

**UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

Departamento de Informática y Automática



**SISTEMAS DE CLASIFICACIÓN AUTOMÁTICOS CON CONFIANZA Y
CREDIBILIDAD EN FUSIÓN TERMONUCLEAR**

TESIS DOCTORAL

Lázaro Emilio Makili

Licenciado en Ciencias de Educación en la Especialidad de Física y Astronomía

2014

**UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

Departamento de Informática y Automática



**SISTEMAS DE CLASIFICACIÓN AUTOMÁTICOS CON CONFIANZA Y
CREDIBILIDAD EN FUSIÓN TERMONUCLEAR**

TESIS DOCTORAL

**Autor: Lázaro Emilio Makili, Licenciado en Ciencias de Educación en la
Especialidad de Física y Astronomía**

Directores: Dr. Sebastián Dormido Canto

Dr. Jesús Antonio Vega Sánchez

2014

DEDICATORIA

A mi esposa, *Maria do Céu*, e filhos, *Lucky*, *Wode* y *Josy*, por el sacrificio, la comprensión, la infinita paciencia y el apoyo moral prestado durante tan larga ausencia.

A la memoria de *Afonso Alves dos Santos*, alias *Checho* o *Verón*, compañero de recorrido que no tuvo la oportunidad de llegar a la ansiada cumbre.

Labor omnia vincit.

Moçâmedes

AGRADECIMIENTOS

La grandeza de cualquier obra se mide por la dimensión y fortaleza de sus pilares. Sin ser una excepción, deseamos expresar nuestros agradecimientos a una serie de instituciones e individualidades que con su apoyo y soporte nos permitieron enfrentar durante largos años la formación y presentar este trabajo para culminación de los estudios.

Nuestra palabra de agradecimiento va para el *Instituto Nacional de Gestão de Bolsas de Estudos* (INAGBE) del Gobierno de Angola, por la financiación de nuestra estancia en Madrid casi en su totalidad. Para el *Gobierno de la Provincia de Benguela*, por el aporte financiero que nos permitió dar los primeros pasos para nuestro traslado a España, así como para el *Instituto Superior Politécnico de la Universidad Katyavala Bwila*, por la financiación de parte de nuestros desplazamientos para participación en eventos relacionados con nuestra formación.

No sería posible alcanzar este estadio sin conducción de nuestros orientadores, *Dr. Sebastián Dormido Canto* y *Dr. Jesús Vega*, quienes con su ejemplo, amabilidad, sabiduría y paciencia supieron motivarnos y ayudarnos a encontrar los mejores caminos para alcanzar los objetivos que nos habíamos propuesto. Estamos profundamente agradecidos y el tiempo de trabajo que compartimos dejó marcas que nos acompañarán durante el resto de la vida.

Quisiéramos extender los agradecimientos a los colegas de la *Unidad de Adquisición de Datos* del CIEMAT, que siempre que necesario nos prestaron todo el apoyo necesario para el desarrollo de nuestro trabajo. También, al personal del *Departamento de Informática y Automática* de la ETSI Informática de la UNED y de la *Facultad de Ciencias Físicas* de la Universidad Complutense, en especial a *Pilar*, en la UNED, y *Pedro*, en la Universidad Complutense, que todo hicieron para tornar nuestra estancia más agradable y nuestras gestiones admi-

nistrativas más fluidas y amenas permitiéndonos dedicar toda la atención requerida al proceso de formación.

