados fueron seleccionados teniendo en cuenta el trabajo de Wu y otros [113] donde se presenta un listado con los 10 algoritmos mas influyentes en el campo de la minería de datos según un estudio donde se consultó la opinión de diversos expertos de reconocido prestigio en este campo. Entre

³Específicamente la versión 2010-11-12, disponible en <http://sci2s.ugr.es/keel/source/2010-11-12.zip>. Es importante señalar que en KEEL, las versiones de estos métodos (exceptuando a ENN) que aparecen en el software ejecutable están concebidas como clasificadores y llevan empotrado a KNN. Para nuestros propósitos empleamos las versiones generales (equivalentes a las anteriores) que aparecen en la categoría “preprocesamiento” en las fuentes, por lo que fue necesario compilarlas e incorporarlas al software ejecutable. Los archivos jar correspondientes, junto a las instrucciones para su incorporación a la plataforma están disponibles en <http://isg.ugr.es/descargas/descargar/138/>.

⁴Esta implementación es compatible con KEEL y puede descargarse junto con las instrucciones para su incorporación a la plataforma de <http://isg.ugr.es/descargas/descargar/141/>.

⁵Esta base de datos puede obtenerse en <http://isg.ugr.es/descargas/descargar/140/>.

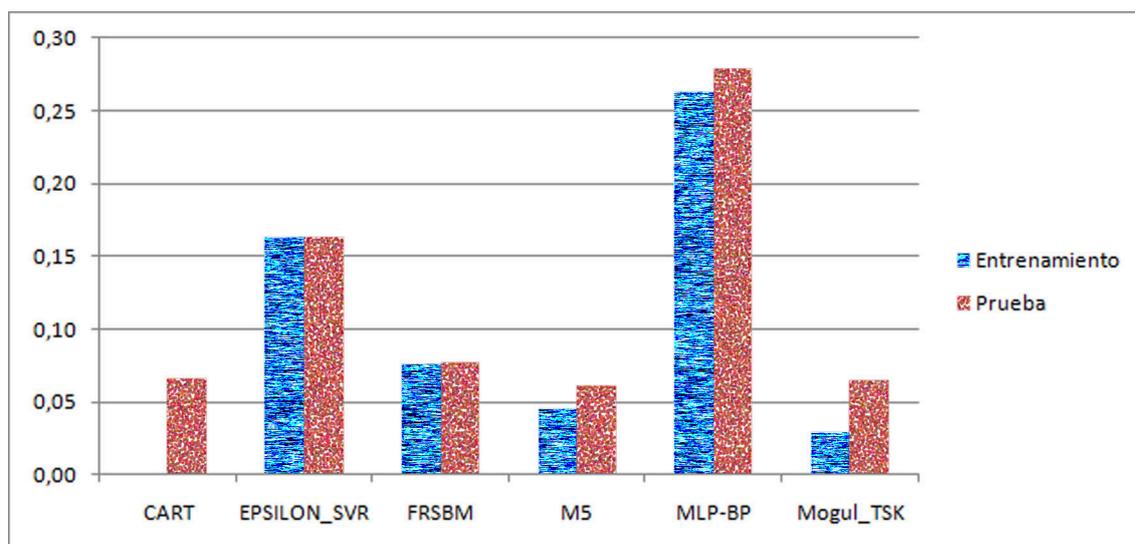


Figura 4.4: Desempeño de varios algoritmos de aprendizaje sobre la base de datos ISP empleando validación cruzada con 10 particiones. Promedio general por algoritmo de la raíz del error cuadrático medio en los conjuntos de entrenamiento y prueba.

estos 10 algoritmos, nos resultaban de utilidad sólo aquellos desarrollados para aprender modelos de regresión, y de estos, escogimos las implementaciones disponibles en KEEL: CART [18], EPSILON_SVR (máquinas de vectores de soporte) [37] y M5 (versión de C4.5 para regresión) [89]. Incluimos además a MLP-BP (Perceptrón multicapas) [94], FRSBM (aprendizaje de reglas difusas) [95] y MOGUL_TSK (aprendizaje genético de reglas difusas) [4] considerando que representan otros paradigmas de aprendizaje que exhiben buenos resultados en este campo. En todos los casos se emplearon los valores que KEEL asigna por defecto a sus parámetros.

La tabla 4.3 muestra la raíz del error cuadrático medio de cada algoritmo de aprendizaje con cada uno de los subconjuntos de ISP ⁶ y el gráfico de la figura 4.4 resume estos resultados por medio de sus promedios generales.

Dos algoritmos resaltan por sus malos desempeños, EPSILON_SVR y MLP-BP. en su defensa puede argumentarse que son precisamente los más sensibles a la elección de sus parámetros, que por demás son numerosos. No puede descartarse que un estudio más exhaustivo permita encontrar los valores adecuados para estos parámetros que conlleven un aprendizaje de mayor calidad. Sin embargo, las marcadas diferencias entre estos y los restantes métodos nos llevaron a considerar innecesario este esfuerzo por lo que los desechamos a ambos.

Entre los cuatro métodos con mejores resultados las diferencias son relativamente pequeñas, con los mejores resultados en los conjuntos de prueba para M5 seguido por MOGUL_TSK, mientras

⁶No confundir estos subconjuntos con las particiones producto de la validación cruzada. Cada subconjunto corresponde a un método de selección de ejemplos y a un objetivo (reducción o precisión), por ejemplo: accCNN contiene los ejemplos necesarios para aprender el meta-modelo de predicción de la precisión de CNN, mientras redCNN el de reducción del mismo método. Por cada uno de ellos se obtuvieron 10 particiones en la validación cruzada.

Tabla 4.3: Desempeño de varios algoritmos de aprendizaje sobre la base de datos ISP empleando validación cruzada con 10 particiones. Raíz del error cuadrático medio en los conjuntos de entrenamiento y prueba.

Base de datos	CART		EPSILON_SVR		FRSBM		M5		MLP-BP		Mogul_TSK	
	tra	tst	tra	tst	tra	tst	tra	tst	tra	tst	tra	tst
accCNN	0,0000	0,1051	0,2352	0,2349	0,0885	0,0908	0,0783	0,0909	0,2750	0,2945	0,0476	0,0938
accCoreHYB	0,0000	0,1042	0,1865	0,1865	0,1171	0,1194	0,0794	0,1355	0,3132	0,3379	0,0470	0,0975
accCpruner	0,0000	0,0972	0,2248	0,2249	0,1096	0,1115	0,0715	0,0918	0,3039	0,3308	0,0451	0,0826
accDROP3	0,0000	0,1158	0,1787	0,1782	0,0989	0,1008	0,0745	0,0857	0,2899	0,3061	0,0494	0,0959
accENN	0,0000	0,0811	0,1710	0,1710	0,0819	0,0835	0,0604	0,0805	0,3002	0,3162	0,0364	0,0763
accIB3	0,0000	0,1111	0,1943	0,1943	0,0939	0,0956	0,0805	0,0887	0,2948	0,3100	0,0510	0,0919
accICF	0,0000	0,1207	0,1782	0,1786	0,1043	0,1073	0,0842	0,0969	0,2828	0,3021	0,0570	0,1024
redCNN	0,0000	0,0334	0,1893	0,1895	0,0619	0,0634	0,0209	0,0350	0,1714	0,1797	0,0148	0,0253
redCoreHYB	0,0000	0,0101	0,0714	0,0715	0,0137	0,0141	0,0039	0,0056	0,2372	0,2526	0,0036	0,0571
redCpruner	0,0000	0,0261	0,0579	0,0580	0,0344	0,0350	0,0156	0,0228	0,3440	0,3598	0,0124	0,0635
redDROP3	0,0000	0,0229	0,1068	0,1069	0,0537	0,0544	0,0138	0,0201	0,2202	0,2191	0,0105	0,0349
redENN	0,0000	0,0231	0,1865	0,1868	0,0450	0,0458	0,0127	0,0218	0,2007	0,2082	0,0079	0,0187
redIB3	0,0000	0,0340	0,1360	0,1360	0,0692	0,0712	0,0225	0,0355	0,1693	0,1890	0,0149	0,0241
redICF	0,0000	0,0423	0,1704	0,1705	0,0935	0,0953	0,0179	0,0453	0,2820	0,3031	0,0152	0,0362
Promedio	0,0000	0,0662	0,1634	0,1634	0,0761	0,0777	0,0454	0,0611	0,2632	0,2792	0,0295	0,0643

Tabla 4.4: Desempeño de M5 y Mogul_TSK sobre la base de datos ISP. Raíz del error cuadrático medio en el conjunto de entrenamiento.

Base de datos	M5	Mogul_TSK
accCNN	0,0711	0,0488
acccoreHYB	0,0732	0,0496
accCpruner	0,0694	0,0458
accDROP3	0,0705	0,0519
accENN	0,0558	0,0360
accIB3	0,0811	0,0531
accICF	0,0805	0,0614
redCNN	0,0178	0,0150
redcoreHYB	0,0043	0,0036
redCpruner	0,0160	0,0124
redDROP3	0,0140	0,0114
redENN	0,0129	0,0078
redIB3	0,0214	0,0153
redICF	0,0186	0,0140
Promedio	0,0433	0,0304

con los conjuntos de entrenamiento el mejor desempeño lo alcanza CART con precisión total. Esto último, aunque a primera vista pudiera parecer un buen desempeño por parte de este método, no lo vemos como tal. La razón es que la combinación de esta ausencia de error en entrenamiento, con la mayor de las diferencias entre los desempeños de entrenamiento y prueba, es síntoma de un sobreajuste en el aprendizaje del meta-modelo.

Finalmente, atendiendo a los resultados obtenidos en la validación cruzada, decidimos desarrollar dos versiones del sistema; una crisp, donde los meta-modelos serían aprendidos por M5, y otra difusa, donde el algoritmo empleado sería MOGUL_TSK. La tabla 4.4 muestra la raíz del error cuadrático medio que estos obtuvieron al ser entrenados con cada base de datos, como es de esperar, los resultados fueron consistentes con los obtenidos en la validación cruzada.

4.4. Resultados

En este epígrafe analizaremos los resultados obtenidos por el sistema KBIS cuya concepción y entrenamiento hemos descrito. Mostraremos que estos superan a los de los métodos individuales que el sistema emplea, lo que consideramos un indicador de la utilidad de las medidas empleadas y de las potencialidades del meta-aprendizaje para el campo de la selección de ejemplos. Primeramente expondremos los resultados obtenidos con el clasificador KNN por ser este el empleado en la construcción de la base de datos ISP. Posteriormente analizaremos el comportamiento del sistema con otros clasificadores representativos de diversos paradigmas de aprendizaje.

Tabla 4.5: Ejemplo de tabla resumen de los tests de Holm de comparaciones múltiples entre varios métodos.

	veces mejor	veces peor
método ₁	m ₁	p ₁
método ₂	m ₂	p ₂
...
método _n	m _n	p _n

4.4.1. Diseño experimental

Los experimentos cuyos resultados mostramos en este epígrafe se han llevado a cabo siguiendo el procedimiento convencional de la validación cruzada con 10 particiones en la selección de ejemplos. Es decir, cada base de datos ha sido particionada en 10 subconjuntos, la unión de 9 de ellos (conjunto de entrenamiento) se ha suministrado a los métodos de selección de ejemplos y las salidas de estos a un clasificador que ha sido empleado para clasificar los ejemplos presentes en el décimo subconjunto (conjunto de prueba). Este procedimiento se ha repetido 10 veces de manera tal que cada subconjunto ha sido empleado como conjunto de prueba exactamente una vez. Las bases de datos empleadas en todos los casos han sido las presentadas previamente en la tabla 4.1.

Los parámetros que empleamos para medir el desempeño de cada método de selección de ejemplos son: el porcentaje de reducción logrado sobre el conjunto de entrenamiento, el porcentaje de aciertos (precisión) logrado por el clasificador sobre el conjunto de prueba al ser entrenado con la salida del método de selección, y el balance de ambos parámetros. Entendiendo este último como la combinación lineal

$$b * \text{reducción} + (1 - b) * \text{precisión}$$

el que además será analizado para distintos valores de b . En todos los casos, para cada base de datos el valor que se presenta de cada parámetro es el promedio obtenido en las 10 ejecuciones.

En el análisis de los resultados hemos empleado el test de Friedman [34] tomando como hipótesis nula la ausencia de diferencias en el desempeño de los distintos métodos. Para identificar estas diferencias una vez rechazada tal hipótesis, hemos empleado el test de Holm [34] para comparaciones múltiples entre varios métodos. En todos los casos fijamos el nivel de significación en $\alpha = 0,05$.

En cuanto a la visualización de los resultados de estos tests⁷, usamos el formato que se ilustra en la tabla 4.5, donde los valores m_i y p_i indican el número de métodos que fueron superados significativamente por método _{i} y el número de métodos que superaron significativamente a este respectivamente.

⁷A causa del gran número de ejecuciones llevadas a cabo, sólo se ofrecen en el capítulo las gráficas y tablas esenciales para presentar los resultados. Para ver estos en detalle puede remitirse al apéndice A, donde ofrecemos el resultado de cada método en cada base de datos, así como los ranking y demás datos de interés arrojados por los test estadísticos aplicados.

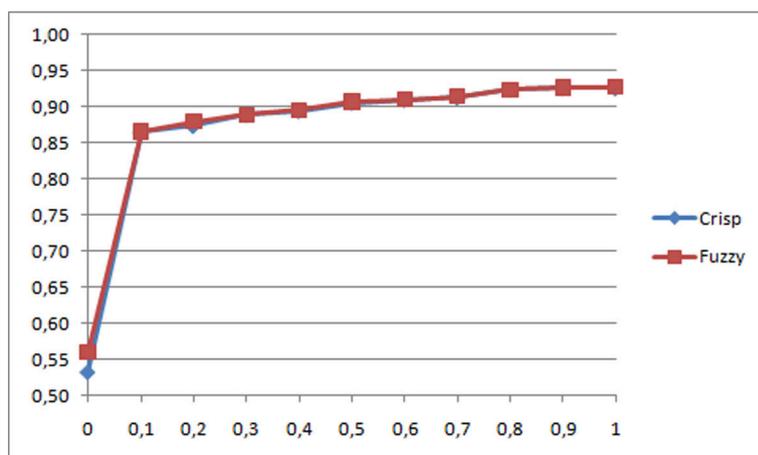


Figura 4.5: Reducción (eje vertical) de ambas versiones del sistema a distintos valores del parámetro b (eje horizontal).

4.4.2. Desempeño del sistema ante el clasificador KNN

Si el conjunto de medidas, el modelo de aplicación del meta-aprendizaje a la selección de ejemplos y la arquitectura para un sistema KBIS que hemos presentado son propuestas efectivas, es de esperar que el desempeño de este último supere al de cualquiera de los métodos individuales al seleccionar ejemplos para su uso por el clasificador KNN. Nos centramos en este clasificador, porque fue el empleado para crear el conjunto de datos con que se entrenó el sistema (ISP).

4.4.2.1. Reducción y precisión ante distintos valores del parámetro b

Para analizar como responden los valores de precisión y reducción ante distintos valores del parámetro b hemos ejecutado cada versión del sistema 11 veces, asignando a b valores desde 0 hasta 1 con un incremento de 0,1.

La figura 4.5 muestra los valores de reducción alcanzados por ambas versiones del sistema para los valores de b antes mencionados. Nótese como la mayor diferencia en el comportamiento de este parámetro entre dos valores consecutivos de b se observa entre 0 y 0,1. Este comportamiento responde a que al no conferírsele peso alguno a la reducción para $b = 0$, el sistema tiende a favorecer la elección del método ENN, con muy buenos resultados en precisión pero reducciones muy pobres. A partir de 0,1 se observa un crecimiento más suave de la reducción hasta alcanzar su máximo valor en 1.

De este gráfico pueden extraerse dos conclusiones fundamentales: el comportamiento de la reducción en función de b es coherente con el objetivo de este último y no muestra diferencias apreciables entre ambas versiones del sistema.

Por su parte la figura 4.6 muestra los valores de precisión alcanzados por KNN con las salidas de

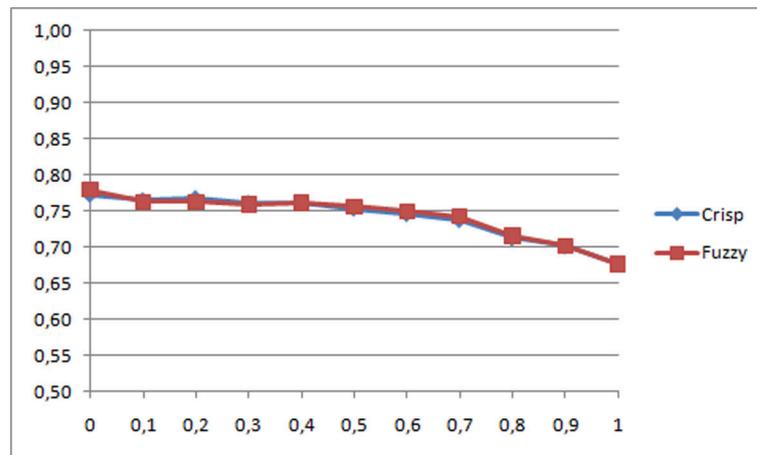


Figura 4.6: Precisión (eje vertical) de ambas versiones del sistema con KNN a distintos valores del parámetro b (eje horizontal).

ambas versiones del sistema para los mismos valores de b . En este caso no se observa ninguna variación brusca entre valores consecutivos de b y la tendencia es decreciente con un tramo central (entre $b = 0,1$ y $b = 0,7$) donde la pendiente es más suave que en los extremos. Nuevamente el comportamiento del parámetro en función de b es coherente con el objetivo de este último y no muestra diferencias apreciables entre ambas versiones del sistema.

Los resultados en ambos parámetros señalan que el sistema responde del modo esperado ante los distintos valores de b . A continuación analizaremos si esta respuesta es suficiente para superar los resultados de los métodos individuales.

4.4.2.2. Comparación con los métodos individuales empleados

En la comparación de los resultados de ambas versiones del sistema con los de los métodos individuales empleados hemos proyectado los resultados de estos últimos en reducción y precisión (con KNN) en las gráficas de las figuras 4.5 y 4.6 respectivamente. Obviamente, a diferencia de lo que ocurre con nuestra propuesta, los resultados en reducción y precisión de los métodos individuales se mantienen constantes independientemente del valor de b , simplemente porque para ellos no existe tal parámetro.

En el caso de la reducción (figura 4.7), puede apreciarse como a partir de $b = 0,1$ nuestra propuesta supera a todos, excepto dos de los restantes métodos (coreHYB y CPruner), a los que iguala o supera a partir de $b = 0,5$. Por su parte la precisión (figura 4.8), para valores de b inferiores a $0,7$ supera a todos los métodos individuales excepto a ENN, al que iguala en $b = 0$.

Observemos entonces estos resultados desde la perspectiva integral que ofrece el parámetro balance (figura 4.9). En este caso, los valores correspondientes a los métodos individuales no se mantienen constantes, sino que crecen o decrecen linealmente en función de b dependiendo de si es su reduc-

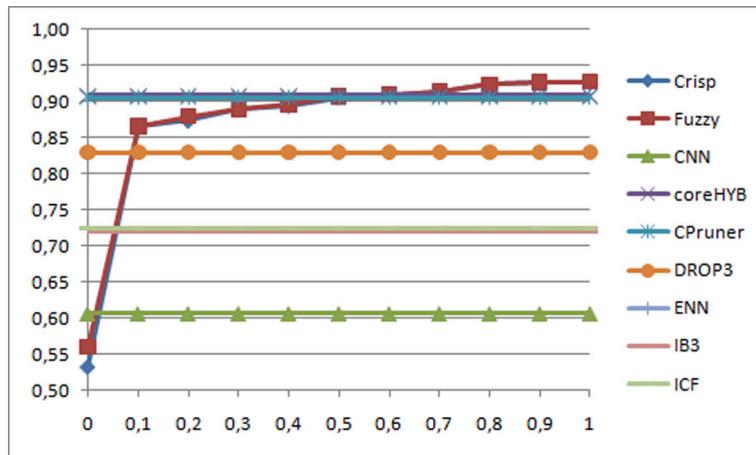


Figura 4.7: Niveles de reducción alcanzados por ambas versiones del sistema a distintos valores del parámetro b . Comparación con las reducciones de los métodos individuales que estas emplean. La reducción de ENN no aparece por ser inferior a 0,5, específicamente 0,2195.

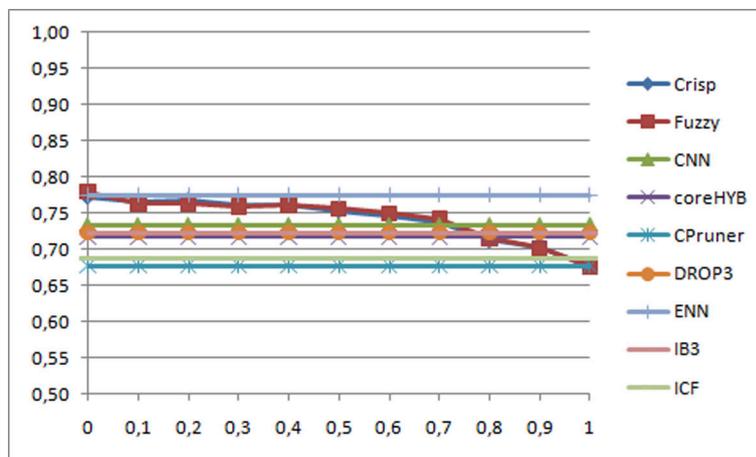


Figura 4.8: Niveles de precisión alcanzados por KNN empleando los resultados de ambas versiones del sistema a distintos valores del parámetro b . Comparación con las precisiones correspondientes a los métodos individuales que estas emplean.

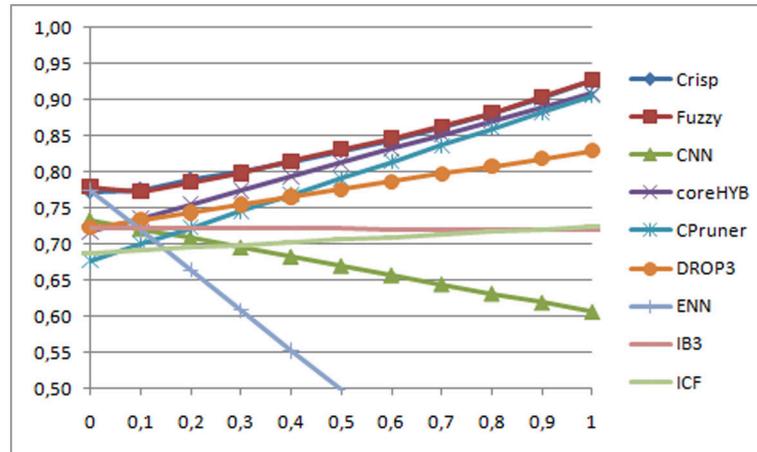


Figura 4.9: Niveles de balance alcanzados por ambas versiones del sistema (con KNN como clasificador) a distintos valores del parámetro b . Comparación con los balances correspondientes a los métodos individuales que estas emplean.

ción mayor o menor que su precisión. Esto se debe a que, aunque ambos parámetros tienen valores fijos para cada método individual independientemente de b , el peso que se le da a cada uno sí varía conforme este lo hace.

El primer aspecto relevante sobre el que queremos llamar la atención en esta gráfica es que para ningún valor de b los resultados de nuestra propuesta son superados por los de ningún método individual. Más aún, sólo tres métodos individuales (ENN, coreHYB y CPruner) parecen rivalizar con nuestra propuesta y esto sólo ocurre para algunos valores de b . Entre ellos, el método con un desempeño más equilibrado es coreHYB, pero aún así a medida que se le resta peso al valor de la reducción, las diferencias entre este y nuestra propuesta se hacen más marcadas a favor de esta última.

Otro aspecto interesante es que la mayor diferencia entre nuestra propuesta y el mejor de los restantes métodos no se da en los valores de b donde la primera tiene los mayores balances, sino entre $b = 0, 1$ y $b = 0, 2$. Precisamente allí donde ningún método individual es claramente superior al resto y se da una especie de convergencia entre los que tienen una tendencia creciente respecto a b y los que la tienen decreciente.

Consideramos que es en estos dos valores donde con más claridad se pone de manifiesto el poder discriminante de los meta-modelos, por el hecho de que en promedio ningún método individual supera claramente al resto y sólo una correcta elección del método a nivel de base de datos puede justificar que nuestra propuesta muestre ventajas tan claras.

Ahora bien ¿son realmente significativas las diferencias que se aprecian?. Para responder esta pregunta llevamos a cabo un conjunto de test estadísticos no paramétricos sobre los datos correspondientes al desempeño de todos los métodos para $b = 0, 1$ y $b = 0, 2$. Estos test, como comentamos al inicio del epígrafe, fueron el de Friedman y el de Holm, recomendados por Demsar [34] para comparaciones que involucren múltiples clasificadores en múltiples bases de datos. En todos los

Tabla 4.6: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en reducción con $b = 0,1$. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	4	CNN	0	5
coreHYB	5	0	coreHYB	5	0
CPruner	4	0	CPruner	4	0
DROP3	4	1	DROP3	4	1
ENN	0	6	ENN	0	6
IB3	1	4	IB3	2	4
ICF	1	4	ICF	1	4
Crisp	4	0	Fuzzy	4	0

Tabla 4.7: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en reducción con $b = 0,2$. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	4	CNN	0	5
coreHYB	5	0	coreHYB	5	0
CPruner	4	0	CPruner	4	0
DROP3	4	1	DROP3	4	1
ENN	0	6	ENN	0	6
IB3	1	4	IB3	2	4
ICF	1	4	ICF	1	4
Crisp	4	0	Fuzzy	4	0

casos que presentaremos el primero de ellos permitió rechazar la hipótesis de igualdad, por lo que nos centraremos en exponer los resultados de los test de Holm.

Siguiendo el mismo orden que en los análisis anteriores comenzaremos por comentar las tablas 4.6 y 4.7, donde se resumen los tests de Holm aplicados a los resultados en reducción con $b = 0,1$ y $b = 0,2$ respectivamente. Los resultados a primera vista parecen corroborar el buen desempeño de nuestra propuesta. Para ninguno de los dos valores de b ni ninguna de las dos versiones implementadas, hubo ningún método individual que superara a nuestra propuesta. Sin embargo, un análisis más detenido pudiera sembrar cierta duda acerca de su supremacía si consideramos que tres de los métodos tampoco son superados por esta. Incluso uno de ellos (coreHYB) es el que globalmente supera a más métodos.

En realidad, si en apariencia no ha quedado clara la ventaja de nuestra propuesta es porque hemos enfocado sólo parcialmente el problema que nos ocupa. No debemos perder de vista que ni la re-

Tabla 4.8: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,1$ y el clasificador *KNN*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	2	CNN	0	2
coreHYB	0	1	coreHYB	0	1
CPruner	0	2	CPruner	0	2
DROP3	0	2	DROP3	0	2
ENN	6	0	ENN	6	0
IB3	0	2	IB3	0	2
ICF	0	2	ICF	0	2
Crisp	5	0	Fuzzy	5	0

Tabla 4.9: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,2$ y el clasificador *KNN*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	2	CNN	0	1
coreHYB	0	2	coreHYB	0	1
CPruner	0	2	CPruner	0	2
DROP3	0	2	DROP3	0	2
ENN	6	0	ENN	6	0
IB3	0	2	IB3	0	2
ICF	0	2	ICF	0	2
Crisp	6	0	Fuzzy	4	0

ducción, ni la precisión aisladas pueden darnos una idea del desempeño de un método de selección de ejemplos.

Analícemos entonces las tablas 4.8 y 4.9, donde se resumen los tests de Holm aplicados a los resultados en precisión (con *KNN*) para $b = 0,1$ y $b = 0,2$ respectivamente. Nuevamente, para ninguno de los dos valores de b ni ninguna de las dos versiones implementadas, hubo ningún método individual que superara a nuestra propuesta. En este caso el método individual que parece competir con esta es ENN, pero precisamente es ahí donde radica la gran ventaja del uso del meta-aprendizaje ¡en reducción ENN fue el método de peores resultados! ¿y qué sucede con los métodos que mostraron buenos resultados en ese parámetro?, pues no son capaces de superar a ningún otro método en precisión.

Esto evidencia que nuestra propuesta fue la única que logró un equilibrio entre los dos objetivos, lo que se percibe con total claridad cuando analizamos los test correspondientes al balance (tablas 4.10 y 4.11), donde esta no es superada por ningún método, supera a 6 de ellos y el que más se le

Tabla 4.10: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,1$ y el clasificador KNN . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	1	CNN	0	1
coreHYB	2	0	coreHYB	2	0
CPruner	0	1	CPruner	0	1
DROP3	0	1	DROP3	0	1
ENN	0	2	ENN	0	2
IB3	0	1	IB3	0	1
ICF	0	2	ICF	0	2
Crisp	6	0	Fuzzy	6	0

Tabla 4.11: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,2$ y el clasificador KNN . Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	2	CNN	0	2
coreHYB	3	0	coreHYB	3	0
CPruner	1	1	CPruner	1	1
DROP3	2	1	DROP3	2	1
ENN	0	5	ENN	0	5
IB3	1	1	IB3	1	1
ICF	0	3	ICF	0	3
Crisp	6	0	Fuzzy	6	0

acerca en desempeño sólo supera a 3.

Los elementos expuestos confirman nuestra hipótesis de que el conjunto de medidas propuestas permite explotar el meta-aprendizaje en beneficio de una selección de ejemplos adecuada a las características de cada base de datos. Hemos podido apreciar cómo el sistema KBIS implementado fue capaz de obtener resultados iguales o mejores que los de cualquiera de los métodos individuales en todos los valores de b , y lo que es más significativo, en los valores donde la selección del mejor método individual se torna más compleja fue precisamente donde más ventaja obtuvo nuestra propuesta.

A continuación analizaremos si estos resultados son generalizables a otros clasificadores no involucrados en la construcción de la base de datos ISP.

4.4.3. Desempeño del sistema ante otros clasificadores

Como explicamos al describir el proceso de construcción de la base de datos ISP, decidimos emplear en este estudio como medida de desempeño en precisión, los resultados obtenidos por KNN al usar las salidas de los métodos individuales de selección de ejemplos para clasificar los ejemplos de los correspondientes conjuntos de prueba. Hasta ahora hemos analizado como se comporta el sistema KBIS obtenido en reducción, precisión y balance, entendiendo por precisión precisamente el porcentaje de aciertos asociado a KNN. Sin embargo es interesante estudiar si el buen desempeño mostrado se extiende también a otros clasificadores.

En principio nada garantiza que esto ocurra, en caso de ser así, un sólo meta-modelo para predecir la precisión de cada método individual permitiría buenos resultados con la mayoría de los clasificadores. De lo contrario, sería necesario el aprendizaje de meta-modelos específicos por cada clasificador o grupo de clasificadores con desempeños similares.

Los clasificadores que incluimos en este estudio fueron C4.5 [90], LDA [78], y PART [40], en representación de tres conocidos paradigmas de aprendizaje: árboles de decisión, aprendizaje estadístico y reglas respectivamente. Atendiendo al precedente observado con KNN, decidimos centrar el análisis en los resultados obtenidos para $b = 0, 1$ y $b = 0, 2$. Las figuras 4.10 y 4.11 muestran los promedios generales de las precisiones obtenidas por estos clasificadores con las salidas de cada método de selección de ejemplos, también aparecen las obtenidas por KNN para facilitar el contraste de los resultados.

Como puede apreciarse, tanto LDA como PART logran peores precisiones que KNN con todos los métodos de selección de ejemplos. En el caso de C4.5, los resultados con ENN, CNN, ICF e IB3 son iguales o mejores que los de KNN, mientras con los otros métodos se aprecia un peor desempeño. El clasificador donde mayores diferencias se aprecian entre los resultados asociados a los distintos métodos es LDA. En cuanto a nuestra propuesta, a diferencia de lo que ocurre con KNN, ambas versiones se ven superadas por varios métodos con estos tres clasificadores. Estos resultados parecen indicar que la precisión asociada a KNN no es un buen estimador para el trabajo con otros clasificadores.

Finalmente compararemos los resultados desde el punto de vista del balance, los que aparecen gráficamente representados en las figuras 4.12 y 4.13. Las gráficas muestran que combinar precisión y reducción en un sólo parámetro tiene el efecto de atenuar las diferencias entre los diferentes métodos de selección de ejemplos en todos los clasificadores. Esto se debe fundamentalmente a que los métodos con los mejores resultados en uno de los dos parámetros no suelen tener buenos resultados en el otro. Tales son los casos por ejemplo de Cpruner y ENN, el primero se destaca en reducción y el segundo en precisión, pero son superados por los otros métodos en la precisión y reducción respectivamente.

Independientemente de lo anterior, es notable la diferencia entre lo que ocurre con KNN y el resto de los clasificadores estudiados. Mientras en el primero, ambas versiones de nuestra propuesta muestran una clara ventaja respecto a los métodos individuales, en los otros clasificadores se comportan como uno más de ellos, incluso con resultados inferiores a la media en el caso LDA.

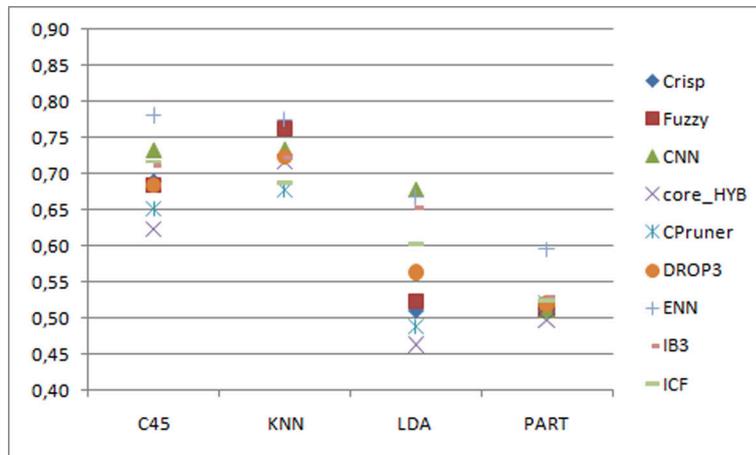


Figura 4.10: Niveles de precisión alcanzados por varios clasificadores empleando los resultados de ambas versiones del sistema con $b = 0,1$. Comparación con las precisiones correspondientes a los métodos individuales que estas emplean.

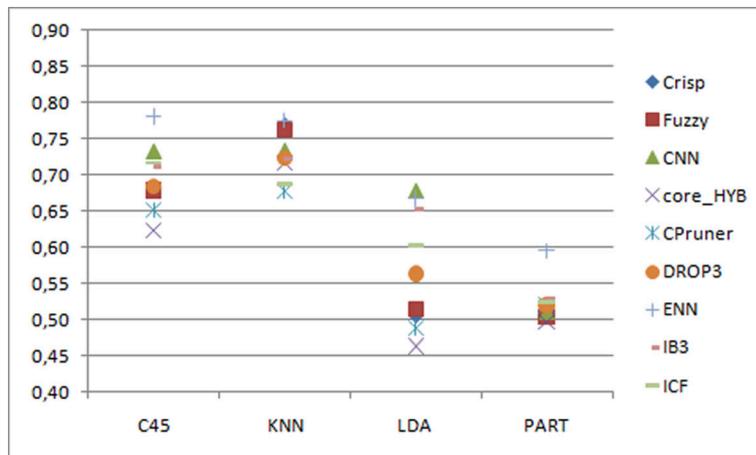


Figura 4.11: Niveles de precisión alcanzados por varios clasificadores empleando los resultados de ambas versiones del sistema con $b = 0,2$. Comparación con las precisiones correspondientes a los métodos individuales que estas emplean.

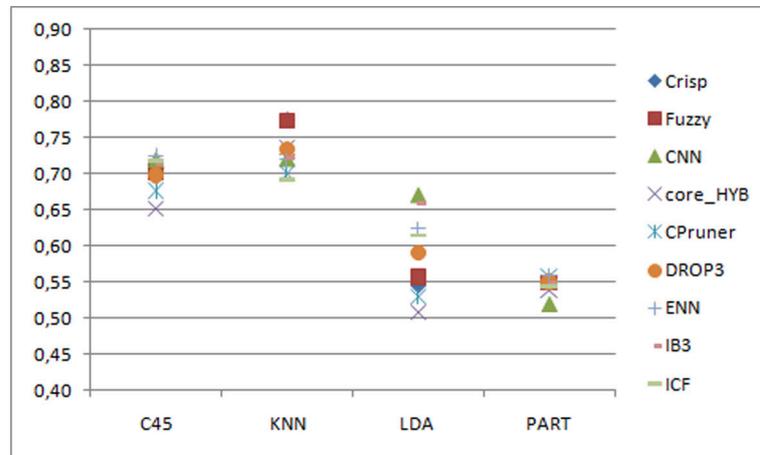


Figura 4.12: Niveles de balance alcanzados por ambas versiones del sistema (con varios clasificadores) para $b = 0,1$. Comparación con los balances correspondientes a los métodos individuales que estas emplean.

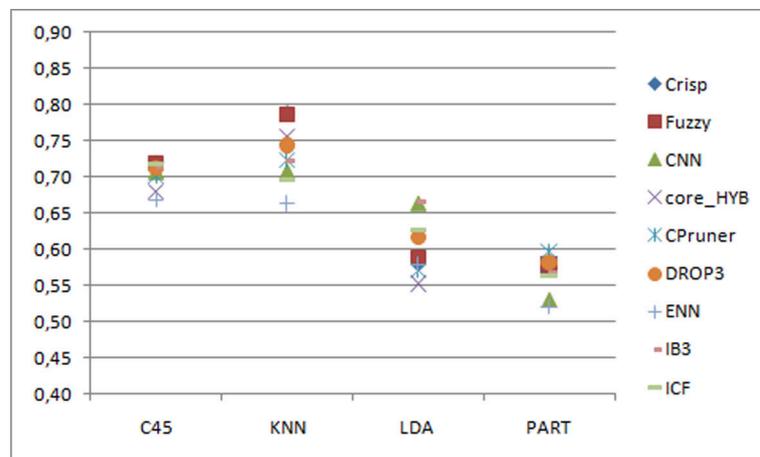


Figura 4.13: Niveles de balance alcanzados por ambas versiones del sistema (con varios clasificadores) para $b = 0,2$. Comparación con los balances correspondientes a los métodos individuales que estas emplean.

4.5. Conclusiones Parciales

En este capítulo hemos presentado un modelo para el uso del meta-aprendizaje en la selección de ejemplos así como la arquitectura de un sistema de selección de ejemplos basado en el conocimiento (KBIS). Empleando las medidas propuestas en el capítulo 3 hemos implementado el modelo y la arquitectura para obtener un sistema KBIS, con el que desarrollamos estudios que nos han permitido llegar a algunas conclusiones de vital importancia para esta investigación.

En primer lugar, comprobamos que es posible extraer meta-modelos de predicción acertados a partir de bases de datos donde las medidas propuestas en el capítulo 3, junto al desempeño experimentado por diversos métodos de selección de ejemplos, dan lugar a meta-ejemplos que describen el comportamiento de estos últimos. De los seis algoritmos de aprendizaje analizados para aprender los meta-modelos, cuatro se comportaron satisfactoriamente en la validación cruzada y no puede descartarse que los otros dos, con un ajuste adecuado de sus parámetros también lo hagan.

Al emplear dos de estos algoritmos en el entrenamiento del sistema, obtuvimos dos versiones del mismo, una crisp y otra difusa. Los experimentos mostraron que el desempeño de ambas es equivalente. Al analizar el balance, que combina linealmente los dos objetivos básicos de la selección de ejemplos (reducción y precisión) con un parámetro (b) que pondera el peso relativo de cada uno de ellos, para todos los valores de b estudiados la propuesta obtiene resultados superiores a todos los métodos individuales que emplea. Este comportamiento fue más significativo allí donde precisamente es más útil, en los valores de b donde ningún método individual aventaja claramente en promedio al resto.

Finalmente, comprobamos que estos resultados no son directamente generalizables a otros clasificadores distintos al empleado en la creación del conjunto de meta-ejemplos, lo que sugiere la necesidad de contar con meta-ejemplos y aprender meta-modelos específicos para cada clasificador o grupo de ellos con desempeños similares.

Capítulo 5

Reutilizando en la Selección las Estructuras de Datos Creadas en la Etapa de Caracterización

En este capítulo proponemos varios métodos de selección de ejemplos basados en conjuntos locales y clusters. Al emplear estos en la instanciación de la arquitectura presentada en el capítulo anterior, se explota una de las potencialidades que presenta el sistema de medidas propuesto en esta memoria; la reutilización de gran parte de las estructuras de datos creadas durante su cómputo. Con esto se gana en eficiencia; pero más aún, en la mayoría de los casos se hace innecesario estimar la reducción que logrará el método en la base de datos por disponerse del valor exacto que esta tendrá, con lo que se reduce el margen de error al estimar el correspondiente *balance*. Los nuevos métodos además, por no estar diseñados con el propósito de cubrir individualmente el mayor número de dominios posible, tienen la potencialidad de ser más especializados y el resultado es que su aplicación selectiva ofrece excelentes resultados en disímiles entornos.

5.1. Nuevos métodos de selección de ejemplos basados en conjuntos locales y clusters

Al implementar los métodos de cómputo para las medidas propuestas el capítulo 3, incorporamos métodos de búsqueda rápida para aumentar su eficiencia computacional. Otra manera de hacer mas eficiente nuestra propuesta, es reducir el tiempo de ejecución de los métodos de selección en sí. Una alternativa para lograrlo es aprovechando la información contenida en las estructuras de datos que se crean en la etapa de caracterización, o sea: los conjuntos locales y los clusters. En la propuesta presentada en el capítulo anterior, la mayoría de los métodos de selección ignoran la información contenida en estas estructuras. Es decir, estamos ante una arquitectura con dos etapas totalmente independientes (caracterización y selección) cada una de las cuales tiene un costo computacional en el orden de $O(mn^2)$.

Resulta lógico suponer que un esquema donde todas las alternativas por las que pueda optar el sistema, aprovechen la información contenida en las estructuras de datos previamente creadas por la etapa de caracterización, debe ser en la mayoría de los casos más rápido que el sistema anterior. Esto no significa que el costo total del sistema sea de un orden inferior a $O(mn^2)$, pero está claro que el tiempo real de ejecución sí debe ser menor, lo que es vital para el tipo de problemas que se pretende resolver.

Ya en el capítulo anterior, al seleccionar los métodos individuales que incluiríamos en el sistema, planteábamos que dos de ellos mostraban estas características. Se trata de ENN y coreHYB. En el caso del primero, su aplicación como filtro de ruido en la etapa de caracterización hace que, de ser elegido, ya el proceso de selección esté efectuado y el costo adicional sea nulo. Por su parte, coreHYB toma los núcleos de los clusters construidos en la etapa de caracterización como entrada para HYB, con lo que el costo de obtener los code-books es absorbido por esta. En ambos casos se añade la ventaja de que la reducción que logrará el sistema no necesita ser estimada, con lo que se reduce el margen de error asociado a la estimación del balance.

Aunque ENN logra niveles de precisión muy buenos, reduce muy poco la base de datos. Esto hace que su utilidad en el sistema se limite a los casos donde el usuario elija valores de b muy cercanos a 0. El método coreHYB por el contrario, logra un buen balance para prácticamente todos los valores de b .

Con estos métodos ya tenemos una base de partida para nuestro objetivo, pero sólo dos métodos no ofrecen la diversidad que necesita un sistema KBIS. El reto entonces es encontrar nuevos métodos de selección de ejemplos que, tomando como entrada un conjunto de ejemplos con sus correspondientes conjuntos locales y el conjunto de clusters obtenidos a partir de ellos, haga una adecuada selección en un tiempo menor que $O(mn^2)$. A diferencia de los métodos del estado del arte, estos no necesitan ser efectivos en todos los dominios sino que basta con que para cada dominio al menos uno de ellos sea suficientemente efectivo. Esta característica encierra una gran potencialidad, pues permite diseñar métodos especializados y complementarios, cuya aplicación selectiva en el marco de la arquitectura que proponemos ofrezca excelentes resultados en disímiles entornos.

5.1.1. Selección de núcleos

La idea más simple y eficiente consiste en seleccionar los núcleos de los clusters. Su eficiencia es insuperable porque no requiere cómputo alguno, pero además tiene una ventaja colateral para la arquitectura que proponemos, no es necesario estimar qué reducción producirá. De este modo sólo hay que predecir su precisión lo que reduce el margen de incertidumbre al seleccionar el método.

Este método, al que llamaremos “core”, se caracteriza por obtener niveles de reducción muy altos. En bases de datos donde la separación entre las clases sea significativa esta puede ser una alternativa muy efectiva, aunque en general los resultados de precisión que logra son inferiores a muchos métodos del estado del arte. Una manera de mejorar estos resultados de precisión manteniendo los excelentes niveles de reducción es refinando el resultado por medio de un post-procesamiento con LVQ3, que no es más que el ya presentado coreHYB.

La tercera alternativa basada en selección de núcleos que hemos incluido consiste en retener estos sólo si no constituyen puntos invasores. A esta variante la llamaremos “coreNI”, y puede tener dos efectos fundamentales. El primero es cuando la mayoría de los clusters son de alta cardinalidad pero unos pocos puntos invasores (y por ende poco representativos) distorsionan los resultados al clasificar, en este caso se espera una mejora poco significativa en la reducción pero un aumento en la precisión. El segundo es en bases de datos con un elevado número de puntos invasores, donde la reducción mejora significativamente si estos se ignoran, aunque el impacto en la precisión puede ser tanto positivo como negativo. Esta técnica también puede ver mejorados en algunos casos sus niveles de precisión por medio del post-procesamiento con LVQ3, lo que da lugar a un cuarto método de esta familia, que llamaremos “coreNIHYB”.

En la figura 5.1 se muestra la base de datos Banana (del repositorio KEEL [6]) en su estado original (5.1(a)) y luego de ser procesada por ENN y los métodos antes descritos (5.1(b)-5.1(f)). Nótese como ENN elimina prácticamente todo el solapamiento entre las clases mientras apenas afecta las regiones homogéneas, cumpliendo así su papel como filtro de ruido. Sin este filtrado, nuestra estrategia de agrupamiento obtendría muchos clusters en las regiones de solapamiento.

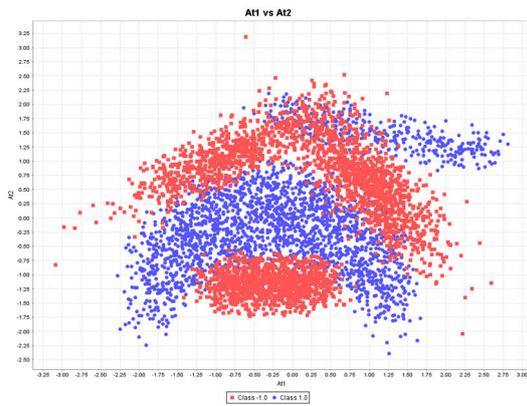
La selección de núcleos (figura 5.1(c)) por su parte arroja un resultado interesante. Como puede apreciarse no sólo retiene ejemplos prototípicos en el sentido tradicional de la palabra, sino que una buena parte de los puntos seleccionados están muy cercanos a las fronteras de clases. Este comportamiento no es para nada una anomalía, se debe a que al estar tan cercanas entre sí las clases, en las fronteras se ubican muchos ejemplos que no son absorbidos por ninguno de los grandes clusters con núcleos en las regiones centrales. Estos ejemplos se agrupan en pequeños clusters que describen las irregularidades en forma de bahías que se presentan en las fronteras.

En situaciones como la antes descrita, suele ocurrir que una parte de los clusters fronterizos son tan pequeños que sólo contienen a su núcleo. Si son elegidos estos ofrecen, en el mejor de los casos, muy poco aporte a la precisión en la clasificación pues son muy poco representativos. En otros casos más perjudiciales, pueden tener una influencia negativa comportándose de modo muy similar al ruido. La estrategia de selección de núcleos no invasores (figura 5.1(d)) puede ser muy efectiva para contrarrestar este efecto.

Finalmente, como puede apreciarse en las figuras 5.1(d) y 5.1(f), el post-procesamiento con LVQ3 en ambos casos tiene el efecto de ensanchar las fronteras y distribuir más uniformemente los puntos seleccionados, aunque siempre respetando las regiones originalmente presentes.

5.1.2. Selección de puntos fronterizos

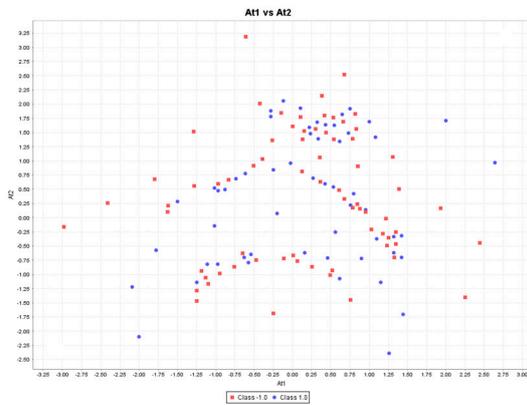
Los cuatro métodos basados en selección de núcleos logran niveles de reducción muy buenos, pero en muchos casos los métodos del estado del arte son claramente mejores que todos ellos en términos de precisión. Por su parte ENN como ya hemos planteado, por su pobre reducción ve limitada su efectividad a valores de b muy cercanos a 0. Por supuesto, si se pretende que el sistema elija en cada caso el método adecuado, debe contar con opciones que se adapten a una amplia gama de contextos. De modo que es necesario incluir al menos otro método que sea efectivo allí donde fracasan los anteriores.



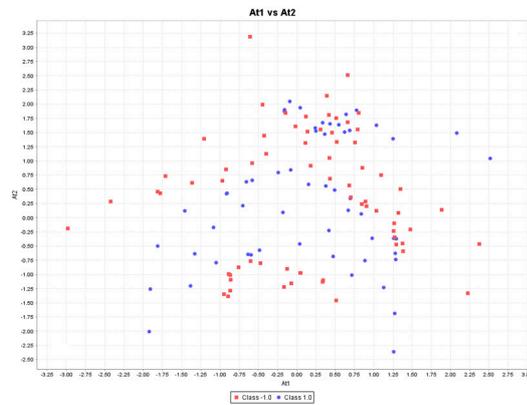
(a) Base de datos banana.



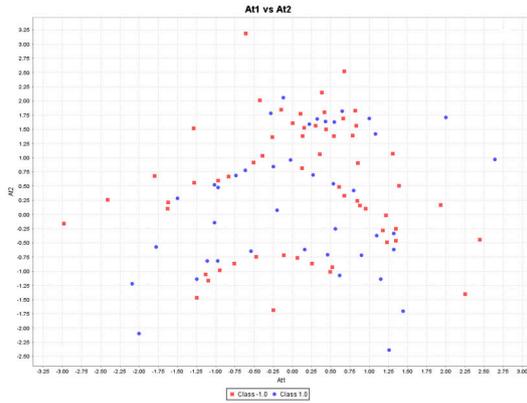
(b) Banana. Procesada por ENN.



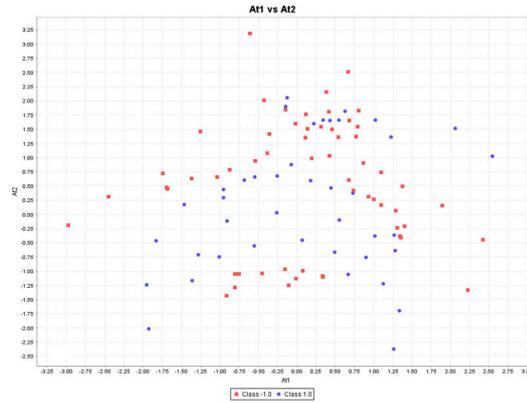
(c) Banana. Procesada por core.



(d) Banana. Procesada por coreHYB.



(e) Banana. Procesada por coreNI.



(f) Banana. Procesada por coreNIHYB.

Figura 5.1: Resultados de someter la base de datos Banana a ENN y distintas estrategias de selección de núcleos.

```

procedure GetBorderPoints(T)
begin
   $S = \{\}$ 
   $Sort(T)$  //ordenar los ejemplos ascendentemente por cardinalidad de sus conjuntos locales
  foreach( $e \in T$ )
    if( $e.LS \cap S = \{\}$ )
      //Agregar ejemplo para garantizar su correcta clasificación
       $S = S \cup \{e\}$ 
  return  $S$ 
end

```

Figura 5.2: Procedimiento para la selección de ejemplos fronterizos. Dado un ejemplo e , $e.LS$ denota su conjunto local.

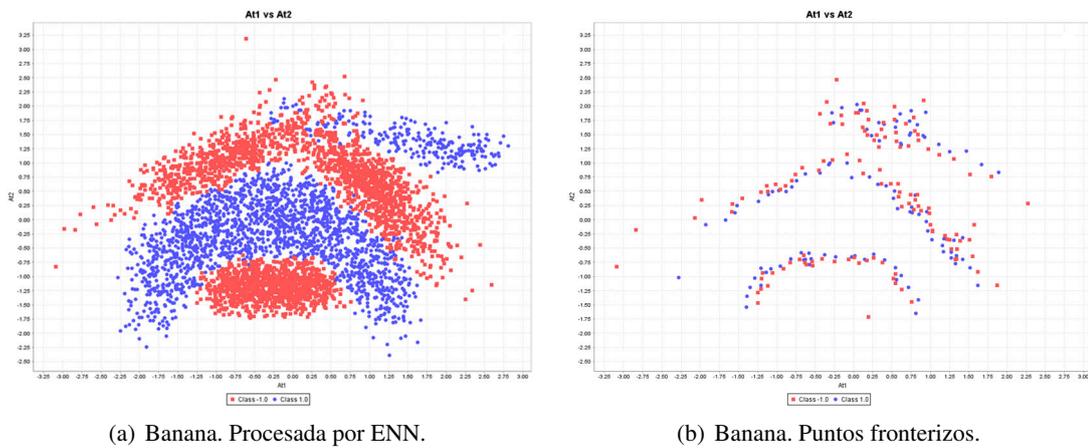


Figura 5.3: Base de datos Banana procesada por ENN y por el método de selección de puntos fronterizos.

Puesto que todos estos salvo ENN se basan en la elección de puntos prototípicos, es lógico pensar que es necesario abordar la selección de puntos fronterizos, máxime cuando algunos de los métodos del estado del arte con mejores resultados emplean esta estrategia de selección ¹. Para esto diseñamos un método que hemos denominado “ClusterBorder” y emplea una regla de selección similar a la de CNN, pero evitando el cálculo de distancias lo que la hace más eficiente. También es necesario tener presente que aunque CNN tiene como principal deficiencia su conocida sensibilidad al ruido, este método procesará bases de datos que han sido sometidas previamente a ENN quedando así libres de ruido.

La figura 3.10 muestra el procedimiento encargado de seleccionar los ejemplos fronterizos. El ordenamiento inicial tiene como objetivo comenzar el procesamiento por aquellos ejemplos más cercanos a las fronteras de clases, favoreciendo así su inclusión en S , este se puede implementar con un costo $O(n \log n)$. En el bucle que aparece a continuación, para cada ejemplo e en T se verifica si algún miembro de su conjunto local ($e.LS$) ya está contenido en S , lo que garantizaría su correcta clasificación. De no ocurrir esto, se incluye e en S garantizando así que sea correctamente clasificado.

Es conocido que la intersección de dos conjuntos A y B tiene un costo $O(|A| * |B|)$, sin embargo en este caso se puede lograr con costo lineal simplemente marcando los ejemplos una vez que son añadidos a S . Así para verificar si la intersección de este conjunto y $e.LS$ es vacía no es necesario recorrer el primero, basta con comprobar si están marcados o no los ejemplos del segundo. De este modo, como tampoco es necesario el cálculo de distancia alguna, el costo del bucle es $O(n * LS \text{ card})$ donde $LS \text{ card}$ es la cardinalidad media de los conjuntos locales que suele ser muy inferior a n .

En la figura 5.3 puede apreciarse el efecto de aplicar esta estrategia de selección. Nótese como quedan perfectamente delineadas las fronteras mientras prácticamente ningún punto interior es conservado. A diferencia de las estrategias anteriores, consideramos que en este caso no tiene mucho sentido el procesamiento de los resultados con LVQ3 pues precisamente el efecto sería perder la nitidez de las fronteras.

Este método produce una reducción algo menor que los basados en núcleos y su precisión generalmente es mejor. A diferencia de estos, para él sí es necesario estimar la reducción que producirá, pues no hay manera directa de calcularla a partir de los datos obtenidos al caracterizar la base de datos. Finalmente, sus resultados tanto en reducción como en precisión son similares a los de los métodos del estado del arte lo que puede considerarse bueno pues el objetivo no es que individualmente los supere.

5.1.3. Estudio del costo computacional

Al inicio del presente epígrafe planteábamos dos motivaciones fundamentales para emplear los métodos descritos en un sistema KBIS que emplee las medidas propuestas en el capítulo 3. Una de ellas es que en la mayoría de los casos no es necesario estimar la reducción que producirán, lo que es consecuencia lógica de su propia definición. La otra motivación está relacionada con el aumento de la eficiencia computacional que puede lograrse al aprovechar una de las potencialidades de estas

¹DROP3 y C-pruner son ejemplos de ello

medidas que ya anunciábamos en dicho capítulo, reutilizar las estructuras de datos obtenidas en su cómputo para la propia selección de ejemplos.

A continuación analizaremos mediante un estudio empírico si efectivamente estos métodos logran traducir esa potencialidad en una mayor eficiencia computacional. Con este fin, compararemos el tiempo de ejecución de cada uno de ellos en las 40 bases de datos empleadas en los experimentos precedentes (véase tabla 4.1) con el de los métodos convencionales incluidos en el sistema presentado en el capítulo 4 que no reutilizan ninguna de las estructuras de datos creadas al computar las medidas.

Es preciso aclarar que estos tiempos se corresponden con la ejecución de cada uno de los métodos como parte de un sistema KBIS basado en estas medidas, por lo que en todos los casos está descontado el costo de computar las mismas. Por este motivo, en el caso de los métodos convencionales el tiempo reflejado coincide con el de su aplicación aislada, pero los métodos que reutilizan estructuras de datos creadas al computar las medidas tienen asociado un tiempo menor si forman parte de un sistema de estas características que si se aplican de manera independiente. La razón de este comportamiento es que estamos analizando el tiempo adicional al del cómputo de las medidas, asociado a cada método.