Nuestros agradecimientos van también para una serie de familiares, colegas y amigos que durante todo este tiempo nos fueron aportando su apoyo incondicional. Al matrimonio *Alice y Amaro*, quienes en los momentos más decisivos nos han ayudado a mantenernos en nuestra ruta de la ilusión, al matrimonio *Maria y Nelson Lima* que nos han acogido y prestado toda la ayuda necesaria durante nuestra estancia en Madrid, así como al matrimonio *Mónica y Tino Gordo*. A *Paulo y Mário Makili*, hermanos, así como a nuestra cuñada *Ana Maria* por todo el apoyo prestado durante nuestra estancia fuera del país. Agradecer a *Eugenio Novais, Jesús Baptista, Benevides Costa, David Aragão, Mário Brito, José Maria Cruz, Gaspar Júnior, Ângelo Samuango*, colegas y amigos que mucho aportaron en las diferentes fases relacionadas a los estudios conducentes a la escritura de esta Tesis.

A todos que de alguna manera se sientan reflejados en este trabajo, y que por el hecho de no haber sido mencionados explícitamente no son menos importantes, dirigimos nuestra más sincera palabra de agradecimiento.

Muchas gracias a todos!

RESUMEN

En esta Tesis se aborda la problemática de asociar medidas de fiabilidad a las predicciones efectuadas al clasificar datos generados en los dispositivos de fusión termonuclear. Con tal fin, se analiza la aplicación de una serie de clasificadores, implementados en base a la teoría de la predicción conformal, a los datos generados en el *estellarator* TJ – II, tomándose como caso de estudio un conjunto de imágenes captadas por su diagnóstico de esparcimiento Thomson.

Por otro lado, se estudia el problema de la selección de las muestras de datos más adecuadas para implementar un clasificador. Este estudio se realiza desde una perspectiva de aprendizaje activo. Se implementó un algoritmo que permite, con una cantidad reducida de datos, alcanzar en el proceso de clasificación tanto tasas de acierto como medidas de fiabilidad cualitativamente superiores (o al mismo nivel) de las alcanzadas al entrenar los clasificadores con una cantidad de datos mucho mayor, cuando se seleccionan de forma aleatoria.

En los dispositivos de fusión, cada vez es más necesario implementar sistemas de aprendizaje automático que puedan funcionar en tiempo real. Por otro lado, la metodología de clasificación utilizada en la Tesis es transductiva, lo que lleva implícita una alta exigencia desde el punto de vista computacional al clasificar los datos. Considerando estos hechos, en la Tesis se ha puesto un énfasis particular en la reducción de la carga computacional en el proceso de clasificación analizado. Para lograrlo se han investigado dos alternativas: (i) la implementación de variantes del método de clasificación y (ii) la reducción del tamaño de los conjuntos de entrenamiento empleados, pero manteniendo siempre la efectividad, en el proceso de clasificación.

Palabras claves: predicción conformal, clasificación, confianza, credibilidad, aprendizaje activo, máquinas de vectores soporte.

ABSTRACT

In this Thesis, the problem of associating reliability measures to automatic classification tasks in thermonuclear fusion devices has been tackled. To this end, a set of classifiers, whose implementation has been based on the conformal prediction theory, have been analyzed and applied to data generated in a specific fusion device, the TJ – II stellarator. In particular, the focus has been put on the set of images corresponding to the TJ – II’s Thomson scattering diagnostic.

Also, it is important to note that the big amount of data stored in the databases of fusion devices makes essential the problem of selecting the most suitable samples to train a classifier. The aforementioned problem has been tackled in this Thesis from an active learning perspective. An active learning algorithm has been implemented to allow, with a reduced amount of training data, reaching both success rates and reliability measures that are better than, or as good as, the ones reached training the classifiers using a much bigger amount of randomly selected data.

In fusion devices, more and more needs of real time signal classification are present. Bearing this kind of requirement in mind, methodologies for the reduction of the computational overload in the classification processes have been emphasized in this Thesis. This goal has been reached by two different ways. On the one hand, the implementation of variants of the classification method has been taken into account. On the other hand, the use of reduced but effective training sets has been analyzed.

Keywords: conformal prediction, classification, confidence, credibility, active learning, support vector machines.