Entre los métodos que analizamos, hay tres cuyo costo asociado es nulo, estos son: ENN, core y coreNI. La razón es que el cálculo de las medias lleva implícita la ejecución de estos métodos, de manera que su aplicación posterior no implica costo adicional alguno. En estos casos es evidente que su uso implica una eficiencia máxima. Veamos entonces que ocurre con los otros tres: ClusterBorder, coreHYB y coreNIHYB.

La figura 5.4 muestra los tiempos de ejecución promedio de estos métodos así como los de CNN, Cpruner, DROP3, IB3 e ICF en las condiciones antes explicadas, mientras en la figura 5.5 aparecen los valores por cada base de datos en orden creciente del costo promedio. Como puede apreciarse, el costo de ClusterBorder es mucho menor que el de cualquier otro método, con lo que su eficiencia resulta incuestionable. Sin embargo, los casos de coreHYB y coreNIHYB merecen un análisis más cuidadoso, pues son superados claramente por tres de los métodos convencionales. Esto pudiera sugerir que no es conveniente su empleo, siendo más eficiente el uso de cualquiera de los métodos que los aventajan en costo. Sin embargo, hay que tener en cuenta que se trata de métodos que no sólo seleccionan ejemplos, sino que además los modifican buscando un posicionamiento óptimo de los mismos en el espacio m -dimensional donde están definidos.

Esta clase de métodos, conocidos como “creativos” [59] o “de generación de prototipos” [105] se caracteriza por tener un costo computacional relativamente alto. Recientemente, Triguero y otros [105] publicaron un abarcador estudio que incluye 24 métodos de esta familia y en él puede comprobarse como ante bases de datos de más de 2000 ejemplos sólo tres de los métodos tuvieron tiempos de ejecución claramente inferiores a HYB, ninguno de estos sin embargo estuvo entre los 10 mejores en precisión. En ese contexto, obtener métodos de esta familia a un costo inferior al de CNN puede considerarse un resultado bastante eficiente. Claro está, una posibilidad sería no incluir ningún método de generación de prototipos, sin embargo consideramos que su capacidad de generar soluciones inalcanzables para los métodos de selección “pura” representa una ventaja a la que no se debe renunciar.

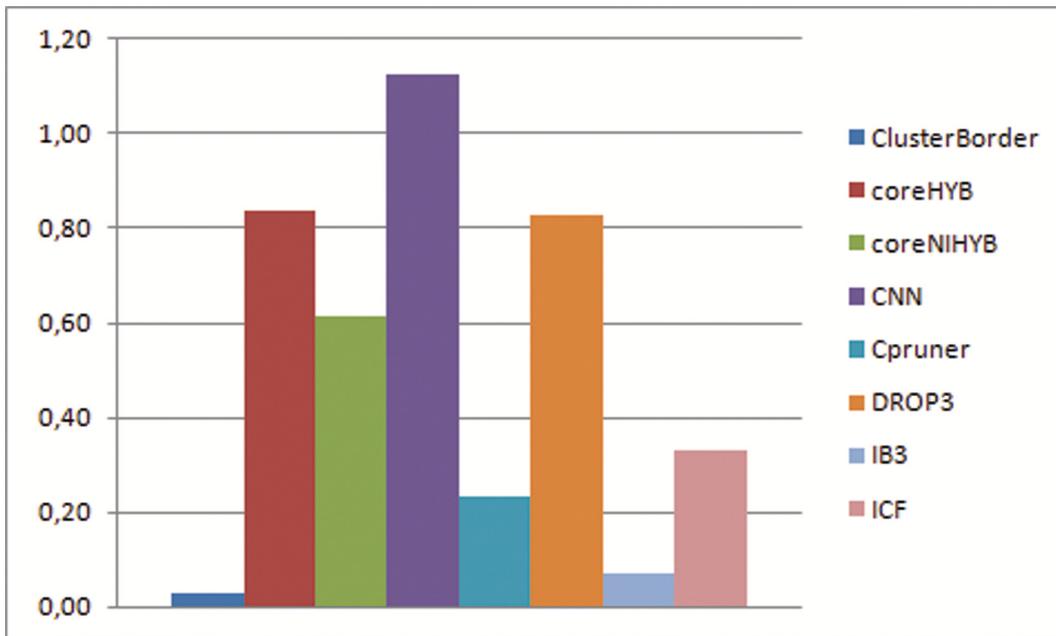


Figura 5.4: Promedio de los tiempos de ejecución (en segundos) sobre 40 bases de datos de tres de los métodos que reutilizan las estructuras de datos creadas durante el cómputo de las medidas y de los cuatro métodos empleados en el capítulo 4 que no las aprovechan.

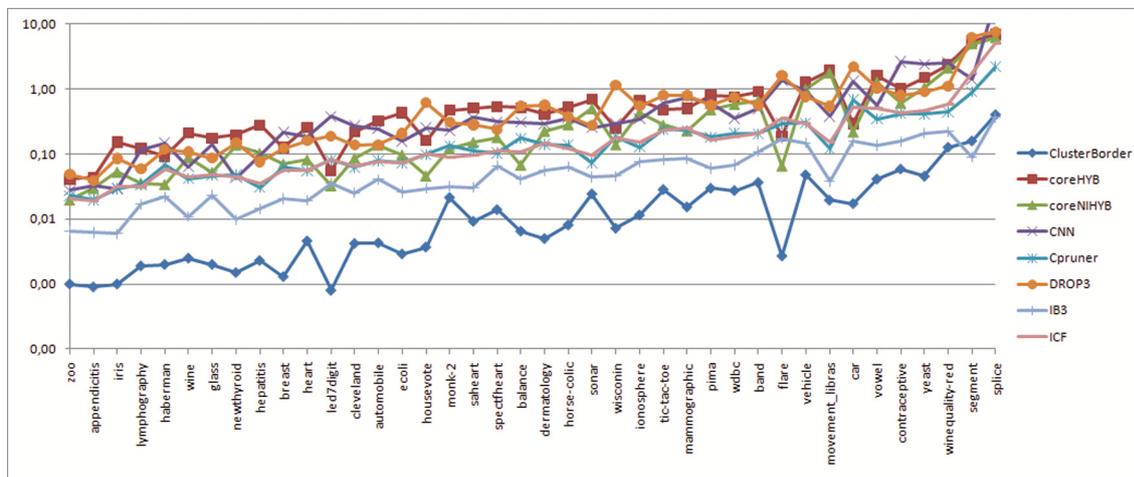


Figura 5.5: Tiempos de ejecución (en segundos) sobre 40 bases de datos de tres de los métodos que reutilizan las estructuras de datos creadas durante el cómputo de las medidas y de los cuatro métodos empleados en el capítulo 4 que no las aprovechan. Nótese que en el eje vertical la escala empleada es logarítmica.

5.2. Meta-aprendizaje. Aprendiendo meta-modelos específicos para cada clasificador

En el capítulo 4 se evidenció la necesidad de aprender meta-modelos específicos para cada clasificador o grupo de ellos con desempeños similares, por tanto para el sistema KBIS desarrollado a partir de los métodos antes descritos hemos introducido una modificación a la arquitectura propuesta con anterioridad. Esta puede apreciarse en la figura 5.6, y consiste en la inclusión del clasificador para el cual se pretenden seleccionar los ejemplos como un parámetro. Esto permite emplear meta-modelos específicos para cada clasificador, pero a su vez implica el aprendizaje de un mayor número de ellos.

En realidad consideramos que no es necesario aprender meta-modelos para cada clasificador, pues es de esperar que entre estos puedan encontrarse clusters con desempeños similares ante bases de datos y métodos de selección similares. De este modo, los meta-modelos de un clasificador serían aplicables a todos los clasificadores de su cluster. Sin embargo, la verificación de esta hipótesis requiere un extenso estudio que se sale del alcance de este trabajo.

En este caso hemos seleccionado cuatro clasificadores para la construcción de la base de datos ISP, estos son: C4.5, KNN, LDA y PART. En su elección no empleamos como criterio sus desempeños, pues el objetivo no es compararlos entre sí. En lugar de esto, tuvimos en cuenta que representaran distintos paradigmas de aprendizaje, aceptaran atributos nominales, enteros y reales, fueran métodos bien conocidos y estuvieran implementados en la plataforma KEEL. Esta última condición la consideramos de gran utilidad práctica, tanto para desarrollar nuestros experimentos, como para facilitar la reproducción de estos a otros investigadores interesados en el tema.

Las bases de datos usadas para construir los meta-ejemplos son las mismas empleadas en el capítulo 4 (tabla 4.1), no así los métodos individuales de selección de ejemplos, que obviamente fueron los presentados en el epígrafe 5.1.

Siguiendo el procedimiento descrito en el capítulo 4 se construyó la nueva versión de la base de datos ISP a la que llamaremos ISP2 en lo sucesivo. Cada meta-ejemplo en ISP2² no sólo está asociado a un método de selección de ejemplos, sino también a un clasificador.

En el meta-aprendizaje como tal, seguimos el mismo procedimiento que con ISP para seleccionar los algoritmos de aprendizaje y empleamos los mismos candidatos (ver epígrafe 4.3). La tabla 5.1 muestra la raíz del error cuadrático medio de cada algoritmo de aprendizaje con cada uno de los subconjuntos de ISP2 y el gráfico de la figura 5.7 resume estos resultados por medio de sus promedios generales. Como puede apreciarse, al igual que en el caso previo, los algoritmos con mejores desempeños son M5 y MOGUL_TSK. Sin embargo en este caso el valor del error para todos los algoritmos es mayor, por lo que analizaremos a continuación las causas de este comportamiento aparentemente peor.

En ISP2 sólo aparece un subconjunto asociado a la predicción de la reducción (redCB, asociado a ClusterBorder), el resto de los casos se omitió por no ser necesario predecir la reducción, pero esto significa que el error en todos ellos sería 0. En la precisión, el error es mayor que en la reducción

²Esta base de datos puede obtenerse en <http://isg.ugr.es/descargas/descargar/142/>.

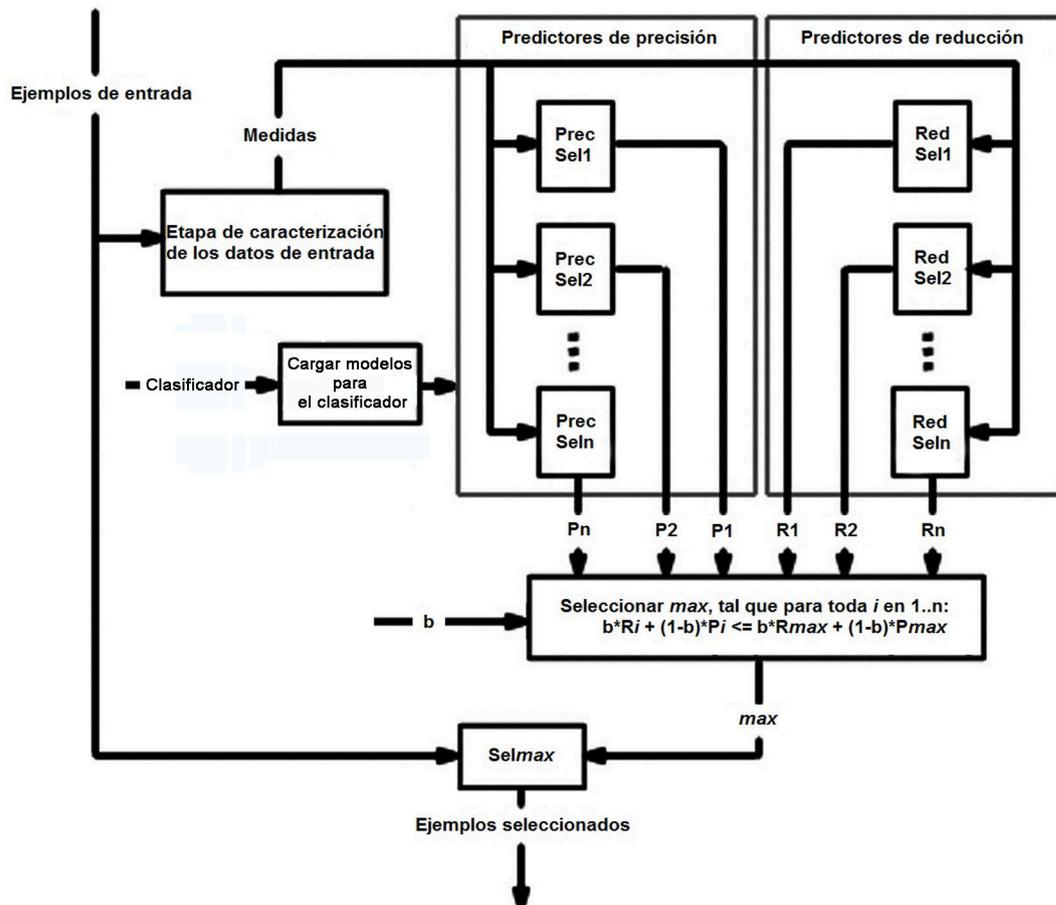


Figura 5.6: Arquitectura de un sistema KBIS con selección del clasificador. Nótese que a diferencia de la versión original propuesta en el capítulo 4, en este caso hay un nuevo parámetro que indica el clasificador para el cual se pretenden seleccionar los ejemplos.

Tabla 5.1: Desempeño de varios algoritmos de aprendizaje sobre la base de datos ISP2 empleando validación cruzada con 10 particiones. Raíz del error cuadrático medio en los conjuntos de entrenamiento y prueba. Salvo *redCB* que corresponde a la reducción lograda por ClusterBorder, todos los subconjuntos se corresponden con la precisión asociada a un clasificador y un método de selección.

Base de datos	CART		EPSILON_SVR		FRSBM		M5		MLP-BP		Mogul_TSK	
	tra	tst	tra	tst	tra	tst	tra	tst	tra	tst	tra	tst
c45CB	0,0000	0,1421	0,1951	0,1951	0,1355	0,1381	0,0968	0,1232	0,3398	0,3556	0,0642	0,1400
c45core	0,0000	0,1503	0,1885	0,1883	0,1404	0,1430	0,1092	0,1252	0,2925	0,3103	0,0724	0,1339
c45coreHYB	0,0000	0,1530	0,1877	0,1873	0,1410	0,1444	0,1079	0,1237	0,2982	0,3155	0,0718	0,1375
c45coreNIHYB	0,0000	0,1570	0,1942	0,1940	0,1467	0,1504	0,1122	0,1274	0,2928	0,3126	0,0675	0,1396
c45coreNI	0,0000	0,1438	0,1876	0,1876	0,1403	0,1439	0,1006	0,1237	0,2791	0,3028	0,0671	0,1273
c45ENN	0,0000	0,0799	0,1623	0,1622	0,0767	0,0790	0,0685	0,0782	0,3286	0,3346	0,0381	0,0715
knnCB	0,0000	0,1053	0,2203	0,2204	0,1181	0,1202	0,0718	0,0946	0,3218	0,3423	0,0461	0,0883
knncore	0,0000	0,1300	0,1715	0,1715	0,1205	0,1229	0,0886	0,1106	0,3171	0,3502	0,0587	0,1050
knncoreHYB	0,0000	0,1083	0,2009	0,2011	0,1148	0,1176	0,0758	0,0929	0,3246	0,3339	0,0477	0,0970
knncoreNIHYB	0,0000	0,1085	0,1889	0,1890	0,1195	0,1220	0,0695	0,0840	0,2812	0,3021	0,0467	0,0949
knncoreNI	0,0000	0,1364	0,1823	0,1828	0,1249	0,1274	0,0920	0,1167	0,2986	0,3144	0,0594	0,1206
knnENN	0,0000	0,0835	0,1710	0,1710	0,0820	0,0835	0,0628	0,1041	0,3213	0,3418	0,0367	0,0763
ldaCB	0,0000	0,1516	0,2934	0,2934	0,2277	0,2329	0,0739	0,2181	0,3493	0,3535	0,0465	0,1007
ldacore	0,0000	0,1350	0,2649	0,2649	0,2321	0,2366	0,0615	0,1489	0,4933	0,5172	0,0457	0,1086
ldacoreHYB	0,0000	0,1239	0,2663	0,2663	0,2347	0,2393	0,0769	0,2784	0,4844	0,5044	0,0477	0,1072
ldacoreNIHYB	0,0000	0,1200	0,2690	0,2689	0,2355	0,2405	0,0622	0,1310	0,4965	0,5186	0,0410	0,0903
ldacoreNI	0,0000	0,1321	0,2687	0,2687	0,2342	0,2392	0,0638	0,1296	0,5016	0,5257	0,0412	0,0799
ldaENN	0,0000	0,1422	0,3319	0,3319	0,2094	0,2132	0,0916	0,1226	0,3097	0,3417	0,0448	0,1643
partCB	0,0000	0,1863	0,2255	0,2254	0,1514	0,1539	0,1351	0,1470	0,3053	0,3163	0,0828	0,1459
partcore	0,0000	0,1636	0,2364	0,2364	0,1596	0,1633	0,1186	0,1413	0,3386	0,3587	0,0784	0,1570
partcoreHYB	0,0000	0,1641	0,2349	0,2349	0,1657	0,1699	0,1121	0,1405	0,3435	0,3580	0,0752	0,1558
partcoreNIHYB	0,0000	0,1635	0,2286	0,2286	0,1632	0,1664	0,1109	0,1369	0,3396	0,3446	0,0760	0,1453
partcoreNI	0,0000	0,1621	0,2327	0,2327	0,1594	0,1629	0,1208	0,1403	0,3464	0,3544	0,0751	0,1472
partENN	0,0000	0,1347	0,2237	0,2237	0,1426	0,1454	0,0883	0,1136	0,3101	0,3199	0,0588	0,1122
redCB	0,0000	0,0268	0,1616	0,1617	0,0348	0,0361	0,0091	0,0167	0,2548	0,2769	0,0075	0,0168
Promedio	0,0000	0,1322	0,2195	0,2195	0,1524	0,1557	0,0872	0,1268	0,3427	0,3603	0,0559	0,1145

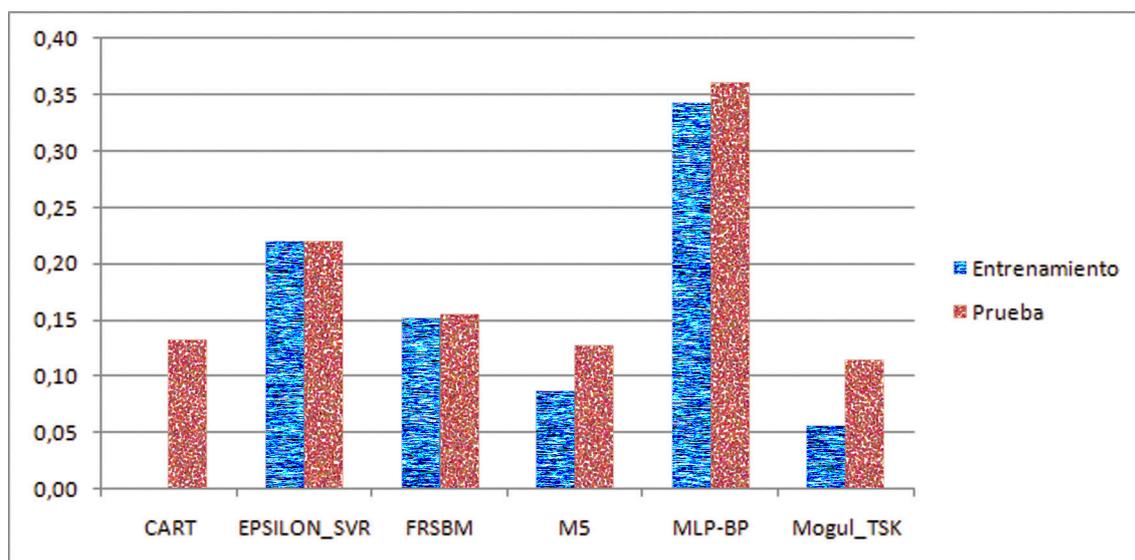


Figura 5.7: Desempeño de varios algoritmos de aprendizaje sobre la base de datos ISP2 empleando validación cruzada con 10 particiones. Promedio general por algoritmo de la raíz del error cuadrático medio en los conjuntos de entrenamiento y prueba.

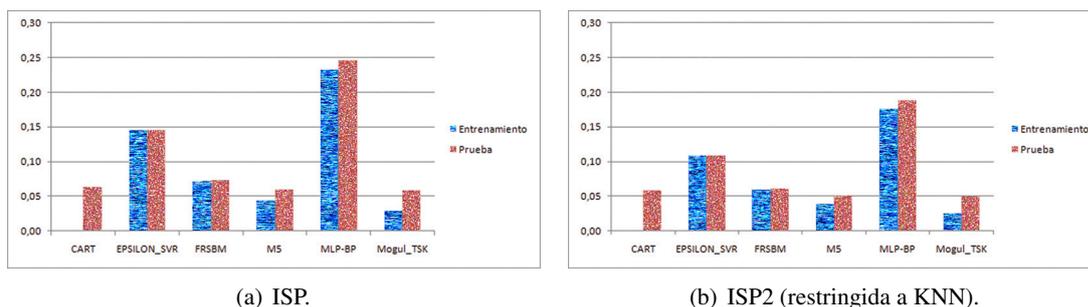
como puede comprobarse tanto en la tabla 4.3 como en la 5.1, con lo que una mayor proporción de casos asociados a la primera evidentemente implicará un mayor error promedio. Este efecto se acentúa más porque en ISP2 se incluyen nuevos clasificadores donde el error es mayor que en KNN, el único empleado en ISP.

Para poder hacer una comparación justa entre los errores cuadráticos medios en ISP y en ISP2, hemos restringido el análisis en la segunda a los subconjuntos asociados a KNN y hemos añadido los casos correspondientes a la reducción, donde el error es 0. La tabla 5.2 muestra los resultados desde esta perspectiva, mientras la figura 5.8 muestra las gráficas correspondientes a los promedios del error cuadrático medio en ISP e ISP2 restringida a KNN y considerando los casos asociados a la reducción. En esta última puede comprobarse que en realidad los resultados de todos los algoritmos son superiores con ISP2.

Finalmente, al igual que ocurrió con ISP, con ISP2 los resultados de la validación cruzada, indican la conveniencia de emplear M5 para la versión crisp, y MOGUL_TSK para la versión difusa del sistema KBIS. La tabla 5.3 muestra la raíz del error cuadrático medio que ambos algoritmos obtuvieron al ser entrenados con cada base de datos, como puede apreciarse estos son consistentes con los obtenidos en la validación cruzada.

Tabla 5.2: Desempeño de varios algoritmos de aprendizaje sobre las bases de datos ISP e ISP2 (restringida a KNN) empleando validación cruzada con 10 particiones. A los casos donde se puede determinar con certeza el valor esperado, se les ha asignado error 0.

Base de datos	CART		EPSILON_SVR		FRSBM		M5		MLP-BP		Mogul_TSK		
	tra	tst	tra	tst	tra	tst	tra	tst	tra	tst	tra	tst	
ISP	accCNN	0,0000	0,1051	0,2352	0,2349	0,0885	0,0908	0,0783	0,0909	0,2750	0,2945	0,0476	0,0938
	acccoreHYB	0,0000	0,1042	0,1865	0,1865	0,1171	0,1194	0,0794	0,1355	0,3132	0,3379	0,0470	0,0975
	accCpruner	0,0000	0,0972	0,2248	0,2249	0,1096	0,1115	0,0715	0,0918	0,3039	0,3308	0,0451	0,0826
	accDROP3	0,0000	0,1158	0,1787	0,1782	0,0989	0,1008	0,0745	0,0857	0,2899	0,3061	0,0494	0,0959
	accENN	0,0000	0,0811	0,1710	0,1710	0,0819	0,0835	0,0604	0,0805	0,3002	0,3162	0,0364	0,0763
	accIB3	0,0000	0,1111	0,1943	0,1943	0,0939	0,0956	0,0805	0,0887	0,2948	0,3100	0,0510	0,0919
	accICF	0,0000	0,1207	0,1782	0,1786	0,1043	0,1073	0,0842	0,0969	0,2828	0,3021	0,0570	0,1024
	redCNN	0,0000	0,0334	0,1893	0,1895	0,0619	0,0634	0,0209	0,0350	0,1714	0,1797	0,0148	0,0253
	redcoreHYB	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	redCpruner	0,0000	0,0261	0,0579	0,0580	0,0344	0,0350	0,0156	0,0228	0,3440	0,3598	0,0124	0,0635
redDROP3	0,0000	0,0229	0,1068	0,1069	0,0537	0,0544	0,0138	0,0201	0,2202	0,2191	0,0105	0,0349	
redENN	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	
redIB3	0,0000	0,0340	0,1360	0,1360	0,0692	0,0712	0,0225	0,0355	0,1693	0,1890	0,0149	0,0241	
redICF	0,0000	0,0423	0,1704	0,1705	0,0935	0,0953	0,0179	0,0453	0,2820	0,3031	0,0152	0,0362	
Promedio	0,0000	0,0638	0,1449	0,1449	0,0719	0,0734	0,0443	0,0592	0,2319	0,2463	0,0287	0,0589	
ISP2	knnCB	0,0000	0,1053	0,2203	0,2204	0,1181	0,1202	0,0718	0,0946	0,3218	0,3423	0,0461	0,0883
	knncore	0,0000	0,1300	0,1715	0,1715	0,1205	0,1229	0,0886	0,1106	0,3171	0,3502	0,0587	0,1050
	knncoreHYB	0,0000	0,1083	0,2009	0,2011	0,1148	0,1176	0,0758	0,0929	0,3246	0,3339	0,0477	0,0970
	knncoreNIHYB	0,0000	0,1085	0,1889	0,1890	0,1195	0,1220	0,0695	0,0840	0,2812	0,3021	0,0467	0,0949
	knncoreNI	0,0000	0,1364	0,1823	0,1828	0,1249	0,1274	0,0920	0,1167	0,2986	0,3144	0,0594	0,1206
	knnENN	0,0000	0,0835	0,1710	0,1710	0,0820	0,0835	0,0628	0,1041	0,3213	0,3418	0,0367	0,0763
	redCB	0,0000	0,0268	0,1616	0,1617	0,0348	0,0361	0,0091	0,0167	0,2548	0,2769	0,0075	0,0168
	redcore	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	redcoreHYB	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
	redcoreNIHYB	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
redcoreNI	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	
redENN	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	
Promedio	0,0000	0,0582	0,1080	0,1081	0,0596	0,0608	0,0391	0,0516	0,1766	0,1885	0,0252	0,0499	



(a) ISP.

(b) ISP2 (restringida a KNN).

Figura 5.8: Error cuadrático medio ajustado de varios algoritmos de aprendizaje sobre las bases de datos ISP e ISP2 (restringida a KNN) empleando validación cruzada con 10 particiones.

Tabla 5.3: Desempeño de M5 y Mogul_TSK sobre la base de datos ISP2. Raíz del error cuadrático medio en el conjunto de entrenamiento.

Base de datos	M5	Mogul_TSK
c45cb	0,0984	0,0693
c45co	0,1065	0,0750
c45coh	0,1002	0,0738
c45conh	0,1087	0,0686
c45coni	0,1088	0,0678
c45enn	0,0775	0,0388
knncb	0,0801	0,0460
knnco	0,0931	0,0604
knncoh	0,0772	0,0490
knnconh	0,0666	0,0487
knnconi	0,0884	0,0620
knnenn	0,0656	0,0354
ldacb	0,0585	0,0486
ldaco	0,0583	0,0408
ldacoh	0,0684	0,0485
ldaconh	0,0649	0,0422
ldaconi	0,0708	0,0423
ldaenn	0,0970	0,0436
partcb	0,1338	0,0859
partco	0,1257	0,0796
partcoh	0,1043	0,0813
partconh	0,0994	0,0785
partconi	0,1134	0,0760
partenn	0,0947	0,0544
redcb	0,0078	0,0075
Promedio	0,0867	0,0570

5.3. Resultados

En este epígrafe estudiamos el desempeño del sistema KBIS que emplea a ENN y los métodos basados en conjuntos locales presentados con anterioridad. En este caso, la arquitectura del sistema es la ilustrada en la figura 5.6 que admite el manejo de meta-modelos específicos para cada clasificador. Como clasificadores se emplean C4.5, KNN, LDA y PART, que son aquellos para los que ISP2 contiene meta-ejemplos y por tanto para los que construimos meta-modelos de predicción.

Las comparaciones desarrolladas incluyen, además de los métodos bien conocidos y empleados en el capítulo 4, algunas de las más recientes propuestas en esta área, así como varios métodos basados en técnicas meta-heurísticas. En todas las comparaciones se emplea un diseño experimental similar al empleado en el capítulo anterior³.

Comenzaremos por analizar como responden la precisión y la reducción ante distintos valores del parámetro b y distintos clasificadores. Posteriormente compararemos los resultados que obtienen las dos versiones del sistema con los de los métodos CCIS, CNN, Cpruner, DROP3, FCNN, IB3, ICF, NRMCS y PSC, y finalmente los compararemos con los resultados de Explore, GGA, PBIL, SGA, SSMA y ZhangTS[115], como representantes de los métodos de selección de ejemplos basados en técnicas meta-heurísticas. En esta última comparación incluimos el tiempo de ejecución de cada uno de los métodos por su relevancia para ilustrar la ventaja que representa el uso de un sistema KBIS basado en la arquitectura que proponemos.

5.3.1. Reducción y precisión ante distintos valores del parámetro b

Al igual que en el capítulo 4 hemos ejecutado cada versión del sistema 11 veces, asignando a b valores desde 0 hasta 1 con un incremento de 0,1.

La figura 5.9 muestra los valores de reducción alcanzados por ambas versiones del sistema para los valores de b antes mencionados. Nótese que a diferencia de lo que ocurría en los experimentos documentados en el capítulo anterior, ahora para cada clasificador hay un comportamiento diferente en la reducción. La causa es que en este caso, el clasificador pasa a ser un parámetro más para el sistema KBIS, que cuenta con meta-modelos específicos para cada uno de ellos. Como puede apreciarse, las gráficas siguen tendencias crecientes respecto a b , lo que es coherente con el objetivo de este parámetro.

Resulta interesante que mientras en el intervalo $[0;0,4]$ hay diferencias palpables entre los resultados asociados a los diferentes clasificadores, a partir de 0,5 prácticamente no hay diferencia entre estos independientemente del clasificador y el valor de b . Consideramos que este comportamiento se debe a que en el conjunto de bases de datos estudiado el rango de valores en que se encuentra la reducción que logran los métodos individuales de selección de ejemplos es mas alto que el correspondiente a los valores de precisión, lo que causa que el peso de la primera sea mayor que el de la segunda al computar el balance con $b=0,5$. Esto hace que en los valores de b inferiores a 0,5 se logre un mejor

³Al igual que en este, en el presente capítulo se muestran las gráficas y tablas que consideramos esenciales para presentar los resultados, mientras el apéndice B muestra una amplia y detallada información acerca de los mismos.

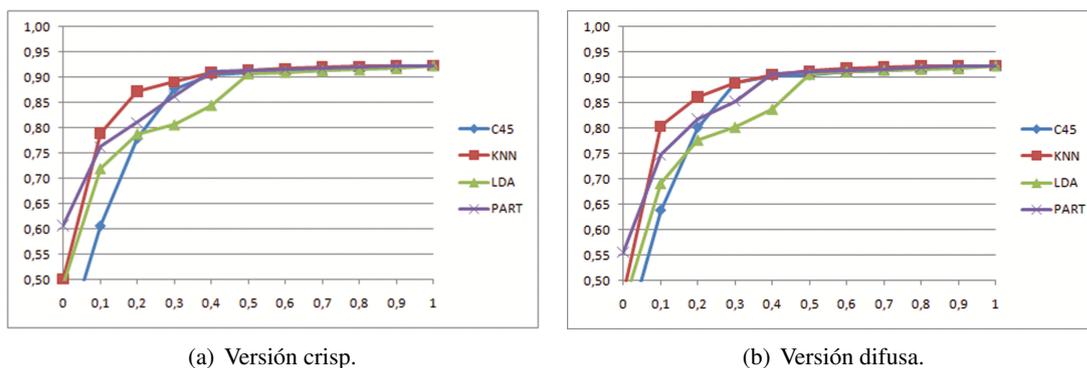


Figura 5.9: Reducción (eje vertical) de ambas versiones del sistema a distintos valores del parámetro b (eje horizontal) y varios clasificadores (series).

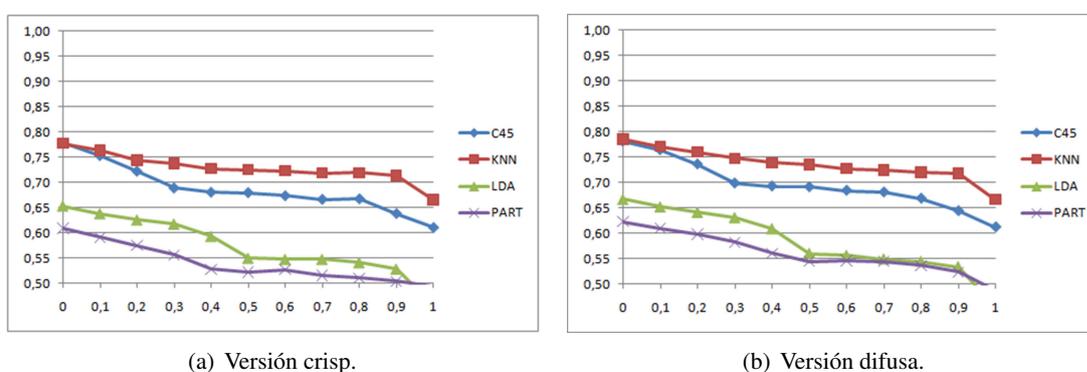


Figura 5.10: Precisión (eje vertical) de ambas versiones del sistema a distintos valores del parámetro b (eje horizontal) y varios clasificadores (series).

equilibrio entre ambos parámetros, mientras en los superiores prácticamente la precisión no juegue papel alguno y generalmente se acabe seleccionando al método con mejor reducción.

En la figura 5.10 aparecen los valores de precisión alcanzados por ambas versiones del sistema para los distintos valores de b y los cuatro clasificadores incluidos en este estudio. Como puede apreciarse, las gráficas siguen tendencias decrecientes respecto a b , lo que es coherente con el objetivo de este parámetro. Puede observarse además que, como en la reducción, ambas versiones del sistema prácticamente no difieren en sus resultados.

Todo lo anterior revela que el sistema responde al parámetro b del modo esperado y que es capaz de elegir adecuadamente un método de selección de ejemplos que responda al objetivo del usuario. Corresponde entonces verificar si estas características se traducen en una ventaja que le permita superar los resultados de los métodos individuales empleados.

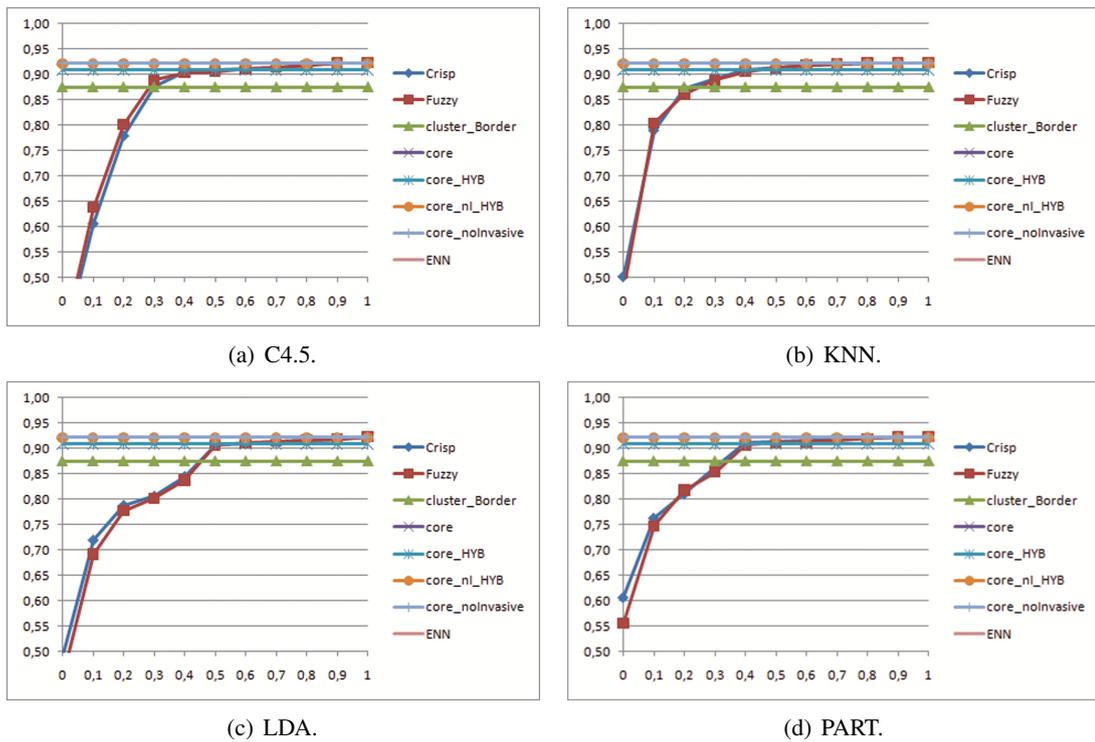


Figura 5.11: Reducción (eje vertical) de ambas versiones del sistema y los métodos individuales empleados por estas a distintos valores del parámetro b (eje horizontal) y varios clasificadores (sub-figuras).

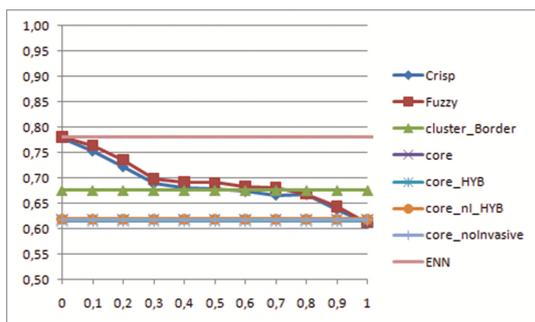
5.3.2. Comparación con los métodos individuales empleados

Para comparar los resultados de ambas versiones del sistema con los de los métodos individuales empleados⁴, proyectamos los resultados de estos últimos en reducción y precisión en las gráficas de las figuras 5.11 y 5.12 respectivamente. Con el objetivo de facilitar el análisis, hemos construido una gráfica independiente por cada clasificador, aunque en el caso de la reducción no debe olvidarse que el valor de este parámetro para los métodos individuales es independiente del clasificador. Sin embargo, aún así es conveniente separarlos en gráficas diferentes pues el sistema KBIS sí ofrece reducciones diferentes para cada clasificador.

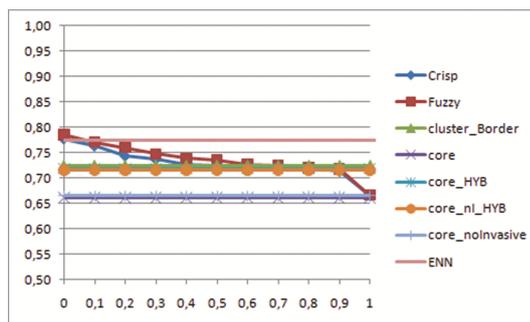
En el caso de la reducción (figura 5.11), los valores correspondientes a ENN no aparecen en los gráficos porque son muy bajos, aunque es importante señalar que los resultados de ambas versiones del sistema KBIS los superan en todos los valores de b .

Puede apreciarse como independientemente del clasificador, para los valores superiores de b se alcanzan reducciones equivalentes a las de los métodos individuales con mejores desempeños en este aspecto. Las gráficas sugieren que a partir de $b = 0,5$ el sistema tiende a seleccionar alguno de

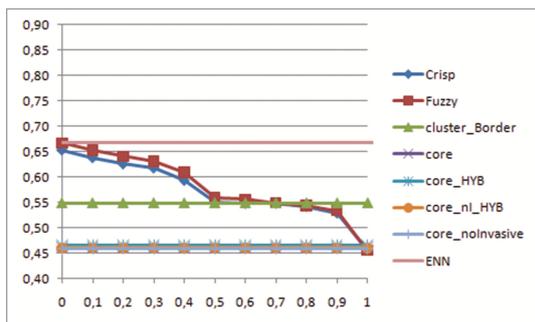
⁴Las implementaciones de estos métodos y el sistema en versiones compatibles con KEEL, pueden descargarse junto con las instrucciones para su incorporación a la plataforma de <http://isg.ugr.es/descargas/descargar/141/>.



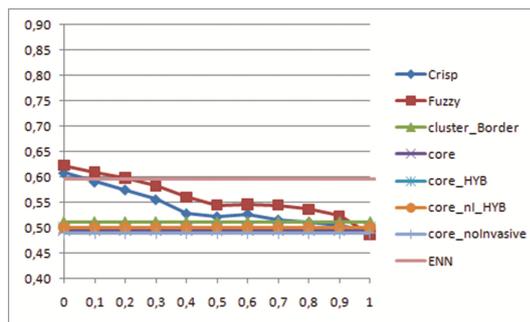
(a) C4.5.



(b) KNN.



(c) LDA.



(d) PART.

Figura 5.12: Precisión (eje vertical) de ambas versiones del sistema y los métodos individuales empleados por estas a distintos valores del parámetro b (eje horizontal) y varios clasificadores (sub-figuras).

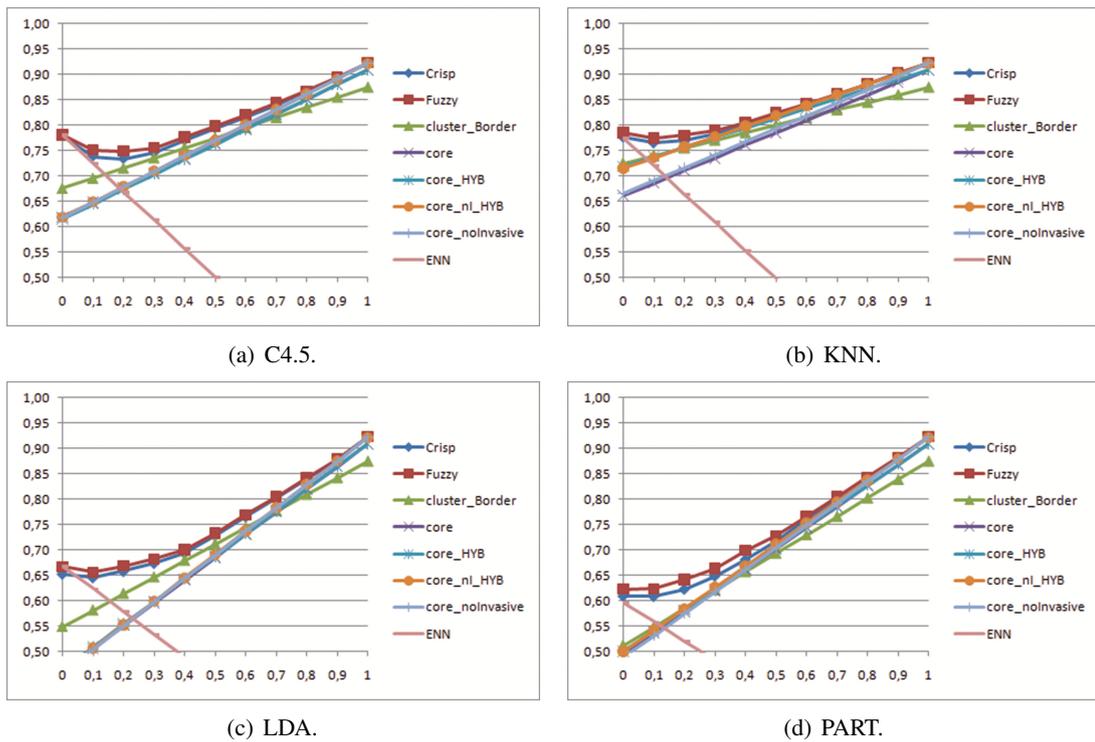


Figura 5.13: Balance (eje vertical) de ambas versiones del sistema y los métodos individuales empleados por estas a distintos valores del parámetro b (eje horizontal) y varios clasificadores (subfiguras).

los métodos de selección de núcleos, mientras en los valores inferiores aumenta la probabilidad de que sean elegidos los métodos ClusterBorder y ENN.

Sin embargo, al observar la precisión (figura 5.12), puede uno percatarse de que la selección del método es más compleja de lo que sugiere la reducción. Ya que si bien en esta última hay una convergencia a partir de $b = 0,5$, la precisión hasta $b = 0,8$ se mantiene en niveles superiores a los correspondientes a los métodos de selección de núcleos. Esto sólo es posible si para algunas de las bases de datos son elegidos ClusterBorder o ENN.

Observando estos resultados desde la perspectiva integral que ofrece el parámetro balance (figura 5.13), se aprecia que para ningún valor de b los resultados del sistema KBIS son superados por los de ningún método individual. Al igual que ocurrió con el sistema KBIS basado en métodos del estado del arte, para estas nuevas versiones las mayores diferencias respecto a los métodos individuales no se dan en los valores de b donde tienen los mayores balances, sino en un rango que va desde $b = 0,1$ hasta $b = 0,4$ y que varía ligeramente en dependencia del clasificador.

Estos resultados sugieren que nuevamente el sistema es capaz de seleccionar adecuadamente un método individual idóneo para cada base de datos. A continuación los compararemos con los de varios métodos del estado del arte para comprobar si los métodos basados en conjuntos locales

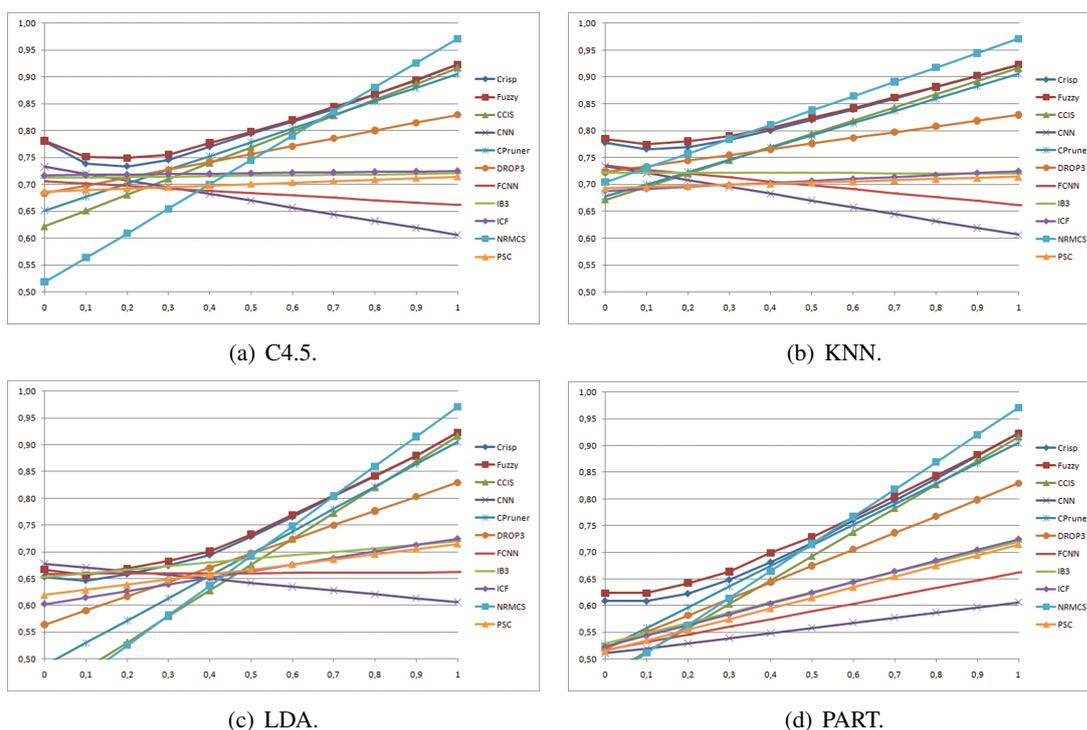


Figura 5.14: Balance (eje vertical) de ambas versiones del sistema y varios métodos convencionales a distintos valores del parámetro b (eje horizontal) y varios clasificadores (subfiguras).

junto a ENN, constituyen una combinación capaz de ofrecer mejores resultados que estos al ser empleados en un sistema KBIS como el que proponemos.

5.3.3. Comparación con métodos convencionales

La figura 5.14 muestra en sendas gráficas los resultados en balance de ambas versiones del sistema y de varios métodos del estado del arte con los cuatro clasificadores incluidos en el estudio. A estos métodos del estado del arte los llamaremos “convencionales” para distinguirlos de aquellos basados en meta-heurísticas y que analizaremos en la próxima sección. En esta comparación, además de los métodos que hemos venido empleando con estos fines, seleccionados por sus buenos resultados y por haber sido ampliamente usados en estudios similares, se añaden los métodos CCIS, FCNN, NRMCS y PSC. Estos últimos son algunos de los aportes más recientes de este campo. En todos los casos empleamos las implementaciones que provee la plataforma KEEL⁵ con los valores por

⁵Específicamente la versión 2010-11-12, disponible en <http://sci2s.ugr.es/keel/source/2010-11-12.zip>. Es importante señalar que en KEEL, las versiones de estos métodos (exceptuando a ENN) que aparecen en el software ejecutable están concebidas como clasificadores y llevan empotrado a KNN. Para nuestros propósitos empleamos las versiones generales (equivalentes a las anteriores) que aparecen en la categoría “preprocesamiento” en las fuentes, por lo que fue necesario compilarlas e incorporarlas al software ejecutable. Los archivos jar correspondientes, junto a las instrucciones para su incorporación a la plataforma están disponibles en <http://isg.ugr.es/descargas/descargar/138/>.

defecto en todos sus parámetros.

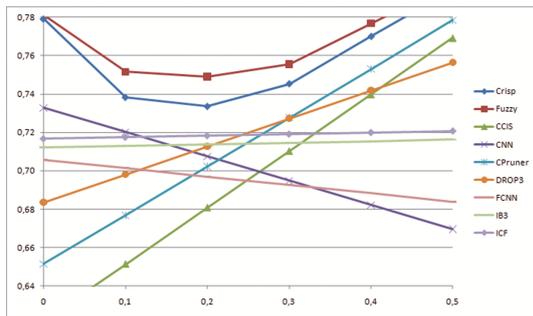
Como puede apreciarse, de modo general ambas versiones del sistema KBIS superan a los restantes métodos en todos los clasificadores y valores de b . Como excepción a esta regla, NRMCS supera al sistema KBIS a partir de cierto valor de b que depende del clasificador. Sin embargo este método, salvo con KNN, es precisamente el que peor desempeño muestra en los valores inferiores de b , y aún en este clasificador es superado en tales valores por el sistema KBIS. La otra excepción se presenta en LDA, clasificador con el que CNN supera ligeramente a ambas versiones del sistema para $b = 0$ y $b = 0,1$.

Aunque la inspección visual de las gráficas anteriores sugiere que el sistema KBIS cumple los objetivos para los que fue concebido, es importante verificar estadísticamente este comportamiento. Para esto, como en el capítulo previo, aplicamos los tests de Friedman y Holm. Dado el gran número de test que implicaría verificar cada valor de b , hemos restringido el análisis a los valores 0,1, 0,2, 0,3 y 0,4; que como se comprobó anteriormente, son aquellos donde el empleo del sistema KBIS propuesto resulta más interesante. Asimismo, por el elevado número de métodos involucrados en el estudio resulta muy difícil que un test estadístico encuentre diferencias significativas, de modo que decidimos incluir en cada test sólo una de las versiones del sistema KBIS junto a los cuatro métodos convencionales con mejor desempeño para el valor de b y clasificador correspondientes. Al ignorar los métodos con desempeños pobres no favorecemos a nuestra propuesta y sí evitamos que estos distorsionen los resultados del test.

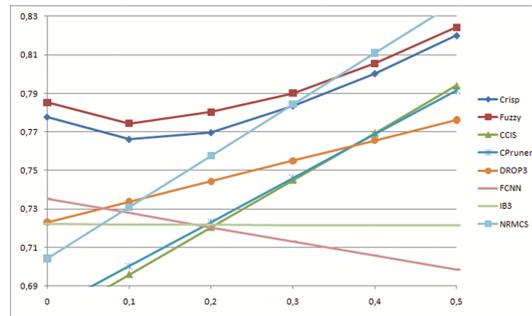
La figura 5.15 es una versión de la figura 5.14 donde se han suprimido los valores de b superiores a 0,5 así como los métodos convencionales que no están entre los cuatro mejores para ninguno de los valores de b mencionados en el párrafo anterior. Por su parte la figura 5.16 muestra los resultados de los métodos de selección de ejemplos con los distintos clasificadores agrupados por los valores de b . Ambas figuras ofrecen perspectivas que consideramos apropiadas para los test que se documentan a continuación.

La tabla 5.4 muestra un resumen de los test de Holm agrupados por clasificador, mientras en la tabla 5.5 se agrupan por valor de b . Por su parte la tabla 5.6 muestra el resumen general. Nótese que al tratarse de cuatro valores de b y cuatro clasificadores, el número de test ejecutados fue de 16 por cada versión del sistema. Como cada test involucra a cuatro métodos convencionales y una versión del sistema, puede aportar a lo sumo el valor 4 para cualquier celda de la columna “veces mejor”. Otro tanto ocurre con la columna “veces peor”. Esto significa que el valor de las celdas en estas columnas está comprendido entre 0 y 64 para el resumen general (tabla 5.6), mientras para los resúmenes por clasificador (tabla 5.4) y por valor de b (tabla 5.5) está entre 0 y 16.

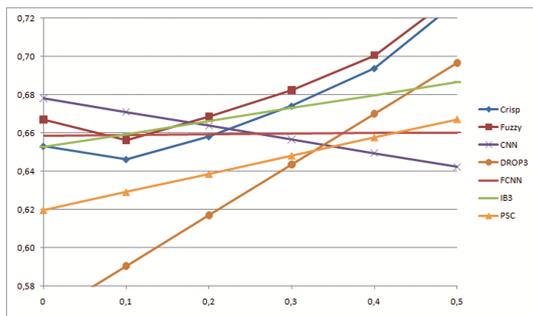
Haciendo un análisis de los resultados de los test agrupados por clasificador, se puede apreciar que con C4.5 (tabla 5.4(a) y 5.4(b)) ambas versiones del sistema KBIS muestran un amplio dominio sobre los otros métodos, entre los que sólo CPruner es mejor que algún método y esto sólo una vez. En cuanto a KNN (tabla 5.4(c) y 5.4(d)), entre los métodos convencionales sólo NRMCS supera a algún método, esto lo hace en varias ocasiones pero son menos que las que lo hace el sistema KBIS. Con el clasificador LDA (tabla 5.4(e) y 5.4(f)) el dominio por parte del sistema KBIS es amplio nuevamente, en este caso es IB3 el único método convencional que es mejor que algún otro. Por último, con PART (tabla 5.4(g) y 5.4(h)) CPruner y NRMCS son los únicos métodos convencionales



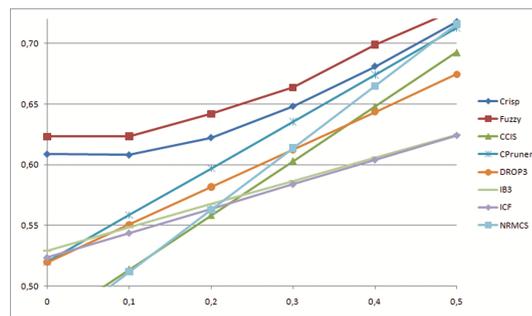
(a) C4.5.



(b) KNN.



(c) LDA.



(d) PART.

Figura 5.15: Balance (eje vertical) de ambas versiones del sistema y varios métodos convencionales con varios clasificadores (subfiguras) para los valores del parámetro b (eje horizontal) comprendidos en el intervalo $[0;0,5]$.

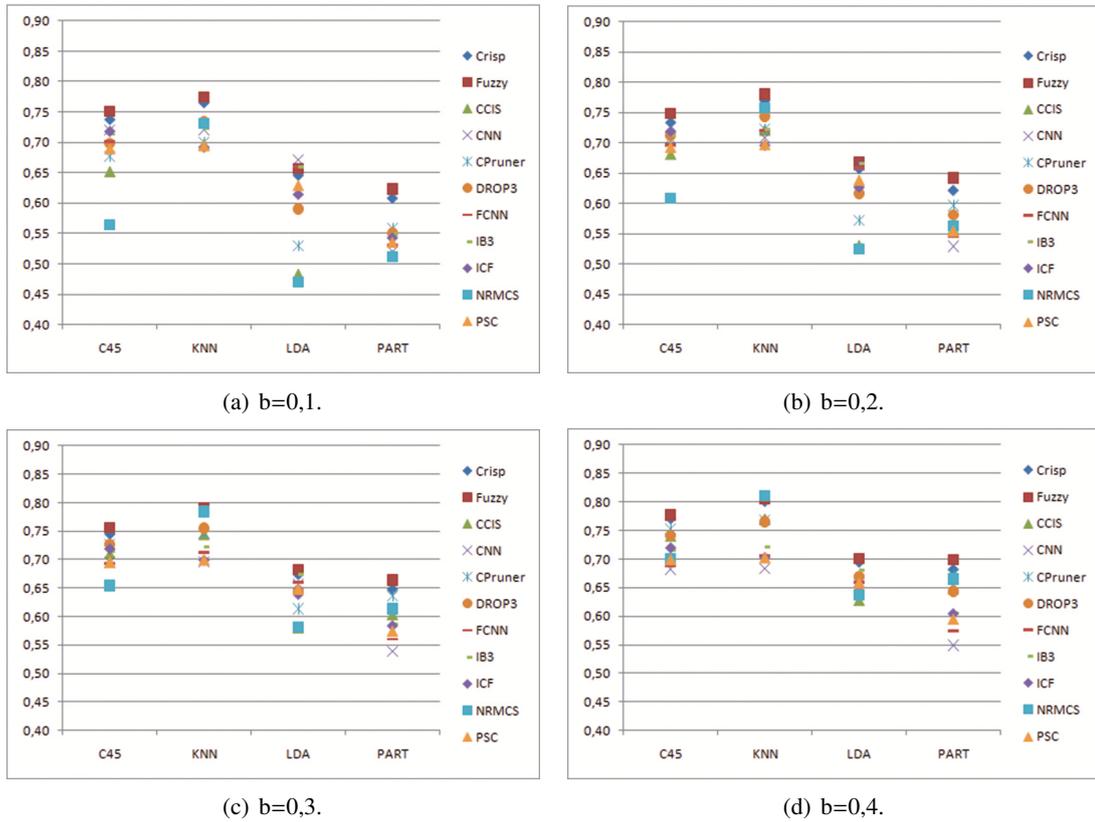


Figura 5.16: Balance (eje vertical) de ambas versiones del sistema y varios métodos convencionales a distintos valores del parámetro b (subfiguras) y varios clasificadores (eje horizontal).