ÍNDICE GENERAL

Capítulo 1: Introducción.....	1
1.1. Motivación.....	1
1.2. Objetivos.....	5
1.3. Principales contribuciones.....	6
1.3.1. Componentes <i>software</i> desarrollados.....	6
1.3.2. Publicaciones en revistas científicas y técnicas.....	7
1.3.3. Participaciones en congresos.....	8
1.4. Estructura de la memoria.....	9
1.5. Convenciones a la hora de escribir la Tesis.....	11
Capítulo 2: Introducción a la fusión termonuclear.....	13
2.1. La fusión termonuclear en el contexto global de la producción de energía.....	13
2.2. Fusión termonuclear y física del plasma.....	17
2.3. Fusión por confinamiento magnético.....	24
2.3.1. <i>Tokamak</i>	25
2.3.2. <i>Estellarator</i>	26
2.4. El <i>estellarator</i> TJ – II.....	27
2.4.1. Confinamiento.....	29
2.4.2. Calentamiento.....	30
2.4.3. <i>Diagnósticos</i> y sistema de adquisición de datos.....	31
2.5. Conclusiones.....	33

Capítulo 3: Estimación de la confianza en el aprendizaje automático.....	35
3.1. Planteamiento y problemática de la estimación de la confianza	35
3.1.1. Métodos para la estimación de la confianza	38
3.1.2. Formas de representación de la confianza	40
3.1.3. Evaluación de la calidad de las predicciones.....	42
3.2. Principales teorías para la estimación de la confianza	45
3.2.1. Teoría del aprendizaje estadístico.....	46
3.2.2. Aprendizaje <i>bayesiano</i>	48
3.2.3. Estimación <i>hold-out</i>	50
3.2.4. Predicción conformal.....	52
3.3. Conclusiones	52
Capítulo 4: Predictores conformales.....	53
4.1. Introducción.....	53
4.2. Teoría algorítmica de aleatoriedad y tests estadísticos	56
4.3. Escenarios de aprendizaje <i>online</i> y <i>offline</i>	58
4.4. Transducción e inducción.....	59
4.5. Definición general de predictor conformal	61
4.6. Clasificación con confianza y credibilidad	63
4.7. Propiedades de los predictores conformales	64
4.8. Máquinas de Vectores Soporte y valores de no conformidad.....	66
4.9. Limitaciones de los predictores conformales	70
4.10. Variantes de clasificadores conformales	71

Índice general	xi
4.10.1. Clasificación con función <i>hash</i>	72
4.10.2. Cálculo de valores de no conformidad con <i>SVM</i> incremental	75
A. Enfoque de Syed, Liu y Sung (SVMi_SLS).....	76
B. Enfoque de Cawenberghs y Poggio (SVMi_CP).....	77
C. Algoritmo de clasificación.....	79
4.10.3. Predictor conformal inductivo.....	80
4.11. Conclusiones.....	83
Capítulo 5: Aplicaciones a los datos de fusión termonuclear.....	85
5.1. Sistema de clasificación de imágenes	85
5.1.1. Introducción	86
5.1.2. <i>Experimentos</i> y resultados.....	88
A. Implementación de los clasificadores.....	88
B. Metodología	90
C. Resultados.....	96
Precisión de la clasificación.....	96
Valores de confianza y credibilidad.....	99
Comparación de la eficiencia temporal.....	108
5.1.3. Implementación práctica	112
5.1.4. Conclusiones	117
5.2. Selección activa de conjuntos de entrenamiento	122
5.2.1. Introducción	122

5.2.2. Aprendizaje activo	124
A. Combinación de múltiples criterios	129
B. Condición de parada.....	135
C. Algoritmo de aprendizaje activo	138
5.2.3. <i>Experimentos</i> y resultados	140
A. Implementación de los algoritmos	140
B. Metodología.....	140
Resultados	142
5.2.4. Conclusiones.....	159
Capítulo 6: Conclusiones y trabajo futuro	161
6.1. Conclusiones	161
6.2. Trabajo futuro.....	164
Capítulo 7: Bibliografía	169
Anexo A: Conjunto de datos.....	203
Anexo B: ejemplo de utilización del método de clasificación con función <i>hash</i>	207
Anexo C: Figuras y Tablas adicionales	211
Anexo D: Librería de <i>software</i>	233
Anexo E: Tutorial sobre utilización de la librería Spider en un entorno de 64 bits.....	255