Tabla 5.4: Test de Holm. Resumen por clasificador.

(a) Crisp, C4.5.			(b) Fuzzy, C4.5.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	5	0	Fuzzy	12	0
CCIS	0	0	CCIS	0	0
CNN	0	1	CNN	0	2
CPruner	1	0	CPruner	1	0
DROP3	0	1	DROP3	0	2
FCNN	0	0	FCNN	0	1
IB3	0	1	IB3	0	3
ICF	0	3	ICF	0	5
NRMCS	0	0	NRMCS	0	0
PSC	0	0	PSC	0	0

(c) Crisp, KNN.			(d) Fuzzy, KNN.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	8	0	Fuzzy	10	0
CCIS	0	0	CCIS	0	0
CNN	0	0	CNN	0	0
CPruner	0	5	CPruner	0	6
DROP3	0	5	DROP3	0	6
FCNN	0	1	FCNN	0	1
IB3	0	3	IB3	0	3
ICF	0	0	ICF	0	0
NRMCS	6	0	NRMCS	6	0
PSC	0	0	PSC	0	0

(e) Crisp, LDA.			(f) Fuzzy, LDA.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	6	0	Fuzzy	10	0
CCIS	0	0	CCIS	0	0
CNN	0	3	CNN	0	4
CPruner	0	0	CPruner	0	0
DROP3	0	0	DROP3	0	1
FCNN	0	1	FCNN	0	1
IB3	1	0	IB3	1	1
ICF	0	0	ICF	0	0
NRMCS	0	0	NRMCS	0	0
PSC	0	3	PSC	0	4

(g) Crisp, PART.			(h) Fuzzy, PART.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	8	0	Fuzzy	10	0
CCIS	0	0	CCIS	0	2
CNN	0	0	CNN	0	0
CPruner	4	0	CPruner	4	0
DROP3	0	7	DROP3	0	7
FCNN	0	0	FCNN	0	0
IB3	0	3	IB3	0	3
ICF	0	3	ICF	0	3
NRMCS	1	0	NRMCS	1	0
PSC	0	0	PSC	0	0

Tabla 5.5: Test de Holm. Resumen por valor de b.

(a) Crisp, b=0,1.			(b) Fuzzy, b=0,1.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	6	0	Fuzzy	12	0
CCIS	0	0	CCIS	0	0
CNN	0	0	CNN	0	2
CPruner	0	0	CPruner	0	0
DROP3	0	2	DROP3	0	2
FCNN	0	1	FCNN	0	2
IB3	0	2	IB3	0	3
ICF	0	1	ICF	0	2
NRMCS	0	0	NRMCS	0	0
PSC	0	0	PSC	0	1

(c) Crisp, b=0,2.			(d) Fuzzy, b=0,2.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	8	0	Fuzzy	11	0
CCIS	0	0	CCIS	0	0
CNN	0	2	CNN	0	2
CPruner	2	2	CPruner	2	2
DROP3	0	1	DROP3	0	2
FCNN	0	0	FCNN	0	0
IB3	0	4	IB3	0	5
ICF	0	2	ICF	0	3
NRMCS	2	0	NRMCS	2	0
PSC	0	1	PSC	0	1

(e) Crisp, b=0,3.			(f) Fuzzy, b=0,3.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	6	0	Fuzzy	9	0
CCIS	0	0	CCIS	0	1
CNN	0	2	CNN	0	2
CPruner	1	1	CPruner	1	2
DROP3	0	4	DROP3	0	5
FCNN	0	0	FCNN	0	0
IB3	1	1	IB3	1	1
ICF	0	1	ICF	0	1
NRMCS	2	0	NRMCS	2	0
PSC	0	1	PSC	0	1

(g) Crisp, b=0,4.			(h) Fuzzy, b=0,4.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	7	0	Fuzzy	10	0
CCIS	0	0	CCIS	0	1
CNN	0	0	CNN	0	0
CPruner	2	2	CPruner	2	2
DROP3	0	6	DROP3	0	7
FCNN	0	1	FCNN	0	1
IB3	0	0	IB3	0	1
ICF	0	2	ICF	0	2
NRMCS	3	0	NRMCS	3	0
PSC	0	1	PSC	0	1

Tabla 5.6: Test de Holm. Resumen general.

(a) Crisp.			(b) Fuzzy.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	27	0	Fuzzy	42	0
CCIS	0	0	CCIS	0	2
CNN	0	4	CNN	0	6
CPruner	5	5	CPruner	5	6
DROP3	0	13	DROP3	0	16
FCNN	0	2	FCNN	0	3
IB3	1	7	IB3	1	10
ICF	0	6	ICF	0	8
NRMCS	7	0	NRMCS	7	0
PSC	0	3	PSC	0	4

que superan a algún otro método, CPruner lo hace en varias ocasiones pero el número es apenas la mitad del correspondiente a la versión del sistema KBIS con desempeño más pobre.

Un análisis similar, pero agrupando los resultados por valor de b , permite apreciar que con $b = 0, 1$ (tabla 5.5(a) y 5.5(b)) ningún método convencional supera a algún otro mientras el sistema KBIS lo hace en varias ocasiones. Con $b = 0, 2$ (tabla 5.5(c) y 5.5(d)) CPruner y NRMCS superan en dos ocasiones a algún método, sin embargo están lejos de igualar el número de veces que lo hace el sistema KBIS y el primero de ellos es superado en dos ocasiones. Para $b = 0, 3$ (tabla 5.5(e) y 5.5(f)) se tiene que CPruner, IB3 y NRMCS superan a algún método, sin embargo el número de veces que lo hacen es muy inferior al correspondiente al sistema KBIS y los dos primeros son superados al menos en una ocasión. Finalmente, con $b = 0, 4$ (tabla 5.5(g) y 5.5(h)) CPruner y NRMCS superan en dos y tres ocasiones respectivamente a algún método, sin embargo nuevamente se quedan muy lejos de igualar el número de veces que lo hace el sistema KBIS y el primero de ellos es superado en dos ocasiones.

En un análisis general (tabla 5.6(a) y 5.6(b)), se aprecia que la versión difusa parece tener mejor desempeño. Algo muy importante que puede apreciarse también, es que ninguna de las versiones del sistema KBIS es superada en ningún caso por un método convencional. Por el contrario, para todos los valores de b y todos los clasificadores estas son las que a más métodos superan. Esto es una evidencia bastante fuerte a favor de la hipótesis de que su desempeño es muy bueno en disímiles entornos. Finalmente, entre los métodos convencionales el de mejor desempeño es NRMCS. Sin embargo, el número de veces que es mejor que otro método es muy inferior a los correspondientes a ambas versiones del sistema KBIS. Por otro lado, si bien es cierto que este método no es superado en ningún test, también lo es que en los correspondientes a C4.5 y a LDA no fue incluido por no estar entre los cuatro mejores para ninguno de los valores de b analizados.

5.3.4. Comparación con métodos basados en meta-heurísticas

En el capítulo 2 comentábamos que los métodos de selección de ejemplos basados en enfoques meta-heurísticos, gracias a su capacidad para explorar el espacio de soluciones, logran aportar soluciones adaptadas a las características de cada dominio concreto. Esta propiedad, los hace desem-

peñarse muy bien en disímiles entornos y los dota de una gran flexibilidad. Esto sin embargo no está libre de inconvenientes, esa gran capacidad exploratoria se logra a un costo computacional prohibitivo que los pone en desventaja con respecto a otros enfoques a la hora de su aplicación práctica.

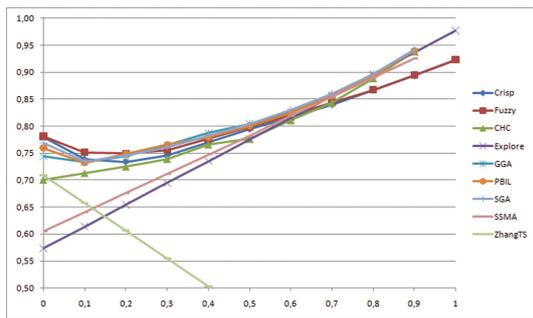
Con la arquitectura para sistemas KBIS y sus instanciaciones que proponemos en este trabajo, pretendemos ofrecer una alternativa con similar efectividad pero mayor eficiencia que estos enfoques. Para comprobar si estos objetivos han sido cumplidos, comparamos su desempeño con el de varios de los métodos de selección de ejemplos basados en meta-heurísticas que encontramos en la literatura especializada. Estos son: CHC, Explore, GGA, PBIL, SGA, SSMA y ZhangTS. En todos los casos empleamos las implementaciones que provee la plataforma KEEL⁶ con los valores por defecto en todos sus parámetros, a excepción del parámetro “Alfa Equilibrate Factor” cuyo propósito es el mismo que el de b pero se interpreta de modo complementario, es decir, para $b = 0$ su valor equivalente es 1 y para $b = 1$ es 0. A este parámetro le hemos dado un tratamiento idéntico al que le dimos a b asignándole valores desde 0 hasta 1 con un incremento de 0,1. En las comparaciones, por su interpretación complementaria a b , se empleó siempre como valor de “Alfa Equilibrate Factor” $1 - b$.

La figura 5.17 muestra en sendas gráficas los resultados en balance de ambas versiones del sistema y de los métodos antes mencionados con los cuatro clasificadores incluidos en el estudio⁷. Como puede apreciarse, el comportamiento no es homogéneo para todos los clasificadores. Con C4.5 el sistema KBIS se desempeña al nivel de los mejores método meta-heurísticos entre $b = 0$ y $b = 0,7$, mientras para los valores superiores es ligeramente superado por estos. Con KNN supera a ZhangTS y es ligeramente superado por el resto de los métodos. Con LDA supera a todos los métodos para $b \in [0; 0,6]$, converge con estos en $b = 0,7$ y a partir de este valor es ligeramente superado por ellos. Finalmente, con PART el comportamiento es muy parecido al anterior, pero la convergencia ocurre entre $b = 0,5$ y $b = 0,6$.

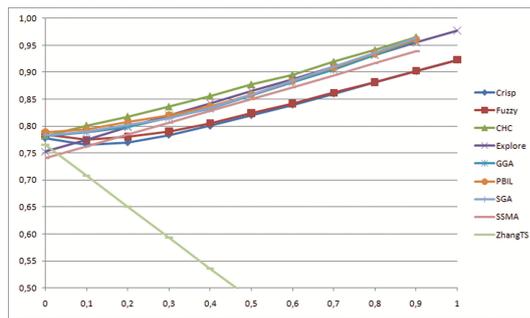
Nuevamente se observa que el rango de valores de b donde resulta de mayor interés el desempeño del sistema KBIS está en la mitad inferior por lo que seguiremos un esquema similar al empleado previamente y centraremos el análisis en el intervalo $b \in [0,1; 0,4]$. En esta ocasión deseamos ZhangTS y SSMA pues en ningún valor de este rango sus resultados son de los cuatro mejores entre los métodos meta-heurísticos. En el caso de Explore, su desempeño sólo cumple esta condición con el clasificador KNN y en los valores de $b = 0,3$ y $b = 0,4$, lo que consideramos muy poco representativo. En consecuencia, optamos por incluir en todos los test de Friedman y Holm a los métodos CHC, GGA, PBIL y SGA junto a una de las versiones del sistema KBIS. La comparación con estos métodos resulta además de especial interés por la flexibilidad que les confiere el parámetro “Alfa Equilibrate Factor” que como hemos analizado antes, juega un papel similar a b .

⁶Específicamente la versión 2010-11-12, disponible en <http://sci2s.ugr.es/keel/source/2010-11-12.zip>. Es importante señalar que en KEEL, las versiones de Explore, SSMA y ZhangTS que aparecen en el software ejecutable están concebidas como clasificadores y llevan empujado a KNN. Para nuestros propósitos empleamos las versiones generales (equivalentes a las anteriores) que aparecen en la categoría “preprocesamiento” en las fuentes, por lo que fue necesario compilarlas e incorporarlas al software ejecutable. Los archivos jar correspondientes, junto a las instrucciones para su incorporación a la plataforma están disponibles en <http://isg.ugr.es/descargas/descargar/138/>.

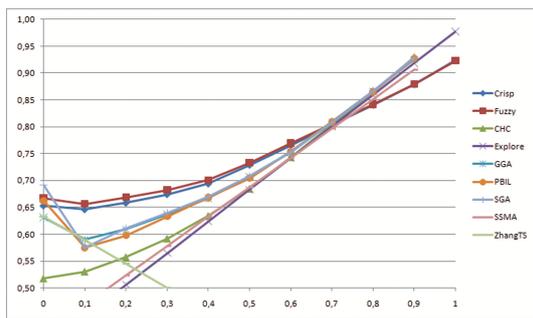
⁷Para algunos métodos se han omitido los resultados correspondientes a $b = 1$ porque dan como resultado subconjuntos vacíos.



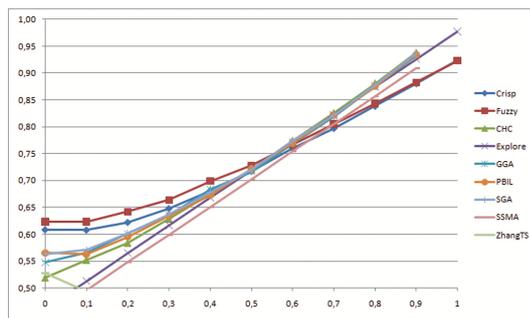
(a) C4.5.



(b) KNN.

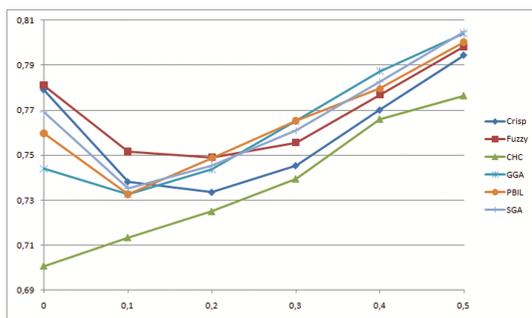


(c) LDA.

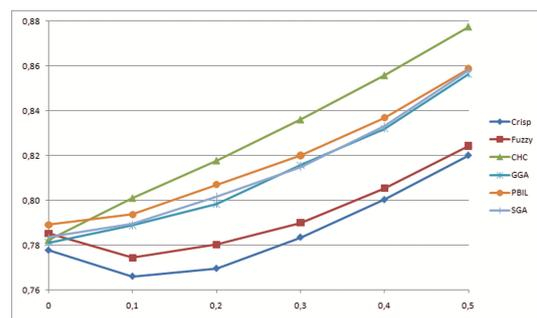


(d) PART.

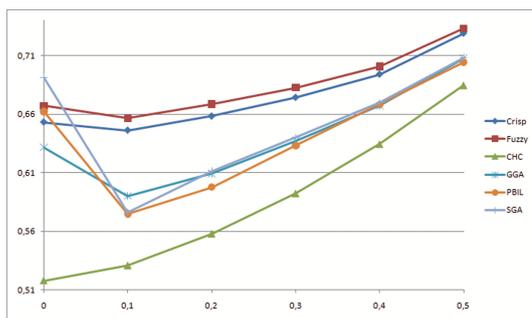
Figura 5.17: Balance (eje vertical) de ambas versiones del sistema y varios métodos basados en meta-heurísticas a distintos valores del parámetro b (eje horizontal) y varios clasificadores (subfiguras).



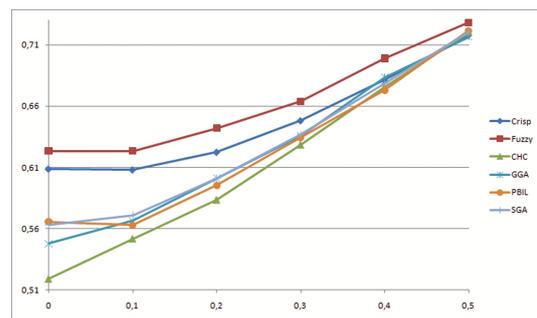
(a) C4.5.



(b) KNN.

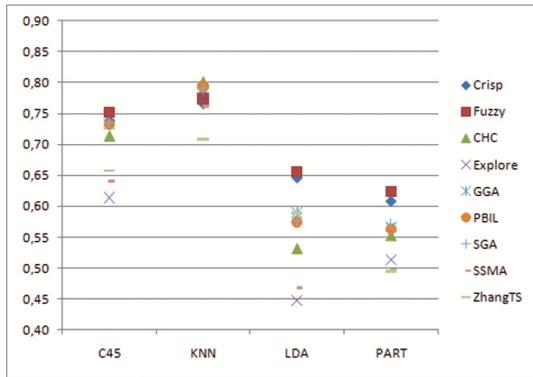


(c) LDA.

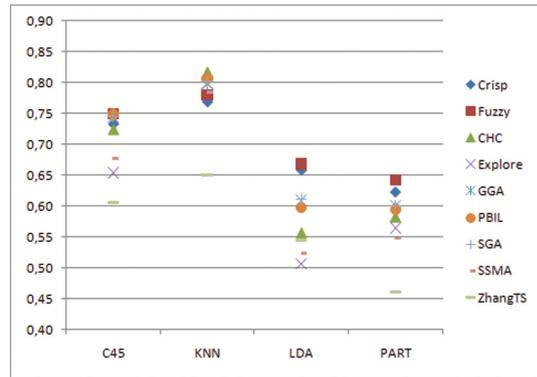


(d) PART.

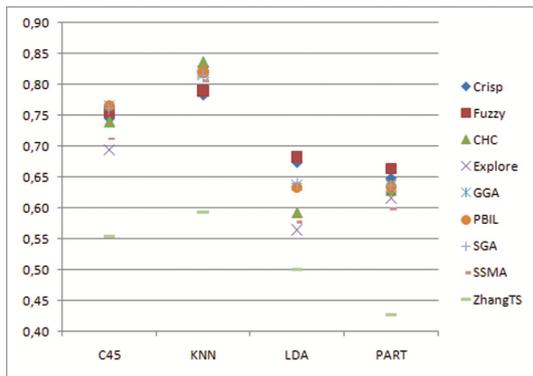
Figura 5.18: Balance (eje vertical) de ambas versiones del sistema y varios métodos basados en meta-heurísticas con varios clasificadores (subfiguras) para los valores del parámetro b (eje horizontal) comprendidos en el intervalo $[0;0,5]$.



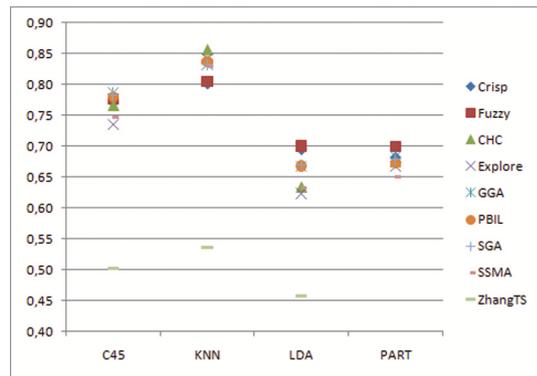
(a) $b=0,1$.



(b) $b=0,2$.



(c) $b=0,3$.



(d) $b=0,4$.

Figura 5.19: Balance (eje vertical) de ambas versiones del sistema y varios métodos basados en meta-heurísticas a distintos valores del parámetro b (subfiguras) y varios clasificadores (eje horizontal).

Tabla 5.7: Test de Holm. Resumen por clasificador.

(a) Crisp, C4.5.			(b) Fuzzy, C4.5.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	1	0	Fuzzy	3	0
CHC	0	4	CHC	0	6
GGA	1	0	GGA	1	0
PBIL	1	0	PBIL	2	0
SGA	1	0	SGA	0	0

(c) Crisp, KNN.			(d) Fuzzy, KNN.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	0	6	Fuzzy	0	3
CHC	13	0	CHC	12	0
GGA	0	5	GGA	0	5
PBIL	3	2	PBIL	1	2
SGA	0	3	SGA	0	3

(e) Crisp, LDA.			(f) Fuzzy, LDA.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	7	0	Fuzzy	13	0
CHC	0	2	CHC	0	3
GGA	0	1	GGA	0	3
PBIL	0	1	PBIL	0	3
SGA	0	3	SGA	0	4

(g) Crisp, PART.			(h) Fuzzy, PART.		
Método	Holm		Método	Holm	
	veces mejor	veces peor		veces mejor	veces peor
Crisp	0	0	Fuzzy	11	0
CHC	0	0	CHC	0	2
GGA	0	0	GGA	0	3
PBIL	0	0	PBIL	0	3
SGA	0	0	SGA	0	3

De igual modo que en la comparación con los métodos convencionales, en la figura 5.18 se ofrece una versión de la figura 5.17 donde se han suprimido los valores de b superiores a 0,5 así como los métodos Explore, SSMA y ZhangTS. Por su parte la figura 5.19 muestra los resultados de los métodos de selección de ejemplos con los distintos clasificadores agrupados por los valores de b . La tabla 5.7 muestra un resumen de los test de Holm agrupados por clasificador, mientras en la tabla 5.8 se agrupan por valor de b . Finalmente, la tabla 5.9 muestra el resumen general. Al igual que en la comparación previa, el valor de las celdas en estas tablas está comprendido entre 0 y 64 para el resumen general (tabla 5.9), mientras para los resúmenes por clasificador (tabla 5.7) y por valor de b (tabla 5.8) está entre 0 y 16.

El análisis de los resultados de los test agrupados por clasificador, permite apreciar que con C4.5 (tabla 5.7(a) y 5.7(b)) ambas versiones del sistema KBIS muestran un desempeño similar al de los otros métodos salvo CHC, que resulta sensiblemente inferior al resto. En contraste, con KNN (tabla 5.7(c) y 5.7(d)) es precisamente CHC el método de mejor desempeño, seguido por PBIL, mientras GGA, SGA y el sistema KBIS se desempeñan de un modo relativamente similar. Con el clasificador

Tabla 5.8: Test de Holm. Resumen por valor de b.

(a) Crisp, b=0,1.			(b) Fuzzy, b=0,1.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	2	1	Fuzzy	9	0
CHC	2	4	CHC	1	4
GGA	0	1	GGA	0	3
PBIL	1	0	PBIL	1	2
SGA	1	0	SGA	0	2

(c) Crisp, b=0,2.			(d) Fuzzy, b=0,2.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	4	2	Fuzzy	9	1
CHC	3	1	CHC	3	4
GGA	0	3	GGA	0	4
PBIL	2	1	PBIL	2	2
SGA	0	2	SGA	0	3

(e) Crisp, b=0,3.			(f) Fuzzy, b=0,3.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	1	1	Fuzzy	8	1
CHC	4	0	CHC	4	2
GGA	0	1	GGA	0	3
PBIL	0	1	PBIL	0	3
SGA	0	2	SGA	0	3

(g) Crisp, b=0,4.			(h) Fuzzy, b=0,4.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	1	2	Fuzzy	1	1
CHC	4	1	CHC	4	1
GGA	1	1	GGA	1	1
PBIL	1	1	PBIL	0	1
SGA	0	2	SGA	0	2

Tabla 5.9: Test de Holm. Resumen general.

(a) Crisp.			(b) Fuzzy.		
Holm			Holm		
Método	veces mejor	veces peor	Método	veces mejor	veces peor
Crisp	8	6	Fuzzy	27	3
CHC	13	6	CHC	12	11
GGA	1	6	GGA	1	11
PBIL	4	3	PBIL	3	8
SGA	1	6	SGA	0	10

LDA (tabla 5.7(e) y 5.7(f)) el dominio por parte del sistema KBIS es amplio, siendo el único método que supera a otros, lo que además ocurre en varias ocasiones. Por último, con PART (tabla 5.7(g) y 5.7(h)) ocurre un fenómeno curioso, ya que los test arrojan que la versión crisp del sistema KBIS se desempeña de modo equivalente al resto de los métodos mientras la versión difusa es muy superior.

En cuanto al análisis agrupando los resultados por valor de b . Se aprecia que con $b = 0,1$ (tabla 5.8(a) y 5.8(b)) la versión crisp del sistema KBIS muestra un desempeño similar a los otros métodos, mientras la versión difusa resulta superior. Con $b = 0,2$ (tabla 5.8(c) y 5.8(d)) GGA y SGA resultan ligeramente inferiores al resto, la versión crisp del sistema KBIS se comporta similar a CHC y PBIL, y la versión difusa resulta nuevamente superior. Para $b = 0,3$ (tabla 5.8(e) y 5.8(f)) se tiene que la versión crisp se desempeña de un modo similar a GGA, PBIL y SGA, mientras resulta superada por CHC, al que a su vez la versión difusa supera. Finalmente, con $b = 0,4$ (tabla 5.8(g) y 5.8(h)) CHC muestra el mejor desempeño, siendo el del sistema KBIS en sus dos versiones equivalente a los de GGA, PBIL y SGA.

En un análisis general (tabla 5.9(a) y 5.9(b)), nuevamente la versión difusa muestra un desempeño superior a la versión crisp y a todos los métodos con que se comparó. Por su parte el desempeño de la versión crisp es superior al de GGA, PBIL y SGA, y algo inferior al de CHC. En sentido general puede decirse que el desempeño del sistema KBIS es similar al de los métodos basados en meta-heurísticas, ya que no existe un dominio claro por parte de ninguno de manera general. A diferencia de lo que se observó en la comparación con los métodos convencionales, en esta ocasión aún el método con mejores resultados (que es CHC) es superado varias veces por otros métodos.

El análisis anterior refleja que se cumple uno de los objetivos del sistema KBIS, obtener un desempeño similar al de los métodos basados en meta-heurísticas. Sin embargo, esto no sería un resultado significativo de no estar conjugado con una eficiencia superior, que permita su aplicación en aquellos problemas donde tales métodos resultan inviables. A continuación analizaremos los tiempos de ejecución del sistema KBIS y estos métodos para comprobar si se cumple este otro objetivo.

Las gráficas de la figura 5.20 muestran los tiempos de ejecución de ambas versiones del sistema KBIS y de los métodos meta-heurísticos estudiados⁸. Estos se corresponden con la ejecución de todos ellos en un mismo procesador y en igualdad de condiciones, por lo que si bien su análisis no constituye un análisis formal de sus complejidades computacionales, sirve para de modo empírico comparar la eficiencia computacional de los métodos involucrados. En el caso del sistema KBIS, en cada gráfica se representan simultáneamente los tiempos de una versión con los cuatro clasificadores incluidos en el estudio, las series correspondientes se denotan con el nombre de cada clasificador.

Como puede apreciarse en la figura 5.20(a) y 5.20(b), se pueden identificar claramente tres comportamientos de acuerdo al tiempo de ejecución. El método más costoso resulta sin dudas ZhangTS, en un segundo nivel se encuentran los enfoques evolutivos incluyendo a SSMA que es el menos costoso entre estos, y en el nivel inferior aparecen el sistema KBIS y Explore. Resulta significativo que, descontando a Explore, el método meta-heurístico menos costoso resultó ser unas siete veces más lento que cualquiera de las versiones del sistema KBIS con cualquiera de los clasificadores. No obstante, aún entre Explore y el sistema KBIS existen diferencias que no se logran apreciar con la escala logarítmica de las gráficas anteriores. Por tal motivo se incluyeron las gráficas de la figura

⁸Nótese que los resultados para $b = 1$ se han omitido en algunos métodos por la razón comentada con anterioridad.

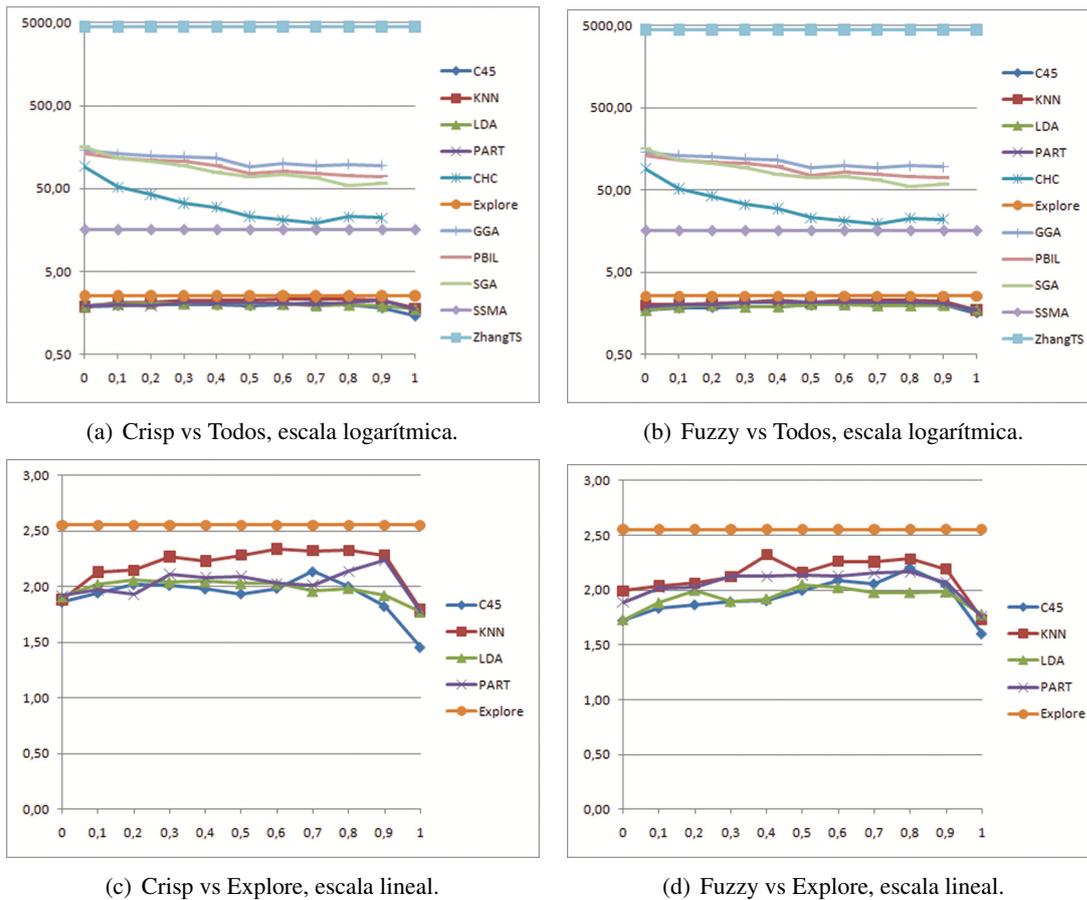


Figura 5.20: Tiempos de ejecución promedio (eje vertical) de ambas versiones del sistema y varios métodos basados en meta-heurísticas a distintos valores del parámetro b (eje horizontal) y varios clasificadores.

5.20(c) y 5.20(d) donde se emplea una escala lineal. En estas se evidencia que con todos los valores de b y todos los clasificadores, ambas versiones del sistema KBIS mantuvieron sus tiempos de ejecución promedio por debajo del correspondiente a Explore.

Los resultados anteriores indican que además de desempeñarse de un modo similar a los enfoques meta-heurísticos en cuanto a balance, el sistema KBIS lo hace a un menor costo computacional. Al respecto es justo señalar que los métodos evolutivos estudiados podrían especializarse para cada clasificador, para esto bastaría con modificar la función de ajuste sustituyendo a KNN como estimador de la precisión por cualquier otro clasificador. De este modo, es razonable suponer que su desempeño ante clasificadores distintos a KNN puede ser mejor que el actual, pero esto sería a costa de un incremento en sus ya abultados tiempos de ejecución como se evidencia en [27].

Por otro lado, Explore parece una buena alternativa dentro de este grupo de métodos, no es el de mejor desempeño pero es mucho más rápido que los otros meta-heurísticos y sus resultados son

buenos con KNN. Al igual que los métodos evolutivos, pudiera adaptarse a otros clasificadores y es posible que sus resultados sean entonces buenos con ellos también. Sin embargo, al igual que con los métodos evolutivos, es lógico suponer que esto incrementaría su costo computacional. Añádase a lo anterior, que en [45] se verifica que su escalabilidad no es buena, lo que hecha por tierra la ventaja relativa que en costo computacional presenta este método.

5.4. Conclusiones Parciales

En este capítulo hemos propuesto varios métodos de selección de ejemplos basados en conjuntos locales y clusters. Estos, junto a ENN, fueron empleados en la instanciación de una nueva versión de la arquitectura presentada en el capítulo anterior que permite el aprendizaje de meta-modelos específicos para cada clasificador. El nuevo conjunto de métodos individuales explota una de las potencialidades que presenta el sistema de medidas propuesto en esta memoria, la reutilización de gran parte de las estructuras de datos creadas durante su cómputo, con lo que se gana en eficiencia. Además, en la mayoría de los casos se hace innecesario estimar la reducción que logrará el método en la base de datos por disponerse del valor exacto que esta tendrá, con lo que se reduce el margen de error al estimar el *balance*.

Al igual que en el capítulo anterior, implementamos dos versiones del sistema KBIS, una crisp y otra difusa. A partir de los resultados experimentales obtenidos por ambas desarrollamos estudios que nos han permitido llegar a algunas conclusiones relevantes para esta investigación.

En primer lugar, comprobamos que nuevamente el sistema KBIS se comportó mejor que los métodos individuales que emplea. A esto se suma que también superó a nueve métodos del estado del arte entre los que se encuentran algunos de los más reconocidos y representantes de los aportes más recientes en este campo.

Comprobamos también que el desempeño logrado es similar al de varios métodos basados en meta-heurísticas, mientras el costo computacional fue muy inferior.

Finalmente, se pudo apreciar que el buen desempeño del sistema KBIS no se limita a KNN, sino que el aprendizaje de meta-modelos específicos por cada clasificador permitió obtener resultados similarmente buenos en los cuatro clasificadores estudiados.

Capítulo 6

Conclusiones y Trabajos Futuros

Al emprender este trabajo nos propusimos como objetivo desarrollar un modelo teórico para la aplicación del meta-aprendizaje al problema de la selección de ejemplos. Partimos de la hipótesis de que es posible obtener de este modo métodos de selección de ejemplos cuyos desempeños sean similares a los de los métodos basados en meta-heurísticas, pero su costo computacional sea sensiblemente inferior. Alcanzar desempeños similares a este tipo de métodos significa lograr resultados comparables a los mejores ante disímiles bases de datos y diversos clasificadores, algo que ningún otro método de selección de ejemplos ofrece.

A lo largo de esta memoria hemos ido describiendo las etapas transitadas durante la investigación para cumplir el objetivo establecido, así como los resultados obtenidos. A continuación exponemos las conclusiones fundamentales a las que hemos llegado como resultado de este trabajo y algunas de las tareas de investigación futuras que consideramos que se derivan del mismo.

En primer lugar, constatamos la carencia de un conjunto de atributos para construir meta-ejemplos que describan el desempeño de métodos de selección de ejemplos. Aunque existen algunas propuestas generales en el campo del meta-aprendizaje, encontramos que presentan algunas limitaciones para su empleo en esta área específica. En consecuencia, propusimos un sistema de medidas de complejidad de los datos, diseñadas para su empleo como atributos en el meta-aprendizaje para la selección de ejemplos. En correspondencia con sus fines, es un sistema compacto y su cálculo no es costoso, adicionalmente, ofrece un conjunto de estructuras de datos potencialmente útiles para la posterior etapa de selección de ejemplos.

Otra aportación fue la elaboración del modelo teórico para la aplicación del meta-aprendizaje al problema de la selección de ejemplos, cuya carencia era, a nuestro juicio, uno de los frenos al desarrollo de propuestas en esta línea. Para demostrar su validez, propusimos una arquitectura que instanciamos dos veces obteniendo sendos sistemas KBIS.

La diferencia fundamental entre estos sistemas es que el primero de ellos emplea como métodos de selección candidatos varios representantes del estado del arte, mientras el segundo emplea fundamentalmente nuevas propuestas que aprovechan las estructuras de datos creadas al calcular las medidas de complejidad. Adicionalmente, con este último introdujimos una ligera modificación en

la arquitectura para adaptar la solución al trabajo con distintos clasificadores.

Para comprobar la eficacia de nuestras propuestas, desarrollamos un amplio estudio experimental que abarcó más de 20 métodos de selección de ejemplos, más de 40 bases de datos y cuatro clasificadores. Los resultados obtenidos confirman la hipótesis inicial, pues nuestras propuestas superan a los métodos convencionales, son similares a los meta-heurísticos y mucho más eficientes que estos.

El trabajo desarrollado abre muchos caminos de investigación en esta línea. Entre estos, hay cuatro que a nuestro juicio resultan los más relevantes y que describiremos a continuación.

En primer lugar, es necesario incrementar el número de meta-ejemplos en ISP. Esto, fundamentalmente, por medio de la inclusión de nuevas bases de datos, preferiblemente aquellas que contribuyan a aumentar la diversidad en el conjunto.

Esta diversidad facilitará otra tarea de vital importancia, la identificación de nichos donde el desempeño del sistema KBIS sea pobre. Esto debe complementarse con el diseño de nuevos métodos individuales de selección de ejemplos especializados en tales nichos de problemas. Los que a su vez darán lugar a nuevos meta-ejemplos y meta-modelos.

Una tercera dirección de trabajo puede ser el perfeccionamiento de la función para el cálculo del balance. Los resultados experimentales confirman que hay un sesgo a favor de la reducción causado por la diferencia entre los rangos de valores en que se encuentran esta y la precisión.

Una alternativa pudiera ser añadir un nuevo parámetro a la ecuación con el propósito de normalizar la precisión con respecto a un máximo esperado. Esto sin embargo, no está exento de riesgos, el más significativo surge de la estimación de dicho máximo.

Finalmente, es importante aumentar el número de clasificadores a los que el sistema KBIS brinde soporte. En este sentido, es importante no perder de vista que la variedad de tales algoritmos existentes constituye un verdadero escollo que hace imposible desde el punto de vista práctico obtener meta-modelos especializados en cada uno de ellos.

Consideramos que hay dos vertientes fundamentales en las que trabajar para aumentar el número de clasificadores. La primera es repetir el procedimiento por medio del cual incorporamos los cuatro actuales. El principal problema asociado a esta estrategia es que provocaría un incremento del número de meta-modelos que pudiera llegar a ser inmanejable. La segunda alternativa es agrupar los clasificadores según su desempeño ante los distintos métodos de selección de ejemplos candidatos y de este modo mantener en el sistema meta-modelos para sólo un representante de cada cluster. La desventaja en este caso es que aumenta la incertidumbre asociada a la elección del método candidato, lo que puede incidir en un deterioro del desempeño del sistema. En ambos casos, el volumen de experimentos necesarios para obtener los meta-ejemplos correspondientes demanda un considerable esfuerzo y consumo de tiempo.

Apéndice A

Detalles de los experimentos del capítulo 4

A.1. Desempeño del sistema ante KNN

Tabla A.1: Reducciones logradas por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8854	0,9696	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780
automobile	0,7414	0,7338	0,7338	0,7360	0,7793	0,8438	0,8438	0,8612	0,8824	0,8824	0,8824
balance	0,8306	0,8587	0,9333	0,9559	0,9321	0,9534	0,9534	0,9534	0,9534	0,9536	0,9527
bands	0,7718	0,7718	0,7718	0,7891	0,8073	0,8732	0,9231	0,9231	0,9231	0,9231	0,9231
breast	0,7452	0,8587	0,8626	0,9258	0,9207	0,9269	0,9281	0,9340	0,9285	0,9391	0,9541
car	0,8627	0,8601	0,8557	0,8640	0,8616	0,8601	0,8561	0,8586	0,9820	0,9831	0,9824
cleveland	0,8239	0,9091	0,8977	0,8940	0,8940	0,8940	0,8940	0,8940	0,8940	0,8940	0,8940
contraceptive	0,5744	0,9013	0,9068	0,9337	0,9311	0,9479	0,9478	0,9484	0,9454	0,9470	0,9467
dermatology	0,1161	0,9095	0,9095	0,9092	0,9211	0,9302	0,9356	0,9356	0,9356	0,9490	0,9490
ecoli	0,2737	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,8975
flare	0,4604	0,7472	0,9364	0,9424	0,9385	0,9391	0,9458	0,9452	0,9723	0,9745	0,9750
glass	0,7506	0,7517	0,7506	0,8645	0,8645	0,8686	0,8780	0,8801	0,8847	0,8832	0,8769
haberman	0,5402	0,8707	0,8682	0,8733	0,8707	0,8682	0,8718	0,8711	0,8856	0,8965	0,9073
heart	0,3510	0,8551	0,8477	0,8477	0,8477	0,8593	0,8708	0,8708	0,8831	0,8831	0,8951
hepatitis	0,3919	0,9032	0,9125	0,9104	0,9104	0,9104	0,9104	0,9104	0,9104	0,9104	0,9104
horse-colic	0,9064	0,9118	0,9118	0,9118	0,9266	0,9266	0,9203	0,8925	0,8925	0,8925	0,8925
housevotes	0,5031	0,9292	0,9305	0,9305	0,9315	0,9277	0,9221	0,9591	0,9775	0,9770	0,9775
ionosphere	0,1450	0,8677	0,8677	0,8677	0,8677	0,8677	0,8778	0,8968	0,9623	0,9623	0,9623
iris	0,4141	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533
led7digit	0,9536	0,9702	0,9733	0,9744	0,9682	0,9736	0,9769	0,9773	0,9747	0,9742	0,9642
lymphography	0,5904	0,8606	0,7304	0,8023	0,8889	0,8912	0,8905	0,9039	0,9228	0,9370	0,9295
mammographic	0,6573	0,8994	0,9006	0,8986	0,8934	0,8954	0,8997	0,9112	0,9281	0,9258	0,9251
monk-2	0,5554	0,7778	0,7845	0,7930	0,8012	0,8001	0,8347	0,8446	0,8346	0,8408	0,8377
movement_libras	0,6556	0,7673	0,7627	0,7608	0,7608	0,8019	0,8049	0,8049	0,8265	0,8451	0,8451
newthyroid	0,6716	0,9359	0,9359	0,9359	0,9359	0,9488	0,9705	0,9705	0,9705	0,9705	0,9705
pima	0,4455	0,8929	0,8929	0,8922	0,8922	0,8974	0,8983	0,8983	0,8983	0,8983	0,9029
saheart	0,4567	0,7822	0,8959	0,8999	0,9074	0,9110	0,9137	0,9137	0,9137	0,9137	0,9137
segment	0,2190	0,9315	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503
sonar	0,6294	0,6963	0,7601	0,8312	0,8312	0,8333	0,8269	0,8269	0,8269	0,8269	0,8275
spectfheart	0,7670	0,9625	0,9625	0,9625	0,9625	0,9625	0,9625	0,9625	0,9625	0,9625	0,9621
splice	0,0519	0,7838	0,7986	0,9369	0,9369	0,9370	0,9369	0,9369	0,9768	0,9771	0,9769
tic-tac-toe	0,5431	0,7401	0,7269	0,7343	0,7941	0,9313	0,9299	0,9264	0,9275	0,9199	0,9136
vehicle	0,3491	0,8546	0,8538	0,8566	0,8566	0,8566	0,8566	0,8729	0,9325	0,9325	0,9325
vowel	0,0305	0,7954	0,7954	0,7954	0,7954	0,8056	0,8130	0,8130	0,8130	0,8180	0,8121
wdbc	0,7750	0,9619	0,9619	0,9619	0,9619	0,9619	0,9619	0,9619	0,9619	0,9635	0,9775
wine	0,0399	0,9076	0,9076	0,9145	0,9164	0,9164	0,9176	0,9176	0,9176	0,9195	0,9195
winequality-red	0,5468	0,8737	0,8826	0,9309	0,9309	0,9309	0,9309	0,9309	0,9309	0,9309	0,9309
wisconsin	0,0342	0,9750	0,9723	0,9730	0,9730	0,9762	0,9782	0,9798	0,9795	0,9797	0,9797
yeast	0,5918	0,8970	0,8893	0,8854	0,9069	0,9400	0,9400	0,9400	0,9400	0,9400	0,9400
zoo	0,6328	0,8726	0,8726	0,8726	0,8726	0,8750	0,8828	0,9088	0,9198	0,9198	0,9198
Promedio	0,5321	0,8654	0,8739	0,8892	0,8947	0,9060	0,9101	0,9134	0,9242	0,9261	0,9260

Tabla A.2: Reducciones logradas por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,3963	0,9696	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780
automobile	0,7387	0,7409	0,7431	0,7436	0,7551	0,8395	0,8390	0,8596	0,8824	0,8824	0,8824
balance	0,5743	0,8610	0,9029	0,9545	0,9547	0,9508	0,9529	0,9502	0,9524	0,9524	0,9520
bands	0,7588	0,7588	0,7588	0,7588	0,7914	0,9231	0,9231	0,9231	0,9231	0,9231	0,9231
breast	0,9071	0,9118	0,9173	0,9129	0,9281	0,9351	0,9332	0,9250	0,9514	0,9479	0,9386
car	0,8315	0,8259	0,8539	0,8603	0,8607	0,8605	0,8567	0,8601	0,9823	0,9808	0,9826
cleveland	0,7336	0,9211	0,9200	0,9186	0,9178	0,9160	0,9160	0,9160	0,9134	0,9050	0,9050
contraceptive	0,8770	0,8799	0,8810	0,9009	0,9483	0,9480	0,9461	0,9491	0,9468	0,9489	0,9491
dermatology	0,1161	0,9044	0,9077	0,9265	0,9265	0,9265	0,9265	0,9356	0,9444	0,9490	0,9490
ecoli	0,4184	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,8886
flare	0,4589	0,7934	0,8907	0,9382	0,9469	0,9395	0,9443	0,9444	0,9518	0,9780	0,9769
glass	0,6352	0,7511	0,8686	0,8697	0,8686	0,8738	0,8738	0,8764	0,8759	0,8759	0,8769
haberman	0,7106	0,8711	0,8736	0,8783	0,8780	0,8776	0,8802	0,8787	0,8852	0,8910	0,9095
heart	0,8029	0,8601	0,8502	0,8424	0,8424	0,8424	0,8424	0,8691	0,8683	0,9058	0,9058
hepatitis	0,6892	0,9068	0,9068	0,9068	0,9068	0,9025	0,9018	0,9061	0,9061	0,9097	0,9097
horse-colic	0,2497	0,9112	0,9040	0,9049	0,9049	0,8991	0,8904	0,8868	0,8868	0,8868	0,8868
horse-colic	0,4918	0,9448	0,9510	0,9487	0,9484	0,9538	0,9466	0,9545	0,9630	0,9768	0,9770
housevotes	0,8753	0,8689	0,8677	0,8677	0,8677	0,8677	0,8677	0,8784	0,9541	0,9623	0,9623
ionosphere	0,5067	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533
iris	0,9322	0,9136	0,9584	0,8844	0,9023	0,9511	0,9756	0,9753	0,9758	0,9751	0,9742
led7digit	0,5317	0,8135	0,8844	0,8821	0,9023	0,8860	0,8972	0,9189	0,9264	0,9317	0,9347
lymphography	0,3058	0,8617	0,8505	0,8535	0,8769	0,9050	0,9251	0,9240	0,9281	0,9268	0,9275
mammographic	0,6398	0,7865	0,7931	0,7894	0,7916	0,8110	0,8161	0,8369	0,8462	0,8416	0,8441
monk-2	0,7083	0,7630	0,7694	0,7707	0,7904	0,8049	0,8049	0,8049	0,8139	0,8451	0,8451
movement_libras	0,9359	0,9359	0,9359	0,9359	0,9370	0,9411	0,9705	0,9705	0,9705	0,9705	0,9705
newthyroid	0,7038	0,8929	0,8929	0,8929	0,8929	0,8929	0,8929	0,9044	0,9029	0,9029	0,9029
pima	0,5016	0,8923	0,9004	0,9077	0,9110	0,9137	0,9137	0,9137	0,9137	0,9137	0,9137
saheart	0,0414	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503	0,9503
segment	0,6337	0,7574	0,8237	0,8253	0,8253	0,8253	0,8216	0,8195	0,8227	0,8318	0,8350
sonar	0,8241	0,9530	0,9530	0,9563	0,9621	0,9621	0,9621	0,9621	0,9621	0,9621	0,9621
spectheart	0,0521	0,7842	0,7995	0,9369	0,9369	0,9369	0,9369	0,9368	0,9727	0,9770	0,9770
splice	0,7206	0,7308	0,7497	0,7625	0,7704	0,8992	0,9280	0,9290	0,9234	0,9179	0,9255
tic-tac-toe	0,5732	0,8566	0,8566	0,8566	0,8566	0,8566	0,8566	0,9111	0,9325	0,9325	0,9325
vehicle	0,1077	0,7981	0,7981	0,7992	0,8003	0,8012	0,8108	0,8130	0,8130	0,8130	0,8121
vowel	0,4041	0,9619	0,9619	0,9619	0,9619	0,9619	0,9619	0,9619	0,9619	0,9635	0,9775
wdbc	0,2968	0,8595	0,8682	0,8882	0,9195	0,9207	0,9195	0,9195	0,9195	0,9195	0,9195
wine	0,5068	0,8737	0,8988	0,9252	0,9252	0,9309	0,9309	0,9309	0,9309	0,9309	0,9309
winequality-red	0,5043	0,9396	0,9510	0,9576	0,9293	0,9797	0,9797	0,9792	0,9793	0,9795	0,9814
wisconsin	0,4612	0,8860	0,8854	0,8911	0,9198	0,9400	0,9400	0,9400	0,9400	0,9400	0,9400
yeast	0,2258	0,8680	0,8648	0,8648	0,8714	0,9040	0,9198	0,9198	0,9198	0,9198	0,9198
zoo	0,5596	0,8657	0,8798	0,8888	0,8952	0,9069	0,9100	0,9146	0,9235	0,9267	0,9271
Promedio											

Tabla A.3: Precisiones logradas (con KNN) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8756	0,8656	0,8756	0,8756	0,8756	0,8756	0,8756	0,8756	0,8756	0,8756	0,8756
automobile	0,6094	0,6187	0,6084	0,6190	0,5902	0,5047	0,4870	0,4762	0,3478	0,3478	0,3478
balance	0,8554	0,8585	0,8505	0,8407	0,8473	0,8390	0,8407	0,8536	0,8456	0,8390	0,8358
bands	0,7167	0,7148	0,7091	0,7110	0,7054	0,6865	0,6598	0,6598	0,6598	0,6598	0,6598
breast	0,7245	0,6907	0,7464	0,6803	0,7058	0,7057	0,7056	0,6812	0,7170	0,6844	0,6440
car	0,8533	0,8650	0,8661	0,8684	0,8510	0,8644	0,8591	0,8580	0,4888	0,5023	0,5128
cleveland	0,5324	0,5223	0,5530	0,5530	0,5530	0,5530	0,5530	0,5530	0,5530	0,5530	0,5530
contraceptive	0,4586	0,4375	0,4375	0,4519	0,4368	0,4375	0,4409	0,4409	0,4423	0,4410	0,4478
dermatology	0,9579	0,9411	0,9411	0,9242	0,9019	0,8908	0,8851	0,8851	0,8851	0,8236	0,8236
ecoli	0,8378	0,8164	0,8194	0,8164	0,8255	0,8164	0,8102	0,8224	0,8133	0,8193	0,7301
flare	0,5985	0,5758	0,5654	0,5587	0,5605	0,5513	0,4612	0,4327	0,3723	0,3306	0,3399
glass	0,6623	0,6725	0,7095	0,6984	0,6925	0,6932	0,6256	0,6113	0,6073	0,5534	0,5307
haberman	0,7228	0,7297	0,7297	0,7297	0,7297	0,7297	0,7297	0,7297	0,7197	0,7059	0,7062
heart	0,8000	0,7808	0,7769	0,7769	0,7769	0,7769	0,7769	0,7769	0,7692	0,7692	0,7500
hepatitis	0,8005	0,8057	0,8057	0,7848	0,7990	0,7781	0,7781	0,7924	0,7990	0,7848	0,7786
horse-colic	0,8070	0,8153	0,8153	0,8153	0,8098	0,8098	0,8070	0,7790	0,7790	0,7790	0,7790
housevotes	0,9054	0,8983	0,8867	0,8983	0,9007	0,8798	0,8750	0,8868	0,7219	0,7219	0,7195
ionosphere	0,8475	0,8766	0,8767	0,8738	0,8797	0,8767	0,8708	0,8620	0,8386	0,8386	0,8386
iris	0,9500	0,9714	0,9571	0,9571	0,9643	0,9571	0,9643	0,9429	0,9571	0,9571	0,9571
led7digit	0,4592	0,4816	0,5082	0,4490	0,4653	0,4714	0,4388	0,3980	0,4714	0,4245	0,2408
lymphography	0,7307	0,7241	0,7406	0,7752	0,7389	0,7461	0,7472	0,6285	0,6444	0,6648	0,5648
mammographic	0,7824	0,7625	0,7645	0,7572	0,7940	0,7468	0,7740	0,7582	0,7456	0,7582	0,7477
monk-2	0,7987	0,8244	0,8380	0,7911	0,8276	0,8098	0,8014	0,7340	0,7355	0,7093	0,6769
movement_libras	0,6800	0,6629	0,6486	0,6543	0,6543	0,6943	0,7000	0,6971	0,6029	0,4514	0,4514
newthyroid	0,9271	0,9174	0,9174	0,9174	0,9174	0,8933	0,8636	0,8543	0,8590	0,8493	0,8543
pima	0,7508	0,7455	0,7415	0,7375	0,7375	0,7310	0,7271	0,7258	0,7284	0,7258	0,7152
saheart	0,6748	0,6815	0,6793	0,6948	0,6971	0,6969	0,6947	0,6947	0,6947	0,6947	0,6947
segment	0,9557	0,9274	0,9439	0,9470	0,9452	0,9448	0,9470	0,9443	0,9461	0,9443	0,9443
sonar	0,8274	0,8424	0,8121	0,7871	0,7921	0,7871	0,7716	0,7666	0,7766	0,7716	0,7621
spectfheart	0,7746	0,7631	0,7631	0,7631	0,7631	0,7631	0,7631	0,7631	0,7631	0,7631	0,7631
splice	0,7531	0,7079	0,7069	0,6764	0,6777	0,6774	0,6761	0,6767	0,5252	0,5261	0,5283
tic-tac-toe	0,8132	0,7890	0,7901	0,7794	0,7224	0,6193	0,6087	0,6298	0,6087	0,6329	0,6277
vehicle	0,6830	0,6723	0,6699	0,6734	0,6746	0,6734	0,6758	0,6364	0,5158	0,5158	0,5158
vowel	0,9694	0,9306	0,9306	0,9306	0,9306	0,9296	0,9173	0,9224	0,9214	0,8653	0,4429
wdbc	0,9659	0,9696	0,9678	0,9624	0,9660	0,9678	0,9642	0,9677	0,9678	0,9606	0,8766
wine	0,9526	0,9349	0,9408	0,9290	0,9114	0,9173	0,9290	0,9110	0,9232	0,9349	0,9467
winequality-red	0,5802	0,5739	0,5689	0,5431	0,5431	0,5431	0,5431	0,5431	0,5431	0,5431	0,5431
wisconsin	0,9666	0,9593	0,9637	0,9637	0,9637	0,9579	0,9492	0,9376	0,9347	0,9332	0,9376
yeast	0,5597	0,5238	0,5387	0,5401	0,5503	0,5604	0,5604	0,5604	0,5604	0,5604	0,5604
zoo	0,7863	0,7595	0,7595	0,7595	0,7595	0,7809	0,7809	0,8442	0,8533	0,8533	0,8533
Promedio	0,7727	0,7653	0,7681	0,7616	0,7609	0,7534	0,7460	0,7378	0,7128	0,7017	0,6769

Tabla A.4: Precisiones logradas (con KNN) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,9178	0,8656	0,8756	0,8756	0,8756	0,8756	0,8756	0,8756	0,8756	0,8756	0,8756
automobile	0,6037	0,6042	0,6085	0,6085	0,5938	0,5097	0,5158	0,4731	0,3478	0,3478	0,3478
balance	0,8716	0,8619	0,8680	0,8375	0,8489	0,8617	0,8439	0,8392	0,8438	0,8521	0,8277
bands	0,7053	0,7129	0,7053	0,7072	0,7091	0,6598	0,6598	0,6598	0,6598	0,6598	0,6598
breast	0,6809	0,6589	0,6948	0,6795	0,7140	0,6309	0,6408	0,6802	0,6180	0,6075	0,6444
car	0,8743	0,8667	0,8638	0,8720	0,8597	0,8772	0,8644	0,8638	0,4947	0,5517	0,5214
cleveland	0,5534	0,5603	0,5601	0,5636	0,5532	0,5633	0,5599	0,5599	0,5668	0,5737	0,5737
contraceptive	0,4538	0,4538	0,4490	0,4600	0,4355	0,4444	0,4485	0,4410	0,4416	0,4478	0,4403
dermatology	0,9579	0,9242	0,9214	0,8962	0,8962	0,8962	0,8962	0,8851	0,8541	0,8236	0,8264
ecoli	0,8438	0,8134	0,8164	0,8164	0,8133	0,8193	0,8134	0,8163	0,8224	0,8164	0,7272
flare	0,5966	0,5758	0,5568	0,5208	0,5568	0,5605	0,4952	0,4422	0,4261	0,3031	0,3305
glass	0,6720	0,6475	0,6603	0,6801	0,6730	0,6535	0,6647	0,6476	0,5545	0,5545	0,5307
haberman	0,7331	0,7297	0,7297	0,7331	0,7331	0,7331	0,7331	0,7331	0,7331	0,7331	0,6961
heart	0,7962	0,7923	0,7962	0,7846	0,7846	0,7846	0,7846	0,7769	0,7769	0,7538	0,7538
hepatitis	0,8548	0,8200	0,8133	0,8200	0,8200	0,8129	0,8124	0,8052	0,8052	0,8052	0,8052
horse-colic	0,8239	0,7987	0,7816	0,7959	0,7959	0,8042	0,7985	0,7874	0,7874	0,7874	0,7874
housevotes	0,9081	0,9081	0,8937	0,8915	0,9151	0,8986	0,8962	0,9057	0,8467	0,7172	0,7219
ionosphere	0,8679	0,8796	0,8738	0,8708	0,8767	0,8767	0,8738	0,8708	0,8386	0,8386	0,8386
iris	0,9643	0,9571	0,9429	0,9643	0,9571	0,9571	0,9571	0,9500	0,9643	0,9571	0,9571
led7digit	0,4388	0,4388	0,4714	0,4367	0,4633	0,4490	0,4633	0,4755	0,4694	0,4510	0,2082
lymphography	0,7567	0,7393	0,7184	0,7413	0,6819	0,7378	0,7181	0,6807	0,6319	0,6157	0,6148
mammographic	0,8097	0,7698	0,7740	0,7541	0,7593	0,7666	0,7603	0,7594	0,7384	0,7792	0,7561
monk-2	0,8226	0,8438	0,8123	0,7801	0,8079	0,7943	0,7792	0,7210	0,7048	0,6979	0,6525
movement_libras	0,6686	0,6771	0,6600	0,6600	0,7114	0,6943	0,7000	0,6971	0,6514	0,4514	0,4514
newthyroid	0,9174	0,9174	0,9174	0,9174	0,9224	0,9269	0,8540	0,8638	0,8540	0,8540	0,8638
pima	0,7521	0,7376	0,7389	0,7454	0,7467	0,7520	0,7349	0,7179	0,7152	0,7152	0,7152
saheart	0,6726	0,6858	0,6925	0,7014	0,6969	0,6947	0,6947	0,6947	0,6947	0,6947	0,6947
segment	0,9552	0,9457	0,9470	0,9457	0,9443	0,9439	0,9478	0,9422	0,9439	0,9439	0,9439
sonar	0,8324	0,8268	0,8268	0,8118	0,8068	0,8118	0,7866	0,7916	0,7771	0,7316	0,7371
spectheart	0,7745	0,7629	0,7668	0,7629	0,7631	0,7631	0,7631	0,7631	0,7669	0,7631	0,7631
splice	0,7531	0,7075	0,7050	0,6777	0,6777	0,6761	0,6764	0,6767	0,5434	0,5286	0,5255
tic-tac-toe	0,8037	0,7890	0,7932	0,7615	0,7542	0,6616	0,6023	0,6277	0,6287	0,6276	0,6181
vehicle	0,6818	0,6687	0,6710	0,6770	0,6710	0,6722	0,6770	0,5600	0,5158	0,5158	0,5158
vowel	0,9633	0,9235	0,9235	0,9265	0,9194	0,9255	0,9173	0,9235	0,9214	0,9204	0,4429
wdbc	0,9695	0,9659	0,9624	0,9642	0,9678	0,9660	0,9660	0,9696	0,9660	0,9642	0,8766
wine	0,9585	0,9162	0,9224	0,9165	0,9283	0,9165	0,9349	0,9408	0,9408	0,9290	0,9287
winequality-red	0,5834	0,5733	0,5576	0,5450	0,5443	0,5431	0,5431	0,5431	0,5431	0,5431	0,5431
wisconsin	0,9564	0,9477	0,9217	0,9404	0,9390	0,9391	0,9391	0,9420	0,9420	0,9332	0,9260
yeast	0,5597	0,5333	0,5428	0,5475	0,5652	0,5604	0,5604	0,5604	0,5604	0,5604	0,5604
zoo	0,8750	0,7139	0,7870	0,7870	0,7870	0,8371	0,8533	0,8533	0,8533	0,8533	0,8533
Promedio	0,7796	0,7629	0,7631	0,7594	0,7617	0,7563	0,7501	0,7429	0,7155	0,7020	0,6764

A.2. Desempeño de los métodos individuales ante KNN

Tabla A.5: Reducciones logradas por los métodos individuales que el sistema emplea.

Bases de datos	CNN	coreHYB	CPruner	DROP3	ENN	IB3	ICF
appendicitis	0,6269	0,9780	0,9004	0,8774	0,1593	0,6708	0,8512
automobile	0,4959	0,8390	0,8954	0,5724	0,4064	0,5511	0,5171
balance	0,6363	0,9524	0,8078	0,8676	0,1559	0,7154	0,9396
bands	0,5275	0,7588	0,9520	0,7267	0,2698	0,6627	0,6116
breast	0,4600	0,9537	0,9514	0,7844	0,3500	0,6325	0,6791
car	0,7689	0,9817	0,8285	0,8834	0,0775	0,8373	0,8555
cleveland	0,3829	0,9222	0,9388	0,8317	0,4448	0,4664	0,7972
contraceptive	0,2624	0,8809	0,9892	0,7307	0,5498	0,4433	0,6881
dermatology	0,8558	0,9490	0,8819	0,9232	0,0310	0,8880	0,7034
ecoli	0,5890	0,9160	0,9183	0,8423	0,1958	0,6802	0,7460
flare	0,3892	0,9743	0,9405	0,8394	0,3809	0,6498	0,5422
glass	0,5104	0,8718	0,9242	0,7415	0,3214	0,6107	0,6775
haberman	0,4738	0,9386	0,9354	0,8003	0,3032	0,6347	0,6543
heart	0,5926	0,9387	0,9296	0,8259	0,2206	0,7457	0,8037
hepatitis	0,6201	0,9118	0,9054	0,8566	0,1828	0,7275	0,7728
horse-colic	0,5954	0,8599	0,8940	0,8593	0,1769	0,7491	0,8083
housevotes	0,8082	0,9773	0,9126	0,9351	0,0672	0,9208	0,8927
ionosphere	0,7680	0,8677	0,8598	0,9079	0,1421	0,8398	0,8072
iris	0,8407	0,9533	0,8985	0,9230	0,0474	0,9089	0,6422
led7digit	0,2360	0,9729	0,9858	0,8278	0,5660	0,6338	0,6178
lymphography	0,5398	0,8859	0,9294	0,8356	0,2146	0,6922	0,7530
mammographic	0,5625	0,9240	0,8836	0,8432	0,2040	0,7784	0,6616
monk-2	0,6842	0,7927	0,8642	0,8670	0,0411	0,7755	0,6880
movement_libras	0,5818	0,8049	0,8917	0,6914	0,2275	0,6552	0,5074
newthyroid	0,8749	0,9705	0,8569	0,8853	0,0579	0,8739	0,8527
pima	0,5041	0,8929	0,9510	0,8228	0,2603	0,6913	0,7734
saheart	0,4543	0,8836	0,9565	0,7987	0,3141	0,6455	0,7453
segment	0,8843	0,9503	0,8487	0,9156	0,0443	0,9100	0,7846
sonar	0,6891	0,8237	0,8184	0,7591	0,1699	0,7238	0,7121
spectfheart	0,5368	0,9130	0,9443	0,8339	0,2863	0,7041	0,8922
splice	0,5405	0,9770	0,9415	0,7647	0,2357	0,6755	0,6969
tic-tac-toe	0,6671	0,9303	0,8334	0,9287	0,2284	0,7308	0,7090
vehicle	0,5051	0,8566	0,9551	0,7727	0,2931	0,6106	0,6916
vowel	0,7753	0,8130	0,8331	0,6721	0,0312	0,7851	0,2878
wdbc	0,8573	0,9619	0,8903	0,9594	0,0307	0,9235	0,9695
wine	0,8458	0,9195	0,8596	0,9101	0,0337	0,8951	0,8377
winequality-red	0,3342	0,8737	0,9715	0,7641	0,4745	0,5273	0,7024
wisconsin	0,8841	0,9798	0,9126	0,9747	0,0337	0,9585	0,9528
yeast	0,3216	0,8854	0,9773	0,7844	0,4668	0,4608	0,7311
zoo	0,7725	0,9198	0,8501	0,8351	0,0815	0,8328	0,4360
Promedio	0,6064	0,9089	0,9055	0,8294	0,2195	0,7205	0,7248

Tabla A.6: Precisiones logradas (con KNN) por los métodos individuales que el sistema emplea.

Bases de datos	CNN	coreHYB	CPruner	DROP3	ENN	IB3	ICF
appendicitis	0,8027	0,8756	0,8018	0,6536	0,8973	0,7355	0,5464
automobile	0,7418	0,5097	0,3673	0,6281	0,5383	0,7313	0,5369
balance	0,7040	0,8439	0,8544	0,8177	0,8560	0,7152	0,7026
bands	0,5921	0,7072	0,5603	0,6012	0,6198	0,5992	0,5474
breast	0,6152	0,5613	0,6990	0,6577	0,7133	0,6184	0,6646
car	0,8802	0,5245	0,7552	0,6221	0,8513	0,8744	0,7842
cleveland	0,5083	0,5636	0,5412	0,4947	0,5643	0,4916	0,5019
contraceptive	0,4176	0,4675	0,4182	0,4406	0,4528	0,4033	0,4480
dermatology	0,9346	0,8236	0,9402	0,9293	0,9591	0,9318	0,8852
ecoli	0,7502	0,8134	0,5955	0,7263	0,8248	0,7236	0,7676
flare	0,4652	0,3333	0,4774	0,6398	0,6632	0,4437	0,4907
glass	0,6940	0,6519	0,4716	0,6571	0,6919	0,6549	0,6021
haberman	0,6339	0,6594	0,7353	0,6697	0,6990	0,6272	0,6732
heart	0,7630	0,7423	0,7815	0,6926	0,7852	0,6889	0,6519
hepatitis	0,7625	0,7971	0,7933	0,8129	0,8200	0,7308	0,7621
horse-colic	0,7366	0,7069	0,7794	0,7769	0,8315	0,7309	0,6900
housevotes	0,8918	0,7195	0,8985	0,9102	0,9193	0,9125	0,8410
ionosphere	0,8660	0,8738	0,8317	0,7521	0,8433	0,8233	0,8175
iris	0,9200	0,9571	0,7733	0,9267	0,9533	0,9067	0,8933
led7digit	0,3700	0,4857	0,2160	0,4060	0,4920	0,3840	0,3720
lymphography	0,6361	0,6407	0,6569	0,7674	0,7547	0,6427	0,7339
mammographic	0,6962	0,7677	0,7961	0,7243	0,8096	0,7003	0,7816
monk-2	0,8340	0,6857	0,6543	0,6571	0,7817	0,8074	0,6946
movement_libras	0,7750	0,6971	0,3306	0,7583	0,7139	0,7361	0,7278
newthyroid	0,9630	0,8593	0,9026	0,9491	0,9494	0,9489	0,7626
pima	0,6472	0,7454	0,6524	0,6992	0,7462	0,6538	0,6692
saheart	0,6276	0,6704	0,6537	0,6492	0,6949	0,6168	0,6342
segment	0,9407	0,9448	0,9100	0,9100	0,9545	0,9286	0,8121
sonar	0,8552	0,7911	0,6743	0,7779	0,8212	0,7643	0,6633
spectfheart	0,6594	0,7397	0,7942	0,6973	0,7419	0,6809	0,4792
splice	0,6853	0,5258	0,6574	0,7025	0,7530	0,6818	0,6589
tic-tac-toe	0,8122	0,6150	0,6670	0,5731	0,7129	0,7788	0,7297
vehicle	0,6536	0,6758	0,4515	0,6099	0,6927	0,6855	0,6336
vowel	0,9576	0,9194	0,4222	0,9313	0,9687	0,9535	0,9313
wdbc	0,9280	0,9642	0,9297	0,8928	0,9648	0,9332	0,7169
wine	0,8984	0,9228	0,9150	0,9105	0,9552	0,9209	0,9493
winequality-red	0,4828	0,5733	0,5372	0,5472	0,5847	0,4697	0,5097
wisconsin	0,9399	0,9390	0,9571	0,8913	0,9642	0,9327	0,8170
yeast	0,4684	0,5380	0,4730	0,5331	0,5505	0,4448	0,5088
zoo	0,8433	0,8533	0,7753	0,9264	0,9114	0,8769	0,9322
Promedio	0,7338	0,7171	0,6775	0,7231	0,7751	0,7221	0,6881

A.3. Comparación del desempeño del sistema (con KNN) con los de los métodos individuales que emplea. Detalles de los test de Friedman y Holm

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 220,675.

Método	Ranking
CNN	6,825
coreHYB	1,725
CPruner	2,65
DROP3	3,475
ENN	7,825
IB3	5,325
ICF	5,4
Crisp01	2,775

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ENN	0	0
2	CPruner vs ENN	0	0
3	CNN vs coreHYB	0	0
4	ENN vs Crisp01	0	0
5	DROP3 vs ENN	0	0
6	CNN vs CPruner	0	0
7	CNN vs Crisp01	0	0
8	coreHYB vs ICF	0	0
9	coreHYB vs IB3	0	0
10	CNN vs DROP3	0	0
11	CPruner vs ICF	0,000001	0,000009
12	CPruner vs IB3	0,000001	0,000018
13	ICF vs Crisp01	0,000002	0,000026
14	IB3 vs Crisp01	0,000003	0,000048
15	ENN vs IB3	0,000005	0,00007
16	ENN vs ICF	0,00001	0,000124
17	DROP3 vs ICF	0,00044	0,005286
18	DROP3 vs IB3	0,000731	0,008043
19	coreHYB vs DROP3	0,001398	0,013981
20	CNN vs IB3	0,00617	0,055529
21	CNN vs ICF	0,009277	0,074214
22	coreHYB vs Crisp01	0,055234	0,38664
23	CNN vs ENN	0,067889	0,407335
24	coreHYB vs CPruner	0,091256	0,456278
25	CPruner vs DROP3	0,132006	0,528025
26	DROP3 vs Crisp01	0,201243	0,603728
27	CPruner vs Crisp01	0,819477	1,638954
28	IB3 vs ICF	0,891086	1,638954

Tabla A.7: Comparación de las reducciones obtenidas por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 217,15.

Método	Ranking
CNN	6,825
coreHYB	1,75
CPruner	2,6
DROP3	3,475
ENN	7,825
IB3	5,25
ICF	5,375
Fuzzy01	2,9

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ENN	0	0
2	CPruner vs ENN	0	0
3	CNN vs coreHYB	0	0
4	ENN vs Fuzzy01	0	0
5	DROP3 vs ENN	0	0
6	CNN vs CPruner	0	0
7	CNN vs Fuzzy01	0	0
8	coreHYB vs ICF	0	0
9	coreHYB vs IB3	0	0
10	CNN vs DROP3	0	0
11	CPruner vs ICF	0	0,000007
12	CPruner vs IB3	0,000001	0,000022
13	ENN vs IB3	0,000003	0,000041
14	ICF vs Fuzzy01	0,000006	0,000093
15	ENN vs ICF	0,000008	0,000108
16	IB3 vs Fuzzy01	0,000018	0,000232
17	DROP3 vs ICF	0,000523	0,006271
18	DROP3 vs IB3	0,001192	0,013116
19	coreHYB vs DROP3	0,001636	0,01636
20	CNN vs IB3	0,004033	0,0363
21	CNN vs ICF	0,008113	0,064905
22	coreHYB vs Fuzzy01	0,035764	0,250346
23	CNN vs ENN	0,067889	0,407335
24	CPruner vs DROP3	0,110149	0,550745
25	coreHYB vs CPruner	0,120691	0,550745
26	DROP3 vs Fuzzy01	0,293809	0,881428
27	CPruner vs Fuzzy01	0,583882	1,167765
28	IB3 vs ICF	0,819477	1,167765

Tabla A.8: Comparación de las reducciones obtenidas por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 219,864583.

Método	Ranking
CNN	6,825
coreHYB	1,8375
CPruner	2,675
DROP3	3,525
ENN	7,825
IB3	5,325
ICF	5,4
Crisp02	2,5875

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ENN	0	0
2	ENN vs Crisp02	0	0
3	CPruner vs ENN	0	0
4	CNN vs coreHYB	0	0
5	DROP3 vs ENN	0	0
6	CNN vs Crisp02	0	0
7	CNN vs CPruner	0	0
8	coreHYB vs ICF	0	0
9	coreHYB vs IB3	0	0
10	CNN vs DROP3	0	0
11	ICF vs Crisp02	0	0,000005
12	IB3 vs Crisp02	0,000001	0,00001
13	CPruner vs ICF	0,000001	0,00001
14	CPruner vs IB3	0,000001	0,00002
15	ENN vs IB3	0,000005	0,00007
16	ENN vs ICF	0,00001	0,000124
17	DROP3 vs ICF	0,000619	0,007425
18	DROP3 vs IB3	0,001015	0,011165
19	coreHYB vs DROP3	0,002063	0,020635
20	CNN vs IB3	0,00617	0,055529
21	CNN vs ICF	0,009277	0,074214
22	CNN vs ENN	0,067889	0,475224
23	DROP3 vs Crisp02	0,086964	0,521786
24	CPruner vs DROP3	0,120691	0,603454
25	coreHYB vs CPruner	0,12625	0,603454
26	coreHYB vs Crisp02	0,170904	0,603454
27	CPruner vs Crisp02	0,873076	1,746152
28	IB3 vs ICF	0,891086	1,746152

Tabla A.9: Comparación de las reducciones obtenidas por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 218,791667.

Método	Ranking
CNN	6,825
coreHYB	1,825
CPruner	2,675
DROP3	3,575
ENN	7,825
IB3	5,3
ICF	5,375
Fuzzy02	2,6

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ENN	0	0
2	ENN vs Fuzzy02	0	0
3	CPruner vs ENN	0	0
4	CNN vs coreHYB	0	0
5	DROP3 vs ENN	0	0
6	CNN vs Fuzzy02	0	0
7	CNN vs CPruner	0	0
8	coreHYB vs ICF	0	0
9	coreHYB vs IB3	0	0
10	CNN vs DROP3	0	0
11	ICF vs Fuzzy02	0	0,000007
12	IB3 vs Fuzzy02	0,000001	0,000014
13	CPruner vs ICF	0,000001	0,000014
14	CPruner vs IB3	0,000002	0,000025
15	ENN vs IB3	0,000004	0,000056
16	ENN vs ICF	0,000008	0,0001
17	DROP3 vs ICF	0,001015	0,01218
18	coreHYB vs DROP3	0,001398	0,015379
19	DROP3 vs IB3	0,001636	0,01636
20	CNN vs IB3	0,005365	0,048286
21	CNN vs ICF	0,008113	0,064905
22	CNN vs ENN	0,067889	0,475224
23	DROP3 vs Fuzzy02	0,07506	0,475224
24	CPruner vs DROP3	0,100348	0,501741
25	coreHYB vs CPruner	0,120691	0,501741
26	coreHYB vs Fuzzy02	0,157083	0,501741
27	CPruner vs Fuzzy02	0,891086	1,782171
28	IB3 vs ICF	0,891086	1,782171

Tabla A.10: Comparación de las reducciones obtenidas por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 79,277083.

Método	Ranking
CNN	4,8
coreHYB	4,7
CPruner	5,3875
DROP3	4,825
ENN	1,975
IB3	5,4
ICF	5,8125
Crisp01	3,1

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs ICF	0	0
2	ENN vs IB3	0	0
3	CPruner vs ENN	0	0
4	DROP3 vs ENN	0	0,000005
5	CNN vs ENN	0	0,000006
6	coreHYB vs ENN	0,000001	0,000015
7	ICF vs Crisp01	0,000001	0,000016
8	IB3 vs Crisp01	0,000027	0,000562
9	CPruner vs Crisp01	0,00003	0,000592
10	DROP3 vs Crisp01	0,001636	0,031085
11	CNN vs Crisp01	0,001911	0,034394
12	coreHYB vs Crisp01	0,003487	0,059279
13	ENN vs Crisp01	0,03998	0,639674
14	coreHYB vs ICF	0,042241	0,639674
15	CNN vs ICF	0,064521	0,903292
16	DROP3 vs ICF	0,071401	0,92821
17	coreHYB vs IB3	0,201243	2,414911
18	coreHYB vs CPruner	0,209407	2,414911
19	CNN vs IB3	0,273322	2,733217
20	CNN vs CPruner	0,28344	2,733217
21	DROP3 vs IB3	0,293809	2,733217
22	CPruner vs DROP3	0,30443	2,733217
23	CPruner vs ICF	0,437784	2,733217
24	IB3 vs ICF	0,451379	2,733217
25	coreHYB vs DROP3	0,819477	3,277908
26	CNN vs coreHYB	0,855132	3,277908
27	CNN vs DROP3	0,963594	3,277908
28	CPruner vs IB3	0,981792	3,277908

Tabla A.11: Comparación de las precisiones obtenidas (con KNN) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 78,65.

Método	Ranking
CNN	4,825
coreHYB	4,65
CPruner	5,4125
DROP3	4,85
ENN	2,0375
IB3	5,35
ICF	5,8375
Fuzzy01	3,0375

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs ICF	0	0
2	CPruner vs ENN	0	0
3	ENN vs IB3	0	0
4	DROP3 vs ENN	0	0,000007
5	ICF vs Fuzzy01	0	0,000008
6	CNN vs ENN	0	0,000008
7	coreHYB vs ENN	0,000002	0,000041
8	CPruner vs Fuzzy01	0,000015	0,000305
9	IB3 vs Fuzzy01	0,000024	0,000484
10	DROP3 vs Fuzzy01	0,000936	0,01778
11	CNN vs Fuzzy01	0,0011	0,019807
12	coreHYB vs Fuzzy01	0,00324	0,055078
13	coreHYB vs ICF	0,030153	0,482456
14	CNN vs ICF	0,064521	0,967813
15	ENN vs Fuzzy01	0,067889	0,967813
16	DROP3 vs ICF	0,071401	0,967813
17	coreHYB vs CPruner	0,163884	1,966603
18	coreHYB vs IB3	0,201243	2,213669
19	CNN vs CPruner	0,28344	2,834401
20	CPruner vs DROP3	0,30443	2,834401
21	CNN vs IB3	0,337803	2,834401
22	DROP3 vs IB3	0,36131	2,834401
23	IB3 vs ICF	0,373439	2,834401
24	CPruner vs ICF	0,437784	2,834401
25	coreHYB vs DROP3	0,715001	2,860003
26	CNN vs coreHYB	0,749344	2,860003
27	CPruner vs IB3	0,909151	2,860003
28	CNN vs DROP3	0,963594	2,860003

Tabla A.12: Comparación de las precisiones obtenidas (con KNN) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 83,285417.

Método	Ranking
CNN	4,85
coreHYB	4,7
CPruner	5,4125
DROP3	4,85
ENN	2
IB3	5,425
ICF	5,8375
Crisp02	2,925

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs ICF	0	0
2	ENN vs IB3	0	0
3	CPruner vs ENN	0	0
4	ICF vs Crisp02	0	0,000003
5	CNN vs ENN	0	0,000005
6	DROP3 vs ENN	0	0,000005
7	coreHYB vs ENN	0,000001	0,000018
8	IB3 vs Crisp02	0,000005	0,000105
9	CPruner vs Crisp02	0,000006	0,000112
10	CNN vs Crisp02	0,00044	0,008369
11	DROP3 vs Crisp02	0,00044	0,008369
12	coreHYB vs Crisp02	0,001192	0,020271
13	coreHYB vs ICF	0,037822	0,605148
14	DROP3 vs ICF	0,071401	1,071011
15	CNN vs ICF	0,071401	1,071011
16	ENN vs Crisp02	0,091256	1,186323
17	coreHYB vs IB3	0,185615	2,227381
18	coreHYB vs CPruner	0,193313	2,227381
19	DROP3 vs IB3	0,293809	2,938094
20	CNN vs IB3	0,293809	2,938094
21	CPruner vs DROP3	0,30443	2,938094
22	CNN vs CPruner	0,30443	2,938094
23	CPruner vs ICF	0,437784	2,938094
24	IB3 vs ICF	0,451379	2,938094
25	CNN vs coreHYB	0,784191	3,136765
26	coreHYB vs DROP3	0,784191	3,136765
27	CPruner vs IB3	0,981792	3,136765
28	CNN vs DROP3	1	3,136765

Tabla A.13: Comparación de las precisiones obtenidas (con KNN) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 79,14375.

Método	Ranking
CNN	4,775
coreHYB	4,6
CPruner	5,4125
DROP3	4,875
ENN	1,95
IB3	5,375
ICF	5,8375
Fuzzy02	3,175

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs,ICF	0	0
2	CPruner vs,ENN	0	0
3	ENN vs,IB3	0	0
4	DROP3 vs,ENN	0	0,000002
5	CNN vs,ENN	0	0,000006
6	ICF vs,Fuzzy02	0,000001	0,000027
7	coreHYB vs,ENN	0,000001	0,000029
8	CPruner vs,Fuzzy02	0,000044	0,000925
9	IB3 vs,Fuzzy02	0,000059	0,001181
10	DROP3 vs,Fuzzy02	0,001911	0,036305
11	CNN vs,Fuzzy02	0,003487	0,062766
12	coreHYB vs,Fuzzy02	0,009277	0,157705
13	coreHYB vs,ICF	0,023861	0,38178
14	ENN vs,Fuzzy02	0,025317	0,38178
15	CNN vs,ICF	0,052398	0,73357
16	DROP3 vs,ICF	0,078871	1,025319
17	coreHYB vs,CPruner	0,137964	1,655569
18	coreHYB vs,IB3	0,157083	1,727915
19	CNN vs,CPruner	0,24446	2,444603
20	CNN vs,IB3	0,273322	2,459895
21	CPruner vs,DROP3	0,326427	2,611416
22	DROP3 vs,IB3	0,36131	2,611416
23	IB3 vs,ICF	0,398443	2,611416
24	CPruner vs,ICF	0,437784	2,611416
25	coreHYB vs,DROP3	0,615612	2,611416
26	CNN vs,coreHYB	0,749344	2,611416
27	CNN vs,DROP3	0,855132	2,611416
28	CPruner vs,IB3	0,945415	2,611416

Tabla A.14: Comparación de las precisiones obtenidas (con KNN) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 62,791667.

Método	Ranking
CNN	5,15
coreHYB	3,625
CPruner	4,7
DROP3	4,25
ENN	5,3
IB3	5,025
ICF	5,8
Crisp01	2,15

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ICF vs,Crisp01	0	0
2	ENN vs,Crisp01	0	0
3	CNN vs,Crisp01	0	0,000001
4	IB3 vs,Crisp01	0	0,000004
5	CPruner vs,Crisp01	0,000003	0,000078
6	coreHYB vs,ICF	0,000072	0,001646
7	DROP3 vs,Crisp01	0,000126	0,002773
8	coreHYB vs,ENN	0,002227	0,046774
9	DROP3 vs,ICF	0,004656	0,093125
10	CNN vs,coreHYB	0,005365	0,101936
11	coreHYB vs,Crisp01	0,007082	0,127474
12	coreHYB vs,IB3	0,010587	0,179981
13	CPruner vs,ICF	0,04461	0,713755
14	coreHYB vs,CPruner	0,049684	0,745264
15	DROP3 vs,ENN	0,055234	0,77328
16	CNN vs,DROP3	0,100348	1,304527
17	IB3 vs,ICF	0,157083	1,884998
18	DROP3 vs,IB3	0,157083	1,884998
19	CNN vs,ICF	0,235333	2,353333
20	coreHYB vs,DROP3	0,253833	2,353333
21	CPruner vs,ENN	0,273322	2,353333
22	ENN vs,ICF	0,36131	2,529173
23	CPruner vs,DROP3	0,411314	2,529173
24	CNN vs,CPruner	0,411314	2,529173
25	CPruner vs,IB3	0,552936	2,529173
26	ENN vs,IB3	0,615612	2,529173
27	CNN vs,ENN	0,784191	2,529173
28	CNN vs,IB3	0,819477	2,529173

Tabla A.15: Comparación de los balances obtenidos (con KNN) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 56,766667.

Método	Ranking
CNN	5,125
coreHYB	3,575
CPruner	4,7
DROP3	4,2
ENN	5,275
IB3	4,975
ICF	5,8
Fuzzy01	2,35

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ICF vs Fuzzy01	0	0
2	ENN vs Fuzzy01	0	0,000003
3	CNN vs Fuzzy01	0	0,000011
4	IB3 vs Fuzzy01	0,000002	0,000041
5	CPruner vs Fuzzy01	0,000018	0,000428
6	coreHYB vs ICF	0,000049	0,001118
7	DROP3 vs Fuzzy01	0,000731	0,016085
8	coreHYB vs ENN	0,001911	0,040126
9	DROP3 vs ICF	0,003487	0,06974
10	CNN vs coreHYB	0,004656	0,088469
11	coreHYB vs IB3	0,010587	0,190568
12	coreHYB vs Fuzzy01	0,025317	0,430386
13	coreHYB vs CPruner	0,03998	0,639674
14	CPruner vs ICF	0,04461	0,669146
15	DROP3 vs ENN	0,049684	0,695579
16	CNN vs DROP3	0,091256	1,186323
17	IB3 vs ICF	0,132006	1,584076
18	DROP3 vs IB3	0,157083	1,727915
19	CNN vs ICF	0,217809	2,178088
20	coreHYB vs DROP3	0,253833	2,284497
21	CPruner vs ENN	0,293809	2,350475
22	ENN vs ICF	0,337803	2,364624
23	CPruner vs DROP3	0,36131	2,364624
24	CNN vs CPruner	0,437784	2,364624
25	ENN vs IB3	0,583882	2,364624
26	CPruner vs IB3	0,615612	2,364624
27	CNN vs IB3	0,784191	2,364624
28	CNN vs ENN	0,784191	2,364624

Tabla A.16: Comparación de los balances obtenidos (con KNN) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 103,35.

Método	Ranking
CNN	5,375
coreHYB	3,375
CPruner	4,3
DROP3	3,875
ENN	6,85
IB3	4,7
ICF	5,55
Crisp02	1,975

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs Crisp02	0	0
2	ICF vs Crisp02	0	0
3	coreHYB vs ENN	0	0
4	CNN vs Crisp02	0	0
5	DROP3 vs ENN	0	0,000001
6	IB3 vs Crisp02	0,000001	0,000015
7	CPruner vs ENN	0,000003	0,000071
8	CPruner vs Crisp02	0,000022	0,000459
9	coreHYB vs ICF	0,000072	0,001432
10	ENN vs IB3	0,000087	0,001646
11	CNN vs coreHYB	0,000261	0,004693
12	DROP3 vs Crisp02	0,000523	0,008884
13	DROP3 vs ICF	0,002227	0,035637
14	CNN vs DROP3	0,00617	0,092548
15	CNN vs ENN	0,007082	0,099146
16	coreHYB vs Crisp02	0,010587	0,137633
17	coreHYB vs IB3	0,015559	0,186703
18	ENN vs ICF	0,017622	0,193843
19	CPruner vs ICF	0,022479	0,224789
20	CNN vs CPruner	0,049684	0,447158
21	coreHYB vs CPruner	0,091256	0,730045
22	IB3 vs ICF	0,120691	0,844836
23	DROP3 vs IB3	0,132006	0,844836
24	CNN vs IB3	0,217809	1,089044
25	coreHYB vs DROP3	0,36131	1,445242
26	CPruner vs DROP3	0,437784	1,445242
27	CPruner vs IB3	0,465209	1,445242
28	CNN vs ICF	0,749344	1,445242

Tabla A.17: Comparación de los balances obtenidos (con KNN) por los métodos individuales y la versión crisp del sistema con $b = 0,2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 97,033333.

Método	Ranking
CNN	5,375
coreHYB	3,25
CPruner	4,3
DROP3	3,85
ENN	6,85
IB3	4,625
ICF	5,525
Fuzzy02	2,225

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs Fuzzy02	0	0
2	coreHYB vs ENN	0	0
3	ICF vs Fuzzy02	0	0
4	CNN vs Fuzzy02	0	0
5	DROP3 vs ENN	0	0,000001
6	CPruner vs ENN	0,000003	0,000074
7	IB3 vs Fuzzy02	0,000012	0,000259
8	coreHYB vs ICF	0,000033	0,000687
9	ENN vs IB3	0,000049	0,000972
10	CNN vs coreHYB	0,000105	0,001987
11	CPruner vs Fuzzy02	0,000152	0,002729
12	DROP3 vs ICF	0,002227	0,037864
13	DROP3 vs Fuzzy02	0,003009	0,048142
14	CNN vs DROP3	0,005365	0,080476
15	CNN vs ENN	0,007082	0,099146
16	coreHYB vs IB3	0,01206	0,156775
17	ENN vs ICF	0,015559	0,186703
18	CPruner vs ICF	0,025317	0,278485
19	CNN vs CPruner	0,049684	0,496842
20	coreHYB vs CPruner	0,055234	0,497108
21	coreHYB vs Fuzzy02	0,061292	0,497108
22	IB3 vs ICF	0,100348	0,702438
23	DROP3 vs IB3	0,157083	0,942499
24	CNN vs IB3	0,170904	0,942499
25	coreHYB vs DROP3	0,273322	1,093287
26	CPruner vs DROP3	0,411314	1,233941
27	CPruner vs IB3	0,552936	1,233941
28	CNN vs ICF	0,784191	1,233941

Tabla A.18: Comparación de los balances obtenidos (con KNN) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

A.4. Desempeño del sistema ante C4.5, LDA y PART

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 100,95625.

Método	Ranking
CNN	3,75
coreHYB	6,425
CPruner	5,6125
DROP3	5,3
ENN	1,6
IB3	4,6625
ICF	3,825
Crisp01	4,825

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ENN	0	0
2	CPruner vs ENN	0	0
3	DROP3 vs ENN	0	0
4	ENN vs Crisp01	0	0
5	ENN vs IB3	0	0,000001
6	CNN vs coreHYB	0,000001	0,000024
7	coreHYB vs ICF	0,000002	0,000045
8	ENN vs ICF	0,000049	0,001021
9	CNN vs ENN	0,000087	0,001732
10	CNN vs CPruner	0,000673	0,012783
11	CPruner vs ICF	0,0011	0,019807
12	coreHYB vs IB3	0,001291	0,021955
13	coreHYB vs Crisp01	0,003487	0,055792
14	CNN vs DROP3	0,004656	0,069844
15	DROP3 vs ICF	0,007082	0,099146
16	coreHYB vs DROP3	0,03998	0,519735
17	CNN vs Crisp01	0,049684	0,596211
18	ICF vs Crisp01	0,067889	0,746781
19	CPruner vs IB3	0,082837	0,828374
20	CNN vs IB3	0,095716	0,86144
21	IB3 vs ICF	0,12625	1,009999
22	coreHYB vs CPruner	0,137964	1,009999
23	CPruner vs Crisp01	0,150499	1,009999
24	DROP3 vs IB3	0,24446	1,222302
25	DROP3 vs Crisp01	0,385817	1,54327
26	CPruner vs DROP3	0,568309	1,704926
27	IB3 vs Crisp01	0,766708	1,704926
28	CNN vs ICF	0,891086	1,704926

Tabla A.19: Comparación de las precisiones obtenidas (con C4.5) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 100,952083.

Método	Ranking
CNN	3,75
coreHYB	6,375
CPruner	5,6125
DROP3	5,3
ENN	1,575
IB3	4,6875
ICF	3,825
Fuzzy01	4,875

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ENN	0	0
2	CPruner vs ENN	0	0
3	DROP3 vs ENN	0	0
4	ENN vs Fuzzy01	0	0
5	ENN vs IB3	0	0
6	CNN vs coreHYB	0,000002	0,000038
7	coreHYB vs ICF	0,000003	0,000071
8	ENN vs ICF	0,00004	0,000838
9	CNN vs ENN	0,000072	0,001432
10	CNN vs CPruner	0,000673	0,012783
11	CPruner vs ICF	0,0011	0,019807
12	coreHYB vs IB3	0,002063	0,035079
13	CNN vs DROP3	0,004656	0,0745
14	coreHYB vs Fuzzy01	0,00617	0,092548
15	DROP3 vs ICF	0,007082	0,099146
16	CNN vs Fuzzy01	0,03998	0,519735
17	coreHYB vs DROP3	0,049684	0,596211
18	ICF vs Fuzzy01	0,055234	0,607577
19	CNN vs IB3	0,086964	0,869643
20	CPruner vs IB3	0,091256	0,869643
21	IB3 vs ICF	0,115325	0,922601
22	coreHYB vs CPruner	0,163884	1,147185
23	CPruner vs Fuzzy01	0,178146	1,147185
24	DROP3 vs IB3	0,263453	1,317265
25	DROP3 vs Fuzzy01	0,437784	1,751137
26	CPruner vs DROP3	0,568309	1,751137
27	IB3 vs Fuzzy01	0,732105	1,751137
28	CNN vs ICF	0,891086	1,751137

Tabla A.20: Comparación de las precisiones obtenidas (con C4.5) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 100,952083.

Método	Ranking
CNN	3,775
coreHYB	6,4125
CPruner	5,6125
DROP3	5,275
ENN	1,525
IB3	4,6875
ICF	3,85
Crisp02	4,8625

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ENN	0	0
2	CPruner vs ENN	0	0
3	DROP3 vs ENN	0	0
4	ENN vs Crisp02	0	0
5	ENN vs IB3	0	0
6	CNN vs coreHYB	0,000001	0,000034
7	coreHYB vs ICF	0,000003	0,000064
8	ENN vs ICF	0,000022	0,000459
9	CNN vs ENN	0,00004	0,000798
10	CNN vs CPruner	0,000794	0,01509
11	CPruner vs ICF	0,001291	0,023246
12	coreHYB vs IB3	0,001636	0,027813
13	coreHYB vs Crisp02	0,004656	0,0745
14	CNN vs DROP3	0,00617	0,092548
15	DROP3 vs ICF	0,009277	0,129875
16	coreHYB vs DROP3	0,037822	0,491683
17	CNN vs Crisp02	0,04709	0,565074
18	ICF vs Crisp02	0,064521	0,709729
19	CPruner vs IB3	0,091256	0,912556
20	CNN vs IB3	0,095716	0,912556
21	IB3 vs ICF	0,12625	1,009999
22	coreHYB vs CPruner	0,144127	1,009999
23	CPruner vs Crisp02	0,170904	1,025421
24	DROP3 vs IB3	0,28344	1,417201
25	DROP3 vs Crisp02	0,451379	1,805515
26	CPruner vs DROP3	0,537771	1,805515
27	IB3 vs Crisp02	0,749344	1,805515
28	CNN vs ICF	0,891086	1,805515

Tabla A.21: Comparación de las precisiones obtenidas (con C4.5) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 103,516667.

Método	Ranking
CNN	3,675
coreHYB	6,3625
CPruner	5,5625
DROP3	5,325
ENN	1,55
IB3	4,5625
ICF	3,825
Fuzzy02	5,1375

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ENN	0	0
2	CPruner vs ENN	0	0
3	DROP3 vs ENN	0	0
4	ENN vs Fuzzy02	0	0
5	ENN vs IB3	0	0,000001
6	CNN vs coreHYB	0,000001	0,000021
7	coreHYB vs ICF	0,000004	0,000079
8	ENN vs ICF	0,000033	0,000687
9	CNN vs ENN	0,000105	0,002092
10	CNN vs CPruner	0,000569	0,010807
11	coreHYB vs IB3	0,001015	0,01827
12	CPruner vs ICF	0,001513	0,025717
13	CNN vs DROP3	0,002591	0,04146
14	DROP3 vs ICF	0,00617	0,092548
15	CNN vs Fuzzy02	0,007582	0,106145
16	ICF vs Fuzzy02	0,016562	0,215308
17	coreHYB vs Fuzzy02	0,025317	0,303802
18	coreHYB vs DROP3	0,058198	0,640173
19	CPruner vs IB3	0,067889	0,678892
20	CNN vs IB3	0,105158	0,946422
21	coreHYB vs CPruner	0,144127	1,153016
22	DROP3 vs IB3	0,163884	1,153016
23	IB3 vs ICF	0,178146	1,153016
24	IB3 vs Fuzzy02	0,293809	1,469047
25	CPruner vs Fuzzy02	0,437784	1,751137
26	CPruner vs DROP3	0,664569	1,993707
27	DROP3 vs Fuzzy02	0,732105	1,993707
28	CNN vs ICF	0,784191	1,993707

Tabla A.22: Comparación de las precisiones obtenidas (con C4.5) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

Tabla A.23: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,1$ y el clasificador $C4,5$. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	2	1	CNN	2	1
coreHYB	0	4	coreHYB	0	4
CPruner	0	3	CPruner	0	3
DROP3	0	1	DROP3	0	1
ENN	7	0	ENN	7	0
IB3	1	1	IB3	1	1
ICF	2	1	ICF	2	1
Crisp	0	1	Fuzzy	0	1

Tabla A.24: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,2$ y el clasificador $C4,5$. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	2	1	CNN	3	1
coreHYB	0	4	coreHYB	0	4
CPruner	0	3	CPruner	0	3
DROP3	0	1	DROP3	0	2
ENN	7	0	ENN	7	0
IB3	1	1	IB3	1	1
ICF	2	1	ICF	2	1
Crisp	0	1	Fuzzy	0	1

(c) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 14,335417.

Método	Ranking
CNN	4,25
coreHYB	5,5125
CPruner	4,975
DROP3	4,75
ENN	4,275
IB3	4,425
ICF	3,825
Crisp01	3,9875

(d) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ICF	0,002063	0,057778
2	coreHYB vs Crisp01	0,005365	0,144857
3	CNN vs coreHYB	0,021167	0,550332
4	coreHYB vs ENN	0,023861	0,596532
5	CPruner vs ICF	0,035764	0,85833
6	coreHYB vs IB3	0,04709	1,083059
7	CPruner vs Crisp01	0,071401	1,570817
8	DROP3 vs ICF	0,091256	1,916368
9	DROP3 vs Crisp01	0,163884	3,277672
10	coreHYB vs DROP3	0,163884	3,277672
11	CNN vs CPruner	0,185615	3,341071
12	CPruner vs ENN	0,201243	3,421125
13	IB3 vs ICF	0,273322	4,373147
14	CPruner vs IB3	0,315302	4,729537
15	coreHYB vs CPruner	0,326427	4,729537
16	CNN vs DROP3	0,36131	4,729537
17	DROP3 vs ENN	0,385817	4,729537
18	ENN vs ICF	0,411314	4,729537
19	IB3 vs Crisp01	0,424428	4,729537
20	CNN vs ICF	0,437784	4,729537
21	DROP3 vs IB3	0,552936	4,729537
22	ENN vs Crisp01	0,599652	4,729537
23	CNN vs Crisp01	0,631756	4,729537
24	CPruner vs DROP3	0,681225	4,729537
25	CNN vs IB3	0,749344	4,729537
26	ICF vs Crisp01	0,766708	4,729537
27	ENN vs IB3	0,784191	4,729537
28	CNN vs ENN	0,963594	4,729537

Tabla A.25: Comparación de los balances obtenidos (con C4.5) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 14,835417.

Método	Ranking
CNN	4,225
coreHYB	5,5375
CPruner	4,975
DROP3	4,775
ENN	4,275
IB3	4,4
ICF	3,85
Fuzzy01	3,9625

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	coreHYB vs ICF	0,002063	0,057778
2	coreHYB vs Fuzzy01	0,004033	0,1089
3	CNN vs coreHYB	0,016562	0,430616
4	coreHYB vs ENN	0,021167	0,529166
5	coreHYB vs IB3	0,037822	0,907722
6	CPruner vs ICF	0,03998	0,919531
7	CPruner vs Fuzzy01	0,064521	1,419459
8	DROP3 vs ICF	0,091256	1,916368
9	DROP3 vs Fuzzy01	0,137964	2,759282
10	coreHYB vs DROP3	0,163884	3,113788
11	CNN vs CPruner	0,170904	3,113788
12	CPruner vs ENN	0,201243	3,421125
13	CPruner vs IB3	0,293809	4,70095
14	coreHYB vs CPruner	0,30443	4,70095
15	IB3 vs ICF	0,315302	4,70095
16	CNN vs DROP3	0,315302	4,70095
17	DROP3 vs ENN	0,36131	4,70095
18	IB3 vs Fuzzy01	0,424428	4,70095
19	ENN vs ICF	0,437784	4,70095
20	CNN vs ICF	0,493563	4,70095
21	DROP3 vs IB3	0,493563	4,70095
22	ENN vs Fuzzy01	0,568309	4,70095
23	CNN vs Fuzzy01	0,631756	4,70095
24	CPruner vs DROP3	0,715001	4,70095
25	CNN vs IB3	0,749344	4,70095
26	ENN vs IB3	0,819477	4,70095
27	ICF vs Fuzzy01	0,837263	4,70095
28	CNN vs ENN	0,927264	4,70095

Tabla A.26: Comparación de los balances obtenidos (con C4.5) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 38,24375.

Método	Ranking
CNN	4,85
coreHYB	4,8625
CPruner	4,3
DROP3	3,95
ENN	6,4
IB3	4,2
ICF	4,05
Crisp02	3,3875

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs Crisp02	0	0,000001
2	DROP3 vs ENN	0,000008	0,000208
3	ENN vs ICF	0,000018	0,000464
4	ENN vs IB3	0,000059	0,001476
5	CPruner vs ENN	0,000126	0,003025
6	CNN vs ENN	0,004656	0,107094
7	coreHYB vs ENN	0,004999	0,109985
8	coreHYB vs Crisp02	0,007082	0,148719
9	CNN vs Crisp02	0,007582	0,151636
10	coreHYB vs DROP3	0,095716	1,818595
11	CPruner vs Crisp02	0,095716	1,818595
12	CNN vs DROP3	0,100348	1,818595
13	IB3 vs Crisp02	0,137964	2,207426
14	coreHYB vs ICF	0,137964	2,207426
15	CNN vs ICF	0,144127	2,207426
16	ICF vs Crisp02	0,22645	2,943851
17	coreHYB vs IB3	0,22645	2,943851
18	CNN vs IB3	0,235333	2,943851
19	coreHYB vs CPruner	0,30443	3,0443
20	DROP3 vs Crisp02	0,30443	3,0443
21	CNN vs CPruner	0,315302	3,0443
22	CPruner vs DROP3	0,522817	3,659717
23	DROP3 vs IB3	0,648077	3,888461
24	CPruner vs ICF	0,648077	3,888461
25	IB3 vs ICF	0,784191	3,888461
26	DROP3 vs ICF	0,855132	3,888461
27	CPruner vs IB3	0,855132	3,888461
28	CNN vs coreHYB	0,981792	3,888461

Tabla A.27: Comparación de los balances obtenidos (con C4.5) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 35,95625.

Método	Ranking
CNN	4,8
coreHYB	4,8375
CPruner	4,275
DROP3	3,925
ENN	6,4
IB3	4,2
ICF	3,975
Fuzzy02	3,5875

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs Fuzzy02	0	0,000008
2	DROP3 vs ENN	0,000006	0,000168
3	ENN vs ICF	0,00001	0,000248
4	ENN vs IB3	0,000059	0,001476
5	CPruner vs ENN	0,000105	0,00251
6	CNN vs ENN	0,003487	0,080201
7	coreHYB vs ENN	0,004335	0,095362
8	coreHYB vs Fuzzy02	0,022479	0,472056
9	CNN vs Fuzzy02	0,026849	0,536972
10	coreHYB vs DROP3	0,095716	1,818595
11	CNN vs DROP3	0,110149	1,982681
12	coreHYB vs ICF	0,115325	1,982681
13	CNN vs ICF	0,132006	2,112101
14	CPruner vs Fuzzy02	0,209407	3,141106
15	coreHYB vs IB3	0,24446	3,422444
16	IB3 vs Fuzzy02	0,263453	3,424889
17	CNN vs IB3	0,273322	3,424889
18	coreHYB vs CPruner	0,30443	3,424889
19	CNN vs CPruner	0,337803	3,424889
20	ICF vs Fuzzy02	0,479271	4,313442
21	CPruner vs DROP3	0,522817	4,313442
22	DROP3 vs Fuzzy02	0,537771	4,313442
23	CPruner vs ICF	0,583882	4,313442
24	DROP3 vs IB3	0,615612	4,313442
25	IB3 vs ICF	0,681225	4,313442
26	CPruner vs IB3	0,891086	4,313442
27	DROP3 vs ICF	0,927264	4,313442
28	CNN vs coreHYB	0,945415	4,313442

Tabla A.28: Comparación de los balances obtenidos (con C4.5) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

Tabla A.29: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,1$ y el clasificador $C4,5$. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	0	CNN	0	0
coreHYB	0	0	coreHYB	0	0
CPruner	0	0	CPruner	0	0
DROP3	0	0	DROP3	0	0
ENN	0	0	ENN	0	0
IB3	0	0	IB3	0	0
ICF	0	0	ICF	0	0
Crisp	0	0	Fuzzy	0	0

Tabla A.30: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,2$ y el clasificador $C4,5$. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	0	CNN	0	0
coreHYB	0	0	coreHYB	0	0
CPruner	1	0	CPruner	1	0
DROP3	1	0	DROP3	1	0
ENN	0	5	ENN	0	5
IB3	1	0	IB3	1	0
ICF	1	0	ICF	1	0
Crisp	1	0	Fuzzy	1	0

(c) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 92,325.

Método	Ranking
CNN	2,9125
coreHYB	6,05
CPruner	6,4375
DROP3	4,725
ENN	2,6
IB3	3,6375
ICF	4,2875
Crisp01	5,35

(d) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CPruner vs ENN	0	0
2	CNN vs CPruner	0	0
3	coreHYB vs ENN	0	0
4	CNN vs coreHYB	0	0
5	CPruner vs IB3	0	0,000008
6	ENN vs Crisp01	0,000001	0,000012
7	CNN vs Crisp01	0,000009	0,000189
8	coreHYB vs IB3	0,000011	0,000223
9	CPruner vs ICF	0,000087	0,001732
10	DROP3 vs ENN	0,000105	0,001987
11	CNN vs DROP3	0,000936	0,016844
12	coreHYB vs ICF	0,001291	0,021955
13	CPruner vs DROP3	0,001769	0,028296
14	IB3 vs Crisp01	0,001769	0,028296
15	ENN vs ICF	0,002063	0,028889
16	CNN vs ICF	0,01206	0,156775
17	coreHYB vs DROP3	0,015559	0,186703
18	CPruner vs Crisp01	0,04709	0,517985
19	DROP3 vs IB3	0,04709	0,517985
20	ICF vs Crisp01	0,052398	0,517985
21	ENN vs IB3	0,058198	0,517985
22	CNN vs IB3	0,185615	1,299305
23	coreHYB vs Crisp01	0,201243	1,299305
24	IB3 vs ICF	0,235333	1,299305
25	DROP3 vs Crisp01	0,253833	1,299305
26	DROP3 vs ICF	0,424428	1,299305
27	coreHYB vs CPruner	0,479271	1,299305
28	CNN vs ENN	0,568309	1,299305

Tabla A.31: Comparación de las precisiones obtenidas (con LDA) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 92,9375.

Método	Ranking
CNN	2,8875
coreHYB	6,075
CPruner	6,4375
DROP3	4,725
ENN	2,625
IB3	3,5875
ICF	4,3375
Fuzzy01	5,325

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CPruner vs ENN	0	0
2	CNN vs CPruner	0	0
3	coreHYB vs ENN	0	0
4	CNN vs coreHYB	0	0
5	CPruner vs IB3	0	0,000005
6	ENN vs Fuzzy01	0,000001	0,000019
7	coreHYB vs IB3	0,000006	0,000123
8	CNN vs Fuzzy01	0,000009	0,00018
9	DROP3 vs ENN	0,000126	0,002521
10	CPruner vs ICF	0,000126	0,002521
11	CNN vs DROP3	0,000794	0,014296
12	coreHYB vs ICF	0,001513	0,025717
13	IB3 vs Fuzzy01	0,001513	0,025717
14	ENN vs ICF	0,001769	0,026528
15	CPruner vs DROP3	0,001769	0,026528
16	CNN vs ICF	0,008113	0,105471
17	coreHYB vs DROP3	0,013711	0,16453
18	DROP3 vs IB3	0,037822	0,416039
19	CPruner vs Fuzzy01	0,042241	0,42241
20	ICF vs Fuzzy01	0,071401	0,642607
21	ENN vs IB3	0,078871	0,642607
22	coreHYB vs Fuzzy01	0,170904	1,196325
23	IB3 vs ICF	0,170904	1,196325
24	CNN vs IB3	0,201243	1,196325
25	DROP3 vs Fuzzy01	0,273322	1,196325
26	DROP3 vs ICF	0,479271	1,437814
27	coreHYB vs CPruner	0,508079	1,437814
28	CNN vs ENN	0,631756	1,437814

Tabla A.32: Comparación de las precisiones obtenidas (con LDA) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 95,645833.

Método	Ranking
CNN	2,8875
coreHYB	6,025
CPruner	6,4375
DROP3	4,775
ENN	2,575
IB3	3,5625
ICF	4,2625
Crisp02	5,475

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CPruner vs ENN	0	0
2	CNN vs CPruner	0	0
3	coreHYB vs ENN	0	0
4	CNN vs coreHYB	0	0
5	ENN vs Crisp02	0	0,000003
6	CPruner vs IB3	0	0,000004
7	CNN vs Crisp02	0,000002	0,000051
8	coreHYB vs IB3	0,000007	0,000145
9	DROP3 vs ENN	0,000059	0,001181
10	CPruner vs ICF	0,000072	0,00136
11	IB3 vs Crisp02	0,00048	0,008638
12	CNN vs DROP3	0,000569	0,009669
13	coreHYB vs ICF	0,001291	0,020663
14	ENN vs ICF	0,002063	0,030952
15	CPruner vs DROP3	0,002403	0,033642
16	CNN vs ICF	0,01206	0,156775
17	coreHYB vs DROP3	0,022479	0,269746
18	DROP3 vs IB3	0,026849	0,295335
19	ICF vs Crisp02	0,026849	0,295335
20	ENN vs IB3	0,071401	0,642607
21	CPruner vs Crisp02	0,078871	0,642607
22	IB3 vs ICF	0,201243	1,408698
23	DROP3 vs Crisp02	0,201243	1,408698
24	CNN vs IB3	0,217809	1,408698
25	coreHYB vs Crisp02	0,315302	1,408698
26	DROP3 vs ICF	0,349431	1,408698
27	coreHYB vs CPruner	0,451379	1,408698
28	CNN vs ENN	0,568309	1,408698

Tabla A.33: Comparación de las precisiones obtenidas (con LDA) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 94,070833.

Método	Ranking
CNN	2,9125
coreHYB	6,05
CPruner	6,4375
DROP3	4,725
ENN	2,575
IB3	3,5875
ICF	4,3125
Fuzzy02	5,4

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CPruner vs ENN	0	0
2	CNN vs CPruner	0	0
3	coreHYB vs ENN	0	0
4	CNN vs coreHYB	0	0
5	CPruner vs IB3	0	0,000005
6	ENN vs Fuzzy02	0	0,000006
7	CNN vs Fuzzy02	0,000006	0,000123
8	coreHYB vs IB3	0,000007	0,000145
9	DROP3 vs ENN	0,000087	0,001732
10	CPruner vs ICF	0,000105	0,001987
11	CNN vs DROP3	0,000936	0,016844
12	IB3 vs Fuzzy02	0,000936	0,016844
13	ENN vs ICF	0,001513	0,024204
14	coreHYB vs ICF	0,001513	0,024204
15	CPruner vs DROP3	0,001769	0,024759
16	CNN vs ICF	0,010587	0,137633
17	coreHYB vs DROP3	0,015559	0,186703
18	DROP3 vs IB3	0,037822	0,416039
19	ICF vs Fuzzy02	0,04709	0,470895
20	CPruner vs Fuzzy02	0,058198	0,523778
21	ENN vs IB3	0,064521	0,523778
22	IB3 vs ICF	0,185615	1,299305
23	CNN vs IB3	0,217809	1,306853
24	DROP3 vs Fuzzy02	0,217809	1,306853
25	coreHYB vs Fuzzy02	0,235333	1,306853
26	DROP3 vs ICF	0,451379	1,354136
27	coreHYB vs CPruner	0,479271	1,354136
28	CNN vs ENN	0,537771	1,354136

Tabla A.34: Comparación de las precisiones obtenidas (con LDA) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

Tabla A.35: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,1$ y el clasificador *LDA*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	4	0	CNN	4	0
coreHYB	0	4	coreHYB	0	4
CPruner	0	5	CPruner	0	5
DROP3	1	2	DROP3	1	2
ENN	5	0	ENN	5	0
IB3	3	0	IB3	3	0
ICF	2	1	ICF	2	1
Crisp	0	3	Fuzzy	0	3

Tabla A.36: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,2$ y el clasificador *LDA*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	4	0	CNN	4	0
coreHYB	0	4	coreHYB	0	4
CPruner	0	5	CPruner	0	5
DROP3	1	2	DROP3	1	2
ENN	5	0	ENN	5	0
IB3	3	0	IB3	3	0
ICF	2	1	ICF	2	1
Crisp	0	3	Fuzzy	0	3

(c) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 27,55.

Método	Ranking
CNN	3,55
coreHYB	5,075
CPruner	5,725
DROP3	4,05
ENN	5,075
IB3	3,6
ICF	4,3
Crisp01	4,625

(d) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs CPruner	0,000072	0,002004
2	CPruner vs IB3	0,000105	0,002824
3	CPruner vs DROP3	0,002227	0,05791
4	CNN vs coreHYB	0,005365	0,134127
5	CNN vs ENN	0,005365	0,134127
6	coreHYB vs IB3	0,007082	0,162883
7	ENN vs IB3	0,007082	0,162883
8	CPruner vs ICF	0,009277	0,194812
9	CPruner vs Crisp01	0,04461	0,892194
10	CNN vs Crisp01	0,049684	0,944001
11	coreHYB vs DROP3	0,061292	1,10325
12	DROP3 vs ENN	0,061292	1,10325
13	IB3 vs Crisp01	0,061292	1,10325
14	coreHYB vs ICF	0,157083	2,356248
15	ENN vs ICF	0,157083	2,356248
16	CNN vs ICF	0,170904	2,356248
17	IB3 vs ICF	0,201243	2,414911
18	coreHYB vs CPruner	0,235333	2,588666
19	CPruner vs ENN	0,235333	2,588666
20	DROP3 vs Crisp01	0,293809	2,644284
21	CNN vs DROP3	0,36131	2,890483
22	coreHYB vs Crisp01	0,411314	2,890483
23	ENN vs Crisp01	0,411314	2,890483
24	DROP3 vs IB3	0,411314	2,890483
25	ICF vs Crisp01	0,552936	2,890483
26	DROP3 vs ICF	0,648077	2,890483
27	CNN vs IB3	0,927264	2,890483
28	coreHYB vs ENN	1	2,890483

Tabla A.37: Comparación de los balances obtenidos (con LDA) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 28,835417.