LISTA DE FIGURAS

Figura 1: Fuentes primarias de energía comercializable consumida en 2008 (fuente: http://www.eia.gov/forecasts/ieo/ieo_tables.cfm)	15
Figura 2: Energía de enlace por nucleón en función de la cantidad de nucleones en el núcleo (fuente: http://es.wikipedia.org/wiki/Helio)	20
Figura 3: Secciones eficaces en función de la energía cinética para las reacciones D - T, D - ³ He y D - D. La curva para la reacción D - D corresponde a la reacción $^2\text{D} + ^2\text{D} \rightarrow ^3\text{T} + \text{p} + 4.0 \text{ MeV}$	20
Figura 4: Vista esquemática de un <i>tokamak</i> (adaptada de http://www.generalfusion.com/magnetic_fusion.html)	26
Figura 5: Instalaciones del TJ - II en el CIEMAT (fuente: http://www-fusion.ciemat.es)	28
Figura 6: Vista esquemática del sistema de bobinas del TJ – II (fuente: http://www-fusion.ciemat.es)	30
Figura 7: Clasificación binaria con SVM; datos no separables	67
Figura 8: Interfaz gráfica para realización de los <i>experimentos</i>	89
Figura 9: Valores de credibilidad para las imágenes predichas de forma incorrecta; <i>kernel</i> lineal, nivel 4 de descomposición	105
Figura 10: Valores de credibilidad para las imágenes predichas de forma incorrecta; <i>kernel</i> lineal, nivel 5 de descomposición	105
Figura 11: Valores de credibilidad para las imágenes predichas de forma incorrecta; <i>kernel</i> polinómico, nivel 4 de descomposición	106
Figura 12: Valores de credibilidad para las imágenes predichas de forma incorrecta; <i>kernel</i> polinómico, nivel 5 de descomposición	106
Figura 13: Valores de credibilidad para las imágenes predichas de forma incorrecta; <i>kernel</i> RBF, nivel 4 de descomposición	107

Figura 14: Valores de credibilidad para las imágenes predichas de forma incorrecta; <i>kernel</i> RBF, nivel 5 de descomposición	107
Figura 15: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> lineal, nivel 4 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador	109
Figura 16: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> polinómico, nivel 4 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador.....	110
Figura 17: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> RBF, nivel 4 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador	110
Figura 18: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> lineal, nivel 5 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador	111
Figura 19: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> polinómico, nivel 5 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador.....	111
Figura 20: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> RBF, nivel 5 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador	112
Figura 21: Interfaz gráfica del servidor de clasificación del TSD.....	113
Figura 22: Ciclo de automatización del TSD	114
Figura 23: Protocolo de comunicación: conexión “test”	115
Figura 24: Protocolo de comunicación: conexión "procesar imagen".....	116
Figura 25: Protocolo de comunicación: conexión “clasificación de usuario”	117

Figura 26: Evolución de la confianza con la cantidad de muestras de entrenamiento; CP (SVMi_SLS uno-contra-uno) entrenado en un <i>escenario online</i>	120
Figura 27: Representación esquemática del ciclo de determinación de la diversidad de un conjunto de datos.....	134
Figura 28: Tasa de acierto media y desviación estándar de la tasa de acierto en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	144
Figura 29: Confianza media y desviación estándar de la confianza en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	145
Figura 30: Credibilidad media y desviación estándar de la credibilidad en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	146
Figura 31: Tasa de acierto media y desviación estándar de la tasa de acierto en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	148
Figura 32: Confianza media y desviación estándar de la confianza en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	149
Figura 33: Credibilidad media y desviación estándar de la credibilidad en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	150
Figura 34: Tasa de acierto media y desviación estándar de la tasa de acierto en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	153
Figura 35: Confianza media y desviación estándar de la confianza en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	154
Figura 36: Credibilidad media y desviación estándar de la credibilidad en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	155
Figura 37: Tasa de acierto media y desviación estándar de la tasa de acierto en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	156

Figura 38: Confianza media y desviación estándar de la confianza en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	157
Figura 39: Credibilidad media y desviación estándar de la credibilidad en función del número de muestras de entrenamiento (después de 10 ejecuciones).....	158
Figura 40: Patrones típicos de imágenes del TSD. De arriba a abajo y de izquierda a derecha: BKGND, ECRH, CUTOFF, NBI y STRAY.....	204
Figura 41: Transformada wavelet discreta de una imagen del TSD. A izquierda: imagen original. A derecha, de arriba a abajo y de izquierda a derecha: aproximación, detalle horizontal, detalle vertical y detalle diagonal.....	206
Figura 42: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVM Uno-Contra-Resto).....	215
Figura 43: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVM Uno-Contra-Uno).....	215
Figura 44: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVMi_SLS Uno-Contra-Resto).....	216
Figura 45: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVMi_SLS Uno-Contra-Uno).....	216
Figura 46: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVMi_CP Uno-Contra-Resto).....	217
Figura 47: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVMi_CP Uno-Contra-Uno).....	217
Figura 48: Valores de credibilidad para las imágenes predichas de forma incorrecta; Hashed CP (partes = 2).....	218
Figura 49: Valores de credibilidad para las imágenes predichas de forma incorrecta; Hashed CP (partes = 4).....	218

Figura 50: Valores de credibilidad para las imágenes predichas de forma incorrecta; ICP (Uno-Contra-Resto).....	219
Figura 51: Valores de credibilidad para las imágenes predichas de forma incorrecta; ICP (Uno-Contra-Uno).....	219

LISTA DE TABLAS

Tabla 1: Funciones <i>kernel</i> empleadas	68
Tabla 2: Variantes de clasificadores utilizados en los <i>experimentos</i> realizados. Los clasificadores se subdividen en tres grupos, los transductivos estándar (en fondo blanco), los transductivos con función <i>hash</i> (en fondo gris claro) y los predictores conformales inductivos (en fondo gris oscuro).....	92
Tabla 3: Espacio de búsqueda de los valores adecuados de los parámetros a emplear en la clasificación.....	94
Tabla 4: Valores adecuados de los parámetros empleados en la clasificación; <i>kernel</i> lineal..	94
Tabla 5: Valores adecuados de los parámetros empleados en la clasificación; <i>kernel</i> polinómico	95
Tabla 6: Valores adecuados de los parámetros empleados en la clasificación; <i>kernel</i> RBF ...	95
Tabla 7: Tasa de acierto correspondiente a una validación cruzada en 10 partes; nivel 4 de descomposición	98
Tabla 8: Tasa de acierto correspondiente a una validación cruzada en 10 partes; nivel 5 de descomposición	98
Tabla 9: Tasa de acierto correspondiente a una validación cruzada en 10 partes; <i>SVM</i> estándar, <i>uno-contra-el-resto</i>	98
Tabla 10: Tasa de acierto correspondiente a una validación cruzada en 10 partes; <i>SVM</i> estándar <i>uno-contra-uno</i>	99
Tabla 11: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; <i>kernel</i> lineal, nivel 4 de descomposición.....	101
Tabla 12: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; <i>kernel</i> lineal, nivel 5 de descomposición.....	102