Método	Ranking
CNN	3,475
coreHYB	5,1375
CPruner	5,75
DROP3	4,125
ENN	5,125
IB3	3,65
ICF	4,275
Fuzzy01	4,4625

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs CPruner	0,000033	0,000917
2	CPruner vs IB3	0,000126	0,003403
3	CNN vs coreHYB	0,002403	0,062478
4	CNN vs ENN	0,002591	0,064782
5	CPruner vs DROP3	0,003009	0,072213
6	coreHYB vs IB3	0,006612	0,152071
7	CPruner vs ICF	0,007082	0,155801
8	ENN vs IB3	0,007082	0,155801
9	CPruner vs Fuzzy01	0,018741	0,37482
10	coreHYB vs DROP3	0,064521	1,225896
11	DROP3 vs ENN	0,067889	1,225896
12	CNN vs Fuzzy01	0,071401	1,225896
13	coreHYB vs ICF	0,115325	1,845203
14	ENN vs ICF	0,120691	1,845203
15	IB3 vs Fuzzy01	0,137964	1,931498
16	CNN vs ICF	0,144127	1,931498
17	coreHYB vs Fuzzy01	0,217809	2,613705
18	ENN vs Fuzzy01	0,22645	2,613705
19	CNN vs DROP3	0,235333	2,613705
20	CPruner vs ENN	0,253833	2,613705
21	IB3 vs ICF	0,253833	2,613705
22	coreHYB vs CPruner	0,263453	2,613705
23	DROP3 vs IB3	0,385817	2,613705
24	DROP3 vs Fuzzy01	0,537771	2,688853
25	ICF vs Fuzzy01	0,732105	2,92842
26	CNN vs IB3	0,749344	2,92842
27	DROP3 vs ICF	0,784191	2,92842
28	coreHYB vs ENN	0,981792	2,92842

Tabla A.38: Comparación de los balances obtenidos (con LDA) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 34,641667.

Método	Ranking
CNN	4,15
coreHYB	4,475
CPruner	5,2
DROP3	3,725
ENN	6,275
IB3	3,725
ICF	4,425
Crisp02	4,025

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs ENN	0,000003	0,00009
2	ENN vs IB3	0,000003	0,00009
3	ENN vs Crisp02	0,00004	0,001038
4	CNN vs ENN	0,000105	0,002615
5	ENN vs ICF	0,000731	0,017548
6	coreHYB vs ENN	0,001015	0,023345
7	CPruner vs DROP3	0,007082	0,155801
8	CPruner vs IB3	0,007082	0,155801
9	CPruner vs Crisp02	0,031933	0,638663
10	CPruner vs ENN	0,049684	0,944001
11	CNN vs CPruner	0,055234	0,994217
12	CPruner vs ICF	0,157083	2,670414
13	coreHYB vs DROP3	0,170904	2,734456
14	coreHYB vs IB3	0,170904	2,734456
15	coreHYB vs CPruner	0,185615	2,734456
16	DROP3 vs ICF	0,201243	2,734456
17	IB3 vs ICF	0,201243	2,734456
18	coreHYB vs Crisp02	0,411314	4,524452
19	CNN vs DROP3	0,437784	4,524452
20	CNN vs IB3	0,437784	4,524452
21	ICF vs Crisp02	0,465209	4,524452
22	CNN vs coreHYB	0,552936	4,524452
23	DROP3 vs Crisp02	0,583882	4,524452
24	IB3 vs Crisp02	0,583882	4,524452
25	CNN vs ICF	0,615612	4,524452
26	CNN vs Crisp02	0,819477	4,524452
27	coreHYB vs ICF	0,927264	4,524452
28	DROP3 vs IB3	1	4,524452

Tabla A.39: Comparación de los balances obtenidos (con LDA) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 34,310417.

Método	Ranking
CNN	4,15
coreHYB	4,4875
CPruner	5,175
DROP3	3,775
ENN	6,275
IB3	3,725
ICF	4,45
Fuzzy02	3,9625

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ENN vs IB3	0,000003	0,00009
2	DROP3 vs ENN	0,000005	0,000135
3	ENN vs Fuzzy02	0,000024	0,000629
4	CNN vs ENN	0,000105	0,002615
5	ENN vs ICF	0,000862	0,020695
6	coreHYB vs ENN	0,0011	0,025309
7	CPruner vs IB3	0,008113	0,178489
8	CPruner vs DROP3	0,010587	0,22233
9	CPruner vs Fuzzy02	0,026849	0,536972
10	CPruner vs ENN	0,04461	0,847585
11	CNN vs CPruner	0,061292	1,10325
12	coreHYB vs IB3	0,163884	2,786021
13	CPruner vs ICF	0,185615	2,969841
14	IB3 vs ICF	0,185615	2,969841
15	coreHYB vs DROP3	0,193313	2,969841
16	coreHYB vs CPruner	0,209407	2,969841
17	DROP3 vs ICF	0,217809	2,969841
18	coreHYB vs Fuzzy02	0,337803	3,715837
19	ICF vs Fuzzy02	0,373439	3,734395
20	CNN vs IB3	0,437784	3,940058
21	CNN vs DROP3	0,493563	3,948502
22	CNN vs coreHYB	0,537771	3,948502
23	CNN vs ICF	0,583882	3,948502
24	IB3 vs Fuzzy02	0,664569	3,948502
25	CNN vs Fuzzy02	0,732105	3,948502
26	DROP3 vs Fuzzy02	0,732105	3,948502
27	DROP3 vs IB3	0,927264	3,948502
28	coreHYB vs ICF	0,945415	3,948502

Tabla A.40: Comparación de los balances obtenidos (con LDA) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

Tabla A.41: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,1$ y el clasificador *LDA*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	1	0	CNN	1	0
coreHYB	0	0	coreHYB	0	0
CPruner	0	2	CPruner	0	2
DROP3	0	0	DROP3	0	0
ENN	0	0	ENN	0	0
IB3	1	0	IB3	1	0
ICF	0	0	ICF	0	0
Crisp	0	0	Fuzzy	0	0

Tabla A.42: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,2$ y el clasificador *LDA*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	1	0	CNN	1	0
coreHYB	1	0	coreHYB	1	0
CPruner	0	0	CPruner	0	0
DROP3	1	0	DROP3	1	0
ENN	0	6	ENN	0	6
IB3	1	0	IB3	1	0
ICF	1	0	ICF	1	0
Crisp	1	0	Fuzzy	1	0

(c) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 20,397917.

Método	Ranking
CNN	5,1125
coreHYB	4,9875
CPruner	4,5625
DROP3	4,3875
ENN	2,975
IB3	4,7625
ICF	4,525
Crisp01	4,6875

(d) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs ENN	0,000095	0,002665
2	coreHYB vs ENN	0,000238	0,006439
3	ENN vs IB3	0,0011	0,02861
4	ENN vs Crisp01	0,001769	0,044213
5	CPruner vs ENN	0,003751	0,090027
6	ENN vs ICF	0,004656	0,107094
7	DROP3 vs ENN	0,009913	0,218079
8	CNN vs DROP3	0,185615	3,897916
9	coreHYB vs DROP3	0,273322	5,466434
10	CNN vs ICF	0,28344	5,466434
11	CNN vs CPruner	0,315302	5,675444
12	coreHYB vs ICF	0,398443	6,773527
13	CNN vs Crisp01	0,437784	7,004548
14	coreHYB vs CPruner	0,437784	7,004548
15	DROP3 vs IB3	0,493563	7,004548
16	CNN vs IB3	0,522817	7,004548
17	coreHYB vs Crisp01	0,583882	7,006589
18	DROP3 vs Crisp01	0,583882	7,006589
19	IB3 vs ICF	0,664569	7,006589
20	coreHYB vs IB3	0,681225	7,006589
21	CPruner vs IB3	0,715001	7,006589
22	CPruner vs DROP3	0,749344	7,006589
23	ICF vs Crisp01	0,766708	7,006589
24	DROP3 vs ICF	0,801784	7,006589
25	CNN vs coreHYB	0,819477	7,006589
26	CPruner vs Crisp01	0,819477	7,006589
27	IB3 vs Crisp01	0,891086	7,006589
28	CPruner vs ICF	0,945415	7,006589

Tabla A.43: Comparación de las precisiones obtenidas (con PART) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 18,141667.

Método	Ranking
CNN	5,0375
coreHYB	5
CPruner	4,5125
DROP3	4,4875
ENN	3,05
IB3	4,7375
ICF	4,525
Fuzzy01	4,65

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs ENN	0,000285	0,007977
2	coreHYB vs ENN	0,000371	0,010006
3	ENN vs IB3	0,002063	0,053651
4	ENN vs Fuzzy01	0,003487	0,087175
5	ENN vs ICF	0,007082	0,169965
6	CPruner vs ENN	0,007582	0,174381
7	DROP3 vs ENN	0,008678	0,190906
8	CNN vs DROP3	0,315302	6,621351
9	CNN vs CPruner	0,337803	6,756068
10	coreHYB vs DROP3	0,349431	6,756068
11	CNN vs ICF	0,349431	6,756068
12	coreHYB vs CPruner	0,373439	6,756068
13	coreHYB vs ICF	0,385817	6,756068
14	CNN vs Fuzzy01	0,479271	7,18907
15	coreHYB vs Fuzzy01	0,522817	7,319433
16	CNN vs IB3	0,583882	7,590471
17	coreHYB vs IB3	0,631756	7,590471
18	DROP3 vs IB3	0,648077	7,590471
19	CPruner vs IB3	0,681225	7,590471
20	IB3 vs ICF	0,698038	7,590471
21	DROP3 vs Fuzzy01	0,766708	7,590471
22	CPruner vs Fuzzy01	0,801784	7,590471
23	ICF vs Fuzzy01	0,819477	7,590471
24	IB3 vs Fuzzy01	0,873076	7,590471
25	DROP3 vs ICF	0,945415	7,590471
26	CNN vs coreHYB	0,945415	7,590471
27	CPruner vs DROP3	0,963594	7,590471
28	CPruner vs ICF	0,981792	7,590471

Tabla A.44: Comparación de las precisiones obtenidas (con PART) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 21,827083.

Método	Ranking
CNN	5,0625
coreHYB	4,9375
CPruner	4,5375
DROP3	4,4125
ENN	2,9
IB3	4,7875
ICF	4,525
Crisp02	4,8375

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs ENN	0,000079	0,002205
2	coreHYB vs ENN	0,000199	0,00538
3	ENN vs Crisp02	0,000404	0,010507
4	ENN vs IB3	0,000569	0,014219
5	CPruner vs ENN	0,002793	0,067031
6	ENN vs ICF	0,003009	0,069204
7	DROP3 vs ENN	0,005755	0,126606
8	CNN vs DROP3	0,235333	4,941999
9	CNN vs ICF	0,326427	6,528539
10	coreHYB vs DROP3	0,337803	6,528539
11	CNN vs CPruner	0,337803	6,528539
12	DROP3 vs Crisp02	0,437784	7,442332
13	coreHYB vs ICF	0,451379	7,442332
14	coreHYB vs CPruner	0,465209	7,442332
15	DROP3 vs IB3	0,493563	7,442332
16	ICF vs Crisp02	0,568309	7,442332
17	CPruner vs Crisp02	0,583882	7,442332
18	CNN vs IB3	0,615612	7,442332
19	IB3 vs ICF	0,631756	7,442332
20	CPruner vs IB3	0,648077	7,442332
21	CNN vs Crisp02	0,681225	7,442332
22	coreHYB vs IB3	0,784191	7,442332
23	CPruner vs DROP3	0,819477	7,442332
24	CNN vs coreHYB	0,819477	7,442332
25	DROP3 vs ICF	0,837263	7,442332
26	coreHYB vs Crisp02	0,855132	7,442332
27	IB3 vs Crisp02	0,927264	7,442332
28	CPruner vs ICF	0,981792	7,442332

Tabla A.45: Comparación de las precisiones obtenidas (con PART) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 21,26875.

Método	Ranking
CNN	5,1125
coreHYB	4,9375
CPruner	4,4625
DROP3	4,4875
ENN	2,925
IB3	4,7875
ICF	4,55
Fuzzy02	4,7375

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs ENN	0,000065	0,001821
2	coreHYB vs ENN	0,000238	0,006439
3	ENN vs IB3	0,000673	0,017492
4	ENN vs Fuzzy02	0,000936	0,023394
5	ENN vs ICF	0,003009	0,072213
6	DROP3 vs ENN	0,004335	0,099697
7	CPruner vs ENN	0,004999	0,109985
8	CNN vs CPruner	0,235333	4,941999
9	CNN vs DROP3	0,253833	5,076661
10	CNN vs ICF	0,30443	5,78417
11	coreHYB vs CPruner	0,385817	6,944714
12	coreHYB vs DROP3	0,411314	6,992334
13	coreHYB vs ICF	0,479271	7,668342
14	CNN vs Fuzzy02	0,493563	7,668342
15	CNN vs IB3	0,552936	7,741106
16	CPruner vs IB3	0,552936	7,741106
17	DROP3 vs IB3	0,583882	7,741106
18	CPruner vs Fuzzy02	0,615612	7,741106
19	DROP3 vs Fuzzy02	0,648077	7,741106
20	IB3 vs ICF	0,664569	7,741106
21	coreHYB vs Fuzzy02	0,715001	7,741106
22	ICF vs Fuzzy02	0,732105	7,741106
23	CNN vs coreHYB	0,749344	7,741106
24	coreHYB vs IB3	0,784191	7,741106
25	CPruner vs ICF	0,873076	7,741106
26	DROP3 vs ICF	0,909151	7,741106
27	IB3 vs Fuzzy02	0,927264	7,741106
28	CPruner vs DROP3	0,963594	7,741106

Tabla A.46: Comparación de las precisiones obtenidas (con PART) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

Tabla A.47: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,1$ y el clasificador *PART*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	1	CNN	0	1
coreHYB	0	1	coreHYB	0	1
CPruner	0	0	CPruner	0	0
DROP3	0	0	DROP3	0	0
ENN	4	0	ENN	2	0
IB3	0	1	IB3	0	0
ICF	0	0	ICF	0	0
Crisp	0	1	Fuzzy	0	0

Tabla A.48: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en precisión con $b = 0,2$ y el clasificador *PART*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	1	CNN	0	1
coreHYB	0	1	coreHYB	0	1
CPruner	0	0	CPruner	0	0
DROP3	0	0	DROP3	0	0
ENN	4	0	ENN	4	0
IB3	0	1	IB3	0	1
ICF	0	0	ICF	0	0
Crisp	0	1	Fuzzy	0	1

(c) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 23,758333.

Método	Ranking
CNN	5,7
coreHYB	4,175
CPruner	3,6
DROP3	4,175
ENN	5,075
IB3	4,875
ICF	4,675
Crisp01	3,725

(d) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs CPruner	0,000126	0,003529
2	CNN vs Crisp01	0,000311	0,008401
3	CNN vs coreHYB	0,005365	0,139492
4	CNN vs DROP3	0,005365	0,139492
5	CPruner vs ENN	0,007082	0,169965
6	ENN vs Crisp01	0,013711	0,315349
7	CPruner vs IB3	0,019922	0,438276
8	IB3 vs Crisp01	0,035764	0,751039
9	CPruner vs ICF	0,049684	0,993685
10	CNN vs ICF	0,061292	1,164542
11	ICF vs Crisp01	0,082837	1,491074
12	coreHYB vs ENN	0,100348	1,70592
13	DROP3 vs ENN	0,100348	1,70592
14	CNN vs IB3	0,132006	1,980095
15	coreHYB vs IB3	0,201243	2,817397
16	DROP3 vs IB3	0,201243	2,817397
17	CNN vs ENN	0,253833	3,045996
18	coreHYB vs CPruner	0,293809	3,231903
19	CPruner vs DROP3	0,293809	3,231903
20	coreHYB vs ICF	0,36131	3,251794
21	DROP3 vs ICF	0,36131	3,251794
22	coreHYB vs Crisp01	0,411314	3,251794
23	DROP3 vs Crisp01	0,411314	3,251794
24	ENN vs ICF	0,465209	3,251794
25	ENN vs IB3	0,715001	3,251794
26	IB3 vs ICF	0,715001	3,251794
27	CPruner vs Crisp01	0,819477	3,251794
28	coreHYB vs DROP3	1	3,251794

Tabla A.49: Comparación de los balances obtenidos (con PART) por los métodos individuales y la versión crisp del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 19.922917.

Método	Ranking
CNN	5.625
coreHYB	4.1875
CPruner	3.575
DROP3	4.2
ENN	5
IB3	4.75
ICF	4.725
Fuzzy01	3.9375

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs .CPruner	0.000182	0.005096
2	CNN vs .Fuzzy01	0.002063	0.055714
3	CNN vs .coreHYB	0.008678	0.225616
4	CNN vs .DROP3	0.009277	0.231919
5	CPruner vs .ENN	0.009277	0.231919
6	CPruner vs .IB3	0.031933	0.734462
7	CPruner vs .ICF	0.035764	0.786803
8	ENN vs .Fuzzy01	0.052398	1.100355
9	CNN vs .ICF	0.100348	2.006965
10	CNN vs .IB3	0.110149	2.09283
11	coreHYB vs .ENN	0.137964	2.483354
12	IB3 vs .Fuzzy01	0.137964	2.483354
13	DROP3 vs .ENN	0.144127	2.483354
14	ICF vs .Fuzzy01	0.150499	2.483354
15	CNN vs .ENN	0.253833	3.553662
16	CPruner vs .DROP3	0.253833	3.553662
17	coreHYB vs .CPruner	0.263453	3.553662
18	coreHYB vs .IB3	0.30443	3.553662
19	DROP3 vs .IB3	0.315302	3.553662
20	coreHYB vs .ICF	0.326427	3.553662
21	DROP3 vs .ICF	0.337803	3.553662
22	CPruner vs .Fuzzy01	0.508079	3.556555
23	ENN vs .ICF	0.615612	3.693672
24	DROP3 vs .Fuzzy01	0.631756	3.693672
25	ENN vs .IB3	0.648077	3.693672
26	coreHYB vs .Fuzzy01	0.648077	3.693672
27	IB3 vs .ICF	0.963594	3.693672
28	coreHYB vs .DROP3	0.981792	3.693672

Tabla A.50: Comparación de los balances obtenidos (con PART) por los métodos individuales y la versión difusa del sistema con $b = 0, 1$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 56,389583.

Método	Ranking
CNN	5,7
coreHYB	3,7875
CPruner	3,025
DROP3	4,075
ENN	6,275
IB3	4,675
ICF	4,875
Crisp02	3,5875

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CPruner vs ENN	0	0
2	ENN vs Crisp02	0,000001	0,000025
3	CNN vs CPruner	0,000001	0,000027
4	coreHYB vs ENN	0,000006	0,00014
5	DROP3 vs ENN	0,000059	0,001417
6	CNN vs Crisp02	0,000115	0,002641
7	CNN vs coreHYB	0,00048	0,010558
8	CPruner vs ICF	0,000731	0,015354
9	CPruner vs IB3	0,002591	0,051826
10	CNN vs DROP3	0,003009	0,057168
11	ENN vs IB3	0,003487	0,062766
12	ENN vs ICF	0,010587	0,179981
13	ICF vs Crisp02	0,018741	0,299856
14	IB3 vs Crisp02	0,04709	0,706343
15	coreHYB vs ICF	0,04709	0,706343
16	CPruner vs DROP3	0,055234	0,718045
17	CNN vs IB3	0,061292	0,7355
18	coreHYB vs IB3	0,105158	1,156738
19	CNN vs ICF	0,132006	1,320063
20	DROP3 vs ICF	0,144127	1,320063
21	coreHYB vs CPruner	0,163884	1,320063
22	DROP3 vs IB3	0,273322	1,913252
23	CNN vs ENN	0,293809	1,913252
24	CPruner vs Crisp02	0,30443	1,913252
25	DROP3 vs Crisp02	0,373439	1,913252
26	coreHYB vs DROP3	0,599652	1,913252
27	coreHYB vs Crisp02	0,715001	1,913252
28	IB3 vs ICF	0,715001	1,913252

Tabla A.51: Comparación de los balances obtenidos (con PART) por los métodos individuales y la versión crisp del sistema con $b = 0, 2$. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 55,072917.

Método	Ranking
CNN	5,7
coreHYB	3,7625
CPruner	3
DROP3	4,125
ENN	6,275
IB3	4,6
ICF	4,825
Fuzzy02	3,7125

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CPruner vs ENN	0	0
2	CNN vs CPruner	0,000001	0,000022
3	ENN vs Fuzzy02	0,000003	0,000075
4	coreHYB vs ENN	0,000004	0,000112
5	DROP3 vs ENN	0,000087	0,002079
6	CNN vs Fuzzy02	0,000285	0,006553
7	CNN vs coreHYB	0,000404	0,008891
8	CPruner vs ICF	0,000862	0,018108
9	ENN vs IB3	0,002227	0,044546
10	CPruner vs IB3	0,003487	0,066253
11	CNN vs DROP3	0,004033	0,0726
12	ENN vs ICF	0,008113	0,137923
13	CPruner vs DROP3	0,03998	0,639674
14	ICF vs Fuzzy02	0,042241	0,639674
15	CNN vs IB3	0,04461	0,639674
16	coreHYB vs ICF	0,052398	0,681172
17	IB3 vs Fuzzy02	0,105158	1,261896
18	CNN vs ICF	0,110149	1,261896
19	coreHYB vs IB3	0,12625	1,262499
20	coreHYB vs CPruner	0,163884	1,474952
21	CPruner vs Fuzzy02	0,193313	1,546503
22	DROP3 vs ICF	0,201243	1,546503
23	CNN vs ENN	0,293809	1,762856
24	DROP3 vs IB3	0,385817	1,929087
25	DROP3 vs Fuzzy02	0,451379	1,929087
26	coreHYB vs DROP3	0,508079	1,929087
27	IB3 vs ICF	0,681225	1,929087
28	coreHYB vs Fuzzy02	0,927264	1,929087

Tabla A.52: Comparación de los balances obtenidos (con PART) por los métodos individuales y la versión difusa del sistema con $b = 0, 2$. Tests de Friedman y Holm.

Tabla A.53: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,1$ y el clasificador *PART*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	2	CNN	0	1
coreHYB	0	0	coreHYB	0	0
CPruner	1	0	CPruner	1	0
DROP3	0	0	DROP3	0	0
ENN	0	0	ENN	0	0
IB3	0	0	IB3	0	0
ICF	0	0	ICF	0	0
Crisp	1	0	Fuzzy	0	0

Tabla A.54: Resumen de los tests de Holm (comparaciones múltiples entre todos los métodos) aplicados a los resultados en balance con $b = 0,2$ y el clasificador *PART*. Cada test incluye una versión del sistema y todos los métodos individuales que este emplea.

(a) Versión crisp.			(b) Versión difusa.		
	veces mejor	veces peor		veces mejor	veces peor
CNN	0	3	CNN	0	3
coreHYB	2	0	coreHYB	2	0
CPruner	3	0	CPruner	3	0
DROP3	1	0	DROP3	1	0
ENN	0	4	ENN	0	5
IB3	0	0	IB3	1	0
ICF	0	1	ICF	0	1
Crisp	2	0	Fuzzy	2	0

Apéndice B

Detalles de los experimentos del capítulo 5

B.1. Desempeño del sistema

Tabla B.1: Reducciones logradas (para C4.5) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,2363	0,9518	0,9518	0,9549	0,9549	0,9654	0,9749	0,9749	0,9759	0,9780	0,9780
automobile	0,2590	0,2579	0,6567	0,8293	0,8293	0,8287	0,8293	0,8336	0,8558	0,8563	0,8580
balance	0,1847	0,2613	0,9545	0,9475	0,9582	0,9605	0,9577	0,9605	0,9614	0,9650	0,9678
bands	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922
breast	0,3965	0,8910	0,9631	0,9510	0,9623	0,9646	0,9576	0,9670	0,9576	0,9658	0,9623
car	0,1372	0,1362	0,0462	0,5085	0,9688	0,9662	0,9659	0,9685	0,9756	0,9754	0,9833
cleveland	0,4481	0,4481	0,7734	0,9134	0,9186	0,9303	0,9362	0,9410	0,9410	0,9410	0,9410
contraceptive	0,8529	0,8534	0,9022	0,9052	0,9045	0,9058	0,9046	0,9053	0,9048	0,9062	0,9050
dermatology	0,7175	0,8907	0,8907	0,8907	0,8907	0,8907	0,8907	0,8907	0,9007	0,9520	0,9520
ecoli	0,4854	0,9160	0,9160	0,9160	0,9160	0,9167	0,9173	0,9180	0,9180	0,9187	0,9190
flare	0,4558	0,4576	0,4591	0,6108	0,9745	0,9781	0,9807	0,9797	0,9795	0,9833	0,9826
glass	0,3089	0,5763	0,8510	0,8515	0,8546	0,8546	0,8604	0,8770	0,8806	0,8806	0,8806
haberman	0,3075	0,9256	0,9285	0,9357	0,9404	0,9426	0,9444	0,9481	0,9477	0,9481	0,9477
heart	0,2206	0,2206	0,9407	0,9407	0,9432	0,9473	0,9527	0,9543	0,9560	0,9560	0,9560
hepatitis	0,8211	0,8961	0,8961	0,8961	0,8961	0,8961	0,8961	0,8982	0,9082	0,9140	0,9147
horse-colic	0,3876	0,7314	0,8946	0,8946	0,8946	0,8946	0,8958	0,8958	0,8958	0,9012	0,9006
housevotes	0,0447	0,9778	0,9755	0,9760	0,9780	0,9780	0,9808	0,9808	0,9808	0,9808	0,9775
ionosphere	0,2877	0,8740	0,8740	0,8740	0,8740	0,8740	0,8740	0,8740	0,8740	0,8990	0,9047
iris	0,2230	0,3978	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9437	0,9533
led7digit	0,8307	0,8313	0,8649	0,9013	0,9276	0,9620	0,9789	0,9702	0,9793	0,9811	0,9838
lymphography	0,2422	0,3094	0,6562	0,7331	0,8964	0,8957	0,9099	0,9159	0,9091	0,9084	0,9106
mammographic	0,8512	0,9210	0,9216	0,9229	0,9218	0,9234	0,9228	0,9270	0,9274	0,9339	0,9356
monk-2	0,5554	0,7698	0,7919	0,7932	0,7950	0,8950	0,8905	0,8904	0,8902	0,8943	0,8948
movement_libras	0,2065	0,2065	0,3333	0,7883	0,7883	0,7883	0,7883	0,8315	0,8315	0,8315	0,8315
newthyroid	0,0579	0,0579	0,8462	0,9364	0,9364	0,9364	0,9364	0,9364	0,9364	0,9540	0,9705
pima	0,6432	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028
saheart	0,7784	0,8901	0,8901	0,8918	0,9000	0,9000	0,9000	0,9000	0,9000	0,9000	0,9000
segment	0,0409	0,9259	0,9259	0,9259	0,9259	0,9259	0,9259	0,9259	0,9259	0,9549	0,9549
sonar	0,1699	0,4742	0,8109	0,8253	0,8360	0,8381	0,8381	0,8381	0,8381	0,8381	0,8381
spectfheart	0,2863	0,9097	0,9097	0,9147	0,9251	0,9322	0,9322	0,9322	0,9322	0,9322	0,9322
splice	0,0521	0,0520	0,0520	0,9565	0,9565	0,9565	0,9565	0,9634	0,9797	0,9797	0,9798
tic-tac-toe	0,2130	0,1329	0,9363	0,9349	0,9355	0,9371	0,9357	0,9355	0,9340	0,9344	0,9357
vehicle	0,2913	0,4048	0,8501	0,8522	0,8545	0,8706	0,8727	0,8743	0,8760	0,8760	0,8760
vowel	0,0294	0,0294	0,1829	0,8148	0,8148	0,8148	0,8148	0,8148	0,8148	0,8148	0,8148
wdbc	0,3873	0,9309	0,9309	0,9309	0,9309	0,9309	0,9309	0,9631	0,9662	0,9662	0,9662
wine	0,2013	0,2838	0,8689	0,8689	0,8689	0,8689	0,8689	0,8689	0,9095	0,9214	0,9214
winequality-red	0,4652	0,8425	0,8897	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924
wisconsin	0,0326	0,9776	0,9792	0,9793	0,9779	0,9792	0,9777	0,9792	0,9804	0,9806	0,9808
yeast	0,4601	0,8848	0,8934	0,8994	0,9033	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049
zoo	0,0603	0,0603	0,1487	0,9142	0,9153	0,9153	0,9153	0,9176	0,9176	0,9176	0,9198
Promedio	0,3605	0,6063	0,7794	0,8773	0,9045	0,9095	0,9116	0,9143	0,9169	0,9219	0,9231

Tabla B.2.: Reducciones logradas (para C4.5) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,5616	0,7965	0,9486	0,9486	0,9518	0,9549	0,9613	0,9644	0,9707	0,9759	0,9780
automobile	0,2585	0,3725	0,7708	0,8319	0,8336	0,8379	0,8395	0,8444	0,8580	0,8580	0,8580
balance	0,2475	0,9378	0,9444	0,9458	0,9431	0,9595	0,9611	0,9646	0,9628	0,9643	0,9678
bands	0,3061	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922
breast	0,6433	0,8360	0,9561	0,9619	0,9658	0,9631	0,9604	0,9643	0,9604	0,9646	0,9643
car	0,0448	0,0453	0,0455	0,0455	0,0455	0,0455	0,0455	0,0455	0,0455	0,0455	0,0455
cleveland	0,5902	0,6825	0,9197	0,9197	0,9219	0,9248	0,9289	0,9347	0,9387	0,9387	0,9387
contraceptive	0,8710	0,8724	0,8868	0,8898	0,8932	0,9016	0,9054	0,9057	0,9071	0,9051	0,9058
dermatology	0,2033	0,8907	0,8907	0,8907	0,8907	0,8907	0,8907	0,8907	0,8907	0,8907	0,8907
ecoli	0,2710	0,8826	0,8948	0,9130	0,9177	0,9177	0,9177	0,9177	0,9177	0,9190	0,9190
flare	0,4578	0,5098	0,7670	0,9754	0,9767	0,9722	0,9755	0,9761	0,9762	0,9808	0,9814
glass	0,5363	0,5888	0,5888	0,8203	0,8770	0,8770	0,8770	0,8770	0,8780	0,8780	0,8806
haberman	0,5002	0,9339	0,9368	0,9335	0,9419	0,9375	0,9430	0,9455	0,9466	0,9473	0,9481
heart	0,3708	0,6543	0,9407	0,9424	0,9477	0,9477	0,9498	0,9514	0,9547	0,9560	0,9560
hepatitis	0,3924	0,8975	0,8975	0,8975	0,8975	0,9040	0,9068	0,9104	0,9147	0,9147	0,9147
horse-colic	0,1809	0,8916	0,8916	0,8916	0,8916	0,8916	0,8916	0,8916	0,8967	0,9000	0,9006
housevotes	0,7933	0,9803	0,9803	0,9798	0,9801	0,9803	0,9798	0,9798	0,9808	0,9808	0,9783
ionosphere	0,1421	0,7259	0,8705	0,8705	0,8705	0,8705	0,8708	0,8718	0,8740	0,8946	0,9047
iris	0,0474	0,4000	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9467	0,9533
led7digit	0,8180	0,8791	0,8293	0,8193	0,8647	0,8793	0,9491	0,9718	0,9720	0,9809	0,9829
lymphography	0,5319	0,6702	0,8867	0,8769	0,8912	0,9046	0,9076	0,9009	0,9054	0,9107	0,9144
mammographic	0,8652	0,9290	0,9308	0,9317	0,9324	0,9283	0,9341	0,9345	0,9376	0,9371	0,9354
monk-2	0,7285	0,8309	0,8106	0,8309	0,8529	0,8601	0,8837	0,8719	0,8832	0,8935	0,8938
movement_libras	0,2065	0,2065	0,5454	0,7830	0,7873	0,7873	0,7988	0,8296	0,8315	0,8315	0,8315
newthyroid	0,1475	0,3269	0,8493	0,9395	0,9395	0,9432	0,9545	0,9545	0,9592	0,9705	0,9705
pima	0,6405	0,8940	0,8967	0,8977	0,8977	0,8984	0,9022	0,9028	0,9028	0,9028	0,9028
saheart	0,7670	0,7670	0,8747	0,8850	0,8971	0,8971	0,8971	0,8985	0,9000	0,9000	0,9000
segment	0,0409	0,0409	0,9259	0,9259	0,9259	0,9259	0,9259	0,9259	0,9259	0,9549	0,9549
sonar	0,2394	0,5075	0,7670	0,8344	0,8344	0,8371	0,8371	0,8381	0,8381	0,8381	0,8381
spectheart	0,5290	0,8381	0,8960	0,9006	0,9080	0,9080	0,9168	0,9243	0,9280	0,9322	0,9322
splice	0,0520	0,0520	0,0520	0,0520	0,0520	0,0520	0,0520	0,0520	0,0520	0,0520	0,0520
tic-tac-toe	0,1322	0,1319	0,8939	0,8985	0,9038	0,9028	0,9062	0,9352	0,9363	0,9371	0,9355
vehicle	0,2913	0,7525	0,8105	0,8501	0,8743	0,8760	0,8760	0,8760	0,8760	0,8760	0,8760
vowel	0,0294	0,0294	0,4818	0,7996	0,7996	0,8062	0,8119	0,8148	0,8148	0,8148	0,8148
wdbc	0,3873	0,9270	0,9270	0,9270	0,9320	0,9391	0,9568	0,9613	0,9652	0,9662	0,9662
wine	0,0337	0,2888	0,8689	0,8689	0,8689	0,8689	0,8689	0,8739	0,9095	0,9214	0,9214
winequality-red	0,5058	0,8737	0,8851	0,8851	0,8851	0,8889	0,8906	0,8924	0,8924	0,8924	0,8924
wisconsin	0,0323	0,9655	0,9650	0,9660	0,9679	0,9679	0,9669	0,9704	0,9773	0,9806	0,9808
yeast	0,4601	0,8544	0,8860	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049
zoo	0,0603	0,0603	0,2366	0,5809	0,9153	0,9153	0,9153	0,9153	0,9153	0,9153	0,9198
Promedio	0,3729	0,6379	0,8016	0,8892	0,9033	0,9055	0,9109	0,9140	0,9172	0,9223	0,9231

Tabla B.3: Precisiones logradas (con C4.5) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8418	0,8691	0,8691	0,8600	0,8600	0,8309	0,8218	0,8218	0,8218	0,7418	0,7418
automobile	0,6887	0,6782	0,5680	0,5171	0,5121	0,5121	0,5071	0,5121	0,5059	0,5223	0,4895
balance	0,7921	0,7745	0,6562	0,6942	0,6640	0,6624	0,6898	0,6416	0,6817	0,6656	0,7120
bands	0,6457	0,6384	0,6458	0,6420	0,6217	0,6328	0,6421	0,6384	0,6329	0,6440	0,6142
breast	0,7202	0,6436	0,5638	0,6434	0,6469	0,6570	0,6897	0,5747	0,6609	0,6486	0,5881
car	0,8884	0,8933	0,9381	0,7332	0,5017	0,4954	0,5290	0,4375	0,4630	0,5099	0,4045
cleveland	0,5644	0,5644	0,4920	0,5023	0,4956	0,5113	0,4682	0,4748	0,4748	0,4748	0,4681
contraceptive	0,5316	0,5207	0,5255	0,5092	0,5044	0,5262	0,5188	0,4990	0,5140	0,5282	0,5065
dermatology	0,9315	0,9179	0,9179	0,9179	0,9179	0,9179	0,9179	0,9179	0,8369	0,5984	0,5984
ecoli	0,7859	0,7565	0,7565	0,7595	0,7595	0,7626	0,7504	0,7416	0,7447	0,7357	0,7595
flare	0,7251	0,7298	0,7270	0,6509	0,4681	0,5253	0,4412	0,4150	0,5079	0,4333	0,4287
glass	0,6493	0,5546	0,5426	0,5378	0,5078	0,5078	0,5442	0,5030	0,5211	0,5211	0,5211
haberman	0,7545	0,6955	0,6887	0,6984	0,7049	0,7085	0,6762	0,6923	0,6886	0,6956	0,6720
heart	0,7815	0,7815	0,7037	0,7037	0,6963	0,6926	0,6963	0,6926	0,6926	0,6926	0,7037
hepatitis	0,8142	0,7942	0,7942	0,7942	0,7942	0,7942	0,7942	0,7875	0,7688	0,7475	0,7808
horse-colic	0,8176	0,8041	0,7442	0,7442	0,7442	0,7442	0,7359	0,7359	0,7359	0,7414	0,7658
housevotes	0,9562	0,8987	0,8624	0,8638	0,8987	0,8987	0,9539	0,9539	0,9539	0,9539	0,8501
ionosphere	0,8689	0,8063	0,8063	0,8063	0,8063	0,8063	0,8063	0,8063	0,8063	0,6243	0,5529
iris	0,9533	0,9400	0,8800	0,8800	0,8800	0,8800	0,8800	0,8800	0,8800	0,7067	0,6133
led7digit	0,3800	0,3480	0,4040	0,2920	0,3680	0,3300	0,2340	0,2540	0,2840	0,2200	0,2060
lymphography	0,7334	0,7048	0,5949	0,6106	0,6096	0,6215	0,5701	0,6489	0,6388	0,5771	0,6380
mammographic	0,8263	0,8273	0,8138	0,8294	0,8263	0,8273	0,8221	0,8294	0,8159	0,8159	0,8242
monk-2	0,9773	0,9621	0,9749	0,9727	0,9681	0,8625	0,8138	0,8206	0,8206	0,8206	0,8138
movement_libras	0,6250	0,6250	0,6000	0,4861	0,4806	0,4861	0,4806	0,4056	0,4306	0,4194	0,3889
newthyroid	0,9310	0,9310	0,8571	0,8143	0,8143	0,8143	0,8143	0,8143	0,8143	0,6762	0,5329
pima	0,7435	0,7383	0,7383	0,7383	0,7383	0,7383	0,7383	0,7383	0,7383	0,7383	0,7383
saheart	0,6970	0,6819	0,6798	0,6862	0,6907	0,6949	0,6864	0,7037	0,6907	0,7036	0,7079
segment	0,9610	0,8104	0,8104	0,8104	0,8104	0,8104	0,8104	0,8104	0,8104	0,6463	0,6550
sonar	0,7212	0,6545	0,6210	0,5824	0,6064	0,6543	0,6448	0,6448	0,6543	0,6400	0,6352
spectfheart	0,7379	0,6520	0,6554	0,6668	0,7085	0,7158	0,6936	0,7010	0,6942	0,6788	0,6000
splice	0,9395	0,9395	0,9404	0,6147	0,6103	0,6116	0,6132	0,5734	0,5514	0,5718	0,5558
tic-tac-toe	0,8090	0,8308	0,5677	0,5585	0,5480	0,5440	0,5679	0,5792	0,5520	0,5709	0,5845
vehicle	0,7092	0,6770	0,6253	0,6217	0,6324	0,6229	0,6146	0,6359	0,6277	0,6442	0,6264
vowel	0,7909	0,7909	0,7475	0,5899	0,5798	0,5899	0,5899	0,5879	0,5899	0,5899	0,5697
wdbc	0,9402	0,9174	0,9227	0,9244	0,9192	0,9227	0,8947	0,8806	0,8578	0,8789	0,7205
wine	0,9261	0,8928	0,8147	0,8147	0,8147	0,8147	0,8147	0,8147	0,7369	0,6873	0,5605
winequality-red	0,5885	0,5641	0,5603	0,5641	0,5691	0,5553	0,5585	0,5566	0,5716	0,5710	0,5535
wisconsin	0,9499	0,8841	0,8741	0,8655	0,8841	0,8427	0,8598	0,8741	0,8641	0,8327	0,8685
yeast	0,5877	0,5405	0,5438	0,5512	0,5452	0,5459	0,5411	0,5405	0,5486	0,5486	0,5175
zoo	0,8811	0,8811	0,8544	0,5008	0,5108	0,5108	0,5108	0,5125	0,5125	0,5125	0,3458
Promedio	0,7792	0,7529	0,7221	0,6888	0,6805	0,6796	0,6734	0,6663	0,6674	0,6382	0,6104

Tabla B.4: Precisiones logradas (con C4.5) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8691	0,8691	0,8691	0,8691	0,8691	0,8600	0,8600	0,8500	0,8318	0,8218	0,7418
automobile	0,6782	0,6826	0,5814	0,5423	0,5380	0,5377	0,5327	0,5277	0,4895	0,4800	0,4895
balance	0,7729	0,6767	0,6498	0,6737	0,7348	0,6877	0,6978	0,7105	0,6773	0,6721	0,6307
bands	0,6421	0,6310	0,6292	0,6328	0,6477	0,6421	0,6235	0,6403	0,6217	0,6329	0,6142
breast	0,6567	0,6197	0,6269	0,5518	0,6273	0,6359	0,5667	0,5802	0,6406	0,6469	0,6128
car	0,9352	0,9387	0,9363	0,4584	0,4543	0,4462	0,5191	0,4704	0,4329	0,4629	0,4503
cleveland	0,5516	0,5583	0,5151	0,5117	0,5117	0,5084	0,4984	0,4751	0,4684	0,4684	0,4616
contraceptive	0,5200	0,5173	0,5234	0,5445	0,5336	0,5160	0,5092	0,5350	0,5350	0,5187	0,5296
dermatology	0,9478	0,9179	0,9179	0,9179	0,9179	0,9179	0,9179	0,9179	0,8874	0,5984	0,5984
ecoli	0,7920	0,7506	0,7388	0,7594	0,7595	0,7625	0,7447	0,7536	0,7478	0,7625	0,7595
flare	0,7176	0,7100	0,5829	0,4691	0,4595	0,4935	0,4888	0,5095	0,4087	0,4431	0,4803
glass	0,6370	0,6270	0,6270	0,5716	0,5516	0,5516	0,5516	0,5516	0,5559	0,5559	0,5211
haberman	0,7644	0,7382	0,7383	0,7383	0,7119	0,7217	0,7215	0,7180	0,7018	0,6825	0,6786
heart	0,7852	0,7852	0,7333	0,7407	0,7296	0,7296	0,7296	0,7148	0,7222	0,7148	0,7037
hepatitis	0,8321	0,8396	0,8396	0,8396	0,8396	0,8329	0,8196	0,7996	0,7933	0,7933	0,7808
horse-colic	0,8475	0,8015	0,8015	0,8015	0,8015	0,8015	0,8015	0,8015	0,7876	0,7741	0,7658
housevotes	0,9471	0,9425	0,9425	0,9425	0,9425	0,9425	0,9425	0,9425	0,9539	0,9539	0,9190
ionosphere	0,8946	0,8548	0,8234	0,8234	0,8234	0,8234	0,8234	0,8234	0,8063	0,6529	0,5529
iris	0,9533	0,9533	0,8800	0,8800	0,8800	0,8800	0,8800	0,8800	0,8800	0,7133	0,6133
led7digit	0,4200	0,3560	0,4320	0,3960	0,3500	0,3660	0,3380	0,2880	0,2500	0,2360	0,2000
lymphography	0,6710	0,6701	0,6557	0,6439	0,6109	0,6163	0,6196	0,6015	0,5839	0,6615	0,6154
mammographic	0,8253	0,8232	0,8179	0,8273	0,8107	0,8273	0,8284	0,8211	0,8148	0,8148	0,8023
monk-2	0,9562	0,9516	0,9539	0,9539	0,9084	0,9148	0,8684	0,9034	0,8706	0,8206	0,8229
movement_libras	0,6250	0,6250	0,5639	0,4889	0,4917	0,4917	0,4833	0,4167	0,4083	0,4056	0,3889
newthyroid	0,9310	0,9032	0,8571	0,7461	0,8143	0,7461	0,6797	0,7478	0,7297	0,6212	0,5329
pima	0,7658	0,7476	0,7411	0,7501	0,7435	0,7370	0,7448	0,7279	0,7292	0,7357	0,7383
saheart	0,7296	0,7296	0,7383	0,7296	0,7144	0,7144	0,7144	0,7144	0,7079	0,7079	0,7079
segment	0,9610	0,9610	0,8104	0,8104	0,8104	0,8104	0,8104	0,8104	0,8104	0,6550	0,6550
sonar	0,7212	0,6924	0,6586	0,6393	0,6393	0,6300	0,6490	0,6636	0,6586	0,6395	0,6352
spectheart	0,7536	0,7350	0,7387	0,6943	0,6943	0,7168	0,6792	0,6830	0,7127	0,6936	0,6000
splice	0,9404	0,9395	0,9404	0,6147	0,6160	0,6025	0,5680	0,5790	0,5696	0,5486	0,5690
tic-tac-toe	0,8360	0,8183	0,6252	0,5834	0,5480	0,5825	0,5970	0,5345	0,5584	0,6138	0,5845
vehicle	0,7092	0,6679	0,6429	0,6583	0,6323	0,6264	0,6264	0,6264	0,6264	0,6264	0,6264
vowel	0,7909	0,7909	0,7010	0,6131	0,6131	0,6081	0,6020	0,5838	0,5990	0,5788	0,5697
wdbc	0,9367	0,9227	0,9279	0,9262	0,9263	0,9263	0,8894	0,8824	0,8841	0,8806	0,7205
wine	0,9320	0,9095	0,8147	0,8147	0,8147	0,8147	0,8147	0,8147	0,7314	0,6412	0,5605
winequality-red	0,5785	0,5616	0,5553	0,5478	0,5603	0,5578	0,5622	0,5616	0,5610	0,5610	0,5535
wisconsin	0,9471	0,9199	0,9070	0,9242	0,9241	0,9313	0,9285	0,9341	0,8712	0,8384	0,8185
yeast	0,5877	0,5465	0,5546	0,5492	0,5411	0,5607	0,5573	0,5587	0,5431	0,5472	0,5175
zoo	0,8811	0,8811	0,8444	0,7542	0,5708	0,5708	0,5708	0,5708	0,5708	0,5708	0,3458
Promedio	0,7811	0,7642	0,7359	0,6983	0,6926	0,6909	0,6840	0,6806	0,6683	0,6437	0,6117

Tabla B.5: Reducciones logradas (para KNN) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8927	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780
automobile	0,7056	0,7061	0,7083	0,8238	0,8244	0,8287	0,8293	0,8461	0,8537	0,8569	0,8580
balance	0,9627	0,9620	0,9629	0,9641	0,9646	0,9655	0,9653	0,9684	0,9659	0,9657	0,9659
bands	0,7813	0,7813	0,7813	0,7813	0,7897	0,7924	0,7922	0,7922	0,7922	0,7922	0,7922
breast	0,3380	0,9588	0,8958	0,8949	0,8956	0,9658	0,9642	0,9631	0,9627	0,9658	0,9635
car	0,2323	0,3242	0,8789	0,9716	0,9754	0,9742	0,9756	0,9769	0,9758	0,9764	0,9826
cleveland	0,4481	0,7792	0,7792	0,9197	0,9248	0,9318	0,9365	0,9387	0,9410	0,9410	0,9410
contraceptive	0,8489	0,8792	0,8797	0,8822	0,8830	0,8855	0,8957	0,9060	0,9053	0,9037	0,9046
dermatology	0,0310	0,8907	0,9414	0,9520	0,9520	0,9520	0,9520	0,9520	0,9520	0,9520	0,9520
ecoli	0,8080	0,8968	0,9021	0,9127	0,9190	0,9190	0,9190	0,9190	0,9190	0,9190	0,9190
flare	0,4573	0,4581	0,4573	0,4563	0,9757	0,9750	0,9783	0,9818	0,9827	0,9827	0,9826
glass	0,5795	0,7490	0,8634	0,8675	0,8675	0,8675	0,8675	0,8675	0,8759	0,8801	0,8806
haberman	0,6224	0,9343	0,9419	0,9426	0,9423	0,9452	0,9459	0,9466	0,9463	0,9473	0,9484
heart	0,7564	0,8309	0,9086	0,9280	0,9560	0,9560	0,9560	0,9560	0,9560	0,9560	0,9560
hepatitis	0,2476	0,9004	0,9025	0,9047	0,9054	0,9097	0,9147	0,9147	0,9147	0,9147	0,9147
horse-colic	0,1809	0,7514	0,8922	0,8922	0,8922	0,8922	0,8922	0,8994	0,9006	0,9006	0,9006
housevotes	0,1404	0,4212	0,9801	0,9803	0,9793	0,9803	0,9803	0,9806	0,9803	0,9801	0,9783
ionosphere	0,7221	0,8677	0,8677	0,8677	0,8715	0,8841	0,8971	0,9047	0,9047	0,9047	0,9047
iris	0,2296	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533
led7digit	0,7947	0,9793	0,9827	0,9798	0,9824	0,9840	0,9818	0,9853	0,9798	0,9827	0,9800
lymphography	0,3984	0,5265	0,8768	0,9008	0,8889	0,9046	0,8889	0,8979	0,9084	0,9039	0,9121
mammographic	0,6555	0,9269	0,9264	0,9282	0,9257	0,9265	0,9255	0,9296	0,9259	0,9326	0,9379
monk-2	0,5152	0,5580	0,5206	0,6414	0,7415	0,8134	0,8581	0,8940	0,8935	0,8956	0,8912
movement_libras	0,6123	0,7793	0,7793	0,7846	0,7914	0,7914	0,8176	0,8315	0,8315	0,8315	0,8315
newthyroid	0,1501	0,9416	0,9416	0,9416	0,9416	0,9416	0,9416	0,9555	0,9705	0,9705	0,9705
pima	0,4534	0,8364	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028
saheart	0,7805	0,8834	0,8834	0,8834	0,8834	0,8855	0,8855	0,8874	0,8976	0,9000	0,9000
segment	0,2241	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549
sonar	0,3005	0,8050	0,8114	0,8114	0,8114	0,8258	0,8323	0,8381	0,8381	0,8381	0,8381
spectfheart	0,6429	0,9076	0,9138	0,9263	0,9322	0,9322	0,9322	0,9322	0,9322	0,9322	0,9322
splice	0,0519	0,0519	0,7920	0,9772	0,9772	0,9773	0,9773	0,9773	0,9786	0,9797	0,9799
tic-tac-toe	0,1356	0,2142	0,9015	0,9108	0,9273	0,9306	0,9367	0,9359	0,9359	0,9356	0,9373
vehicle	0,8034	0,8609	0,8609	0,8609	0,8609	0,8609	0,8609	0,8649	0,8688	0,8760	0,8760
vowel	0,0294	0,7828	0,7828	0,7973	0,8148	0,8148	0,8148	0,8148	0,8148	0,8148	0,8148
wdbc	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662
wine	0,3788	0,9058	0,9214	0,9214	0,9214	0,9214	0,9214	0,9214	0,9214	0,9214	0,9214
winequality-red	0,5025	0,8782	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924
wisconsin	0,6944	0,9806	0,9809	0,9809	0,9806	0,9809	0,9808	0,9806	0,9808	0,9804	0,9808
yeast	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049
zoo	0,0603	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198
Promedio	0,5010	0,7897	0,8723	0,8915	0,9093	0,9147	0,9172	0,9208	0,9220	0,9227	0,9230

Tabla B.6: Reducciones logradas (para KNN) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,3216	0,9696	0,9696	0,9728	0,9728	0,9738	0,9780	0,9780	0,9780	0,9780	0,9780
automobile	0,8211	0,8217	0,8211	0,8217	0,8206	0,8222	0,8222	0,8282	0,8504	0,8585	0,8580
balance	0,6523	0,9554	0,9598	0,9623	0,9625	0,9627	0,9662	0,9685	0,9650	0,9614	0,9623
bands	0,7842	0,7885	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922
breast	0,6462	0,9534	0,9635	0,9553	0,9639	0,9619	0,9650	0,9608	0,9635	0,9654	0,9612
car	0,0455	0,0466	0,0460	0,9758	0,9741	0,9759	0,9809	0,9794	0,9819	0,9829	0,9835
cleveland	0,7746	0,9153	0,9153	0,9197	0,9245	0,9267	0,9333	0,9399	0,9399	0,9410	0,9410
contraceptive	0,8927	0,9031	0,9044	0,9047	0,9053	0,9053	0,9048	0,9051	0,9047	0,9053	0,9041
dermatology	0,0310	0,8907	0,8907	0,8907	0,8968	0,9029	0,9414	0,9517	0,9517	0,9520	0,9520
ecoli	0,3449	0,9114	0,9114	0,9170	0,9170	0,9170	0,9170	0,9170	0,9187	0,9187	0,9190
flare	0,4579	0,4610	0,4533	0,4581	0,9799	0,9813	0,9830	0,9826	0,9829	0,9820	0,9830
glass	0,6947	0,8598	0,8598	0,8634	0,8634	0,8676	0,8707	0,8718	0,8744	0,8785	0,8806
haberman	0,6216	0,9263	0,9263	0,9343	0,9372	0,9415	0,9433	0,9470	0,9473	0,9477	0,9473
heart	0,6938	0,9103	0,9103	0,9103	0,9230	0,9453	0,9560	0,9560	0,9560	0,9560	0,9560
hepatitis	0,5420	0,8290	0,8982	0,8982	0,9004	0,9004	0,9075	0,9104	0,9147	0,9147	0,9147
horse-colic	0,2476	0,6809	0,8898	0,8898	0,8979	0,8979	0,8979	0,8985	0,9000	0,9000	0,9006
housevotes	0,0450	0,2329	0,8872	0,9793	0,9808	0,9803	0,9806	0,9798	0,9798	0,9806	0,9788
ionosphere	0,8677	0,8677	0,8677	0,8712	0,8734	0,8930	0,9047	0,9047	0,9047	0,9047	0,9047
iris	0,4970	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533	0,9533
led7digit	0,9393	0,8973	0,9720	0,9733	0,9782	0,9633	0,9756	0,9776	0,9758	0,9800	0,9791
lymphography	0,4606	0,6052	0,8897	0,8942	0,8941	0,8986	0,8964	0,8963	0,9046	0,9106	0,9129
mammographic	0,3805	0,9257	0,9316	0,9288	0,9265	0,9302	0,9342	0,9314	0,9334	0,9371	0,9360
monk-2	0,5703	0,5134	0,5509	0,5702	0,6536	0,8543	0,8695	0,8951	0,8948	0,8966	0,8927
movement_libras	0,5046	0,8009	0,8065	0,8130	0,8130	0,8145	0,8145	0,8210	0,8315	0,8315	0,8315
newthyroid	0,0579	0,9318	0,9318	0,9318	0,9318	0,9318	0,9468	0,9705	0,9705	0,9705	0,9705
pima	0,5063	0,8896	0,8942	0,8942	0,8942	0,8983	0,9018	0,9028	0,9028	0,9028	0,9028
salheart	0,4958	0,8682	0,8740	0,8824	0,8860	0,8951	0,8990	0,9000	0,9000	0,9000	0,9000
segment	0,0409	0,9502	0,9502	0,9502	0,9508	0,9535	0,9544	0,9549	0,9549	0,9549	0,9549
sonar	0,6037	0,7927	0,7954	0,7954	0,7954	0,8050	0,8242	0,8301	0,8328	0,8376	0,8381
spectheart	0,8663	0,9313	0,9313	0,9322	0,9322	0,9322	0,9322	0,9322	0,9322	0,9322	0,9322
splice	0,0521	0,0521	0,9773	0,9772	0,9770	0,9772	0,9775	0,9774	0,9772	0,9772	0,9799
tic-tac-toe	0,2097	0,9289	0,9055	0,9151	0,9098	0,9103	0,9268	0,9369	0,9349	0,9367	0,9367
vehicle	0,5750	0,8706	0,8706	0,8706	0,8706	0,8706	0,8743	0,8760	0,8760	0,8760	0,8760
vowel	0,0294	0,7864	0,7864	0,7893	0,7960	0,8052	0,8120	0,8148	0,8148	0,8148	0,8148
wdbc	0,4900	0,9559	0,9643	0,9647	0,9647	0,9652	0,9658	0,9662	0,9662	0,9662	0,9662
wine	0,2988	0,9207	0,9207	0,9207	0,9207	0,9207	0,9214	0,9214	0,9214	0,9214	0,9214
winequality-red	0,5097	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924
wisconsin	0,3148	0,9692	0,9709	0,9709	0,9728	0,9758	0,9777	0,9809	0,9809	0,9806	0,9809
yeast	0,8584	0,9026	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049	0,9049
zoo	0,1480	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198
Promedio	0,4723	0,8045	0,8615	0,8890	0,9056	0,9130	0,9180	0,9207	0,9220	0,9229	0,9229