Tabla 13: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; <i>kernel</i> polinómico, nivel 4 de descomposición.....	102
Tabla 14: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; <i>kernel</i> polinómico, nivel 5 de descomposición.....	103
Tabla 15: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; <i>kernel</i> RBF, nivel 4 de descomposición.....	103
Tabla 16: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; <i>kernel</i> RBF, nivel 5 de descomposición.....	104
Tabla 17: Clases de imágenes del TSD del TJ – II.....	204
Tabla 18: Cantidad de imágenes por clase.....	205
Tabla 19: Datos utilizados en el ejemplo descrito. Los mismos fueron extraídos del conjunto de datos <i>wine</i> del repositorio de la UCI (Bache, 2013)	208
Tabla 20: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (<i>SVM</i> Uno-Contra-Resto).....	211
Tabla 21: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (<i>SVM</i> Uno-Contra-Uno)	212
Tabla 22: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (<i>SVMi_SLS</i> Uno-Contra-Resto).....	212
Tabla 23: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (<i>SVMi_SLS</i> Uno-Contra-Uno)	212
Tabla 24: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (<i>SVMi_CP</i> Uno-Contra-Resto).....	213
Tabla 25: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (<i>SVMi_CP</i> Uno-Contra-Uno)	213

Tabla 26: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; Hashed CP (partes = 2)	213
Tabla 27: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; Hashed CP (partes = 4)	214
Tabla 28: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; ICP (Uno-Contra-Resto)	214
Tabla 29: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; ICP (Uno-Contra-Uno)	214
Tabla 30: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; <i>kernel</i> lineal, nivel 4 de descomposición	220
Tabla 31: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; <i>kernel</i> lineal, nivel 5 de descomposición	220
Tabla 32: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; <i>kernel</i> polinómico, nivel 4 de descomposición	220
Tabla 33: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; <i>kernel</i> polinómico, nivel 5 de descomposición	221
Tabla 34: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; <i>kernel</i> RBF, nivel 4 de descomposición	221
Tabla 35: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; <i>kernel</i> RBF, nivel 5 de descomposición	221
Tabla 36: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (<i>SVM</i> Uno-Contra-Resto)	222
Tabla 37: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (<i>SVM</i> Uno-Contra-Uno)	222

Tabla 38: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (SVMi_SLS Uno-Contra-Resto).....	223
Tabla 39: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (SVMi_SLS Uno-Contra-Uno).....	223
Tabla 40: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (SVMi_CP Uno-Contra-Resto).....	224
Tabla 41: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (SVMi_CP Uno-Contra-Uno).....	224
Tabla 42: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; Hashed CP (partes = 2)	225
Tabla 43: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; Hashed CP (partes = 4)	225
Tabla 44: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; ICP (Uno-Contra-Resto)	226
Tabla 45: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; ICP (Uno-Contra-Uno).....	226
Tabla 46: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> lineal, nivel 4 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador	227
Tabla 47: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> polinómico, nivel 4 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador	228
Tabla 48: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> RBF, nivel 4 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador	229

Tabla 49: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> lineal, nivel 5 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador.....	230
Tabla 50: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> polinómico, nivel 5 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador.....	231
Tabla 51: Tiempo de CPU (s) requerido para la clasificación de una imagen; <i>kernel</i> RBF, nivel 5 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador.....	232

GLOSARIO DE TÉRMINOS Y SIGLAS

Algoritmo base. *Underlying algorithm* en inglés. Algoritmo de aprendizaje automático utilizado para la implementación de un predictor conformal. Los predictores conformales se pueden implementar a partir de casi cualquier algoritmo de aprendizaje automático, siendo necesario para tal la definición de una medida de no conformidad adecuada (Vovk, 2005).

Beta. Cociente entre la presión del plasma y la presión del campo magnético. Viene dado por

$$\beta = \frac{p}{\left(\frac{B^2}{\mu_0} \right)}$$

donde p es la presión del plasma, B la intensidad del campo magnético y μ_0

la permeabilidad magnética del vacío, expresadas en unidades del Sistema Internacional.