Tabla B.7: Precisiones logradas (con KNN) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8600	0,8691	0,8691	0,8691	0,8691	0,8691	0,8691	0,8691	0,8691	0,8691	0,5464
automobile	0,6160	0,6160	0,6213	0,5934	0,5934	0,5934	0,5930	0,5644	0,5111	0,4805	0,4805
balance	0,8463	0,8449	0,8433	0,8415	0,8625	0,8543	0,8351	0,8465	0,8401	0,8497	0,7617
bands	0,7033	0,7088	0,7070	0,7051	0,6996	0,7033	0,6940	0,7033	0,7051	0,7014	0,6363
breast	0,7140	0,6350	0,6478	0,6470	0,6545	0,6595	0,5977	0,5936	0,6783	0,6463	0,6603
car	0,8085	0,7863	0,5955	0,5508	0,5167	0,5185	0,5179	0,5103	0,5341	0,4930	0,5028
cleveland	0,5677	0,5678	0,5678	0,5712	0,5645	0,5643	0,5547	0,5481	0,5449	0,5416	0,5216
contraceptive	0,4596	0,4474	0,4542	0,4494	0,4548	0,4575	0,4507	0,4541	0,4555	0,4480	0,4555
dermatology	0,9564	0,9264	0,8364	0,8310	0,8310	0,8338	0,8310	0,8310	0,8310	0,8310	0,8173
ecoli	0,7921	0,7949	0,7981	0,7951	0,8069	0,7951	0,8099	0,8039	0,7950	0,8010	0,7176
flare	0,6070	0,6023	0,6023	0,5957	0,3246	0,3490	0,3256	0,3218	0,3265	0,3312	0,3125
glass	0,7247	0,6973	0,6800	0,6755	0,6815	0,6908	0,6865	0,6753	0,6552	0,6184	0,5197
haberman	0,7057	0,7124	0,6929	0,6930	0,6897	0,6929	0,6833	0,6832	0,6799	0,6800	0,6306
heart	0,7963	0,7926	0,7778	0,7815	0,7556	0,7556	0,7519	0,7519	0,7556	0,7519	0,7481
hepatitis	0,8329	0,8058	0,8125	0,8063	0,8063	0,8063	0,7988	0,7988	0,7988	0,8113	0,7796
horse-colic	0,8261	0,7824	0,7635	0,7635	0,7635	0,7635	0,7635	0,7525	0,7303	0,7303	0,7303
housevotes	0,9076	0,8243	0,7373	0,7395	0,7349	0,7395	0,7395	0,7372	0,7349	0,7373	0,7512
ionosphere	0,8517	0,8460	0,8460	0,8460	0,8460	0,8460	0,8433	0,8433	0,8433	0,8433	0,8433
iris	0,9533	0,9667	0,9667	0,9667	0,9667	0,9667	0,9600	0,9667	0,9600	0,9667	0,8933
led7digit	0,4620	0,4480	0,4120	0,4380	0,4240	0,4020	0,4720	0,4280	0,4740	0,4480	0,3900
lymphography	0,7261	0,7410	0,6630	0,6242	0,6977	0,6504	0,6514	0,6509	0,6696	0,6452	0,6242
mammographic	0,7804	0,7733	0,7753	0,7607	0,7390	0,7463	0,7598	0,7566	0,7505	0,7494	0,5465
monk-2	0,9081	0,8875	0,9103	0,8030	0,7301	0,6628	0,6469	0,6230	0,6280	0,6139	0,6371
movement_libras	0,7056	0,6972	0,6972	0,6889	0,6833	0,6833	0,6583	0,6389	0,6361	0,6389	0,6194
newthyroid	0,9494	0,9123	0,9123	0,9123	0,9123	0,9123	0,9123	0,8885	0,8610	0,8515	0,8095
pima	0,7475	0,7318	0,7318	0,7423	0,7383	0,7383	0,7384	0,7397	0,7344	0,7370	0,7057
saheart	0,6645	0,6602	0,6601	0,6580	0,6602	0,6602	0,6623	0,6667	0,6624	0,6646	0,5914
segment	0,9519	0,9433	0,9450	0,9463	0,9433	0,9468	0,9429	0,9420	0,9459	0,9437	0,8775
sonar	0,8162	0,7871	0,7919	0,7967	0,7871	0,7879	0,7974	0,7926	0,7971	0,7781	0,6726
spectfheart	0,7644	0,7644	0,7681	0,7756	0,7682	0,7717	0,7755	0,7678	0,7681	0,7718	0,6261
splice	0,7533	0,7536	0,5721	0,5288	0,5295	0,5282	0,5317	0,5301	0,5185	0,5069	0,5075
tic-tac-toe	0,7911	0,7776	0,6316	0,6274	0,6211	0,6221	0,6107	0,6117	0,6117	0,6191	0,5909
vehicle	0,6739	0,6702	0,6715	0,6703	0,6762	0,6691	0,6715	0,6786	0,6738	0,6750	0,6632
vowel	0,9697	0,9434	0,9434	0,9364	0,9141	0,9162	0,9172	0,9172	0,9182	0,9162	0,8919
wdbc	0,9683	0,9647	0,9665	0,9630	0,9630	0,9612	0,9647	0,9665	0,9683	0,9665	0,8383
wine	0,9605	0,9379	0,9386	0,9497	0,9608	0,9552	0,9497	0,9497	0,9552	0,9552	0,8869
winequality-red	0,5797	0,5597	0,5804	0,5816	0,5797	0,5747	0,5822	0,5779	0,5804	0,5791	0,5297
wisconsin	0,9557	0,9356	0,9370	0,9385	0,9313	0,9385	0,9428	0,9356	0,9413	0,9356	0,9413
yeast	0,5539	0,5526	0,5586	0,5559	0,5546	0,5640	0,5573	0,5633	0,5499	0,5606	0,5128
zoo	0,8947	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678
Promedio	0,7776	0,7634	0,7438	0,7372	0,7276	0,7254	0,7230	0,7187	0,7190	0,7139	0,6660

Tabla B.8: Precisiones logradas (con KNN) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8973	0,8691	0,8691	0,8691	0,8691	0,8791	0,8691	0,8691	0,8691	0,8691	0,5464
automobile	0,6121	0,6121	0,6121	0,6121	0,6121	0,6226	0,6068	0,6030	0,5311	0,4905	0,4805
balance	0,8768	0,8559	0,8560	0,8559	0,8401	0,8592	0,8561	0,8433	0,8431	0,8528	0,7426
bands	0,7015	0,7051	0,7033	0,6996	0,7014	0,7015	0,7033	0,7014	0,6996	0,7052	0,6363
breast	0,6763	0,6717	0,6395	0,6225	0,6102	0,6557	0,6523	0,6744	0,6609	0,6714	0,6651
car	0,8883	0,8935	0,8929	0,5156	0,4982	0,5382	0,4774	0,5533	0,5260	0,5144	0,4983
cleveland	0,5680	0,5713	0,5813	0,5777	0,5648	0,5748	0,5580	0,5348	0,5347	0,5215	0,5216
contraceptive	0,4460	0,4555	0,4487	0,4514	0,4596	0,4637	0,4596	0,4582	0,4582	0,4623	0,4555
dermatology	0,9564	0,9264	0,9264	0,9264	0,9210	0,9182	0,8580	0,8282	0,8310	0,8338	0,8173
ecoli	0,8395	0,8246	0,8186	0,8247	0,8187	0,8158	0,8217	0,8157	0,8039	0,8009	0,7176
flare	0,6022	0,5976	0,5995	0,5882	0,3236	0,3406	0,3247	0,3199	0,3161	0,3397	0,3143
glass	0,7093	0,7139	0,6995	0,6859	0,6859	0,6774	0,6666	0,6629	0,6592	0,6404	0,5197
haberman	0,6962	0,6928	0,6895	0,6765	0,6865	0,6960	0,6929	0,6932	0,6670	0,6831	0,6274
heart	0,8148	0,8074	0,8074	0,8074	0,8000	0,7667	0,7519	0,7481	0,7481	0,7519	0,7481
hepatitis	0,8775	0,8646	0,8513	0,8513	0,8517	0,8642	0,8254	0,8379	0,7983	0,7921	0,7796
horse-colic	0,8152	0,7610	0,7498	0,7498	0,7417	0,7417	0,7417	0,7417	0,7525	0,7525	0,7303
housevotes	0,9239	0,9031	0,7651	0,7395	0,7395	0,7395	0,7418	0,7418	0,7418	0,7418	0,7396
ionosphere	0,8717	0,8716	0,8660	0,8774	0,8745	0,8519	0,8433	0,8433	0,8433	0,8433	0,8433
iris	0,9600	0,9600	0,9467	0,9533	0,9600	0,9533	0,9467	0,9600	0,9600	0,9600	0,8933
led7digit	0,4440	0,4120	0,4620	0,4740	0,4400	0,4660	0,4700	0,4620	0,4500	0,4500	0,3960
lymphography	0,7461	0,7328	0,6792	0,6503	0,6714	0,6731	0,6580	0,6331	0,6785	0,6660	0,6634
mammographic	0,7971	0,7930	0,7711	0,7784	0,7794	0,7472	0,7567	0,7669	0,7524	0,7390	0,5661
monk-2	0,8801	0,9034	0,8801	0,8756	0,8369	0,6758	0,6428	0,6349	0,6329	0,6283	0,6346
movement_libras	0,7194	0,7083	0,7028	0,7083	0,7000	0,6972	0,7000	0,6917	0,6694	0,6694	0,6194
newthyroid	0,9494	0,9303	0,9303	0,9303	0,9303	0,9303	0,9165	0,8701	0,8563	0,8656	0,8095
pima	0,7422	0,7409	0,7357	0,7370	0,7318	0,7370	0,7435	0,7410	0,7435	0,7422	0,7057
saheart	0,6907	0,6800	0,6798	0,6777	0,6777	0,6646	0,6624	0,6581	0,6603	0,6559	0,5914
segment	0,9550	0,9463	0,9463	0,9446	0,9481	0,9450	0,9442	0,9446	0,9433	0,9442	0,8775
sonar	0,8357	0,8212	0,8307	0,8307	0,8307	0,8402	0,8114	0,8019	0,7924	0,7876	0,6726
spectheart	0,7792	0,7755	0,7868	0,7644	0,7644	0,7681	0,7756	0,7719	0,7868	0,7681	0,6261
splice	0,7533	0,7539	0,5313	0,5317	0,5282	0,5279	0,5273	0,5288	0,5273	0,5285	0,5072
tic-tac-toe	0,7922	0,6013	0,6327	0,6180	0,6557	0,6169	0,6211	0,6003	0,6211	0,6002	0,5919
vehicle	0,6974	0,6844	0,6915	0,6915	0,6856	0,6927	0,6821	0,6798	0,6797	0,6869	0,6632
vowel	0,9697	0,9495	0,9495	0,9465	0,9485	0,9354	0,9242	0,9141	0,9192	0,9162	0,8919
wdbc	0,9683	0,9666	0,9683	0,9683	0,9683	0,9701	0,9718	0,9647	0,9665	0,9682	0,8383
wine	0,9608	0,9271	0,9386	0,9330	0,9441	0,9330	0,9386	0,9386	0,9330	0,9275	0,8869
winequality-red	0,5803	0,5816	0,5829	0,5835	0,5785	0,5772	0,5785	0,5760	0,5879	0,5841	0,5297
wisconsin	0,9600	0,9514	0,9500	0,9485	0,9485	0,9471	0,9442	0,9413	0,9428	0,9399	0,9385
yeast	0,5586	0,5573	0,5586	0,5593	0,5553	0,5519	0,5600	0,5633	0,5593	0,5580	0,5128
zoo	0,8947	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678	0,8678
Promedio	0,7852	0,7710	0,7600	0,7476	0,7387	0,7356	0,7273	0,7245	0,7204	0,7180	0,6667

Tabla B.9: Reducciones logradas (para LDA) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,1593	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780
automobile	0,6007	0,7211	0,7239	0,8341	0,8352	0,8358	0,8336	0,8374	0,8460	0,8438	0,8580
balance	0,9513	0,9644	0,9616	0,9668	0,9664	0,9662	0,9634	0,9648	0,9671	0,9655	0,9673
bands	0,3521	0,7359	0,7813	0,7813	0,7850	0,7850	0,7887	0,7922	0,7922	0,7922	0,7922
breast	0,5140	0,9584	0,9557	0,9608	0,9627	0,9619	0,9546	0,9612	0,9650	0,9635	0,9627
car	0,4144	0,9673	0,9688	0,9700	0,9715	0,9680	0,9686	0,9703	0,9673	0,9688	0,9828
cleveland	0,6411	0,9204	0,9241	0,9289	0,9289	0,9289	0,9332	0,9387	0,9387	0,9410	0,9410
contraceptive	0,8853	0,8959	0,9004	0,9044	0,9044	0,9047	0,9056	0,9046	0,9056	0,9058	0,9048
dermatology	0,0310	0,0310	0,0310	0,0310	0,1234	0,9520	0,9520	0,9520	0,9520	0,9520	0,9520
ecoli	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9160	0,9167	0,9190
flare	0,5611	0,6101	0,6144	0,5621	0,6145	0,9759	0,9788	0,9812	0,9827	0,9827	0,9829
glass	0,5248	0,7494	0,8635	0,8635	0,8635	0,8728	0,8728	0,8775	0,8775	0,8785	0,8806
haberman	0,9484	0,9481	0,9484	0,9484	0,9477	0,9481	0,9473	0,9484	0,9484	0,9477	0,9473
heart	0,6156	0,8885	0,8885	0,8885	0,8885	0,8885	0,9000	0,9000	0,9329	0,9531	0,9560
hepatitis	0,3203	0,3217	0,3217	0,3210	0,3217	0,8419	0,9147	0,9147	0,9147	0,9147	0,9147
horse-colic	0,5773	0,8644	0,8726	0,8726	0,8825	0,8898	0,8925	0,8946	0,9000	0,9000	0,9006
housevotes	0,0452	0,1381	0,1388	0,1388	0,9765	0,9770	0,9775	0,9773	0,9775	0,9768	0,9783
ionosphere	0,2880	0,8015	0,8740	0,8740	0,8740	0,8740	0,8740	0,8740	0,8791	0,9034	0,9047
iris	0,3170	0,9304	0,9304	0,9304	0,9304	0,9304	0,9304	0,9304	0,9304	0,9304	0,9533
led7digit	0,8018	0,8173	0,8104	0,8162	0,8542	0,9020	0,8900	0,9487	0,9640	0,9804	0,9836
lymphography	0,5280	0,3834	0,7492	0,9008	0,9009	0,9009	0,9024	0,9069	0,9047	0,9106	0,9129
mammographic	0,3110	0,9259	0,9281	0,9303	0,9323	0,9346	0,9372	0,9375	0,9363	0,9351	0,9378
monk-2	0,0507	0,5620	0,8223	0,7940	0,8365	0,8976	0,8927	0,8912	0,8909	0,8948	0,8953
movement_libras	0,5627	0,7997	0,8059	0,8059	0,8139	0,8154	0,8275	0,8315	0,8315	0,8315	0,8315
newthyroid	0,0579	0,9318	0,9318	0,9318	0,9318	0,9318	0,9318	0,9318	0,9318	0,9318	0,9705
pima	0,5140	0,6432	0,8997	0,9012	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028
saheart	0,8574	0,8595	0,8648	0,8793	0,8911	0,8959	0,8978	0,8978	0,9000	0,9000	0,9000
segment	0,9259	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549
sonar	0,1699	0,2330	0,2330	0,7009	0,8355	0,8355	0,8355	0,8355	0,8355	0,8381	0,8381
spectfheart	0,2863	0,2863	0,2863	0,2863	0,2863	0,9322	0,9322	0,9322	0,9322	0,9322	0,9322
splice	0,0519	0,0519	0,9567	0,9565	0,9566	0,9565	0,9565	0,9566	0,9566	0,9583	0,9799
tic-tac-toe	0,2130	0,3671	0,8985	0,8982	0,9016	0,9095	0,9210	0,9231	0,9347	0,9359	0,9364
vehicle	0,7881	0,8505	0,8505	0,8652	0,8705	0,8705	0,8727	0,8760	0,8760	0,8760	0,8760
vowel	0,6561	0,8136	0,8136	0,8140	0,8144	0,8146	0,8146	0,8148	0,8148	0,8148	0,8148
wdbc	0,2980	0,9229	0,9229	0,9229	0,9229	0,9229	0,9229	0,9229	0,9229	0,9229	0,9662
wine	0,2837	0,6989	0,7814	0,7814	0,8689	0,8689	0,8689	0,8689	0,8702	0,8770	0,9214
winequality-red	0,8737	0,8764	0,8780	0,8837	0,8837	0,8856	0,8893	0,8893	0,8907	0,8924	0,8924
wisconsin	0,9639	0,9644	0,9646	0,9652	0,9660	0,9650	0,9649	0,9655	0,9685	0,9806	0,9808
yeast	0,6635	0,6635	0,6635	0,6635	0,8822	0,8839	0,8881	0,8964	0,8983	0,9049	0,9049
zoo	0,0603	0,8311	0,9187	0,9187	0,9187	0,9187	0,9198	0,9198	0,9198	0,9198	0,9198
Promedio	0,4895	0,7194	0,7882	0,8060	0,8449	0,9074	0,9101	0,9129	0,9152	0,9177	0,9232

Tabla B.10: Reducciones logradas (para LDA) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,1593	0,6576	0,9759	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780	0,9780
automobile	0,6734	0,8564	0,8564	0,8574	0,8574	0,8580	0,8580	0,8580	0,8580	0,8580	0,8580
balance	0,8646	0,9380	0,9588	0,9605	0,9632	0,9596	0,9641	0,9644	0,9641	0,9646	0,9662
bands	0,3061	0,5387	0,7755	0,7881	0,7881	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922
breast	0,7160	0,7114	0,9576	0,9631	0,9616	0,9573	0,9561	0,9619	0,9635	0,9600	0,9650
car	0,0457	0,9708	0,9716	0,9700	0,9677	0,9706	0,9698	0,9697	0,9726	0,9707	0,9822
cleveland	0,5395	0,7182	0,9065	0,9080	0,9149	0,9168	0,9351	0,9399	0,9399	0,9410	0,9410
contraceptive	0,8357	0,8700	0,8893	0,9044	0,9054	0,9052	0,9064	0,9062	0,9059	0,9054	0,9061
dermatology	0,0310	0,0310	0,0310	0,0310	0,1228	0,9502	0,9514	0,9514	0,9520	0,9520	0,9520
ecoli	0,9173	0,9173	0,9173	0,9173	0,9173	0,9173	0,9180	0,9187	0,9187	0,9190	0,9190
flare	0,4598	0,4557	0,4583	0,4614	0,4611	0,9824	0,9828	0,9833	0,9822	0,9821	0,9829
glass	0,5868	0,6451	0,8733	0,8733	0,8733	0,8733	0,8744	0,8744	0,8744	0,8796	0,8806
haberman	0,5615	0,8830	0,9459	0,9466	0,9462	0,9477	0,9459	0,9463	0,9470	0,9477	0,9473
heart	0,4222	0,8222	0,8897	0,8897	0,8897	0,8955	0,9082	0,9082	0,9428	0,9560	0,9560
hepatitis	0,1757	0,1757	0,1749	0,1757	0,1757	0,9147	0,9147	0,9147	0,9147	0,9147	0,9147
horse-colic	0,2485	0,5187	0,8729	0,8729	0,8729	0,8777	0,8922	0,8991	0,9006	0,9006	0,9006
housevotes	0,0455	0,0444	0,0450	0,0455	0,9791	0,9788	0,9796	0,9796	0,9793	0,9803	0,9788
ionosphere	0,1421	0,8721	0,8721	0,8721	0,8721	0,8721	0,8721	0,8740	0,8822	0,9034	0,9047
iris	0,0474	0,9304	0,9304	0,9304	0,9304	0,9304	0,9304	0,9304	0,9304	0,9304	0,9533
led7digit	0,7847	0,8033	0,7976	0,7856	0,8360	0,8571	0,8862	0,9338	0,9658	0,9773	0,9809
lymphography	0,3772	0,6737	0,6028	0,8278	0,7562	0,8400	0,9136	0,9151	0,9144	0,9106	0,9144
mammographic	0,4502	0,9266	0,9272	0,9251	0,9260	0,9283	0,9266	0,9361	0,9394	0,9366	0,9370
monk-2	0,3206	0,6529	0,7922	0,8742	0,8655	0,8761	0,8987	0,8933	0,8932	0,8966	0,8950
movement_libras	0,7654	0,8315	0,8315	0,8315	0,8315	0,8315	0,8315	0,8315	0,8315	0,8315	0,8315
newthyroid	0,1434	0,9318	0,9318	0,9318	0,9318	0,9318	0,9318	0,9318	0,9318	0,9318	0,9705
pima	0,6882	0,8769	0,8867	0,8867	0,8867	0,8909	0,8994	0,9028	0,9028	0,9028	0,9028
salheart	0,4968	0,8735	0,8766	0,8855	0,8872	0,8963	0,8978	0,8988	0,9000	0,9000	0,9000
segment	0,9466	0,9529	0,9540	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549	0,9549
sonar	0,1699	0,1699	0,1699	0,6403	0,8355	0,8376	0,8376	0,8376	0,8376	0,8381	0,8381
spectheart	0,2863	0,2863	0,2863	0,2863	0,4167	0,9301	0,9322	0,9322	0,9322	0,9322	0,9322
splice	0,0519	0,0519	0,9566	0,9565	0,9565	0,9567	0,9566	0,9565	0,9566	0,9565	0,9797
tic-tac-toe	0,1335	0,5958	0,8968	0,9035	0,9032	0,9037	0,9207	0,9366	0,9361	0,9361	0,9362
vehicle	0,5653	0,8526	0,8654	0,8672	0,8672	0,8693	0,8727	0,8743	0,8760	0,8760	0,8760
vowel	0,6568	0,8137	0,8137	0,8137	0,8137	0,8140	0,8140	0,8145	0,8147	0,8148	0,8148
wdbc	0,2981	0,8334	0,9229	0,9229	0,9229	0,9229	0,9229	0,9229	0,9229	0,9229	0,9662
wine	0,0337	0,5332	0,7814	0,8689	0,8689	0,8689	0,8689	0,8689	0,8689	0,8702	0,9214
winequality-red	0,7960	0,8891	0,8908	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924	0,8924
wisconsin	0,8735	0,9682	0,9657	0,9650	0,9676	0,9676	0,9682	0,9708	0,9701	0,9736	0,9808
yeast	0,5792	0,6681	0,7144	0,8003	0,8866	0,8866	0,8866	0,8984	0,9027	0,9027	0,9027
zoo	0,7481	0,9187	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198
Promedio	0,4486	0,6915	0,7772	0,8021	0,8376	0,9064	0,9116	0,9143	0,9166	0,9178	0,9232

Tabla B.11: Precisiones logradas (con LDA) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8691	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018
automobile	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146
balance	0,8736	0,8752	0,8672	0,8704	0,8689	0,8720	0,8816	0,8688	0,8688	0,8655	0,8415
bands	0,6663	0,6625	0,6606	0,6606	0,6477	0,6440	0,6440	0,6440	0,6421	0,6421	0,6383
breast	0,6957	0,6687	0,6602	0,6433	0,6785	0,6747	0,6431	0,6330	0,6719	0,6757	0,6959
car	0,7899	0,7326	0,7315	0,7307	0,6984	0,7280	0,7599	0,7420	0,7622	0,7181	0,6007
cleveland	0,5516	0,5452	0,5353	0,5352	0,5319	0,5319	0,5252	0,4923	0,4956	0,4956	0,4986
contraceptive	0,5030	0,5071	0,5031	0,5044	0,5037	0,5030	0,5037	0,4963	0,4963	0,5064	0,5159
dermatology	0,9727	0,9727	0,9727	0,9727	0,9133	0,3060	0,3060	0,3060	0,3060	0,3060	0,3060
ecoli	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256
flare	0,5224	0,4972	0,5035	0,5306	0,4400	0,3105	0,3105	0,3105	0,3105	0,3105	0,3105
glass	0,5847	0,5129	0,5276	0,5181	0,5319	0,5197	0,5391	0,5431	0,5612	0,5387	0,4881
haberman	0,7384	0,7319	0,7319	0,7417	0,7418	0,7418	0,7452	0,7352	0,7352	0,7385	0,7418
heart	0,8000	0,8000	0,8000	0,8000	0,8000	0,8000	0,7704	0,7704	0,6556	0,5815	0,5667
hepatitis	0,7200	0,7200	0,7200	0,7200	0,7200	0,2796	0,2063	0,2063	0,2063	0,2063	0,2063
horse-colic	0,8017	0,7199	0,7307	0,7307	0,7064	0,7226	0,6902	0,7064	0,6953	0,6953	0,7034
housevotes	0,9445	0,8957	0,8957	0,8957	0,3862	0,3862	0,3862	0,3862	0,3862	0,3862	0,3862
ionosphere	0,8235	0,7667	0,7810	0,7667	0,7808	0,7810	0,7724	0,7865	0,7893	0,6525	0,6410
iris	0,9800	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467
led7digit	0,3340	0,3760	0,4180	0,4340	0,2960	0,2640	0,3120	0,3580	0,2520	0,1620	0,0900
lymphography	0,2500	0,3095	0,0990	0,0133	0,0133	0,0133	0,0133	0,0133	0,0133	0,0133	0,0133
mammographic	0,8242	0,7950	0,7919	0,7992	0,7961	0,7961	0,7909	0,7940	0,7971	0,7982	0,7992
monk-2	0,8038	0,7852	0,7384	0,7338	0,7397	0,7271	0,7305	0,7350	0,7168	0,7305	0,7373
movement_libras	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667
newthyroid	0,9162	0,8848	0,8848	0,8848	0,8848	0,8848	0,8848	0,8848	0,8848	0,8848	0,1396
pima	0,7658	0,7632	0,7632	0,7632	0,7619	0,7645	0,7632	0,7632	0,7632	0,7658	0,7619
saheart	0,7121	0,7121	0,7077	0,6949	0,6927	0,6863	0,6948	0,6949	0,6926	0,6970	0,6927
segment	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429
sonar	0,7493	0,7064	0,7064	0,5112	0,4662	0,4662	0,4662	0,4662	0,4662	0,4662	0,4662
spectfheart	0,7380	0,7380	0,7380	0,7380	0,7380	0,2058	0,2058	0,2058	0,2058	0,2058	0,2058
splice	0,9160	0,9147	0,7094	0,7119	0,7066	0,7150	0,7129	0,7085	0,7075	0,6793	0,2592
tic-tac-toe	0,6962	0,6649	0,5940	0,5720	0,5666	0,5700	0,5438	0,5555	0,5259	0,5333	0,5197
vehicle	0,7589	0,7565	0,7577	0,7505	0,7400	0,7400	0,7400	0,7435	0,7376	0,7435	0,7364
vowel	0,6303	0,6303	0,6232	0,6253	0,6242	0,6273	0,6283	0,6263	0,6263	0,6242	0,6222
wdbc	0,9472	0,9069	0,9069	0,9069	0,9069	0,9069	0,9069	0,9069	0,9069	0,9069	0,3726
wine	0,9719	0,9490	0,9435	0,9435	0,8879	0,8879	0,8879	0,8879	0,8768	0,8157	0,3317
winequality-red	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
wisconsin	0,9585	0,9571	0,9599	0,9385	0,9356	0,9342	0,9356	0,9357	0,8971	0,8295	0,7938
yeast	0,2459	0,2459	0,2459	0,2459	0,2040	0,2040	0,2040	0,2040	0,2040	0,1644	0,1644
zoo	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189
Promedio	0,6531	0,6380	0,6257	0,6176	0,5932	0,5503	0,5480	0,5482	0,5418	0,5289	0,4561

Tabla B.12: Precisiones logradas (con LDA) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8691	0,8591	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018
automobile	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146
balance	0,8816	0,8767	0,8672	0,8704	0,8719	0,8816	0,8768	0,8767	0,8688	0,8608	0,8496
bands	0,6700	0,6755	0,6494	0,6439	0,6439	0,6402	0,6383	0,6383	0,6402	0,6402	0,6383
breast	0,6857	0,7242	0,6904	0,6754	0,6747	0,6688	0,6110	0,6965	0,6604	0,6333	0,6971
car	0,8119	0,7130	0,6504	0,7136	0,7396	0,6899	0,7656	0,7135	0,7303	0,7158	0,5741
cleveland	0,5647	0,5747	0,5553	0,5619	0,5586	0,5552	0,5153	0,5119	0,5153	0,5053	0,4986
contraceptive	0,4881	0,5030	0,5139	0,5091	0,5031	0,5078	0,5159	0,5031	0,4977	0,4990	0,4970
dermatology	0,9727	0,9727	0,9727	0,9727	0,9133	0,3060	0,3060	0,3060	0,3060	0,3060	0,3060
ecoli	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256
flare	0,5882	0,5629	0,5900	0,5872	0,5863	0,3105	0,3105	0,3105	0,3105	0,3105	0,3105
glass	0,6310	0,6360	0,5836	0,5931	0,5883	0,5888	0,5879	0,5780	0,5923	0,5786	0,4881
haberman	0,7710	0,7482	0,7452	0,7385	0,7417	0,7449	0,7452	0,7451	0,7483	0,7452	0,7418
heart	0,8222	0,8074	0,8037	0,8037	0,8037	0,8000	0,7741	0,7444	0,6556	0,5815	0,5667
hepatitis	0,8325	0,8325	0,8325	0,8325	0,8325	0,2063	0,2063	0,2063	0,2063	0,2063	0,2063
horse-colic	0,8423	0,8395	0,7821	0,7821	0,7821	0,7604	0,7361	0,7115	0,7034	0,7034	0,7034
housevotes	0,9445	0,9445	0,9445	0,9445	0,3862	0,3862	0,3862	0,3862	0,3862	0,3862	0,3862
ionosphere	0,8378	0,7752	0,7894	0,7867	0,7838	0,7979	0,7838	0,7921	0,7467	0,6582	0,6410
iris	0,9800	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467	0,9467	0,5267
led7digit	0,3580	0,3820	0,4500	0,4400	0,3740	0,4400	0,4760	0,3220	0,2880	0,1340	0,0900
lymphography	0,4029	0,1710	0,2567	0,0990	0,1562	0,1067	0,0133	0,0133	0,0133	0,0133	0,0133
mammographic	0,8096	0,8086	0,8054	0,8034	0,7981	0,8085	0,8044	0,7930	0,7940	0,7982	0,7877
monk-2	0,8015	0,7717	0,7763	0,7191	0,7442	0,7177	0,7167	0,7304	0,7181	0,7183	0,7324
movement_libras	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667
newthyroid	0,9162	0,8848	0,8848	0,8848	0,8848	0,8848	0,8848	0,8848	0,8848	0,8848	0,1396
pima	0,7787	0,7748	0,7683	0,7683	0,7683	0,7644	0,7579	0,7567	0,7567	0,7567	0,7619
saheart	0,7121	0,7078	0,7099	0,7078	0,7056	0,6969	0,6926	0,6992	0,6948	0,6992	0,6927
segment	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429
sonar	0,7493	0,7493	0,7493	0,5710	0,4662	0,4662	0,4662	0,4662	0,4662	0,4662	0,4662
spectheart	0,7380	0,7380	0,7380	0,7380	0,6232	0,2058	0,2058	0,2058	0,2058	0,2058	0,2058
splice	0,9150	0,9157	0,7060	0,7094	0,7141	0,7097	0,7082	0,7078	0,7078	0,7144	0,2624
tic-tac-toe	0,7035	0,6628	0,5908	0,6075	0,5834	0,6075	0,6054	0,5052	0,5250	0,5208	0,5167
vehicle	0,7683	0,7553	0,7529	0,7411	0,7435	0,7447	0,7495	0,7482	0,7412	0,7447	0,7364
vowel	0,6242	0,6253	0,6202	0,6232	0,6253	0,6253	0,6253	0,6253	0,6293	0,6273	0,6222
wdbc	0,9560	0,9297	0,9069	0,9069	0,9069	0,9069	0,9069	0,9069	0,9069	0,9069	0,3726
wine	0,9775	0,9546	0,9435	0,8879	0,8879	0,8879	0,8879	0,8879	0,8879	0,8768	0,3317
winequality-red	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
wisconsin	0,9614	0,9571	0,9642	0,9599	0,9614	0,9571	0,9585	0,9599	0,9585	0,9285	0,7938
yeast	0,2472	0,2472	0,2472	0,2472	0,2054	0,2054	0,2054	0,2054	0,2054	0,2054	0,2054
zoo	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189
Promedio	0,6670	0,6524	0,6414	0,6312	0,6094	0,5599	0,5560	0,5489	0,5442	0,5337	0,4558

Tabla B.13: Reducciones logradas (para PART) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,3287	0,7395	0,8988	0,9738	0,9738	0,9738	0,9738	0,9780	0,9780	0,9780	0,9780
automobile	0,8379	0,8379	0,8374	0,8384	0,8384	0,8417	0,8412	0,8520	0,8569	0,8580	0,8580
balance	0,9650	0,9652	0,9643	0,9620	0,9662	0,9643	0,9644	0,9648	0,9669	0,9650	0,9671
bands	0,5982	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922
breast	0,8297	0,9549	0,9592	0,9639	0,9627	0,9518	0,9565	0,9604	0,9565	0,9670	0,9631
car	0,0455	0,4196	0,9833	0,9832	0,9822	0,9817	0,9814	0,9827	0,9826	0,9819	0,9833
cleveland	0,5442	0,9219	0,9219	0,9219	0,9219	0,9292	0,9358	0,9410	0,9410	0,9410	0,9410
contraceptive	0,8540	0,8537	0,8528	0,8531	0,8534	0,8701	0,8917	0,9061	0,9039	0,9062	0,9061
dermatology	0,0310	0,0310	0,0310	0,0310	0,9520	0,9520	0,9520	0,9520	0,9520	0,9520	0,9520
ecoli	0,1994	0,4155	0,9187	0,9187	0,9187	0,9187	0,9190	0,9190	0,9190	0,9190	0,9190
flare	0,8223	0,9799	0,9796	0,9780	0,9784	0,9792	0,9789	0,9810	0,9822	0,9829	0,9841
glass	0,3678	0,8796	0,8796	0,8801	0,8801	0,8801	0,8801	0,8801	0,8806	0,8806	0,8806
haberman	0,8032	0,9248	0,9259	0,9321	0,9324	0,9332	0,9397	0,9459	0,9452	0,9473	0,9484
heart	0,9473	0,9473	0,9473	0,9498	0,9498	0,9498	0,9539	0,9539	0,9560	0,9560	0,9560
hepatitis	0,6163	0,8385	0,9126	0,9126	0,9126	0,9126	0,9126	0,9126	0,9126	0,9140	0,9147
horse-colic	0,7212	0,8680	0,8680	0,8680	0,8795	0,8877	0,8877	0,8877	0,9012	0,9012	0,9006
housevotes	0,3266	0,8870	0,8877	0,9783	0,9783	0,9785	0,9783	0,9783	0,9791	0,9785	0,9783
ionosphere	0,1421	0,1421	0,1421	0,8297	0,9047	0,9047	0,9047	0,9047	0,9047	0,9047	0,9047
iris	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9511	0,9533
led7digit	0,7984	0,9176	0,9856	0,9796	0,9824	0,9780	0,9798	0,9802	0,9844	0,9824	0,9827
lymphography	0,5216	0,8994	0,8979	0,9009	0,9106	0,9084	0,9114	0,9114	0,9114	0,9121	0,9136
mammographic	0,9321	0,9302	0,9308	0,9335	0,9295	0,9305	0,9302	0,9349	0,9347	0,9365	0,9380
monk-2	0,7881	0,7937	0,7860	0,7953	0,7860	0,8353	0,8541	0,8616	0,8971	0,8948	0,8956
movement_libras	0,4531	0,7574	0,8179	0,8179	0,8201	0,8238	0,8238	0,8253	0,8269	0,8315	0,8315
newthyroid	0,9318	0,9318	0,9318	0,9318	0,9318	0,9411	0,9613	0,9705	0,9705	0,9705	0,9705
pima	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028
saheart	0,8834	0,8834	0,8834	0,8853	0,8865	0,8976	0,8990	0,9000	0,9000	0,9000	0,9000
segment	0,0409	0,0409	0,0409	0,9502	0,9502	0,9502	0,9502	0,9502	0,9502	0,9526	0,9549
sonar	0,8312	0,8344	0,8344	0,8371	0,8371	0,8371	0,8381	0,8381	0,8381	0,8381	0,8381
spectfheart	0,9043	0,9097	0,9155	0,9243	0,9243	0,9243	0,9263	0,9305	0,9322	0,9322	0,9322
splice	0,9565	0,9567	0,9566	0,9566	0,9566	0,9565	0,9564	0,9566	0,9566	0,9629	0,9799
tic-tac-toe	0,1326	0,9069	0,9078	0,9217	0,9261	0,9288	0,9316	0,9327	0,9320	0,9309	0,9352
vehicle	0,8567	0,8567	0,8588	0,8654	0,8725	0,8760	0,8760	0,8760	0,8760	0,8760	0,8760
vowel	0,7828	0,7828	0,7828	0,7828	0,7936	0,8148	0,8148	0,8148	0,8148	0,8148	0,8148
wdbc	0,2182	0,2182	0,6856	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662	0,9662
wine	0,8689	0,8689	0,8689	0,8689	0,8689	0,8689	0,8689	0,8689	0,9039	0,9207	0,9214
winequality-red	0,5025	0,8860	0,8874	0,8874	0,8874	0,8874	0,8874	0,8924	0,8924	0,8924	0,8924
wisconsin	0,1272	0,9800	0,9801	0,9806	0,9800	0,9804	0,9800	0,9803	0,9800	0,9804	0,9808
yeast	0,8848	0,8848	0,8848	0,8872	0,8894	0,8917	0,8917	0,8958	0,8978	0,9049	0,9049
zoo	0,0603	0,0603	0,0603	0,0603	0,9186	0,9198	0,9198	0,9198	0,9198	0,9198	0,9198
Promedio	0,6071	0,7631	0,8107	0,8632	0,9105	0,9136	0,9159	0,9181	0,9206	0,9225	0,9233

Tabla B.14: Reducciones logradas (para PART) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	Bases de datos \ b										
	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,2424	0,8133	0,9654	0,9717	0,9717	0,9759	0,9759	0,9759	0,9759	0,9759	0,9780
automobile	0,8384	0,8363	0,8363	0,8363	0,8363	0,8379	0,8379	0,8433	0,8547	0,8580	0,8580
balance	0,9529	0,9511	0,9531	0,9413	0,9517	0,9541	0,9584	0,9584	0,9675	0,9668	0,9641
bands	0,7891	0,7891	0,7924	0,7924	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922	0,7922
breast	0,7740	0,9588	0,9596	0,9627	0,9623	0,9611	0,9612	0,9662	0,9584	0,9639	0,9674
car	0,2305	0,5139	0,9774	0,9774	0,9772	0,9785	0,9799	0,9810	0,9819	0,9828	0,9824
cleveland	0,7382	0,9267	0,9267	0,9267	0,9267	0,9299	0,9299	0,9340	0,9399	0,9410	0,9410
contraceptive	0,9062	0,9045	0,9057	0,9050	0,9065	0,9059	0,9056	0,9052	0,9049	0,9050	0,9054
dermatology	0,0310	0,0310	0,0310	0,0310	0,8044	0,9023	0,9023	0,9223	0,9408	0,9520	0,9520
ecoli	0,2700	0,2700	0,7010	0,9183	0,9183	0,9183	0,9183	0,9183	0,9183	0,9190	0,9190
flare	0,7223	0,9783	0,9781	0,9784	0,9794	0,9761	0,9797	0,9796	0,9806	0,9812	0,9834
glass	0,3089	0,7567	0,8734	0,8785	0,8790	0,8790	0,8790	0,8806	0,8806	0,8806	0,8806
haberman	0,8153	0,9430	0,9393	0,9419	0,9437	0,9430	0,9473	0,9481	0,9484	0,9477	0,9481
heart	0,7749	0,9214	0,9214	0,9272	0,9317	0,9366	0,9453	0,9527	0,9543	0,9560	0,9560
hepatitis	0,4588	0,9061	0,9061	0,9061	0,9061	0,9061	0,9061	0,9133	0,9147	0,9147	0,9147
horse-colic	0,8810	0,8931	0,8931	0,8931	0,8931	0,8931	0,8946	0,8946	0,8946	0,9006	0,9006
housevotes	0,5140	0,7946	0,8872	0,9788	0,9788	0,9788	0,9785	0,9783	0,9788	0,9788	0,9780
ionosphere	0,1421	0,1421	0,1421	0,1421	0,9047	0,9047	0,9047	0,9047	0,9047	0,9047	0,9047
iris	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9237	0,9370	0,9533
led7digit	0,8744	0,8951	0,8998	0,9680	0,9758	0,9724	0,9740	0,9773	0,9702	0,9809	0,9802
lymphography	0,8820	0,8829	0,8874	0,8865	0,8867	0,9017	0,8962	0,9030	0,9129	0,9121	0,9121
mammographic	0,9349	0,9323	0,9336	0,9327	0,9356	0,9312	0,9324	0,9351	0,9371	0,9373	0,9361
monk-2	0,7192	0,7126	0,7955	0,7896	0,7917	0,8159	0,8051	0,8662	0,8915	0,8961	0,8958
movement_libras	0,2664	0,6969	0,7636	0,8265	0,8296	0,8315	0,8315	0,8315	0,8315	0,8315	0,8315
newthyroid	0,1481	0,7721	0,9576	0,9576	0,9576	0,9576	0,9576	0,9576	0,9613	0,9705	0,9705
pima	0,9010	0,9010	0,9010	0,9010	0,9010	0,9028	0,9028	0,9028	0,9028	0,9028	0,9028
sabheart	0,8286	0,8795	0,8834	0,8853	0,8891	0,8920	0,8949	0,8985	0,8985	0,9000	0,9000
segment	0,2231	0,2231	0,3144	0,9507	0,9507	0,9507	0,9507	0,9507	0,9514	0,9518	0,9549
sonar	0,5027	0,6974	0,7588	0,8179	0,8179	0,8179	0,8254	0,8291	0,8328	0,8381	0,8381
spectheart	0,8231	0,8877	0,8877	0,9005	0,9068	0,9130	0,9276	0,9293	0,9313	0,9322	0,9322
splice	0,0520	0,9726	0,9729	0,9748	0,9727	0,9747	0,9748	0,9773	0,9773	0,9776	0,9797
tic-tac-toe	0,1334	0,8159	0,9038	0,8975	0,9014	0,9122	0,9172	0,9146	0,9351	0,9345	0,9353
vehicle	0,6980	0,8137	0,8659	0,8659	0,8739	0,8739	0,8739	0,8739	0,8760	0,8760	0,8760
vowel	0,2615	0,8108	0,8108	0,8136	0,8136	0,8137	0,8137	0,8145	0,8148	0,8148	0,8148
wdbc	0,0307	0,2135	0,9512	0,9512	0,9512	0,9611	0,9611	0,9656	0,9656	0,9660	0,9662
wine	0,4469	0,7858	0,8689	0,8689	0,8726	0,8795	0,8795	0,8795	0,8982	0,9214	0,9214
winequality-red	0,6531	0,8478	0,8478	0,8670	0,8728	0,8854	0,8854	0,8874	0,8924	0,8924	0,8924
wisconsin	0,5965	0,9722	0,9688	0,9704	0,9690	0,9704	0,9676	0,9690	0,9752	0,9776	0,9809
yeast	0,8848	0,8848	0,8848	0,8848	0,8892	0,8958	0,8978	0,9049	0,9049	0,9049	0,9049
zoo	0,0603	0,0603	0,1487	0,4074	0,9154	0,9154	0,9154	0,9154	0,9154	0,9165	0,9198
Promedio	0,5559	0,7478	0,8180	0,8538	0,9065	0,9117	0,9126	0,9164	0,9198	0,9223	0,9231

Tabla B.15: Precisiones logradas (con PART) por la versión crisp del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8509	0,8318	0,7518	0,7418	0,7418	0,7418	0,7418	0,7418	0,7418	0,7418	0,7418
automobile	0,4147	0,4147	0,4147	0,3957	0,4147	0,4102	0,4102	0,3568	0,3564	0,3564	0,3612
balance	0,5595	0,5859	0,6000	0,5915	0,5166	0,5472	0,5836	0,5118	0,5339	0,5259	0,5585
bands	0,6087	0,6179	0,6068	0,6198	0,6050	0,6272	0,6142	0,6161	0,6179	0,6068	0,6179
breast	0,6687	0,5587	0,6307	0,6442	0,6220	0,6517	0,6064	0,6183	0,6312	0,6396	0,6756
car	0,7002	0,6398	0,3448	0,4143	0,3599	0,3546	0,3835	0,4125	0,4147	0,4341	0,4287
cleveland	0,5412	0,5412	0,5345	0,5412	0,5345	0,5278	0,5412	0,5278	0,5445	0,5278	0,4878
contraceptive	0,4270	0,4236	0,4311	0,4270	0,4277	0,4392	0,4372	0,4576	0,4447	0,4345	0,4597
dermatology	0,8247	0,8247	0,8247	0,8247	0,1667	0,1667	0,1667	0,1667	0,1667	0,1667	0,1667
ecoli	0,4912	0,4796	0,3081	0,3111	0,3553	0,3111	0,3111	0,3051	0,3111	0,3051	0,2869
flare	0,2749	0,2545	0,2636	0,2733	0,2758	0,2177	0,2704	0,2290	0,2732	0,2441	0,2787
glass	0,4341	0,3972	0,3927	0,3881	0,3970	0,3927	0,3881	0,3927	0,4018	0,3927	0,3788
haberman	0,7385	0,7218	0,7385	0,7417	0,7417	0,7252	0,7417	0,7385	0,7385	0,7353	0,7353
heart	0,5852	0,8000	0,7875	0,5889	0,5889	0,5889	0,6037	0,6037	0,6037	0,6037	0,6037
hepatitis	0,8000	0,8000	0,7875	0,7875	0,7875	0,7608	0,7875	0,7875	0,7608	0,7875	0,7875
horse-colic	0,6305	0,6305	0,6305	0,6305	0,6359	0,6629	0,6629	0,6629	0,6764	0,6764	0,6764
housevotes	0,9446	0,8727	0,9052	0,8541	0,8517	0,8238	0,8200	0,8541	0,8541	0,8541	0,7468
ionosphere	0,8862	0,8862	0,8862	0,4790	0,4275	0,4275	0,4275	0,4275	0,4275	0,4275	0,4275
iris	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,3333
led7digit	0,1960	0,1140	0,1060	0,1000	0,1060	0,1160	0,1200	0,1000	0,1060	0,0920	0,1080
lymphography	0,6677	0,5812	0,5745	0,5930	0,5822	0,5396	0,6060	0,5244	0,5582	0,6201	0,5649
mammographic	0,8138	0,8138	0,8190	0,7888	0,7963	0,7982	0,8190	0,8190	0,7846	0,7502	0,8065
monk-2	0,9517	0,9103	0,9426	0,9621	0,9517	0,8540	0,8459	0,8012	0,7560	0,7401	0,7401
movement_libras	0,1750	0,1694	0,1722	0,1667	0,1611	0,1750	0,2167	0,1694	0,1667	0,2000	0,1444
newthyroid	0,6162	0,6162	0,6162	0,6162	0,6162	0,6162	0,6171	0,5266	0,5266	0,5266	0,5403
pima	0,7201	0,7201	0,7201	0,7201	0,7201	0,7201	0,7201	0,7201	0,7201	0,7201	0,7201
saheart	0,6884	0,6950	0,6820	0,6842	0,6950	0,6710	0,6667	0,6645	0,6689	0,6602	0,6689
segment	0,6494	0,6494	0,6494	0,4498	0,4537	0,4420	0,4481	0,4030	0,3550	0,4355	0,2844
sonar	0,6252	0,6350	0,6348	0,5900	0,6050	0,6000	0,6300	0,6050	0,6095	0,6090	0,5424
spectfheart	0,7942	0,7942	0,7942	0,7571	0,7571	0,7571	0,7571	0,7793	0,7571	0,7793	0,7195
splice	0,4197	0,4357	0,4041	0,4172	0,4201	0,4157	0,3959	0,4060	0,4235	0,4335	0,3828
tic-tac-toe	0,6535	0,4906	0,5365	0,4622	0,4372	0,5274	0,4155	0,4550	0,3695	0,4436	0,5081
vehicle	0,3805	0,3805	0,3688	0,3783	0,4174	0,4328	0,4328	0,4328	0,4328	0,4328	0,4328
vowel	0,1091	0,1091	0,1091	0,1091	0,1202	0,1232	0,1232	0,1232	0,1232	0,1232	0,1232
wdbc	0,8313	0,8313	0,7344	0,6906	0,6888	0,6432	0,6450	0,6888	0,6379	0,6502	0,6292
wine	0,6520	0,6520	0,6520	0,6520	0,6520	0,6520	0,6520	0,6520	0,4853	0,4493	0,4601
winequality-red	0,5109	0,4846	0,4740	0,4740	0,4740	0,4740	0,4740	0,4596	0,4596	0,4596	0,4596
wisconsin	0,7913	0,7467	0,6452	0,6652	0,7495	0,6667	0,7495	0,6138	0,7195	0,6938	0,6252
yeast	0,3417	0,3586	0,3612	0,3296	0,3302	0,3572	0,3450	0,3566	0,3552	0,3268	0,3269
zoo	0,7514	0,7514	0,7514	0,7514	0,3139	0,2872	0,2872	0,2872	0,2872	0,1831	0,1831
Promedio	0,6088	0,5910	0,5754	0,5561	0,5283	0,5220	0,5275	0,5158	0,5109	0,5037	0,4931

Tabla B.16: Precisiones logradas (con PART) por la versión difusa del sistema a distintos valores del parámetro b .

Bases de datos \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1
appendicitis	0,8600	0,8318	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,8018	0,7418
automobile	0,3657	0,4016	0,4016	0,4016	0,4016	0,3879	0,3879	0,3991	0,3675	0,3342	0,3612
balance	0,6959	0,6565	0,6498	0,6733	0,6097	0,6559	0,6702	0,6928	0,5974	0,6545	0,5523
bands	0,6327	0,6327	0,6346	0,6346	0,6309	0,6290	0,6327	0,6309	0,6346	0,6327	0,6179
breast	0,5827	0,6679	0,6417	0,6097	0,6823	0,6073	0,5902	0,6478	0,6110	0,6364	0,6377
car	0,6025	0,4624	0,3315	0,3673	0,4989	0,2922	0,4716	0,4589	0,3941	0,3858	0,3600
cleveland	0,5412	0,5345	0,5345	0,5345	0,5345	0,5412	0,5412	0,5278	0,5278	0,5278	0,4878
contraceptive	0,4495	0,4583	0,4780	0,4657	0,4697	0,4698	0,4732	0,4365	0,4827	0,4712	0,4535
dermatology	0,8247	0,8247	0,8247	0,8247	0,4060	0,3384	0,3384	0,3107	0,2396	0,1667	0,1667
ecoli	0,4821	0,4821	0,3940	0,3803	0,3111	0,3202	0,3259	0,3379	0,3773	0,3946	0,2869
flare	0,2825	0,2383	0,2505	0,2562	0,2937	0,2775	0,2335	0,2930	0,2908	0,2729	0,2534
glass	0,4432	0,4222	0,4077	0,4023	0,3830	0,3877	0,3923	0,3970	0,3832	0,3970	0,3788
haberman	0,7252	0,7218	0,7252	0,7252	0,7252	0,7218	0,7353	0,7353	0,7353	0,7353	0,7353
heart	0,6778	0,6704	0,6704	0,6852	0,6852	0,6667	0,6556	0,6333	0,6333	0,6037	0,6037
hepatitis	0,8129	0,8133	0,8133	0,8133	0,8133	0,8133	0,8133	0,8008	0,7875	0,7875	0,7875
horse-colic	0,6737	0,6737	0,6710	0,6737	0,6710	0,6710	0,6764	0,6764	0,6764	0,6764	0,6764
housevotes	0,9446	0,9121	0,9052	0,8541	0,8541	0,8541	0,8541	0,8541	0,8820	0,8541	0,7049
ionosphere	0,8862	0,8862	0,8862	0,8862	0,4275	0,4275	0,4275	0,4275	0,4275	0,4275	0,4275
iris	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,6333	0,5267	0,3333
led7digit	0,1280	0,1300	0,1580	0,1140	0,1000	0,1200	0,1360	0,1120	0,1500	0,1180	0,1020
lymphography	0,5801	0,5264	0,5991	0,5275	0,6239	0,5350	0,5371	0,5879	0,5907	0,6014	0,5515
mammographic	0,8190	0,8190	0,8190	0,7846	0,8190	0,7888	0,8190	0,8190	0,8190	0,8190	0,8190
monk-2	0,9471	0,9153	0,9426	0,8893	0,9659	0,9281	0,9176	0,8332	0,7401	0,7401	0,7401
movement_libras	0,2139	0,1944	0,1889	0,1778	0,1889	0,1694	0,1778	0,1528	0,1722	0,1667	0,1444
newthyroid	0,7026	0,7045	0,6307	0,6307	0,6307	0,6307	0,6307	0,6307	0,6307	0,5403	0,5403
pima	0,7450	0,7398	0,7424	0,7359	0,7372	0,7162	0,7149	0,7214	0,7227	0,7292	0,7201
saheart	0,6971	0,6841	0,6951	0,6907	0,6906	0,6906	0,6689	0,6602	0,6471	0,6558	0,6689
segment	0,7022	0,7095	0,7013	0,5390	0,4229	0,4649	0,4957	0,4931	0,4732	0,4208	0,2844
sonar	0,7117	0,6688	0,6688	0,6536	0,6536	0,6488	0,6143	0,6502	0,6200	0,6090	0,5424
spectheart	0,7942	0,7942	0,7942	0,7942	0,7942	0,7942	0,7571	0,7793	0,7793	0,7571	0,7195
splice	0,5188	0,4044	0,4056	0,4141	0,4188	0,3790	0,4113	0,3561	0,4254	0,4197	0,3721
tic-tac-toe	0,6535	0,5792	0,5354	0,5289	0,5920	0,5028	0,5313	0,4862	0,4522	0,4404	0,4852
vehicle	0,4786	0,4705	0,4611	0,4611	0,4446	0,4446	0,4446	0,4446	0,4328	0,4328	0,4328
vowel	0,1525	0,1364	0,1263	0,1263	0,1212	0,1242	0,1273	0,1273	0,1263	0,1263	0,1232
wdbc	0,8541	0,8594	0,7243	0,6909	0,7401	0,6713	0,6134	0,6643	0,6625	0,7081	0,6292
wine	0,7131	0,6797	0,6520	0,6520	0,6284	0,6461	0,6284	0,6284	0,6016	0,4856	0,4601
winequality-red	0,4990	0,4840	0,5003	0,5028	0,5121	0,4896	0,4865	0,4740	0,4596	0,4596	0,4596
wisconsin	0,8010	0,8381	0,8524	0,7424	0,8010	0,8138	0,7610	0,7738	0,7824	0,7281	0,6181
yeast	0,3512	0,3720	0,3382	0,3329	0,3248	0,3552	0,3404	0,3208	0,3113	0,3296	0,3269
zoo	0,7514	0,7514	0,7314	0,6814	0,3814	0,3814	0,3814	0,3814	0,3814	0,3814	0,1831
Promedio	0,6233	0,6096	0,5981	0,5823	0,5606	0,5448	0,5462	0,5448	0,5366	0,5239	0,4872

B.2. Desempeño de los métodos individuales

Tabla B.17: Reducciones logradas por los métodos individuales que el sistema emplea.

Bases de datos	clusterBorder	core	coreHYB	coreNI	coreNIHYB	ENN
appendicitis	0,9486	0,9780	0,9780	0,9780	0,9780	0,1593
automobile	0,8216	0,8384	0,8374	0,8580	0,8585	0,4064
balance	0,8432	0,9524	0,9550	0,9650	0,9653	0,1559
bands	0,7627	0,7543	0,7543	0,7922	0,7922	0,2698
breast	0,9452	0,9581	0,9604	0,9580	0,9666	0,3500
car	0,9688	0,9825	0,9816	0,9839	0,9831	0,0775
cleveland	0,8992	0,9219	0,9219	0,9410	0,9410	0,4448
contraceptive	0,8538	0,8785	0,8797	0,9053	0,9065	0,5498
dermatology	0,8907	0,9490	0,9490	0,9520	0,9520	0,0310
ecoli	0,8614	0,9160	0,9160	0,9190	0,9190	0,1958
flare	0,9685	0,9742	0,9742	0,9827	0,9824	0,3809
glass	0,8354	0,8723	0,8723	0,8806	0,8806	0,3214
haberman	0,9128	0,9383	0,9375	0,9473	0,9473	0,3032
heart	0,8885	0,9387	0,9387	0,9560	0,9560	0,2206
hepatitis	0,8925	0,9126	0,9126	0,9147	0,9147	0,1828
horse-colic	0,8922	0,8566	0,8584	0,8692	0,8692	0,1769
housevotes	0,9663	0,9773	0,9773	0,9808	0,9808	0,0672
ionosphere	0,9047	0,8677	0,8677	0,8740	0,8740	0,1421
iris	0,9237	0,9533	0,9533	0,9533	0,9533	0,0474
led7digit	0,9520	0,9744	0,9729	0,9836	0,9793	0,5660
lymphography	0,8603	0,8986	0,8971	0,9114	0,9106	0,2146
mammographic	0,9218	0,9275	0,9268	0,9378	0,9376	0,2040
monk-2	0,5180	0,7911	0,7937	0,8922	0,8958	0,0411
movement_libras	0,7793	0,8096	0,8096	0,8315	0,8315	0,2275
newthyroid	0,9318	0,9705	0,9705	0,9705	0,9705	0,0579
pima	0,8550	0,8924	0,8924	0,9028	0,9028	0,2603
saheart	0,8487	0,8834	0,8834	0,9000	0,9000	0,3141
segment	0,9259	0,9502	0,9502	0,9549	0,9549	0,0443
sonar	0,7805	0,8237	0,8237	0,8381	0,8381	0,1699
spectfheart	0,8643	0,9147	0,9147	0,9322	0,9322	0,2863
splice	0,9566	0,9770	0,9771	0,9798	0,9796	0,2357
tic-tac-toe	0,8943	0,9330	0,9304	0,9383	0,9371	0,2284
vehicle	0,8031	0,8567	0,8567	0,8760	0,8760	0,2931
vowel	0,7828	0,8130	0,8130	0,8148	0,8148	0,0312
wdbc	0,9229	0,9621	0,9621	0,9662	0,9662	0,0307
wine	0,8689	0,9201	0,9201	0,9214	0,9214	0,0337
winequality-red	0,8407	0,8737	0,8737	0,8924	0,8924	0,4745
wisconsin	0,9642	0,9793	0,9792	0,9806	0,9809	0,0337
yeast	0,8437	0,8848	0,8848	0,9049	0,9049	0,4668
zoo	0,9131	0,9198	0,9198	0,9198	0,9198	0,0815
Promedio	0,8752	0,9094	0,9094	0,9215	0,9217	0,2195

Tabla B.18: Precisiones logradas (con C4.5) por los métodos individuales que el sistema emplea.