Un alto valor de *beta* significa un buen rendimiento del dispositivo de fusión (LNF, 2004a).

Cizalla magnética. *Magnetic shear* en inglés. Es una grandezza que mide la variación que ocurre en el ángulo de inclinación de una línea del campo magnético o en la *transformada rotacional* al moverse desde una superficie magnética dada hasta la siguiente. Se define

como $s = 2 \frac{V}{i} \frac{d\iota}{dV}$, donde V representa al volumen contenido por dicha superficie magnética e $i/2\pi$ la *transformada rotacional*. En general, un valor elevado de la *cizalla magnética* es favorable para la estabilidad del plasma (Wakatani, 1998).

Computable

Función computable. Una función numérica se designa computable, si existe un algoritmo que sirva para calcular sus valores (Cutland, 1980), (Vereshchagin, 2003).

Secuencia computable de funciones computables. Una secuencia de funciones computables de una variable, f_0, f_1, \dots , se designa computable, si la función binaria, F definida por la fórmula $F(i, n) = f_i(n)$, es computable (Vereshchagin, 2003).

Consulta. *Query* en inglés. En aprendizaje activo, consiste en una muestra no etiquetadas que es presentada por el algoritmo de aprendizaje a un experto (por ejemplo, un especialista humano) para que sea etiquetada por este (Settles, 2009).

Consulta por Comité. *Query-By-Committee* en inglés. Estrategia de aprendizaje activo en la cual las muestras son seleccionadas para *consulta* en base al desacuerdo en cuanto a su clasificación existente entre los miembros de un conjunto (un comité) de modelos. La estrategia implica la manutención de un comité de modelos que son entrenados utilizándose el mismo conjunto de datos etiquetados pero representan hipótesis alternativas. Cada miembro del comité se utiliza para clasificar la muestra candidata y la selección para *consulta* se hace teniendo en cuenta el nivel de desacuerdo existente entre dichas clasificaciones (Settles, 2009).

Diagnóstico. Aparato diseñado para medir uno o varios parámetros del plasma (temperatura, densidad, corriente, etc.) (LNF, 2004a).

ECRH. Abreviatura de calentamiento a la frecuencia de resonancia electrón – ciclotrón (del inglés *electron cyclotron resonance heating*). Método de calentamiento del plasma mediante la inyección de ondas a la frecuencia ciclotrónica de los electrones del plasma o a sus múltiplos. Dichas ondas son absorbidas por los electrones gracias a un proceso de resonancia. Para la generación de las ondas (que están en el rango de las microondas con frecuencias entre 50 y 200 GHz) se utilizan generadores denominados *girotrones* especialmente desarrollados para esta aplicación (Thumm, 1995), (LNF, 2004a).

Error estándar. Es la desviación estándar de la distribución muestral de una estadística.

Normalmente, el valor de una estadística relacionada a una población varía de muestra para muestra. La forma en que esa variación ocurre se describe a través de la “*distribución muestral*” de la referida estadística. El *error estándar* expresa la incertidumbre relacionada a la estimación de una estadística y depende de la desviación estándar y del tamaño de la muestra, $SE = SD/\sqrt{n}$, siendo *SE* el *error estándar*, *SD* la desviación estándar y *n* el tamaño de la muestra (Everitt, 2002), (Altman, 2005).

Espacio de características. En los métodos *kernel*, como es el caso de las máquinas de vectores soporte, designa a un espacio de elevadas dimensiones definido a través de un conjunto de funciones de base no lineales. Superficies de decisión lineales en este espacio corresponden a superficies de decisión no lineales en el *espacio de entradas* (Cherkassky, 2007).

Espacio de entradas. Espacio abstracto en el que reside una colección de vectores relacionados a un problema determinado (Cherkassky, 2007).