Bases de datos BD	clusterBorder	core	coreHYB	coreNI	coreNIHYB	ENN
appendicitis	0,8691	0,7418	0,7418	0,7418	0,7418	0,8418
automobile	0,5302	0,5143	0,5216	0,4895	0,4757	0,6247
balance	0,7251	0,6930	0,6960	0,6735	0,6769	0,7711
bands	0,6142	0,6325	0,6103	0,6142	0,6625	0,6829
breast	0,5680	0,6233	0,5890	0,5963	0,6643	0,7273
car	0,4941	0,4277	0,4693	0,4416	0,4553	0,8866
cleveland	0,4663	0,4630	0,4487	0,4681	0,4482	0,5543
contraceptive	0,5350	0,5072	0,5384	0,5194	0,5194	0,5105
dermatology	0,9179	0,5984	0,5984	0,5984	0,5984	0,9480
ecoli	0,7619	0,7535	0,7328	0,7595	0,7299	0,7949
flare	0,4889	0,4989	0,4186	0,5029	0,4643	0,7036
glass	0,5205	0,5071	0,4302	0,5211	0,4382	0,6503
haberman	0,7285	0,6563	0,6951	0,6720	0,6788	0,7448
heart	0,7000	0,7037	0,7185	0,7037	0,6704	0,7815
hepatitis	0,8142	0,7746	0,7296	0,7808	0,7546	0,8392
horse-colic	0,8015	0,6875	0,6820	0,6548	0,6548	0,8530
housevotes	0,6643	0,8248	0,8362	0,9539	0,9539	0,9494
ionosphere	0,5529	0,8177	0,6635	0,8063	0,7035	0,8946
iris	0,8800	0,6133	0,5200	0,6133	0,5267	0,9533
led7digit	0,3420	0,2620	0,3180	0,2260	0,2540	0,5260
lymphography	0,6215	0,6015	0,6174	0,6149	0,6076	0,7523
mammographic	0,8284	0,8159	0,8096	0,8190	0,8159	0,8253
monk-2	0,9795	0,9621	0,9621	0,8206	0,8206	0,9727
movement_libras	0,4806	0,4028	0,3972	0,3889	0,3917	0,6056
newthyroid	0,8000	0,5329	0,5381	0,5329	0,4777	0,9310
pima	0,7290	0,7411	0,7371	0,7383	0,7279	0,7397
saheart	0,7035	0,6925	0,6710	0,7079	0,6993	0,6991
segment	0,8104	0,6372	0,5779	0,6550	0,6121	0,9593
sonar	0,5767	0,5879	0,6164	0,6352	0,6593	0,7212
spectfheart	0,6751	0,6026	0,6783	0,6000	0,6788	0,7379
splice	0,6157	0,5520	0,5658	0,5495	0,5746	0,9367
tic-tac-toe	0,6483	0,5575	0,5574	0,5678	0,5719	0,7797
vehicle	0,6548	0,6195	0,6194	0,6264	0,6064	0,6962
vowel	0,6010	0,5747	0,5889	0,5697	0,5960	0,7879
wdbc	0,9350	0,7874	0,8683	0,7205	0,8789	0,9349
wine	0,8147	0,5663	0,6356	0,5605	0,6595	0,9320
winequality-red	0,5541	0,5497	0,5603	0,5535	0,5528	0,5797
wisconsin	0,9184	0,7956	0,7826	0,8442	0,8641	0,9471
yeast	0,5600	0,5378	0,5310	0,5175	0,5499	0,5741
zoo	0,5508	0,3458	0,3458	0,3458	0,3458	0,8947
Promedio	0,6758	0,6191	0,6155	0,6176	0,6191	0,7811

Tabla B.19: Precisiones logradas (con KNN) por los métodos individuales que el sistema emplea.

Bases de datos	clusterBorder	core	coreHYB	coreNI	coreNIHYB	ENN
appendicitis	0,8791	0,5464	0,8691	0,5464	0,8691	0,8973
automobile	0,6121	0,4812	0,5198	0,4805	0,4955	0,5383
balance	0,7888	0,7519	0,8448	0,7439	0,8417	0,8560
bands	0,7033	0,6494	0,7034	0,6363	0,7108	0,6198
breast	0,6177	0,6217	0,6109	0,6190	0,6705	0,7133
car	0,5624	0,5064	0,5346	0,5387	0,4982	0,8513
cleveland	0,5613	0,5482	0,5547	0,5216	0,5348	0,5643
contraceptive	0,4555	0,4501	0,4467	0,4555	0,4602	0,4528
dermatology	0,9264	0,8146	0,8256	0,8173	0,8310	0,9591
ecoli	0,7918	0,7297	0,8128	0,7176	0,7980	0,8248
flare	0,2439	0,3490	0,3283	0,3443	0,3463	0,6632
glass	0,6904	0,5277	0,6486	0,5197	0,6455	0,6919
haberman	0,6828	0,5852	0,6543	0,6274	0,6768	0,6990
heart	0,7889	0,7111	0,7333	0,7481	0,7556	0,7852
hepatitis	0,8258	0,7796	0,8050	0,7796	0,8050	0,8200
horse-colic	0,7414	0,6983	0,7176	0,6871	0,6818	0,8315
housevotes	0,6663	0,7327	0,7373	0,7395	0,7395	0,9193
ionosphere	0,8433	0,8460	0,8745	0,8431	0,8660	0,8433
iris	0,9533	0,8933	0,9467	0,8933	0,9600	0,9533
led7digit	0,4020	0,3920	0,4780	0,3600	0,4560	0,4920
lymphography	0,6539	0,6647	0,6312	0,6371	0,5961	0,7547
mammographic	0,7700	0,5193	0,7544	0,5736	0,7680	0,8096
monk-2	0,9054	0,6715	0,7145	0,6303	0,6325	0,7817
movement_libras	0,6972	0,6444	0,7056	0,6194	0,6667	0,7139
newthyroid	0,9303	0,8095	0,8610	0,8095	0,8656	0,9494
pima	0,7344	0,6666	0,7434	0,7057	0,7318	0,7462
saheart	0,6667	0,5609	0,6580	0,5914	0,6624	0,6949
segment	0,9390	0,8697	0,9459	0,8775	0,9446	0,9545
sonar	0,8112	0,6771	0,7871	0,6726	0,7879	0,8212
spectfheart	0,7494	0,5537	0,7567	0,6261	0,7681	0,7419
splice	0,5072	0,5313	0,5285	0,5047	0,5078	0,7530
tic-tac-toe	0,6536	0,6044	0,6169	0,6170	0,5992	0,7129
vehicle	0,6702	0,6620	0,6809	0,6632	0,6774	0,6927
vowel	0,9434	0,8889	0,9212	0,8919	0,9172	0,9687
wdbc	0,9596	0,8490	0,9613	0,8383	0,9665	0,9648
wine	0,9435	0,8869	0,9275	0,8869	0,9438	0,9552
winequality-red	0,5291	0,4928	0,5747	0,5297	0,5810	0,5847
wisconsin	0,9542	0,9328	0,9298	0,9370	0,9413	0,9642
yeast	0,5418	0,4778	0,5418	0,5128	0,5579	0,5505
zoo	0,6956	0,8678	0,8678	0,8678	0,8678	0,9114
Promedio	0,7248	0,6611	0,7189	0,6653	0,7156	0,7751

Tabla B.20: Precisiones logradas (con LDA) por los métodos individuales que el sistema emplea.

Bases de datos	clusterBorder	core	coreHYB	coreNI	coreNIHYB	ENN
appendicitis	0,8018	0,8018	0,8018	0,8018	0,8018	0,8691
automobile	0,0146	0,0146	0,0146	0,0146	0,0146	0,0146
balance	0,8768	0,8478	0,8720	0,8399	0,8752	0,8640
bands	0,6346	0,6457	0,6476	0,6383	0,6421	0,6921
breast	0,6429	0,6395	0,6136	0,6640	0,6707	0,7173
car	0,7587	0,5763	0,5902	0,5716	0,5705	0,8003
cleveland	0,5286	0,4923	0,5087	0,4986	0,4926	0,5711
contraceptive	0,5132	0,4969	0,5024	0,4990	0,5078	0,5017
dermatology	0,3060	0,3060	0,3060	0,3060	0,3060	0,9727
ecoli	0,4256	0,4256	0,4256	0,4256	0,4256	0,4256
flare	0,3105	0,3105	0,3105	0,3105	0,3105	0,6848
glass	0,5632	0,4792	0,5533	0,4881	0,5505	0,6024
haberman	0,7059	0,7062	0,7092	0,7418	0,7386	0,7481
heart	0,8000	0,5667	0,6481	0,5667	0,5815	0,8185
hepatitis	0,2063	0,2063	0,2063	0,2063	0,2063	0,8325
horse-colic	0,6980	0,7036	0,6791	0,7334	0,6899	0,8203
housevotes	0,3862	0,3862	0,3862	0,3862	0,3862	0,9422
ionosphere	0,6410	0,7406	0,7806	0,7552	0,7864	0,8378
iris	0,9333	0,5267	0,4867	0,5267	0,5067	0,9800
led7digit	0,3040	0,1260	0,1480	0,0900	0,0900	0,6060
lymphography	0,0133	0,0133	0,0133	0,0133	0,0133	0,1776
mammographic	0,8127	0,7899	0,7971	0,7992	0,7992	0,8304
monk-2	0,7865	0,7345	0,7431	0,7232	0,7161	0,8061
movement_libras	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667
newthyroid	0,8848	0,1396	0,1396	0,1396	0,1396	0,9162
pima	0,7775	0,7619	0,7567	0,7619	0,7593	0,7710
saheart	0,7099	0,7036	0,6992	0,6927	0,6861	0,7209
segment	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429
sonar	0,4662	0,4662	0,4662	0,4662	0,4662	0,7493
spectfheart	0,2058	0,2058	0,2058	0,2058	0,2058	0,7380
splice	0,7157	0,3705	0,3480	0,2652	0,2781	0,8991
tic-tac-toe	0,5972	0,5125	0,5261	0,5375	0,5072	0,7025
vehicle	0,7506	0,7482	0,7293	0,7364	0,7400	0,7612
vowel	0,6121	0,6222	0,6273	0,6222	0,6253	0,5990
wdbc	0,9069	0,3726	0,3726	0,3726	0,3726	0,9525
wine	0,8879	0,3317	0,3317	0,3317	0,3317	0,9775
winequality-red	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
wisconsin	0,9599	0,8510	0,8481	0,8295	0,8424	0,9528
yeast	0,2040	0,2054	0,2047	0,1644	0,1644	0,2482
zoo	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189
Promedio	0,5493	0,4614	0,4657	0,4589	0,4607	0,6683

Tabla B.21: Precisiones logradas (con PART) por los métodos individuales que el sistema emplea.

Bases de datos	clusterBorder	core	coreHYB	coreNI	coreNIHYB	ENN
appendicitis	0,8018	0,7418	0,7418	0,7418	0,7418	0,8509
automobile	0,3779	0,4147	0,4106	0,3612	0,3612	0,3735
balance	0,5580	0,5484	0,6484	0,5119	0,6120	0,5488
bands	0,5735	0,5898	0,5731	0,6179	0,6216	0,5789
breast	0,5617	0,6716	0,5790	0,6727	0,6606	0,6920
car	0,3754	0,3627	0,3748	0,4080	0,4605	0,7002
cleveland	0,4662	0,5345	0,5078	0,4878	0,5145	0,5412
contraceptive	0,4304	0,4284	0,4291	0,4440	0,4684	0,4338
dermatology	0,3384	0,1667	0,1667	0,1667	0,1667	0,8247
ecoli	0,2293	0,2869	0,3290	0,2869	0,3475	0,4912
flare	0,2451	0,2787	0,2430	0,3327	0,2618	0,3105
glass	0,3952	0,3745	0,3791	0,3788	0,3783	0,4428
haberman	0,7353	0,6863	0,7284	0,7353	0,7353	0,7353
heart	0,6222	0,6111	0,6037	0,6037	0,5926	0,5815
hepatitis	0,7875	0,7875	0,7938	0,7875	0,7671	0,8196
horse-colic	0,6764	0,6305	0,6305	0,6305	0,6412	0,6305
housevotes	0,6825	0,7736	0,7736	0,8041	0,8041	0,9306
ionosphere	0,4275	0,3590	0,3590	0,3590	0,3590	0,8862
iris	0,6333	0,3333	0,3333	0,3333	0,3333	0,3333
led7digit	0,1200	0,1100	0,1180	0,0980	0,1160	0,1880
lymphography	0,5862	0,6769	0,6017	0,5611	0,5996	0,5704
mammographic	0,8127	0,7607	0,7899	0,7846	0,8190	0,7442
monk-2	0,4734	0,9621	0,9389	0,7401	0,7401	0,6821
movement_libras	0,1222	0,1583	0,1556	0,1444	0,1861	0,2417
newthyroid	0,6162	0,5403	0,5266	0,5403	0,5266	0,7026
pima	0,7018	0,6756	0,7162	0,7201	0,7135	0,6511
saheart	0,6541	0,6732	0,6950	0,6689	0,6667	0,6602
segment	0,2584	0,3584	0,4952	0,2844	0,4364	0,7260
sonar	0,6155	0,5343	0,6157	0,5424	0,5948	0,6593
spectfheart	0,7942	0,6889	0,7312	0,7195	0,7571	0,7942
splice	0,4263	0,4288	0,4335	0,3633	0,3740	0,5141
tic-tac-toe	0,6093	0,4666	0,3935	0,4446	0,4813	0,6535
vehicle	0,3825	0,3805	0,3673	0,4328	0,3819	0,3436
vowel	0,1091	0,1232	0,1121	0,1232	0,1091	0,1455
wdbc	0,6506	0,6558	0,5327	0,6292	0,6625	0,8541
wine	0,6520	0,4660	0,4507	0,4601	0,4801	0,6510
winequality-red	0,4809	0,4465	0,4690	0,4596	0,4415	0,4909
wisconsin	0,7862	0,6452	0,7281	0,6538	0,5938	0,7899
yeast	0,3242	0,3248	0,3498	0,3269	0,3268	0,3093
zoo	0,3814	0,1831	0,1831	0,1831	0,1831	0,7553
Promedio	0,5119	0,4960	0,5002	0,4886	0,5004	0,5958

B.3. Desempeño de los métodos convencionales

Tabla B.22: Reducciones logradas por los métodos convencionales.

Bases de datos	CCIS	CNN	CPruner	DROP3	FCNN	IB3	ICF	NRMCS	PSC
appendicitis	0,9707	0,6269	0,9004	0,8774	0,6876	0,6708	0,8512	0,9885	0,7400
automobile	0,8472	0,4959	0,8954	0,5724	0,5170	0,5511	0,5171	0,8981	0,5717
balance	0,9449	0,6363	0,8078	0,8676	0,6804	0,7154	0,9396	0,9934	0,7591
bands	0,9561	0,5275	0,9520	0,7267	0,5718	0,6627	0,6116	0,9557	0,6640
breast	0,9413	0,4600	0,9514	0,7844	0,4903	0,6325	0,6791	0,9814	0,6538
car	0,9459	0,7689	0,8285	0,8834	0,8355	0,8373	0,8555	0,9687	0,6782
cleveland	0,6564	0,3829	0,9388	0,8317	0,4005	0,4664	0,7972	0,9806	0,5732
contraceptive	0,7693	0,2624	0,9892	0,7307	0,2873	0,4433	0,6881	0,9738	0,4773
dermatology	0,9572	0,8558	0,8819	0,9232	0,8813	0,8880	0,7034	0,9848	0,9038
ecoli	0,8624	0,5890	0,9183	0,8423	0,6359	0,6802	0,7460	0,9795	0,7130
flare	0,8792	0,3892	0,9405	0,8394	0,5980	0,6498	0,5422	0,9949	0,7868
glass	0,8541	0,5104	0,9242	0,7415	0,5386	0,6107	0,6775	0,9496	0,5800
haberman	0,9564	0,4738	0,9354	0,8003	0,4877	0,6347	0,6543	0,9837	0,6191
heart	0,9556	0,5926	0,9296	0,8259	0,6424	0,7457	0,8037	0,9848	0,7156
hepatitis	0,9842	0,6201	0,9054	0,8566	0,6473	0,7275	0,7728	0,9828	0,7305
horse-colic	0,9647	0,5954	0,8940	0,8593	0,6477	0,7491	0,8083	0,9894	0,7627
housevotes	0,9865	0,8082	0,9126	0,9351	0,8756	0,9208	0,8927	0,9949	0,8825
ionosphere	0,9778	0,7680	0,8598	0,9079	0,8145	0,8398	0,8072	0,9924	0,8430
iris	0,9667	0,8407	0,8985	0,9230	0,8711	0,9089	0,6422	0,9778	0,8667
led7digit	0,8469	0,2360	0,9858	0,8278	0,6382	0,6338	0,6178	0,9984	0,7627
lymphography	0,9362	0,5398	0,9294	0,8356	0,5871	0,6922	0,7530	0,9535	0,6915
mammographic	0,9780	0,5625	0,8836	0,8432	0,6140	0,7784	0,6616	0,9961	0,7554
monk-2	0,9661	0,6842	0,8642	0,8670	0,9578	0,7755	0,6880	0,9547	0,4725
movement_libras	0,6975	0,5818	0,8917	0,6914	0,6238	0,6552	0,5074	0,8988	0,6062
newthyroid	0,9814	0,8749	0,8569	0,8853	0,8884	0,8739	0,8527	0,9897	0,8724
pima	0,9521	0,5041	0,9510	0,8228	0,5327	0,6913	0,7734	0,9763	0,6612
saheart	0,9512	0,4543	0,9565	0,7987	0,4860	0,6455	0,7453	0,9634	0,6328
segment	0,9783	0,8843	0,8487	0,9156	0,8985	0,9100	0,7846	0,9952	0,8736
sonar	0,9615	0,6891	0,8184	0,7591	0,7003	0,7238	0,7121	0,9525	0,7009
spectfheart	0,9879	0,5368	0,9443	0,8339	0,5743	0,7041	0,8922	0,9938	0,7595
splice	0,8983	0,5405	0,9415	0,7647	0,5878	0,6755	0,6969	0,9689	0,6960
tic-tac-toe	0,9618	0,6671	0,8334	0,9287	0,6951	0,7308	0,7090	0,9836	0,6857
vehicle	0,8245	0,5051	0,9551	0,7727	0,5169	0,6106	0,6916	0,9456	0,5678
vowel	0,9382	0,7753	0,8331	0,6721	0,7637	0,7851	0,2878	0,9061	0,6962
wdbc	0,9924	0,8573	0,8903	0,9594	0,8746	0,9235	0,9695	0,9961	0,9211
wine	0,9831	0,8458	0,8596	0,9101	0,8758	0,8951	0,8377	0,9813	0,8839
winequality-red	0,8576	0,3342	0,9715	0,7641	0,4385	0,5273	0,7024	0,9262	0,5740
wisconsin	1,0000	0,8841	0,9126	0,9747	0,9048	0,9585	0,9528	0,9968	0,9450
yeast	0,7838	0,3216	0,9773	0,7844	0,3492	0,4608	0,7311	0,9550	0,5198
zoo	0,8107	0,7725	0,8501	0,8351	0,8592	0,8328	0,4360	0,9571	0,7823
Promedio	0,9166	0,6064	0,9055	0,8294	0,6619	0,7205	0,7248	0,9711	0,7145

Tabla B.23: Precisiones logradas (con C4.5) por los métodos convencionales.

Bases de datos	CCIS	CNN	CPruner	DROP3	FCNN	IB3	ICF	NRMCS	PSC
appendicitis	0,8118	0,7927	0,8109	0,6445	0,7736	0,7645	0,8127	0,8018	0,7518
automobile	0,4640	0,6664	0,3987	0,5564	0,6656	0,6507	0,5479	0,5221	0,6792
balance	0,6176	0,7649	0,6466	0,7408	0,7505	0,7343	0,6320	0,5507	0,7453
bands	0,5064	0,6213	0,5290	0,5862	0,6476	0,5973	0,6344	0,5660	0,5753
breast	0,7097	0,7102	0,6959	0,6482	0,6890	0,6928	0,7198	0,6853	0,6088
car	0,5457	0,8408	0,7269	0,5533	0,8357	0,8270	0,7031	0,7002	0,8658
cleveland	0,4424	0,4917	0,5412	0,5056	0,4885	0,4888	0,4924	0,5412	0,4785
contraceptive	0,4970	0,5133	0,4142	0,5160	0,5105	0,4868	0,5262	0,4949	0,4922
dermatology	0,7241	0,7705	0,8166	0,8441	0,9181	0,8773	0,9098	0,3523	0,7150
ecoli	0,6133	0,7532	0,5203	0,7147	0,7232	0,7417	0,7890	0,4195	0,6376
flare	0,7083	0,6961	0,5657	0,6727	0,7046	0,6989	0,6117	0,3237	0,6952
glass	0,5069	0,6339	0,4816	0,5516	0,5782	0,5478	0,6355	0,4162	0,6334
haberman	0,7353	0,6924	0,7353	0,7252	0,7286	0,7085	0,7444	0,6869	0,7353
heart	0,7296	0,7704	0,7370	0,7815	0,7370	0,6926	0,7148	0,5481	0,7074
hepatitis	0,7938	0,6854	0,7617	0,7800	0,6517	0,7454	0,7992	0,3125	0,5492
horse-colic	0,8094	0,8475	0,8285	0,7876	0,8179	0,8012	0,8096	0,6845	0,5782
housevotes	0,7392	0,9470	0,8735	0,9381	0,9539	0,9354	0,9422	0,3862	0,9332
ionosphere	0,6273	0,7781	0,7976	0,5558	0,7040	0,7663	0,8746	0,6410	0,7468
iris	0,4333	0,7933	0,6467	0,8667	0,5600	0,6467	0,9400	0,3333	0,9067
led7digit	0,5340	0,6860	0,1740	0,5140	0,4800	0,5220	0,5620	0,0980	0,4500
lymphography	0,5859	0,7301	0,6812	0,6061	0,6961	0,7085	0,7131	0,6540	0,6544
mammographic	0,8107	0,8221	0,8127	0,8055	0,8190	0,8179	0,8284	0,6067	0,8190
monk-2	0,8161	1,0000	0,8766	0,9727	0,9864	0,9909	0,7050	0,8065	0,9598
movement_libras	0,4222	0,4694	0,2028	0,4583	0,4889	0,4611	0,5417	0,3056	0,4000
newthyroid	0,4143	0,8054	0,7580	0,6784	0,7959	0,8188	0,7877	0,1623	0,8186
pima	0,6849	0,7057	0,6630	0,6130	0,7006	0,7071	0,7307	0,6550	0,6316
saheart	0,6214	0,6861	0,6537	0,6384	0,6494	0,6255	0,6513	0,6383	0,6276
segment	0,7597	0,8199	0,8900	0,7697	0,8494	0,8229	0,8935	0,3173	0,8407
sonar	0,6243	0,6345	0,6360	0,7021	0,5810	0,6290	0,5688	0,6390	0,6674
spectfheart	0,6756	0,7526	0,7942	0,6853	0,7301	0,7487	0,5766	0,7942	0,7088
splice	0,8248	0,9219	0,6677	0,8940	0,9201	0,9132	0,9129	0,7382	0,9163
tic-tac-toe	0,6335	0,7026	0,6535	0,4990	0,7786	0,6953	0,7536	0,6691	0,6858
vehicle	0,6337	0,6704	0,4716	0,6161	0,6537	0,6819	0,6216	0,5604	0,6713
vowel	0,3717	0,6020	0,4566	0,6616	0,6485	0,6263	0,7444	0,4010	0,6071
wdbc	0,6099	0,9087	0,8560	0,7541	0,8629	0,8701	0,6855	0,3726	0,8505
wine	0,4095	0,8085	0,7696	0,7245	0,7693	0,8131	0,6912	0,3317	0,7758
winequality-red	0,4653	0,5303	0,5103	0,5335	0,5516	0,5128	0,5385	0,5310	0,5047
wisconsin	0,6552	0,8912	0,9128	0,8026	0,8827	0,9084	0,8699	0,6552	0,6662
yeast	0,4515	0,5169	0,3396	0,5425	0,5148	0,4785	0,5547	0,5398	0,4704
zoo	0,8572	0,8772	0,7486	0,9008	0,4339	0,7256	0,9017	0,3058	0,7183
Promedio	0,6219	0,7328	0,6514	0,6835	0,7058	0,7120	0,7168	0,5187	0,6870

Tabla B.24: Precisiones logradas (con KNN) por los métodos convencionales.

Bases de datos	CCIS	CNN	CPruner	DROP3	FCNN	IB3	ICF	NRMCS	PSC
appendicitis	0,8400	0,8027	0,8018	0,6536	0,7745	0,7355	0,5464	0,8118	0,6800
automobile	0,5356	0,7418	0,3673	0,6281	0,7461	0,7313	0,5369	0,5329	0,6534
balance	0,7583	0,7040	0,8544	0,8177	0,7105	0,7152	0,7026	0,8544	0,6786
bands	0,5212	0,5921	0,5603	0,6012	0,5808	0,5992	0,5474	0,5770	0,5677
breast	0,7026	0,6152	0,6990	0,6577	0,6050	0,6184	0,6646	0,7203	0,5064
car	0,7147	0,8802	0,7552	0,6221	0,8970	0,8744	0,7842	0,7460	0,8160
cleveland	0,4488	0,5083	0,5412	0,4947	0,5085	0,4916	0,5019	0,5412	0,4714
contraceptive	0,4155	0,4176	0,4182	0,4406	0,4155	0,4033	0,4480	0,4501	0,3870
dermatology	0,9372	0,9346	0,9402	0,9293	0,9265	0,9318	0,8852	0,8606	0,9101
ecoli	0,6610	0,7502	0,5955	0,7263	0,7472	0,7236	0,7676	0,7234	0,6163
flare	0,5695	0,4652	0,4774	0,6398	0,5121	0,4437	0,4907	0,3612	0,5498
glass	0,5764	0,6940	0,4716	0,6571	0,7123	0,6549	0,6021	0,6250	0,6734
haberman	0,7319	0,6339	0,7353	0,6697	0,6272	0,6272	0,6732	0,7448	0,6109
heart	0,7333	0,7630	0,7815	0,6926	0,7519	0,6889	0,6519	0,7593	0,7111
hepatitis	0,7938	0,7625	0,7933	0,8129	0,7692	0,7308	0,7621	0,7929	0,6917
horse-colic	0,7878	0,7366	0,7794	0,7769	0,7339	0,7309	0,6900	0,7986	0,6577
housevotes	0,8660	0,8918	0,8985	0,9102	0,8893	0,9125	0,8410	0,8575	0,8688
ionosphere	0,6981	0,8660	0,8317	0,7521	0,8606	0,8233	0,8175	0,8091	0,8433
iris	0,9467	0,9200	0,7733	0,9267	0,9000	0,9067	0,8933	0,9333	0,9200
led7digit	0,3920	0,3700	0,2160	0,4060	0,3620	0,3840	0,3720	0,0700	0,3740
lymphography	0,7069	0,6361	0,6569	0,7674	0,7155	0,6427	0,7339	0,7679	0,6839
mammographic	0,7867	0,6962	0,7961	0,7243	0,7337	0,7003	0,7816	0,8013	0,5984
monk-2	0,7906	0,8340	0,6543	0,6571	0,9909	0,8074	0,6946	0,7551	0,8640
movement_libras	0,7000	0,7750	0,3306	0,7583	0,7667	0,7361	0,7278	0,5583	0,7278
newthyroid	0,9303	0,9630	0,9026	0,9491	0,9489	0,9489	0,7626	0,8091	0,9487
pima	0,6941	0,6472	0,6524	0,6992	0,6499	0,6538	0,6692	0,7256	0,5885
saheart	0,6388	0,6276	0,6537	0,6492	0,6017	0,6168	0,6342	0,7014	0,5212
segment	0,8606	0,9407	0,9100	0,9100	0,9481	0,9286	0,8121	0,8087	0,9221
sonar	0,6874	0,8552	0,6743	0,7779	0,8264	0,7643	0,6633	0,7252	0,7645
spectfheart	0,6349	0,6594	0,7942	0,6973	0,6332	0,6809	0,4792	0,7682	0,5957
splice	0,6743	0,6853	0,6574	0,7025	0,6962	0,6818	0,6589	0,6987	0,6649
tic-tac-toe	0,6524	0,8122	0,6670	0,5731	0,7662	0,7788	0,7297	0,6806	0,7401
vehicle	0,6075	0,6536	0,4515	0,6099	0,6489	0,6855	0,6336	0,5993	0,6620
vowel	0,3960	0,9576	0,4222	0,9313	0,9667	0,9535	0,9313	0,6182	0,8626
wdbc	0,9367	0,9280	0,9297	0,8928	0,9227	0,9332	0,7169	0,9261	0,8506
wine	0,6814	0,8984	0,9150	0,9105	0,9033	0,9209	0,9493	0,8931	0,9493
winequality-red	0,4928	0,4828	0,5372	0,5472	0,4822	0,4697	0,5097	0,5322	0,4478
wisconsin	0,0000	0,9399	0,9571	0,8913	0,9371	0,9327	0,8170	0,9513	0,8641
yeast	0,4245	0,4684	0,4730	0,5331	0,4643	0,4448	0,5088	0,5640	0,4279
zoo	0,9281	0,8433	0,7753	0,9264	0,7706	0,8769	0,9322	0,7147	0,8431
Promedio	0,6714	0,7338	0,6775	0,7231	0,7351	0,7221	0,6881	0,7042	0,6929

Tabla B.25: Precisiones logradas (con LDA) por los métodos convencionales.

Bases de datos	CCIS	CNN	CPruner	DROP3	FCNN	IB3	ICF	NRMCS	PSC
appendicitis	0,8018	0,8400	0,8018	0,5582	0,7073	0,7827	0,6673	0,8018	0,6900
automobile	0,0146	0,5613	0,0146	0,4555	0,6117	0,5572	0,4093	0,0146	0,1705
balance	0,7168	0,9040	0,8640	0,8767	0,9168	0,9103	0,7266	0,4607	0,8800
bands	0,4211	0,6680	0,4211	0,6700	0,6329	0,6308	0,6792	0,4211	0,6291
breast	0,6252	0,7134	0,7033	0,6576	0,7168	0,6680	0,7032	0,7033	0,6041
car	0,5134	0,8154	0,7466	0,7581	0,8137	0,8061	0,7604	0,7552	0,8235
cleveland	0,5157	0,5946	0,5412	0,5648	0,5651	0,5552	0,5351	0,5412	0,5252
contraceptive	0,5037	0,5085	0,4270	0,5132	0,5132	0,5038	0,5023	0,4650	0,5119
dermatology	0,3060	0,3060	0,3060	0,3060	0,3060	0,3060	0,8691	0,3060	0,3060
ecoli	0,4256	0,6734	0,4256	0,4256	0,6883	0,6676	0,4256	0,4256	0,6471
flare	0,3105	0,7092	0,3105	0,5375	0,4851	0,4435	0,5010	0,3105	0,5497
glass	0,4939	0,5585	0,4277	0,5269	0,5652	0,6103	0,6198	0,3290	0,5750
haberman	0,6547	0,7513	0,2647	0,7123	0,7384	0,7353	0,7286	0,2647	0,7513
heart	0,5556	0,8148	0,5556	0,7704	0,8296	0,7815	0,8000	0,5556	0,7852
hepatitis	0,2063	0,7738	0,2063	0,4021	0,7217	0,7358	0,6675	0,2063	0,6246
horse-colic	0,6305	0,7771	0,6548	0,7467	0,7390	0,6897	0,7495	0,6305	0,6245
housevotes	0,3862	0,9517	0,3862	0,9126	0,9539	0,9146	0,8900	0,3862	0,9401
ionosphere	0,6410	0,6839	0,6410	0,6410	0,6979	0,6898	0,8290	0,6410	0,7037
iris	0,3333	0,9533	0,7800	0,8933	0,9733	0,9000	0,9600	0,3333	0,9467
led7digit	0,7000	0,7420	0,0900	0,5280	0,7020	0,7000	0,5520	0,0620	0,7020
lymphography	0,0133	0,7982	0,0133	0,0133	0,8039	0,7455	0,0133	0,0133	0,5357
mammographic	0,8044	0,8127	0,6796	0,8169	0,8096	0,8180	0,8169	0,5369	0,8148
monk-2	0,6827	0,7720	0,7377	0,7880	0,5624	0,7747	0,7065	0,5587	0,7491
movement_libras	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667	0,0667
newthyroid	0,1396	0,9450	0,8022	0,9121	0,9353	0,9535	0,8472	0,1396	0,9444
pima	0,7397	0,7697	0,6642	0,7448	0,7670	0,7605	0,7709	0,7137	0,7658
saheart	0,6341	0,7099	0,6537	0,7272	0,7165	0,7121	0,7186	0,6690	0,6251
segment	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429	0,1429
sonar	0,4662	0,4805	0,4662	0,4662	0,4614	0,4662	0,4812	0,4662	0,4862
spectfheart	0,1651	0,7187	0,2058	0,2836	0,6628	0,6885	0,2058	0,2058	0,5776
splice	0,8668	0,9241	0,7147	0,9094	0,9226	0,9169	0,8821	0,6599	0,9116
tic-tac-toe	0,5824	0,5396	0,6576	0,5554	0,6836	0,6701	0,7004	0,6775	0,5657
vehicle	0,7256	0,7399	0,5200	0,7032	0,7458	0,7482	0,7269	0,6584	0,7435
vowel	0,3485	0,6081	0,4636	0,6202	0,6222	0,6293	0,6141	0,4939	0,5141
wdbc	0,3726	0,9103	0,8927	0,3726	0,9155	0,7734	0,3726	0,3726	0,7031
wine	0,2984	0,9275	0,7765	0,4725	0,6850	0,7216	0,8827	0,3317	0,7824
winequality-red	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
wisconsin	0,0000	0,9642	0,9256	0,8455	0,9614	0,9357	0,9057	0,6552	0,8798
yeast	0,1644	0,5789	0,1644	0,2392	0,5843	0,5782	0,2439	0,1644	0,5722
zoo	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189	0,4189
Promedio	0,4347	0,6782	0,4884	0,5639	0,6586	0,6527	0,6023	0,4140	0,6197

Tabla B.26: Precisiones logradas (con PART) por los métodos convencionales.

Bases de datos	CCIS	CNN	CPruner	DROP3	FCNN	IB3	ICF	NRMCS	PSC
appendicitis	0,8018	0,7836	0,8109	0,6245	0,8027	0,7645	0,8127	0,8018	0,8018
automobile	0,2940	0,4102	0,2963	0,3647	0,4239	0,3715	0,3334	0,3780	0,3487
balance	0,6336	0,5534	0,5812	0,5201	0,5204	0,5102	0,4802	0,4560	0,5259
bands	0,5138	0,5733	0,5697	0,5696	0,5752	0,5900	0,4843	0,5457	0,5974
breast	0,6589	0,6964	0,6852	0,6647	0,7032	0,6502	0,6791	0,6683	0,5001
car	0,6939	0,3659	0,7002	0,2644	0,7002	0,7002	0,7002	0,7002	0,5089
cleveland	0,3338	0,5212	0,5412	0,5412	0,5080	0,5045	0,5412	0,5412	0,5020
contraceptive	0,3578	0,4270	0,4155	0,4359	0,4304	0,4284	0,4413	0,4474	0,4236
dermatology	0,3504	0,2438	0,4694	0,2462	0,2072	0,3579	0,6830	0,3060	0,2359
ecoli	0,2739	0,4453	0,4227	0,3341	0,4965	0,4666	0,4227	0,3574	0,3506
flare	0,1979	0,2214	0,3105	0,2383	0,2101	0,2289	0,2833	0,3105	0,2242
glass	0,3055	0,3966	0,3598	0,4668	0,3602	0,4030	0,4469	0,3810	0,4006
haberman	0,7353	0,7191	0,7353	0,7353	0,7353	0,7353	0,7353	0,6869	0,7353
heart	0,5815	0,6778	0,5926	0,6111	0,7037	0,6704	0,5926	0,5111	0,6444
hepatitis	0,7938	0,7688	0,7617	0,8071	0,7504	0,7738	0,8325	0,3188	0,5933
horse-colic	0,7878	0,6305	0,6980	0,6605	0,6305	0,6193	0,5736	0,6305	0,5730
housevotes	0,4962	0,6095	0,6138	0,9538	0,8567	0,8299	0,5601	0,3862	0,6820
ionosphere	0,4818	0,6072	0,7976	0,4444	0,5675	0,5493	0,4104	0,6410	0,7437
iris	0,3333	0,7000	0,3933	0,6933	0,4200	0,5333	0,3333	0,3333	0,8800
led7digit	0,2020	0,1460	0,1600	0,1860	0,1520	0,2220	0,1940	0,0980	0,1640
lymphography	0,5371	0,6582	0,4277	0,5429	0,5891	0,6587	0,7187	0,5797	0,6109
mammographic	0,6825	0,7006	0,5369	0,7846	0,7326	0,6546	0,7773	0,4932	0,5369
monk-2	0,4779	0,5211	0,5266	0,8065	0,5075	0,4780	0,6404	0,5166	0,4720
movement_libras	0,0917	0,0639	0,0917	0,0722	0,0944	0,0917	0,1278	0,0944	0,0722
newthyroid	0,3747	0,7918	0,7167	0,5846	0,7338	0,7762	0,6851	0,1623	0,7677
pima	0,6693	0,6587	0,6446	0,5419	0,6795	0,6748	0,6955	0,6679	0,6224
saheart	0,5957	0,6409	0,6537	0,6492	0,6668	0,6342	0,6516	0,6471	0,6386
segment	0,2550	0,2835	0,2913	0,2571	0,3844	0,2619	0,4238	0,1610	0,3476
sonar	0,4662	0,5571	0,6121	0,6067	0,5479	0,5381	0,4900	0,5662	0,6286
spectfheart	0,6756	0,7645	0,7942	0,7158	0,7905	0,7942	0,6063	0,7942	0,7578
splice	0,2934	0,5232	0,2708	0,5223	0,5298	0,5282	0,4470	0,4821	0,4379
tic-tac-toe	0,6535	0,3476	0,6535	0,3663	0,6535	0,6535	0,6535	0,6597	0,6587
vehicle	0,3642	0,2839	0,3380	0,3142	0,2944	0,2859	0,3096	0,4240	0,3283
vowel	0,1576	0,1232	0,1646	0,1242	0,1071	0,1010	0,1172	0,1303	0,1172
wdbc	0,4239	0,7331	0,6520	0,7664	0,7246	0,7292	0,5638	0,3726	0,7537
wine	0,3373	0,5379	0,5212	0,6013	0,6516	0,6346	0,4000	0,3317	0,5912
winequality-red	0,4040	0,4415	0,4478	0,4966	0,4234	0,4134	0,4290	0,4647	0,4240
wisconsin	0,6552	0,5099	0,8052	0,7798	0,4405	0,6771	0,8799	0,6552	0,5905
yeast	0,3046	0,3228	0,1636	0,3437	0,3167	0,3289	0,3512	0,3073	0,3188
zoo	0,5072	0,4631	0,5725	0,5511	0,0450	0,3500	0,4314	0,4189	0,4972
Promedio	0,4688	0,5106	0,5200	0,5197	0,5167	0,5293	0,5235	0,4607	0,5152

B.4. Desempeño de los métodos basados en meta-heurísticas

Tabla B.27: Reducciones logradas por Explore, SSMA y ZhangTS.

Bases de datos	Explore	SSMA	ZhangTS
appendicitis	0,9790	0,9644	0,0000
automobile	0,9610	0,9225	0,0000
balance	0,9815	0,9764	0,0000
bands	0,9907	0,9672	0,0000
breast	0,9895	0,9790	0,0000
car	0,9859	0,9644	0,0000
cleveland	0,9850	0,9773	0,0000
contraceptive	0,9841	0,9691	0,0000
dermatology	0,9733	0,9663	0,9778
ecoli	0,9636	0,9507	0,0000
flare	0,9802	0,9748	0,0000
glass	0,9647	0,9294	0,0000
haberman	0,9938	0,9789	0,0000
heart	0,9860	0,9724	0,0000
hepatitis	0,9921	0,9778	0,0000
horse-colic	0,9885	1,0000	0,0000
housevotes	0,9893	0,9903	0,1982
ionosphere	0,9801	0,9655	0,0000
iris	0,9778	0,9607	0,8733
led7digit	0,9653	0,9680	0,0000
lymphography	0,9730	0,9512	0,0000
mammographic	0,9927	0,9923	0,0001
monk-2	0,9761	0,9835	0,0000
movement_libras	0,9136	0,8590	0,0000
newthyroid	0,9824	0,9721	0,9829
pima	0,9894	0,9771	0,0000
saheart	0,9887	0,9723	0,0000
segment	0,9811	0,9741	0,9809
sonar	0,9680	0,9274	0,0000
spectfheart	0,9958	0,9813	0,0000
splice	0,9793	0,9704	0,0000
tic-tac-toe	0,9911	0,9407	0,0000
vehicle	0,9594	0,9366	0,0000
vowel	0,9400	0,8521	0,7512
wdbc	0,9939	0,9846	0,9951
wine	0,9757	0,9594	0,9800
winequality-red	0,9791	0,9695	0,0000
wisconsin	0,9960	0,9935	0,9968
yeast	0,9786	0,9675	0,0000
zoo	0,9220	0,8998	0,0000
Promedio	0,9772	0,9605	0,1934

Tabla B.28: Precisiones logradas (con C4.5) por Explore, SSMA y ZhangTS.

Bases de datos	Explore	SSMA	ZhangTS
appendicitis	0,8018	0,8318	0,8327
automobile	0,3822	0,4117	0,8094
balance	0,6256	0,6497	0,7680
bands	0,4878	0,5762	0,6364
breast	0,7102	0,7328	0,7591
car	0,6996	0,6863	0,9155
cleveland	0,5545	0,5150	0,5476
contraceptive	0,4658	0,5173	0,5384
dermatology	0,4754	0,6392	0,3528
ecoli	0,4939	0,6187	0,7947
flare	0,5910	0,7130	0,7430
glass	0,3481	0,4531	0,6744
haberman	0,6000	0,7286	0,7316
heart	0,5963	0,6963	0,7852
hepatitis	0,7338	0,3492	0,8258
horse-colic	0,6661	0,0229	0,8531
housevotes	0,7786	0,7872	0,9040
ionosphere	0,4504	0,5666	0,9119
iris	0,3333	0,6667	0,5600
led7digit	0,4820	0,4560	0,7100
lymphography	0,5632	0,6726	0,7430
mammographic	0,7858	0,7942	0,8179
monk-2	0,8847	0,6823	1,0000
movement_libras	0,3111	0,3500	0,6944
newthyroid	0,3195	0,5740	0,3084
pima	0,6600	0,7030	0,7397
saheart	0,5325	0,6014	0,6709
segment	0,5818	0,6528	0,5645
sonar	0,6393	0,5248	0,7055
spectfheart	0,7942	0,6231	0,7645
splice	0,7210	0,7414	0,9404
tic-tac-toe	0,6930	0,6523	0,8455
vehicle	0,5131	0,5992	0,7411
vowel	0,4182	0,5616	0,6303
wdbc	0,6011	0,7890	0,4077
wine	0,5170	0,6585	0,3928
winequality-red	0,5310	0,5304	0,6072
wisconsin	0,5295	0,7967	0,6552
yeast	0,4806	0,5028	0,5560
zoo	0,5883	0,5953	0,9281
Promedio	0,5735	0,6056	0,7092

Tabla B.29: Precisiones logradas (con KNN) por Explore, SSMA y ZhangTS.

Bases de datos	Explore	SSMA	ZhangTS
appendicitis	0,8982	0,8700	0,7936
automobile	0,5319	0,6605	0,7611
balance	0,8749	0,8849	0,7904
bands	0,5733	0,6999	0,6309
breast	0,5540	0,7409	0,6535
car	0,6271	0,9080	0,8565
cleveland	0,5576	0,5659	0,5314
contraceptive	0,4861	0,4833	0,4277
dermatology	0,9482	0,9383	0,9372
ecoli	0,7680	0,8129	0,8070
flare	0,7214	0,5478	0,6463
glass	0,6297	0,6750	0,7361
haberman	0,7220	0,7027	0,6697
heart	0,7889	0,8185	0,7704
hepatitis	0,7804	0,2933	0,8075
horse-colic	0,8286	0,0000	0,7663
housevotes	0,9193	0,9338	0,9286
ionosphere	0,8267	0,8946	0,8717
iris	0,9533	0,9400	0,9533
led7digit	0,6080	0,3460	0,4020
lymphography	0,8031	0,7810	0,7387
mammographic	0,8023	0,8426	0,7462
monk-2	0,9473	0,9772	0,7791
movement_libras	0,6056	0,6500	0,8194
newthyroid	0,9628	0,9680	0,9537
pima	0,7579	0,7579	0,7033
saheart	0,7142	0,7056	0,6449
segment	0,9294	0,9446	0,9437
sonar	0,6831	0,7733	0,8555
spectfheart	0,7942	0,6161	0,6970
splice	0,7555	0,7536	0,7524
tic-tac-toe	0,5734	0,7327	0,7307
vehicle	0,6051	0,6478	0,7010
vowel	0,6465	0,8778	0,9535
wdbc	0,9613	0,9525	0,9666
wine	0,9546	0,9493	0,9386
winequality-red	0,5929	0,5803	0,5316
wisconsin	0,9671	0,9709	0,9686
yeast	0,5661	0,5708	0,5047
zoo	0,8875	0,8461	0,9281
Promedio	0,7527	0,7404	0,7650

Tabla B.30: Precisiones logradas (con LDA) por Explore, SSMA y ZhangTS.

Bases de datos	Explore	SSMA	ZhangTS
appendicitis	0,8018	0,8018	0,8691
automobile	0,0146	0,0186	0,5959
balance	0,7813	0,8399	0,8624
bands	0,4211	0,3680	0,7126
breast	0,5652	0,7080	0,7033
car	0,5323	0,7842	0,8183
cleveland	0,5412	0,5388	0,5982
contraceptive	0,4291	0,4352	0,5092
dermatology	0,3060	0,3103	0,3060
ecoli	0,4256	0,4256	0,7091
flare	0,3105	0,3105	0,7279
glass	0,3290	0,3355	0,6253
haberman	0,3131	0,6016	0,7482
heart	0,5556	0,5556	0,8481
hepatitis	0,2063	0,0450	0,8183
horse-colic	0,6305	0,0000	0,8258
housevotes	0,3862	0,4706	0,8388
ionosphere	0,6410	0,6410	0,8661
iris	0,3333	0,3333	0,4000
led7digit	0,0900	0,0900	0,7420
lymphography	0,0133	0,0133	0,8600
mammographic	0,5609	0,5146	0,8252
monk-2	0,5279	0,4734	0,7703
movement_libras	0,0667	0,0667	0,0667
newthyroid	0,1396	0,1396	0,1396
pima	0,6615	0,7293	0,7710
saheart	0,6537	0,6755	0,7250
segment	0,1429	0,1429	0,1429
sonar	0,4662	0,4662	0,7540
spectfheart	0,2058	0,1614	0,7494
splice	0,3078	0,6749	0,9248
tic-tac-toe	0,5191	0,6544	0,7066
vehicle	0,5825	0,7091	0,7813
vowel	0,1273	0,5778	0,5960
wdbc	0,3726	0,3726	0,3726
wine	0,3317	0,3317	0,3317
winequality-red	0,0000	0,0000	0,0000
wisconsin	0,6552	0,6501	0,6552
yeast	0,1644	0,1644	0,5910
zoo	0,4189	0,4189	0,4189
Promedio	0,3883	0,4138	0,6327

Tabla B.31: Precisiones logradas (con PART) por Explore, SSMA y ZhangTS.

Bases de datos	Explore	SSMA	ZhangTS
appendicitis	0,8018	0,8018	0,8227
automobile	0,3263	0,3486	0,4182
balance	0,5357	0,6447	0,6324
bands	0,4378	0,5981	0,5770
breast	0,5652	0,6497	0,7033
car	0,7002	0,7002	0,7002
cleveland	0,5412	0,5491	0,5412
contraceptive	0,4481	0,4365	0,4291
dermatology	0,2977	0,2782	0,2814
ecoli	0,3599	0,2766	0,4468
flare	0,3107	0,2599	0,3105
glass	0,3561	0,3686	0,4650
haberman	0,5935	0,7353	0,7353
heart	0,5556	0,5926	0,5778
hepatitis	0,7338	0,3242	0,7938
horse-colic	0,6305	0,0189	0,6413
housevotes	0,5683	0,4759	0,6320
ionosphere	0,4418	0,4523	0,7921
iris	0,3333	0,3333	0,3333
led7digit	0,1120	0,1260	0,2120
lymphography	0,5490	0,5904	0,6510
mammographic	0,5515	0,6256	0,7711
monk-2	0,4961	0,4708	0,5266
movement_libras	0,0750	0,0917	0,1500
newthyroid	0,3104	0,2788	0,3084
pima	0,6224	0,6393	0,6511
saheart	0,5983	0,4825	0,6494
segment	0,2472	0,2169	0,2385
sonar	0,5862	0,5438	0,6052
spectfheart	0,7942	0,6571	0,7942
splice	0,4837	0,4740	0,5172
tic-tac-toe	0,5170	0,5993	0,6535
vehicle	0,3973	0,3725	0,3785
vowel	0,1343	0,1293	0,1162
wdbc	0,3976	0,5498	0,3989
wine	0,3935	0,4039	0,3428
winequality-red	0,4821	0,4984	0,4584
wisconsin	0,5295	0,5924	0,6552
yeast	0,3079	0,3120	0,3221
zoo	0,3422	0,2781	0,8614
Promedio	0,4616	0,4444	0,5274

Tabla B.32: Promedio de los balances logrados por los algoritmos genéticos a distintos valores del parámetro b y varios clasificadores.

Clasificador	Método \ b	0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9
C4.5	CHC	0,7006	0,7133	0,7249	0,7394	0,7661	0,7766	0,8111	0,8439	0,8888	0,9388
	GGA	0,7439	0,7327	0,7436	0,7651	0,7873	0,8042	0,8261	0,8569	0,8949	0,9388
	PBIL	0,7597	0,7325	0,7487	0,7653	0,7796	0,8003	0,8246	0,8567	0,8933	0,9400
	SGA	0,7692	0,7352	0,7454	0,7610	0,7825	0,8048	0,8301	0,8605	0,8963	0,9413
KNN	CHC	0,7824	0,8011	0,8179	0,8362	0,8559	0,8776	0,8957	0,9194	0,9414	0,9656
	GGA	0,7812	0,7888	0,7985	0,8157	0,8320	0,8565	0,8815	0,9048	0,9324	0,9578
	PBIL	0,7891	0,7937	0,8071	0,8201	0,8369	0,8589	0,8837	0,9090	0,9347	0,9607
	SGA	0,7835	0,7895	0,8017	0,8149	0,8332	0,8581	0,8837	0,9100	0,9364	0,9630
LDA	CHC	0,5179	0,5308	0,5578	0,5922	0,6344	0,6846	0,7433	0,8032	0,8658	0,9294
	GGA	0,6316	0,5902	0,6093	0,6373	0,6671	0,7075	0,7530	0,8087	0,8660	0,9250
	PBIL	0,6623	0,5748	0,5978	0,6332	0,6683	0,7044	0,7540	0,8091	0,8653	0,9272
	SGA	0,6914	0,5761	0,6113	0,6400	0,6696	0,7083	0,7532	0,8084	0,8668	0,9277
PART	CHC	0,5191	0,5519	0,5837	0,6285	0,6755	0,7201	0,7740	0,8255	0,8805	0,9374
	GGA	0,5478	0,5664	0,6010	0,6360	0,6835	0,7165	0,7679	0,8190	0,8759	0,9318
	PBIL	0,5656	0,5629	0,5952	0,6343	0,6728	0,7216	0,7693	0,8218	0,8751	0,9346
	SGA	0,5632	0,5709	0,6012	0,6369	0,6782	0,7205	0,7730	0,8218	0,8775	0,9359

B.5. Comparación del desempeño del sistema con los de los métodos convencionales. Detalles de los test de Friedman y Holm

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 7,98.

Método	Ranking
CNN	3,05
FCNN	3,275
IB3	3,35
ICF	2,85
Crisp01	2,475

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Crisp01	0,013328	0,133283
2	FCNN vs Crisp01	0,023652	0,212865
3	CNN vs Crisp01	0,103876	0,831009
4	IB3 vs ICF	0,157299	1,101094
5	FCNN vs ICF	0,229332	1,375992
6	ICF vs Crisp01	0,288844	1,444222
7	CNN vs IB3	0,396144	1,584576
8	CNN vs FCNN	0,524518	1,584576
9	CNN vs ICF	0,571608	1,584576
10	FCNN vs IB3	0,832004	1,584576

Tabla B.33: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 1$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 31,04.

Método	Ranking
CNN	3,275
FCNN	3,45
IB3	3,525
ICF	2,925
Fuzzy01	1,825

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Fuzzy01	0,000002	0,000015
2	FCNN vs Fuzzy01	0,000004	0,000039
3	CNN vs Fuzzy01	0,000041	0,000329
4	ICF vs Fuzzy01	0,001863	0,01304
5	IB3 vs ICF	0,089686	0,538116
6	FCNN vs ICF	0,137564	0,687819
7	CNN vs ICF	0,322199	1,288795
8	CNN vs IB3	0,4795	1,4385
9	CNN vs FCNN	0,620618	1,4385
10	FCNN vs IB3	0,832004	1,4385

Tabla B.34: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 1$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 21,06.

Método	Ranking
DROP3	3,1
FCNN	3,375
IB3	3,6
NRMCS	2,8
Crisp01	2,125

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Crisp01	0,00003	0,000302
2	FCNN vs Crisp01	0,000407	0,003663
3	DROP3 vs Crisp01	0,005821	0,046565
4	IB3 vs NRMCS	0,023652	0,165561
5	NRMCS vs Crisp01	0,056238	0,337427
6	FCNN vs NRMCS	0,103876	0,519381
7	DROP3 vs IB3	0,157299	0,629197
8	DROP3 vs NRMCS	0,396144	1,188432
9	DROP3 vs FCNN	0,436677	1,188432
10	FCNN vs IB3	0,524518	1,188432

Tabla B.35: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 1$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 26,18.

Método	Ranking
DROP3	3,15
FCNN	3,425
IB3	3,6
NRMCS	2,85
Fuzzy01	1,975

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Fuzzy01	0,000004	0,000043
2	FCNN vs Fuzzy01	0,000041	0,00037
3	DROP3 vs Fuzzy01	0,000889	0,007114
4	NRMCS vs Fuzzy01	0,013328	0,093298
5	IB3 vs NRMCS	0,033895	0,203369
6	FCNN vs NRMCS	0,103876	0,519381
7	DROP3 vs IB3	0,203092	0,812367
8	DROP3 vs NRMCS	0,396144	1,188432
9	DROP3 vs FCNN	0,436677	1,188432
10	FCNN vs IB3	0,620618	1,188432

Tabla B.36: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 1$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 8,86.

Método	Ranking
CNN	3,275
FCNN	2,9
IB3	2,8
PSC	3,475
Crisp01	2,55

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PSC vs Crisp01	0,008889	0,08889
2	CNN vs Crisp01	0,040305	0,362745
3	IB3 vs PSC	0,056238	0,449902
4	FCNN vs PSC	0,103876	0,727133
5	CNN vs IB3	0,179109	1,074655
6	CNN vs FCNN	0,288844	1,444222
7	FCNN vs Crisp01	0,322199	1,444222
8	IB3 vs Crisp01	0,4795	1,444222
9	CNN vs PSC	0,571608	1,444222
10	FCNN vs IB3	0,777297	1,444222

Tabla B.37: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 1$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 16,3.

Método	Ranking
CNN	3,325
FCNN	3
IB3	2,9
PSC	3,55
Fuzzy01	2,225

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PSC vs Fuzzy01	0,000178	0,001785
2	CNN vs Fuzzy01	0,001863	0,016766
3	FCNN vs Fuzzy01	0,028377	0,227018
4	IB3 vs Fuzzy01	0,056238	0,393665
5	IB3 vs PSC	0,065992	0,395952
6	FCNN vs PSC	0,119795	0,598975
7	CNN vs IB3	0,229332	0,917328
8	CNN vs FCNN	0,357971	1,073912
9	CNN vs PSC	0,524518	1,073912
10	FCNN vs IB3	0,777297	1,073912

Tabla B.38: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 1$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 22,94.

Método	Ranking
CPruner	2,725
DROP3	3,15
IB3	3,5
ICF	3,525
Crisp01	2,1

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ICF vs Crisp01	0,000056	0,000557
2	IB3 vs Crisp01	0,000075	0,000675
3	DROP3 vs Crisp01	0,002979	0,023836
4	CPruner vs ICF	0,023652	0,165561
5	CPruner vs IB3	0,028377	0,170264
6	CPruner vs Crisp01	0,0771	0,385499
7	CPruner vs DROP3	0,229332	0,917328
8	DROP3 vs ICF	0,288844	0,917328
9	DROP3 vs IB3	0,322199	0,917328
10	IB3 vs ICF	0,943628	0,943628

Tabla B.39: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 1$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 32,86.

Método	Ranking
CPruner	2,775
DROP3	3,275
IB3	3,55
ICF	3,55
Fuzzy01	1,85

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Fuzzy01	0,000002	0,000015
2	ICF vs Fuzzy01	0,000002	0,000015
3	DROP3 vs Fuzzy01	0,000056	0,000445
4	CPruner vs Fuzzy01	0,008889	0,062223
5	CPruner vs IB3	0,028377	0,170264
6	CPruner vs ICF	0,028377	0,170264
7	CPruner vs DROP3	0,157299	0,629197
8	DROP3 vs IB3	0,436677	1,31003
9	DROP3 vs ICF	0,436677	1,31003
10	IB3 vs ICF	1	1,31003

Tabla B.40: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 1$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 10,54.

Método	Ranking
CNN	3,625
DROP3	2,8
IB3	3,05
ICF	3
Crisp02	2,525

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs Crisp02	0,001863	0,018628
2	CNN vs DROP3	0,019624	0,17662
3	CNN vs ICF	0,0771	0,616799
4	CNN vs IB3	0,103876	0,727133
5	IB3 vs Crisp02	0,137564	0,825383
6	ICF vs Crisp02	0,179109	0,895546
7	DROP3 vs Crisp02	0,436677	1,746707
8	DROP3 vs IB3	0,4795	1,746707
9	DROP3 vs ICF	0,571608	1,746707
10	IB3 vs ICF	0,887537	1,746707

Tabla B.41: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 2$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 24,8.

Método	Ranking
CNN	3,75
DROP3	2,975
IB3	3,175
ICF	3,075
Fuzzy02	2,025

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs Fuzzy02	0,000001	0,000011
2	IB3 vs Fuzzy02	0,001143	0,010289
3	ICF vs Fuzzy02	0,002979	0,023836
4	DROP3 vs Fuzzy02	0,00721	0,050467
5	CNN vs DROP3	0,028377	0,170264
6	CNN vs ICF	0,056238	0,281189
7	CNN vs IB3	0,103876	0,415505
8	DROP3 vs IB3	0,571608	1,714823
9	DROP3 vs ICF	0,777297	1,714823
10	IB3 vs ICF	0,777297	1,714823

Tabla B.42: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 2$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 23,26.

Método	Ranking
CPruner	3,35
DROP3	3,25
IB3	3,675
NRMCS	2,375
Crisp02	2,35

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Crisp02	0,000178	0,001785
2	IB3 vs NRMCS	0,000236	0,002124
3	CPruner vs Crisp02	0,004678	0,037422
4	CPruner vs NRMCS	0,005821	0,040745
5	DROP3 vs Crisp02	0,010909	0,065457
6	DROP3 vs NRMCS	0,013328	0,066642
7	DROP3 vs IB3	0,229332	0,917328
8	CPruner vs IB3	0,357971	1,073912
9	CPruner vs DROP3	0,777297	1,554595
10	NRMCS vs Crisp02	0,943628	1,554595

Tabla B.43: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 2$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 31,1.

Método	Ranking
CPruner	3,475
DROP3	3,275
IB3	3,7
NRMCS	2,5
Fuzzy02	2,05

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Fuzzy02	0,000003	0,000031
2	CPruner vs Fuzzy02	0,000056	0,000501
3	DROP3 vs Fuzzy02	0,000531	0,004245
4	IB3 vs NRMCS	0,000689	0,00482
5	CPruner vs NRMCS	0,005821	0,034924
6	DROP3 vs NRMCS	0,028377	0,141886
7	NRMCS vs Fuzzy02	0,203092	0,812367
8	DROP3 vs IB3	0,229332	0,812367
9	CPruner vs IB3	0,524518	1,049037
10	CPruner vs DROP3	0,571608	1,049037

Tabla B.44: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 2$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 18,18.

Método	Ranking
CNN	3,525
FCNN	3,075
IB3	2,7
PSC	3,45
Crisp02	2,25

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs Crisp02	0,000311	0,003107
2	PSC vs Crisp02	0,000689	0,006197
3	FCNN vs Crisp02	0,019624	0,156995
4	CNN vs IB3	0,019624	0,156995
5	IB3 vs PSC	0,033895	0,203369
6	IB3 vs Crisp02	0,203092	1,015459
7	CNN vs FCNN	0,203092	1,015459
8	FCNN vs IB3	0,288844	1,015459
9	FCNN vs PSC	0,288844	1,015459
10	CNN vs PSC	0,832004	1,015459

Tabla B.45: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 2$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 19,56.

Método	Ranking
CNN	3,525
FCNN	3,075
IB3	2,725
PSC	3,475
Fuzzy02	2,2

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs Fuzzy02	0,000178	0,001785
2	PSC vs Fuzzy02	0,000311	0,002796
3	FCNN vs Fuzzy02	0,013328	0,106627
4	CNN vs IB3	0,023652	0,165561
5	IB3 vs PSC	0,033895	0,203369
6	IB3 vs Fuzzy02	0,137564	0,687819
7	CNN vs FCNN	0,203092	0,812367
8	FCNN vs PSC	0,257899	0,812367
9	FCNN vs IB3	0,322199	0,812367
10	CNN vs PSC	0,887537	0,887537

Tabla B.46: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 2$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 28,18.

Método	Ranking
CPruner	2,4
DROP3	3,2
IB3	3,575
ICF	3,625
Crisp02	2,2

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ICF vs Crisp02	0,000056	0,000557
2	IB3 vs Crisp02	0,000101	0,000906
3	CPruner vs ICF	0,000531	0,004245
4	CPruner vs IB3	0,000889	0,006225
5	DROP3 vs Crisp02	0,004678	0,028066
6	CPruner vs DROP3	0,023652	0,118258
7	DROP3 vs ICF	0,229332	0,917328
8	DROP3 vs IB3	0,288844	0,917328
9	CPruner vs Crisp02	0,571608	1,143215
10	IB3 vs ICF	0,887537	1,143215

Tabla B.47: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 2$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 42,66.

Método	Ranking
CPruner	2,5
DROP3	3,3
IB3	3,675
ICF	3,7
Fuzzy02	1,825

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ICF vs Fuzzy02	0	0,000001
2	IB3 vs Fuzzy02	0	0,000002
3	DROP3 vs Fuzzy02	0,00003	0,000242
4	CPruner vs ICF	0,000689	0,00482
5	CPruner vs IB3	0,000889	0,005336
6	CPruner vs DROP3	0,023652	0,118258
7	CPruner vs Fuzzy02	0,056238	0,224951
8	DROP3 vs ICF	0,257899	0,773697
9	DROP3 vs IB3	0,288844	0,773697
10	IB3 vs ICF	0,943628	0,943628

Tabla B.48: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 2$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 12,715.

Método	Ranking
CPruner	2,8625
DROP3	3,0125
IB3	3,4
ICF	3,4
Crisp03	2,325

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Crisp03	0,002361	0,023614
2	ICF vs Crisp03	0,002361	0,023614
3	DROP3 vs Crisp03	0,05183	0,414639
4	CPruner vs IB3	0,128441	0,899085
5	CPruner vs ICF	0,128441	0,899085
6	CPruner vs Crisp03	0,128441	0,899085
7	DROP3 vs IB3	0,273072	1,092288
8	DROP3 vs ICF	0,273072	1,092288
9	CPruner vs DROP3	0,671373	1,342746
10	IB3 vs ICF	1	1,342746

Tabla B.49: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 3$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 21,145.

Método	Ranking
CPruner	2,9625
DROP3	3,0875
IB3	3,5
ICF	3,4
Fuzzy03	2,05

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	IB3 vs Fuzzy03	0,000041	0,000411
2	ICF vs Fuzzy03	0,000134	0,001209
3	DROP3 vs Fuzzy03	0,003341	0,026727
4	CPruner vs Fuzzy03	0,009853	0,068972
5	CPruner vs IB3	0,128441	0,770644
6	CPruner vs ICF	0,215925	1,079625
7	DROP3 vs IB3	0,243321	1,079625
8	DROP3 vs ICF	0,376759	1,130277
9	CPruner vs DROP3	0,723674	1,447347
10	IB3 vs ICF	0,777297	1,447347

Tabla B.50: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 3$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 22,72.

Método	Ranking
CPruner	3,475
DROP3	3,625
CCIS	3,125
NRMCS	2,25
Crisp03	2,525

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs NRMCS	0,000101	0,001006
2	CPruner vs NRMCS	0,000531	0,004775
3	DROP3 vs Crisp03	0,001863	0,014903
4	CPruner vs Crisp03	0,00721	0,050467
5	CCIS vs NRMCS	0,013328	0,07997
6	CCIS vs Crisp03	0,089686	0,44843
7	DROP3 vs CCIS	0,157299	0,629197
8	CPruner vs CCIS	0,322199	0,966596
9	NRMCS vs Crisp03	0,436677	0,966596
10	CPruner vs DROP3	0,671373	0,966596

Tabla B.51: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 3$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 26,24.

Método	Ranking
CPruner	3,525
DROP3	3,65
CCIS	3,175
NRMCS	2,325
Fuzzy03	2,325

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs Fuzzy03	0,000178	0,001785
2	DROP3 vs NRMCS	0,000178	0,001785
3	CPruner vs Fuzzy03	0,000689	0,005508
4	CPruner vs NRMCS	0,000689	0,005508
5	CCIS vs Fuzzy03	0,01621	0,097257
6	CCIS vs NRMCS	0,01621	0,097257
7	DROP3 vs CCIS	0,179109	0,716437
8	CPruner vs CCIS	0,322199	0,966596
9	CPruner vs DROP3	0,723674	1,447347
10	NRMCS vs Fuzzy03	1	1,447347

Tabla B.52: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 3$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 25,8.

Método	Ranking
CNN	3,825
FCNN	3,125
IB3	2,525
PSC	3,3
Crisp03	2,225

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs Crisp03	0,000006	0,00006
2	CNN vs IB3	0,000236	0,002124
3	PSC vs Crisp03	0,002361	0,018891
4	FCNN vs Crisp03	0,010909	0,076366
5	IB3 vs PSC	0,028377	0,170264
6	CNN vs FCNN	0,047715	0,238574
7	FCNN vs IB3	0,089686	0,358744
8	CNN vs PSC	0,137564	0,412692
9	IB3 vs Crisp03	0,396144	0,792288
10	FCNN vs PSC	0,620618	0,792288

Tabla B.53: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 3$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 25,52.

Método	Ranking
CNN	3,8
FCNN	3,15
IB3	2,55
PSC	3,3
Fuzzy03	2,2

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CNN vs Fuzzy03	0,000006	0,00006
2	CNN vs IB3	0,000407	0,003663
3	PSC vs Fuzzy03	0,001863	0,014903
4	FCNN vs Fuzzy03	0,00721	0,050467
5	IB3 vs PSC	0,033895	0,203369
6	CNN vs FCNN	0,065992	0,32996
7	FCNN vs IB3	0,089686	0,358744
8	CNN vs PSC	0,157299	0,471898
9	IB3 vs Fuzzy03	0,322199	0,644398
10	FCNN vs PSC	0,671373	0,671373

Tabla B.54: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 3$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 15,985.

Método	Ranking
CPruner	2,6125
DROP3	3,675
CCIS	3,3875
NRMCS	2,75
Crisp03	2,575

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs Crisp03	0,001863	0,018628
2	CPruner vs DROP3	0,002654	0,023886
3	DROP3 vs NRMCS	0,008889	0,071112
4	CCIS vs Crisp03	0,021556	0,150894
5	CPruner vs CCIS	0,028377	0,170264
6	CCIS vs NRMCS	0,071369	0,356845
7	DROP3 vs CCIS	0,416119	1,664476
8	NRMCS vs Crisp03	0,620618	1,861854
9	CPruner vs NRMCS	0,697344	1,861854
10	CPruner vs Crisp03	0,91553	1,861854

Tabla B.55: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 3$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 21,985.

Método	Ranking
CPruner	2,6875
DROP3	3,7
CCIS	3,4625
NRMCS	2,9
Fuzzy03	2,25

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs Fuzzy03	0,000041	0,000411
2	CCIS vs Fuzzy03	0,000605	0,005443
3	CPruner vs DROP3	0,004186	0,033489
4	DROP3 vs NRMCS	0,023652	0,165561
5	CPruner vs CCIS	0,028377	0,170264
6	NRMCS vs Fuzzy03	0,065992	0,32996
7	CCIS vs NRMCS	0,111612	0,446447
8	CPruner vs Fuzzy03	0,215925	0,647775
9	DROP3 vs CCIS	0,501742	1,003484
10	CPruner vs NRMCS	0,547813	1,003484

Tabla B.56: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 3$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 22,4.

Método	Ranking
CCIS	2,925
CPruner	2,725
DROP3	3,275
ICF	3,825
Crisp04	2,25

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ICF vs Crisp04	0,000008	0,000084
2	CPruner vs ICF	0,001863	0,016766
3	DROP3 vs Crisp04	0,003742	0,029935
4	CCIS vs ICF	0,010909	0,076366
5	CCIS vs Crisp04	0,056238	0,337427
6	CPruner vs DROP3	0,119795	0,598975
7	DROP3 vs ICF	0,119795	0,598975
8	CPruner vs Crisp04	0,179109	0,598975
9	CCIS vs DROP3	0,322199	0,644398
10	CCIS vs CPruner	0,571608	0,644398

Tabla B.57: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0,4$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 28,54.

Método	Ranking
CCIS	2,925
CPruner	2,825
DROP3	3,35
ICF	3,85
Fuzzy04	2,05

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	ICF vs Fuzzy04	0	0,000004
2	DROP3 vs Fuzzy04	0,000236	0,002124
3	CPruner vs ICF	0,003742	0,029935
4	CCIS vs ICF	0,008889	0,062223
5	CCIS vs Fuzzy04	0,013328	0,07997
6	CPruner vs Fuzzy04	0,028377	0,141886
7	CPruner vs DROP3	0,137564	0,550256
8	DROP3 vs ICF	0,157299	0,550256
9	CCIS vs DROP3	0,229332	0,550256
10	CCIS vs CPruner	0,777297	0,777297

Tabla B.58: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0,4$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 29,06.

Método	Ranking
CPruner	3,625
DROP3	3,675
CCIS	3
NRMCS	2,1
Crisp04	2,6

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs NRMCS	0,000008	0,000084
2	CPruner vs NRMCS	0,000016	0,000145
3	DROP3 vs Crisp04	0,002361	0,018891
4	CPruner vs Crisp04	0,003742	0,026193
5	CCIS vs NRMCS	0,010909	0,065457
6	DROP3 vs CCIS	0,056238	0,281189
7	CPruner vs CCIS	0,0771	0,308399
8	NRMCS vs Crisp04	0,157299	0,471898
9	CCIS vs Crisp04	0,257899	0,515798
10	CPruner vs DROP3	0,887537	0,887537

Tabla B.59: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 4$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 31,32.

Método	Ranking
CPruner	3,675
DROP3	3,7
CCIS	3,025
NRMCS	2,175
Fuzzy04	2,425

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs NRMCS	0,000016	0,000161
2	CPruner vs NRMCS	0,000022	0,000199
3	DROP3 vs Fuzzy04	0,000311	0,002485
4	CPruner vs Fuzzy04	0,000407	0,002849
5	CCIS vs NRMCS	0,01621	0,097257
6	DROP3 vs CCIS	0,056238	0,281189
7	CPruner vs CCIS	0,065992	0,281189
8	CCIS vs Fuzzy04	0,089686	0,281189
9	NRMCS vs Fuzzy04	0,4795	0,959
10	CPruner vs DROP3	0,943628	0,959

Tabla B.60: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 4$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 25,48.

Método	Ranking
DROP3	2,875
FCNN	3,675
IB3	2,875
PSC	3,5
Crisp04	2,075

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	FCNN vs Crisp04	0,000006	0,00006
2	PSC vs Crisp04	0,000056	0,000501
3	DROP3 vs Crisp04	0,023652	0,189213
4	FCNN vs IB3	0,023652	0,189213
5	IB3 vs Crisp04	0,023652	0,189213
6	DROP3 vs FCNN	0,023652	0,189213
7	IB3 vs PSC	0,0771	0,308399
8	DROP3 vs PSC	0,0771	0,308399
9	FCNN vs PSC	0,620618	1,241236
10	DROP3 vs IB3	1	1,241236

Tabla B.61: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 4$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 31,9.

Método	Ranking
DROP3	2,95
FCNN	3,725
IB3	2,925
PSC	3,5
Fuzzy04	1,9

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	FCNN vs Fuzzy04	0	0,000002
2	PSC vs Fuzzy04	0,000006	0,000054
3	DROP3 vs Fuzzy04	0,002979	0,023836
4	IB3 vs Fuzzy04	0,003742	0,026193
5	FCNN vs IB3	0,023652	0,14191
6	DROP3 vs FCNN	0,028377	0,14191
7	IB3 vs PSC	0,103876	0,415505
8	DROP3 vs PSC	0,119795	0,415505
9	FCNN vs PSC	0,524518	1,049037
10	DROP3 vs IB3	0,943628	1,049037

Tabla B.62: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 4$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 18,56.

Método	Ranking
CPruner	2,675
DROP3	3,775
CCIS	3,35
NRMCS	2,525
Crisp04	2,675

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs NRMCS	0,000407	0,00407
2	DROP3 vs Crisp04	0,001863	0,016766
3	CPruner vs DROP3	0,001863	0,016766
4	CCIS vs NRMCS	0,019624	0,137371
5	CCIS vs Crisp04	0,056238	0,337427
6	CPruner vs CCIS	0,056238	0,337427
7	DROP3 vs CCIS	0,229332	0,917328
8	CPruner vs NRMCS	0,671373	2,01412
9	NRMCS vs Crisp04	0,671373	2,01412
10	CPruner vs Crisp04	1	2,01412

Tabla B.63: Comparación de los balances obtenidos por algunos métodos convencionales y la versión crisp del sistema con $b = 0, 4$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 23,84.

Método	Ranking
CPruner	2,725
DROP3	3,85
CCIS	3,425
NRMCS	2,575
Fuzzy04	2,425

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	DROP3 vs Fuzzy04	0,000056	0,000557
2	DROP3 vs NRMCS	0,000311	0,002796
3	CPruner vs DROP3	0,001463	0,011702
4	CCIS vs Fuzzy04	0,004678	0,032744
5	CCIS vs NRMCS	0,01621	0,097257
6	CPruner vs CCIS	0,047715	0,238574
7	DROP3 vs CCIS	0,229332	0,917328
8	CPruner vs Fuzzy04	0,396144	1,188432
9	CPruner vs NRMCS	0,671373	1,342746
10	NRMCS vs Fuzzy04	0,671373	1,342746

Tabla B.64: Comparación de los balances obtenidos por algunos métodos convencionales y la versión difusa del sistema con $b = 0, 4$ y el clasificador PART. Tests de Friedman y Holm.

B.6. Comparación del desempeño del sistema con los de los algoritmos genéticos. Detalles de los test de Friedman y Holm

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 16,42.

Método	Ranking
CHC	3,875
GGA	2,95
PBIL	2,725
SGA	2,85
Crisp01	2,6

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Crisp01	0,000311	0,003107
2	CHC vs PBIL	0,001143	0,010289
3	CHC vs SGA	0,003742	0,029935
4	CHC vs GGA	0,008889	0,062223
5	GGA vs Crisp01	0,322199	1,933193
6	SGA vs Crisp01	0,4795	2,397501
7	GGA vs PBIL	0,524518	2,397501
8	PBIL vs SGA	0,723674	2,397501
9	PBIL vs Crisp01	0,723674	2,397501
10	GGA vs SGA	0,777297	2,397501

Tabla B.65: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 1$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 21,62.

Método	Ranking
CHC	3,85
GGA	3,075
PBIL	2,85
SGA	3
Fuzzy01	2,225

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Fuzzy01	0,000004	0,000043
2	CHC vs PBIL	0,004678	0,0421
3	CHC vs SGA	0,01621	0,129676
4	GGA vs Fuzzy01	0,01621	0,129676
5	CHC vs GGA	0,028377	0,170264
6	SGA vs Fuzzy01	0,028377	0,170264
7	PBIL vs Fuzzy01	0,0771	0,308399
8	GGA vs PBIL	0,524518	1,573555
9	PBIL vs SGA	0,671373	1,573555
10	GGA vs SGA	0,832004	1,573555

Tabla B.66: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 1$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 20,02.

Método	Ranking
CHC	2,1
GGA	3,325
PBIL	2,95
SGA	3,05
Crisp01	3,575

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Crisp01	0,00003	0,000302
2	CHC vs GGA	0,000531	0,004775
3	CHC vs SGA	0,00721	0,057677
4	CHC vs PBIL	0,01621	0,113467
5	PBIL vs Crisp01	0,0771	0,462599
6	SGA vs Crisp01	0,137564	0,687819
7	GGA vs PBIL	0,288844	1,155377
8	GGA vs SGA	0,436677	1,31003
9	GGA vs Crisp01	0,4795	1,31003
10	PBIL vs SGA	0,777297	1,31003

Tabla B.67: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 1$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 13,28.

Método	Ranking
CHC	2,225
GGA	3,425
PBIL	3,05
SGA	3,175
Fuzzy01	3,125

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs GGA	0,000689	0,006885
2	CHC vs SGA	0,00721	0,064886
3	CHC vs Fuzzy01	0,010909	0,087276
4	CHC vs PBIL	0,019624	0,137371
5	GGA vs PBIL	0,288844	1,733066
6	GGA vs Fuzzy01	0,396144	1,98072
7	GGA vs SGA	0,4795	1,98072
8	PBIL vs SGA	0,723674	2,171021
9	PBIL vs Fuzzy01	0,832004	2,171021
10	SGA vs Fuzzy01	0,887537	2,171021

Tabla B.68: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 1$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 15,06.

Método	Ranking
CHC	3,6
GGA	3,05
PBIL	2,975
SGA	3,125
Crisp01	2,25

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Crisp01	0,000134	0,001343
2	SGA vs Crisp01	0,013328	0,119955
3	GGA vs Crisp01	0,023652	0,189213
4	PBIL vs Crisp01	0,040305	0,282135
5	CHC vs PBIL	0,0771	0,462599
6	CHC vs GGA	0,119795	0,598975
7	CHC vs SGA	0,179109	0,716437
8	PBIL vs SGA	0,671373	2,01412
9	GGA vs SGA	0,832004	2,01412
10	GGA vs PBIL	0,832004	2,01412

Tabla B.69: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 1$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 30,72.

Método	Ranking
CHC	3,725
GGA	3,125
PBIL	3,075
SGA	3,225
Fuzzy01	1,85

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Fuzzy01	0	0,000001
2	SGA vs Fuzzy01	0,000101	0,000906
3	GGA vs Fuzzy01	0,000311	0,002485
4	PBIL vs Fuzzy01	0,000531	0,003714
5	CHC vs PBIL	0,065992	0,395952
6	CHC vs GGA	0,089686	0,44843
7	CHC vs SGA	0,157299	0,629197
8	PBIL vs SGA	0,671373	2,01412
9	GGA vs SGA	0,777297	2,01412
10	GGA vs PBIL	0,887537	2,01412

Tabla B.70: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 1$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 8,38.

Método	Ranking
CHC	3,175
GGA	3,1
PBIL	3,35
SGA	2,975
Crisp01	2,4

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Crisp01	0,00721	0,072096
2	CHC vs Crisp01	0,028377	0,255395
3	GGA vs Crisp01	0,047715	0,381719
4	SGA vs Crisp01	0,103876	0,727133
5	PBIL vs SGA	0,288844	1,733066
6	GGA vs PBIL	0,4795	2,397501
7	CHC vs SGA	0,571608	2,397501
8	CHC vs PBIL	0,620618	2,397501
9	GGA vs SGA	0,723674	2,397501
10	CHC vs GGA	0,832004	2,397501

Tabla B.71: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0, 1$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 20,46.

Método	Ranking
CHC	3,325
GGA	3,25
PBIL	3,4
SGA	3
Fuzzy01	2,025

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Fuzzy01	0,000101	0,001006
2	CHC vs Fuzzy01	0,000236	0,002124
3	GGA vs Fuzzy01	0,000531	0,004245
4	SGA vs Fuzzy01	0,005821	0,040745
5	PBIL vs SGA	0,257899	1,547394
6	CHC vs SGA	0,357971	1,789853
7	GGA vs SGA	0,4795	1,918
8	GGA vs PBIL	0,671373	2,01412
9	CHC vs PBIL	0,832004	2,01412
10	CHC vs GGA	0,832004	2,01412

Tabla B.72: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0, 1$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 8,36.

Método	Ranking
CHC	3,525
GGA	3,125
PBIL	2,625
SGA	2,7
Crisp02	3,025

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs PBIL	0,010909	0,109095
2	CHC vs SGA	0,019624	0,17662
3	GGA vs PBIL	0,157299	1,258394
4	CHC vs Crisp02	0,157299	1,258394
5	GGA vs SGA	0,229332	1,375992
6	PBIL vs Crisp02	0,257899	1,375992
7	CHC vs GGA	0,257899	1,375992
8	SGA vs Crisp02	0,357971	1,375992
9	GGA vs Crisp02	0,777297	1,554595
10	PBIL vs SGA	0,832004	1,554595

Tabla B.73: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,2$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 16,08.

Método	Ranking
CHC	3,7
GGA	3,25
PBIL	2,7
SGA	2,95
Fuzzy02	2,4

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Fuzzy02	0,000236	0,00236
2	CHC vs PBIL	0,004678	0,0421
3	GGA vs Fuzzy02	0,01621	0,129676
4	CHC vs SGA	0,033895	0,237264
5	SGA vs Fuzzy02	0,119795	0,71877
6	GGA vs PBIL	0,119795	0,71877
7	CHC vs GGA	0,203092	0,812367
8	PBIL vs Fuzzy02	0,396144	1,188432
9	GGA vs SGA	0,396144	1,188432
10	PBIL vs SGA	0,4795	1,188432

Tabla B.74: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,2$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 39,5.

Método	Ranking
CHC	1,8
GGA	3,65
PBIL	2,625
SGA	3,325
Crisp02	3,6

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs GGA	0	0,000002
2	CHC vs Crisp02	0	0,000003
3	CHC vs SGA	0,000016	0,000129
4	GGA vs PBIL	0,003742	0,026193
5	PBIL vs Crisp02	0,005821	0,034924
6	CHC vs PBIL	0,019624	0,098122
7	PBIL vs SGA	0,047715	0,19086
8	GGA vs SGA	0,357971	1,073912
9	SGA vs Crisp02	0,436677	1,073912
10	GGA vs Crisp02	0,887537	1,073912

Tabla B.75: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,2$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 35,14.

Método	Ranking
CHC	1,85
GGA	3,775
PBIL	2,75
SGA	3,4
Fuzzy02	3,225

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs GGA	0	0,000001
2	CHC vs SGA	0,000012	0,000105
3	CHC vs Fuzzy02	0,000101	0,000805
4	GGA vs PBIL	0,003742	0,026193
5	CHC vs PBIL	0,010909	0,065457
6	PBIL vs SGA	0,065992	0,32996
7	GGA vs Fuzzy02	0,119795	0,47918
8	PBIL vs Fuzzy02	0,179109	0,537328
9	GGA vs SGA	0,288844	0,577689
10	SGA vs Fuzzy02	0,620618	0,620618

Tabla B.76: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,2$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 17,46.

Método	Ranking
CHC	3,325
GGA	3,075
PBIL	3,4
SGA	3,1
Crisp02	2,1

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Crisp02	0,000236	0,00236
2	CHC vs Crisp02	0,000531	0,004775
3	SGA vs Crisp02	0,004678	0,037422
4	GGA vs Crisp02	0,005821	0,040745
5	GGA vs PBIL	0,357971	2,147824
6	PBIL vs SGA	0,396144	2,147824
7	CHC vs GGA	0,4795	2,147824
8	CHC vs SGA	0,524518	2,147824
9	CHC vs PBIL	0,832004	2,147824
10	GGA vs SGA	0,943628	2,147824

Tabla B.77: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,2$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 29,54.

Método	Ranking
CHC	3,35
GGA	3,1
PBIL	3,55
SGA	3,175
Fuzzy02	1,825

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Fuzzy02	0,000001	0,000011
2	CHC vs Fuzzy02	0,000016	0,000145
3	SGA vs Fuzzy02	0,000134	0,001075
4	GGA vs Fuzzy02	0,000311	0,002175
5	GGA vs PBIL	0,203092	1,218551
6	PBIL vs SGA	0,288844	1,444222
7	CHC vs GGA	0,4795	1,918
8	CHC vs PBIL	0,571608	1,918
9	CHC vs SGA	0,620618	1,918
10	GGA vs SGA	0,832004	1,918

Tabla B.78: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,2$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 7,1.

Método	Ranking
CHC	3,075
GGA	2,9
PBIL	3,3
SGA	3,25
Crisp02	2,475

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Crisp02	0,019624	0,196244
2	SGA vs Crisp02	0,028377	0,255395
3	CHC vs Crisp02	0,089686	0,717488
4	GGA vs Crisp02	0,229332	1,605324
5	GGA vs PBIL	0,257899	1,605324
6	GGA vs SGA	0,322199	1,610994
7	CHC vs PBIL	0,524518	2,098073
8	CHC vs SGA	0,620618	2,098073
9	CHC vs GGA	0,620618	2,098073
10	PBIL vs SGA	0,887537	2,098073

Tabla B.79: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,2$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 19,3.

Método	Ranking
CHC	3,175
GGA	3,05
PBIL	3,425
SGA	3,3
Fuzzy02	2,05

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Fuzzy02	0,000101	0,001006
2	SGA vs Fuzzy02	0,000407	0,003663
3	CHC vs Fuzzy02	0,001463	0,011702
4	GGA vs Fuzzy02	0,004678	0,032744
5	GGA vs PBIL	0,288844	1,733066
6	CHC vs PBIL	0,4795	2,397501
7	GGA vs SGA	0,4795	2,397501
8	CHC vs SGA	0,723674	2,397501
9	PBIL vs SGA	0,723674	2,397501
10	CHC vs GGA	0,723674	2,397501

Tabla B.80: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,2$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 7,3.

Método	Ranking
CHC	3,55
GGA	2,65
PBIL	2,825
SGA	3
Crisp03	2,975

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs GGA	0,010909	0,109095
2	CHC vs PBIL	0,040305	0,362745
3	CHC vs Crisp03	0,103876	0,831009
4	CHC vs SGA	0,119795	0,838565
5	GGA vs SGA	0,322199	1,933193
6	GGA vs Crisp03	0,357971	1,933193
7	GGA vs PBIL	0,620618	2,482472
8	PBIL vs SGA	0,620618	2,482472
9	PBIL vs Crisp03	0,671373	2,482472
10	SGA vs Crisp03	0,943628	2,482472

Tabla B.81: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,3$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 10,92.

Método	Ranking
CHC	3,6
GGA	2,825
PBIL	2,975
SGA	3,125
Fuzzy03	2,475

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Fuzzy03	0,001463	0,014627
2	CHC vs GGA	0,028377	0,255395
3	SGA vs Fuzzy03	0,065992	0,527936
4	CHC vs PBIL	0,0771	0,539699
5	PBIL vs Fuzzy03	0,157299	0,943795
6	CHC vs SGA	0,179109	0,943795
7	GGA vs Fuzzy03	0,322199	1,288795
8	GGA vs SGA	0,396144	1,288795
9	GGA vs PBIL	0,671373	1,342746
10	PBIL vs SGA	0,671373	1,342746

Tabla B.82: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,3$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 38,46.

Método	Ranking
CHC	1,75
GGA	3,15
PBIL	2,875
SGA	3,45
Crisp03	3,775

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Crisp03	0	0
2	CHC vs SGA	0,000002	0,000014
3	CHC vs GGA	0,000075	0,0006
4	CHC vs PBIL	0,001463	0,010239
5	PBIL vs Crisp03	0,010909	0,065457
6	GGA vs Crisp03	0,0771	0,385499
7	PBIL vs SGA	0,103876	0,415505
8	SGA vs Crisp03	0,357971	1,073912
9	GGA vs SGA	0,396144	1,073912
10	GGA vs PBIL	0,436677	1,073912

Tabla B.83: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,3$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 30,74.

Método	Ranking
CHC	1,825
GGA	3,25
PBIL	2,95
SGA	3,5
Fuzzy03	3,475

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs SGA	0,000002	0,000022
2	CHC vs Fuzzy03	0,000003	0,000028
3	CHC vs GGA	0,000056	0,000445
4	CHC vs PBIL	0,001463	0,010239
5	PBIL vs SGA	0,119795	0,71877
6	PBIL vs Fuzzy03	0,137564	0,71877
7	GGA vs PBIL	0,396144	1,584576
8	GGA vs SGA	0,4795	1,584576
9	GGA vs Fuzzy03	0,524518	1,584576
10	SGA vs Fuzzy03	0,943628	1,584576

Tabla B.84: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,3$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 12,14.

Método	Ranking
CHC	3,05
GGA	3,175
PBIL	3,15
SGA	3,375
Crisp03	2,25

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	SGA vs Crisp03	0,001463	0,014627
2	GGA vs Crisp03	0,008889	0,080001
3	PBIL vs Crisp03	0,010909	0,087276
4	CHC vs Crisp03	0,023652	0,165561
5	CHC vs SGA	0,357971	2,147824
6	PBIL vs SGA	0,524518	2,622591
7	GGA vs SGA	0,571608	2,622591
8	CHC vs GGA	0,723674	2,622591
9	CHC vs PBIL	0,777297	2,622591
10	GGA vs PBIL	0,943628	2,622591

Tabla B.85: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,3$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 17,86.

Método	Ranking
CHC	3,1
GGA	3,2
PBIL	3,225
SGA	3,4
Fuzzy03	2,075

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	SGA vs Fuzzy03	0,000178	0,001785
2	PBIL vs Fuzzy03	0,001143	0,010289
3	GGA vs Fuzzy03	0,001463	0,011702
4	CHC vs Fuzzy03	0,003742	0,026193
5	CHC vs SGA	0,396144	2,376863
6	GGA vs SGA	0,571608	2,858038
7	PBIL vs SGA	0,620618	2,858038
8	CHC vs PBIL	0,723674	2,858038
9	CHC vs GGA	0,777297	2,858038
10	GGA vs PBIL	0,943628	2,858038

Tabla B.86: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,3$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 5,3.

Método	Ranking
CHC	2,775
GGA	3,1
PBIL	3,3
SGA	3,2
Crisp03	2,625

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Crisp03	0,056238	0,562378
2	SGA vs Crisp03	0,103876	0,934885
3	CHC vs PBIL	0,137564	1,100511
4	GGA vs Crisp03	0,179109	1,253764
5	CHC vs SGA	0,229332	1,375992
6	CHC vs GGA	0,357971	1,789853
7	GGA vs PBIL	0,571608	2,286431
8	CHC vs Crisp03	0,671373	2,286431
9	PBIL vs SGA	0,777297	2,286431
10	GGA vs SGA	0,777297	2,286431

Tabla B.87: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,3$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 17,36.

Método	Ranking
CHC	3
GGA	3,25
PBIL	3,35
SGA	3,3
Fuzzy03	2,1

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Fuzzy03	0,000407	0,00407
2	SGA vs Fuzzy03	0,000689	0,006197
3	GGA vs Fuzzy03	0,001143	0,009145
4	CHC vs Fuzzy03	0,010909	0,076366
5	CHC vs PBIL	0,322199	1,933193
6	CHC vs SGA	0,396144	1,98072
7	CHC vs GGA	0,4795	1,98072
8	GGA vs PBIL	0,777297	2,331892
9	GGA vs SGA	0,887537	2,331892
10	PBIL vs SGA	0,887537	2,331892

Tabla B.88: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,3$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 9,64.

Método	Ranking
CHC	3,4
GGA	2,375
PBIL	2,925
SGA	3,175
Crisp04	3,125

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs GGA	0,003742	0,037419
2	GGA vs SGA	0,023652	0,212865
3	GGA vs Crisp04	0,033895	0,271159
4	GGA vs PBIL	0,119795	0,838565
5	CHC vs PBIL	0,179109	1,074655
6	CHC vs Crisp04	0,436677	2,183383
7	PBIL vs SGA	0,4795	2,183383
8	CHC vs SGA	0,524518	2,183383
9	PBIL vs Crisp04	0,571608	2,183383
10	SGA vs Crisp04	0,887537	2,183383

Tabla B.89: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,4$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 10,88.

Método	Ranking
CHC	3,5
GGA	2,45
PBIL	3,05
SGA	3,25
Fuzzy04	2,75

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs GGA	0,002979	0,029795
2	GGA vs SGA	0,023652	0,212865
3	CHC vs Fuzzy04	0,033895	0,271159
4	GGA vs PBIL	0,089686	0,627802
5	SGA vs Fuzzy04	0,157299	0,943795
6	CHC vs PBIL	0,203092	1,015459
7	GGA vs Fuzzy04	0,396144	1,584576
8	PBIL vs Fuzzy04	0,396144	1,584576
9	CHC vs SGA	0,4795	1,584576
10	PBIL vs SGA	0,571608	1,584576

Tabla B.90: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,4$ y el clasificador C4.5. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 54,315.

Método	Ranking
CHC	1,4875
GGA	3,4
PBIL	2,925
SGA	3,25
Crisp04	3,9375

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Crisp04	0	0
2	CHC vs GGA	0	0,000001
3	CHC vs SGA	0,000001	0,000005
4	CHC vs PBIL	0,000048	0,000335
5	PBIL vs Crisp04	0,004186	0,025117
6	SGA vs Crisp04	0,05183	0,25915
7	GGA vs Crisp04	0,128441	0,513763
8	GGA vs PBIL	0,179109	0,537328
9	PBIL vs SGA	0,357971	0,715941
10	GGA vs SGA	0,671373	0,715941

Tabla B.91: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,4$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 48,42.

Método	Ranking
CHC	1,525
GGA	3,4
PBIL	3
SGA	3,3
Fuzzy04	3,775

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs Fuzzy04	0	0
2	CHC vs GGA	0	0,000001
3	CHC vs SGA	0,000001	0,000004
4	CHC vs PBIL	0,00003	0,000211
5	PBIL vs Fuzzy04	0,028377	0,170264
6	SGA vs Fuzzy04	0,179109	0,895546
7	GGA vs PBIL	0,257899	1,031596
8	GGA vs Fuzzy04	0,288844	1,031596
9	PBIL vs SGA	0,396144	1,031596
10	GGA vs SGA	0,777297	1,031596

Tabla B.92: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,4$ y el clasificador KNN. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 11,68.

Método	Ranking
CHC	2,975
GGA	3,175
PBIL	2,975
SGA	3,525
Crisp04	2,35

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	SGA vs Crisp04	0,000889	0,008893
2	GGA vs Crisp04	0,019624	0,17662
3	PBIL vs Crisp04	0,0771	0,616799
4	CHC vs Crisp04	0,0771	0,616799
5	CHC vs SGA	0,119795	0,71877
6	PBIL vs SGA	0,119795	0,71877
7	GGA vs SGA	0,322199	1,288795
8	CHC vs GGA	0,571608	1,714823
9	GGA vs PBIL	0,571608	1,714823
10	CHC vs PBIL	1	1,714823

Tabla B.93: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,4$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 13,86.

Método	Ranking
CHC	3
GGA	3,15
PBIL	3,075
SGA	3,525
Fuzzy04	2,25

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	SGA vs Fuzzy04	0,000311	0,003107
2	GGA vs Fuzzy04	0,010909	0,098185
3	PBIL vs Fuzzy04	0,019624	0,156995
4	CHC vs Fuzzy04	0,033895	0,237264
5	CHC vs SGA	0,137564	0,825383
6	PBIL vs SGA	0,203092	1,015459
7	GGA vs SGA	0,288844	1,155377
8	CHC vs GGA	0,671373	2,01412
9	CHC vs PBIL	0,832004	2,01412
10	GGA vs PBIL	0,832004	2,01412

Tabla B.94: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,4$ y el clasificador LDA. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 4,24.

Método	Ranking
CHC	2,675
GGA	2,775
PBIL	3,25
SGA	3,175
Crisp04	3,125

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	CHC vs PBIL	0,103876	1,038762
2	CHC vs SGA	0,157299	1,415693
3	GGA vs PBIL	0,179109	1,432874
4	CHC vs Crisp04	0,203092	1,432874
5	GGA vs SGA	0,257899	1,547394
6	GGA vs Crisp04	0,322199	1,610994
7	PBIL vs Crisp04	0,723674	2,894694
8	CHC vs GGA	0,777297	2,894694
9	PBIL vs SGA	0,832004	2,894694
10	SGA vs Crisp04	0,887537	2,894694

Tabla B.95: Comparación de los balances obtenidos por los algoritmos genéticos y la versión crisp del sistema con $b = 0,4$ y el clasificador PART. Tests de Friedman y Holm.

(a) Rankings promedios obtenidos al aplicar el Test de Friedman. Estadístico de Friedman: 10,06.

Método	Ranking
CHC	2,8
GGA	3
PBIL	3,425
SGA	3,325
Fuzzy04	2,45

(b) Valores de p ajustados según test de Holm.

i	Hipótesis	p no ajustado	p_{Holm}
1	PBIL vs Fuzzy04	0,005821	0,058207
2	SGA vs Fuzzy04	0,013328	0,119955
3	CHC vs PBIL	0,0771	0,616799
4	GGA vs Fuzzy04	0,119795	0,838565
5	CHC vs SGA	0,137564	0,838565
6	GGA vs PBIL	0,229332	1,14666
7	CHC vs Fuzzy04	0,322199	1,288795
8	GGA vs SGA	0,357971	1,288795
9	CHC vs GGA	0,571608	1,288795
10	PBIL vs SGA	0,777297	1,288795

Tabla B.96: Comparación de los balances obtenidos por los algoritmos genéticos y la versión difusa del sistema con $b = 0,4$ y el clasificador PART. Tests de Friedman y Holm.

Bibliografía

- [1] Aguilar J, Riquelme J, Toro M (2001) Data set editing by ordered projection. *Intelligent Data Analysis* 5(5):405-417
- [2] Aha DW (ed) (1997) *Lazy Learning*. Kluwer Academic Publishers, Norwell, MA
- [3] Aha DW, Kibler D, Albert MK (1991) Instance-based learning algorithms. *Mach Learn* 6(1):37-66
- [4] Alcalá R, Alcalá-Fdez J, Casillas J, Cerdón O, Herrera F (2007) Local identification of prototypes for genetic learning of accurate TSK fuzzy rule-based systems. *Int J of Intell Syst* 22(9):909-941
- [5] Alcalá-Fdez J, Sánchez L, García S, del Jesus MJ, Ventura S, Garrell JM, Otero J, Romero C, Bacardit J, Rivas VM, Fernández JC, Herrera F (2009) KEEL: A Software Tool to Assess Evolutionary Algorithms to Data Mining Problems. *Soft Computing* 13(3):307-318
- [6] Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2010) KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. *Journal of Multiple-Valued Logic and Soft Computing*, in press
- [7] Al-Harbi SH, Rayward-Smith VJ (2006) Adapting k-means for supervised clustering. *Applied Intelligence* 24:219-226
- [8] Ali S, Smith-Miles KA (2006) A meta-learning approach to automatic kernel selection for support vector machines. *Neurocomputing* 70:173-186
- [9] Angiulli F (2007) Fast nearest neighbor condensation for large data sets classification. *IEEE Transactions on Knowledge and Data Engineering* 19(11):1450-1464
- [10] Asimov D (1985) The grand tour: A tool for viewing multidimensional data, *SIAM Journal on Scientific and Statistical Computing* 6(1):128-143
- [11] Asuncion A, Newman DJ (2007) UCI Machine Learning Repository. Available via <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [12] Bernadó-Mansilla E, Llorá X, Garrel J (2002) XCS and gale: A comparative study of two learning classifier. In: Lanzi P, Stolzmann W, Wilson S (eds) *Advances in Learning Classifier Systems*, LNCS, vol 2321. Springer, Heidelberg, pp 115-132

- [13] Bernardó E, Ho TK (2004) On classifier domain of competence, In: proceedings of 17th International Conference on Pattern Recognition. Cambridge, UK, pp 136-139
- [14] Bernadó-Mansilla E, Ho TK (2005) Domain of competence of XCS classifier system in complexity measurement space. *IEEE Transactions on Evolutionary Computation* 9(1):82-104
- [15] Bezdek JC, Hathaway RJ (2002) VAT: A Tool for Visual Assessment of (Cluster) Tendency. In: proceedings of 2002 International Joint Conference on Neural Networks. pp 2225-2230
- [16] Bonarini A (2000) An introduction to learning fuzzy classifier systems. In: Carbonell JG, Siekmann J (eds) *Learning Classifier Systems*, LNCS, vol 1813. Springer, Heidelberg, pp 83-104
- [17] Brazdil P, Giraud-Carrier C, Soares C, Vilalta R (2009) *Metalearning: applications to data mining*. Cognitive Technologies, Springer.
- [18] Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) *Classification and Regression Trees*. Wadsworth, Belmont.
- [19] Brighton H, Mellish C (2002) Advances in instance selection for instance-based learning algorithms. *Data Min and Knowl Discov* 6:153-172
- [20] Caíses Y, González A, Leyva E, Pérez R (2009) A Method for Combining Instance Selection Algorithms. In: proceedings of the IADIS International Conference Intelligent Systems and Agents 2009. Algarve, Portugal, pp 77-84
- [21] Caíses Y, González A, Leyva E, Pérez R (2009) SCIS: Combining Instance Selection Methods to Increase Their Effectiveness over a Wide Range of Domains. In: Corchado E, Yin H (eds) *IDEAL 2009*, LNCS, vol 5788. Springer, Heidelberg, pp 17-24
- [22] Caíses Y, González A, Leyva E, Pérez R (2011) Combining instance selection methods based on data characterization: An approach to increase their effectiveness. *Information Sciences* 181(20):4780-4798.
- [23] Cameron-Jones R (1995) Instance selection by encoding length heuristic with random mutation hill climbing. In: *Proceedings of the Eighth Australian Joint Conference on Artificial Intelligence*. pp 99-106
- [24] Cano JR, Herrera F, Lozano M (2003) Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. *IEEE Trans on Evol Comput* 7(6):561-575
- [25] Cano JR, Herrera F, Lozano M (2005) Stratification for scaling up evolutionary prototype selection. *Pattern Recognit Lett* 26:953-963
- [26] Cano JR, Herrera F, Lozano M (2006) On the combination of evolutionary algorithms and stratified strategies for training set selection in data mining. *Applied Soft Computing* 6:323-332

- [27] Cano JR, Herrera F, Lozano M (2007) Evolutionary Stratified Training Set Selection for Extracting Classification Rules with Trade Off Precision-Interpretability. *Data & Knowledge Engineering* 60(1):90-108
- [28] Cerverón V, Ferri FJ (2001) Another Move Toward the Minimum Consistent Subset: A Tabu Search Approach to the Condensed Nearest Neighbor. *RuleIEEE Transactions on systems, Man, and Cybernetics-Part B* 31(3):408-413
- [29] Chang C (1974) Finding prototypes for nearest neighbor classifiers. *IEEE Trans on Comput* 23(11):1179-1184
- [30] Cover T, Hart P (1967) Nearest neighbor pattern classification. *IEEE Trans on Inf Theory* 13(1):21-27
- [31] Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
- [32] Czarnowski I (2011) Cluster-based instance selection for machine classification. *Knowl Inf Syst* DOI 10.1007/s10115-010-0375-z
- [33] Dasarathy B (1994) Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Trans on Syst, Man, and Cybern* 24(1):511-517
- [34] Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. *J of Mach Learn Res* 7:1-30
- [35] Derrac J, García S, Herrera F (2010) Stratified prototype selection based on a steady-state memetic algorithm: a study of scalability. *Memetic Computation* 2:183-199
- [36] Derrac J, García S, Herrera F (2010) A survey on evolutionary instance selection and Generation. *International Journal of Applied Metaheuristic Computing* 1(1):60-92
- [37] Fan RE, Chen PH, Lin CJ (2005) Working set selection using the second order information for training SVM. *Journal of Machine Learning Research* 6:1889-1918.
- [38] Fayed HA, Hashem SR, Atiya AF (2007) Self-generating prototypes for pattern classification. *Pattern Recognition* 40:1498-1509
- [39] Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery: An overview. In: Fayyad U, Piatetsky-Shapiro G, Smyth P (eds) *Advances in Knowledge Discovery and Data Mining*. AAAI Press/The MIT Press, Menlo Park, CA, pp 495-515
- [40] Frank E, Witten I (1998) Generating accurate rule sets without global optimization. In: *proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, San Francisco, CA, pp 144-151
- [41] Friedman JH, Bentley JL, Finkel RA (1977) An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software* 3(3):209-226

- [42] Furnkranz J (1997) More Efficient Windowing. In: proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97). AAAI Press, Providence pp 509-514
- [43] Furnkranz J (1999) Separate-and-conquer rule learning. *Artif Intell Rev* 1(13):3-54
- [44] García S, Cano JR, Bernardó-Mansilla E, Herrera F (2009) Diagnose Effective Evolutionary Prototype Selection Using an Overlapping Measure. *International Journal of Pattern Recognition and Artificial Intelligence* 23(8):1527-1548
- [45] García S, Cano JR, Herrera F (2008) A memetic algorithm for evolutionary prototype selection: A scaling up approach. *Pattern Recognit* 41:2693-2709
- [46] García-Osorio C, Haro-García A, García-Pedrajas N (2010) Democratic instance selection: A linear complexity instance selection algorithm based on classifier ensemble concepts. *Artificial Intelligence* 174:410-441
- [47] García S, Herrera F (2009) Evolutionary Undersampling for Classification with Imbalanced Datasets: Proposals and Taxonomy. *Evolutionary Computation* 17(3):275-306
- [48] García-Pedrajas N, Romero del Castillo JA, Ortiz-Boyer D (2010) A cooperative coevolutionary algorithm for instance selection for instance-based learning. *Machine Learning* 78(3):381-420.
- [49] Glover F, Laguna M (1997) *Tabu Search*. Kluwer Academic Publishers, Dordrecht
- [50] González A, Pérez R (1999) SLAVE: A genetic learning system based on an iterative approach. *IEEE Trans on Fuzzy Sys* 7:176-191
- [51] Grochowski M, Jankowski N (2004) Comparison of Instance Selection Algorithms II. Results and Comments. In: Rutkowski L, Siekmann J, Tadeusiewicz R, Zadeh LA (eds) *Artificial Intelligence and Soft Computing - ICAISC 2004, LNCS, vol 3070*. Springer, Heidelberg, pp 580-585
- [52] Hamo Y, Markovitch S (2005) The COMPSET Algorithm for Subset Selection. In: *Proceedings of The Nineteenth International Joint Conference for Artificial Intelligence*. Edinburgh, Scotland, pp 728-733
- [53] Hart PE (1968) The condensed nearest neighbor rule. *IEEE Trans on Inf Theory* 14(3):515-516
- [54] Haro-García A, García-Pedrajas N (2009) A divide-and-conquer recursive approach for scaling up instance selection algorithms. *Data Min Knowl Disc* 18:392-418
- [55] Ho TK, Basu M (2002) Complexity Measures of Supervised Classification Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(3):289-300
- [56] Hoekstra A, Duin RPW (1996) On the nonlinearity of pattern classifiers. In: *proceedings of the 13th International Conference on Pattern Recognition*. Vienna, Austria, pp 271-275
- [57] Jensen F (1996) *An introduction to bayesian networks*. UCL Press

- [58] John GH, Langley P (1995) Estimating continuous distributions in bayesian classifiers. In: proceedings of the 11th Conference on Uncertainty in AI. Morgan Kaufmann, San Mateo, pp 338-345
- [59] Kim S, Oommen B (2003) A brief taxonomy and ranking of creative prototype reduction schemes. *Pattern Anal Appl* 6:232-244
- [60] Kim S, Oommen B (2003) Enhancing prototype reduction schemes with LVQ3-type algorithms. *Pattern Recognit* 36(5):1083-1093
- [61] Kim S, Oommen B (2004) Enhancing Prototype Reduction Schemes With Recursion: A Method Applicable for "Large" Data Sets. *IEEE Transactions on Systems, Man and Cybernetics-Part B* 34(3):1384-1397
- [62] Kim S, Oommen B (2009) On using prototype reduction schemes to enhance the computation of volume-based inter-class overlap measures. *Pattern Recognition* 42:2695-2704
- [63] King RD, Feng C, Sutherland A (1995) Statlog: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence* 9(3):289-333
- [64] Kohonen T (2001) *Self-Organizing Maps*, 3th Edition. Springer, Berlin
- [65] Kotsiantis SB, Pintelas P (2004) Recent advances in clustering: a brief survey. *WSEAS Trans on Inf Science and Applications* 1(1):73-81
- [66] Kotsiantis SB (2007) Supervised machine learning: A review of classification techniques. *Inform* 31:249-268
- [67] Krasnogor N, Smith J (2005) A tutorial for competent memetic algorithms: model, taxonomy, and design issues, *IEEE Trans. Evol. Comput.* 9(5):474-488.
- [68] Kuncheva L (1995) Editing for the k-Nearest neighbors rule by a genetic algorithm. *Pattern Recognit Lett* 16:809-814
- [69] Lashkia GV (2002) Learning with Relevant Features and Examples. In: proceedings of ICPR 2002, vol 2, pp 68-71
- [70] Lebourgeois F, Emptoz H (1996) Pretopological Approach for Supervised Learning. In: proceedings of ICPR 96, pp 256-260
- [71] Li G, Japkowicz N, Stocki TJ, Ungar RK (2009) Instance Selection by Border Sampling in Multi-class Domains. In: Huang R, Yang Q, Pei J, Gama J, Meng X, Li X (eds) *ADMA 2009, LNAI*, vol 5678. Springer, Heidelberg, pp 209-221
- [72] Lin J (1991) Divergence Measures Based on the Shannon Entropy. *IEEE Trans on Information Theory* 37(1):145-151
- [73] Liu H, Motoda H (2002) On issues of instance selection. *Data Mining and Knowledge Discovery* 6:115-130

- [74] Luengo J, Fernández A, García S, Herrera F (2010) Addressing data complexity for imbalanced data sets: analysis of SMOTE-based oversampling and evolutionary undersampling. *Soft Computing* DOI 10.1007/s00500-010-0625-8
- [75] Luengo J, Herrera F (2010) Domains of competence of fuzzy rule based classification systems with data complexity measures: A case of study using a fuzzy hybrid genetic based machine learning method. *Fuzzy Sets Systems* 161(1):3-19
- [76] Marchiori E (2008) Hit Miss Networks with Applications to Instance Selection. *Journal of Machine Learning Research* 9:997-1017
- [77] Marchiori E (2010) Class Conditional Nearest Neighbor for Large Margin Instance Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(2):364-370
- [78] McLachlan GJ (2004) *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley and Sons.
- [79] Michalski RS (1983) A Theory and Methodology of Inductive Learning. *Artificial Intelligence* 20(2):111 - 161
- [80] Mollineda R, Sánchez J, Sotoca J (2005) Data characterization for effective prototype selection. In: Marques J, Pérez N, Pina P (eds) *Pattern Recognit and Image Anal, LNCS*, vol 3523. Springer, Heidelberg, pp 27-34
- [81] Moreno F (2004) *Clasificadores eficaces basados en algoritmos rápidos de búsqueda del vecino más cercano*. Tesis doctoral, Departamento de Lenguajes y Sistemas Informáticos, Universidad de Alicante, Alicante, España.
- [82] Murthy SK (1998) Automatic construction of decision trees from data: A multi-disciplinary survey. *Data Min and Knowl Discov* 2:345-389
- [83] Nettleton DF, Orriols-Puig A, Fornells A (2010) A study of the effect of different types of noise on the precision of supervised learning techniques. *Artif Intell Rev* 33:275-306
- [84] Nikolaidis K, Rodriguez E, Goulermas JY, Wu QH (2010) Instance Seriation for Prototype Abstraction. In: proceedings of the IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), pp 1351-1355
- [85] Olvera-López JA, Carrasco-Ochoa JA, Martínez-Trinidad JF (2010) A new fast prototype selection method based on clustering. *Pattern Analysis and Applications* 13:131-141
- [86] Platt J (1999) Fast training of support vector machines using SMO. In: Schölkopf B, Burges C, Smola A (eds) *Advances in kernel methods*. MIT Press, Cambridge, pp 185-208
- [87] Price KV, Storn RM, Lampinen JA (2005) *Differential evolution: a practical approach to global optimization*. Springer-Verlag, Berlin

- [88] Quinlan JR (1983) Learning Efficient Classification Procedures and Their Application to Chess End Games. In: Michalski RS, Carbonell JG, Mitchell TM (eds) Machine Learning: An Artificial Intelligence Approach. Morgan Kaufmann, San Francisco, CA pp 463-482.
- [89] Quinlan JR (1992) Learning with continuous classes. In: proceedings of the 5th Australian Joint Conference on Artificial Intelligence. Singapore, World Scientific, pp 343-348
- [90] Quinlan JR (1993) C4.5: Program for machine learning. Morgan Kaufman, San Mateo, CA
- [91] Reeves C, Bush D (2001): Using genetic algorithms for training data selection in RBF networks. In: Liu, H., Motoda, H. (eds.) Instance Selection and Construction for Data Mining, pp. 339-356. Kluwer, Norwell, MA (2001)
- [92] Reinartz T (2002) A unifying view on instance selection. Data Min and Knowl Discov 6:191-210
- [93] Riquelme J, Aguilar J, Toro M (2003) Finding representative patterns with ordered projections. Pattern Recognit 36(4):1009-1018
- [94] Rojas R, Feldman J (1996) Neural Networks: A Systematic Introduction. Springer-Verlag, Berlin.
- [95] Sánchez L (1998) A Random Sets-Based Method for Identifying Fuzzy Models. Fuzzy Sets and Systems 98(3):343-354.
- [96] Singh S (2003) PRISM - A novel framework for pattern recognition. Pattern Analysis and Applications 6:134-149
- [97] Sivanandam SN, Deepa SN (2007) Introduction to Genetic Algorithms. Springer Verlag
- [98] Smith FW (1968) Pattern classifier design by linear programming, IEEE Transaction on Computers 17:367-372
- [99] Smith-Miles KA (2008) Cross-Disciplinary Perspectives on Meta-Learning for Algorithm Selection. ACM Computing Surveys 41(1):6:1-25
- [100] Sotoca JM, Mollineda RA, Sánchez JS (2006) A meta-learning framework for pattern classification by means of data complexity measures. Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial 29:31-38
- [101] Talbi E (2009) Metaheuristics from design to implementation. John Wiley & Sons, Hoboken, New Jersey.
- [102] Tenorio-González AC, Martínez-Trinidad JF, Carrasco-Ochoa JA (2010) Designing RBFNNs Using Prototype Selection. In: Carrasco-Ochoa JA et al. (eds): MCPR 2010, LNCS vol 6256. Springer, Heidelberg, pp 189-198
- [103] Todorovski L, Dzeroski S (2003) Combining classifiers with meta decision trees. Machine Learning 50:223-249

- [104] Tomek I (1976) An experiment with the edited nearest-neighbor rule. *IEEE Trans on Syst, Man, and Cybern* 6(6):448-452
- [105] Triguero I, Derrac J, García S, Herrera F (2011) A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Trans. on Syst, Man, and Cybern, Part C*, DOI:10.1109/TSMCC.2010.2103939
- [106] Triguero I, Garcia S, Herrera F (2010) A preliminary study on the use of differential evolution for adjusting the position of examples in nearest neighbor classification. In: proceedings of 2010 IEEE Congress on Evolutionary Computation (CEC). Barcelona, pp 1-8
- [107] Veenman CJ, Reinders MJT, Backer E (2002) A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(9):1273-1280
- [108] Veenman CJ, Reinders MJT (2005) The Nearest Sub-class Classifier: a Compromise between the Nearest Mean and Nearest Neighbor Classifier. *IEEE Transactions on PAMI* 27(9):1417-1429
- [109] Wang X, Wu B, He Y, Pei X (2008) NRMCS: Noise removing based on the MCS. In: proceedings of the 7th International Conference on Machine Learning and Cybernetics (ICMLA08). La Jolla Village, USA, pp 89-93
- [110] Wilson D (1972) Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans on Syst, Man, and Cybern* 2(3):408-421
- [111] Wilson D, Martinez T (2000) Reduction techniques for instance-based learning algorithms. *Mach Learn* 38:257-286
- [112] Wolpert D, Macready WG (1997) No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1(1): 67-82
- [113] Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng SK, Liu B, Yu PS, Zhou ZH, Steinbach M, Hand DJ, Steinberg D (2008) Top 10 algorithms in data mining. *Knowl and Inf Syst* 14:1-37
- [114] Yiamilos PN (1993) Data structures and algorithms for nearest neighbor search in general metric spaces. In: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms, Austin, Texas, pp 311-321
- [115] Zhang H, Sun G (2002) Optimal reference subset selection for nearest neighbor classification by tabu search. *Pattern Recognition* 35:1481-1490
- [116] Zhang G (2000) Neural networks for classification: a survey. *IEEE Trans on Syst, Man, and Cybern* 30(4):451-462
- [117] Zhao K, Zhou S, Guan J, Zhou A (2003) C-pruner: An improved instance pruning algorithm. In: *Machine Learning and Cybernetics, 2003 Int Conf*, vol 1, pp 94-99
- [118] Zuhba AV (2010) NP-Completeness of the Problem of Prototype Selection in the Nearest Neighbor Method. *Pattern Recognition and Image Analysis*, 20(4):484-494