Experimento. La disciplina de aprendizaje automático posee una componente empírica muy importante, por el hecho de que la gran mayoría de los algoritmos de aprendizaje son demasiado complejos para un análisis formal. Por *experimento* se entiende, a semejanza de las demás ciencias, la variación sistemática de una o más variables independientes y el examen de su efecto en una variable dependiente determinada. De este modo, un *experimento* en aprendizaje automático requiere un conjunto de ejecuciones de un algoritmo bajo diferentes condiciones con el objetivo de comparar un aspecto dado del comportamiento del sistema. Las mencionadas condiciones pueden estar relacionadas con el algoritmo empleado, el campo de aplicación o el entorno en que el aprendizaje ocurre (Langley, 1988), (Dietterich, 1997).

Girotrón. *Gyrotron* en inglés. Generador de microondas de alta potencia en el rango de frecuencias de la resonancia electrón – ciclotrón y de sus armónicos (desde decenas a centenas de GHz) (LNF, 2004a).

H – 1. Dispositivo de fusión del tipo *estellarator* localizado en la *Research School of Physics and Engineering* de la *Australian National University*, en Canberra, Australia. Se encuentra en funcionamiento desde 1992 (ANU, 2013).

Heliac. Configuración de dispositivos de fusión del tipo *estellarator* con eje magnético helicoidal. El campo magnético se produce por la superposición de los campos creados por un conjunto de bobinas de campo toroidal, colocadas en planos verticales, cuyos centros siguen una curva helicoidal y por la corriente que circula en un conductor central, que constituye el núcleo del sistema. Su versión flexible se diferencia de la original por la incorporación de una bobina helicoidal, enrollada alrededor del conductor central, al núcleo del sistema (Boozer, 1983), (Harris, 1985).

ITER. Abreviatura de *International Thermonuclear Experimental Reactor*. Dispositivo de fusión por confinamiento magnético del tipo *tokamak*. Se encuentra en construcción en la localidad francesa de Cadarache y fue proyectado con una dimensión física cerca de tres veces superior a las del *JET* o del *JT – 60U* (McCracken, 2005), (ITER Org, 2013).

JET. Abreviatura de *Joint European Torus*. Dispositivo de fusión por confinamiento magnético del tipo *tokamak*, siendo la mayor máquina existente de este tipo. Está localizado en Culham, cerca de Oxford, en el Reino Unido (EFDA, 2013).

JT - 60U. Dispositivo de fusión por confinamiento magnético del tipo *tokamak*, construido en Naka, Japón. Entró en funcionamiento en 1985, designándose en esa altura *JT – 60*. Pasó a designarse *JT – 60U* en 1991 después de una profunda actualización (Braams, 2002), (JAEA, s.f.).

LHD. Abreviatura de *Large Helical Device*. Dispositivo de fusión del tipo *estellarator*, con bobinas hechas de material superconductor. Es el mayor dispositivo de este tipo en funcionamiento y se localiza en Toki, Japón. Se encuentra operacional desde 1998 (NIFS, 2013).

Máquina en espejo magnético. *Magnetic mirror machine* en inglés. Configuración de confinamiento magnético en la cual el plasma se confina a través de un campo magnético estacionario cuya intensidad es mayor en los extremos del recipiente. Las regiones de mayor intensidad del campo magnético, designadas espejos magnéticos, permiten atrapar el plasma en la región central, de menor intensidad del campo magnético (McCracken, 2005).

Métodos de ponderación de densidad. *Density weighted methods* en inglés. Estrategia de aprendizaje activo en la cual para la selección de las muestras para *consulta* se considera no solamente la incertidumbre acerca de su clasificación sino también la “representatividad” de dichas muestras. Una muestra se considera representativa cuando la misma proviene de regiones densamente pobladas del *espacio de entradas* (Settles, 2009).

Muestreo por incertidumbre. *Uncertainty sampling* en inglés. Estrategia de aprendizaje activo en la cual el algoritmo selecciona para *consulta* las muestras cuyo etiquetado le resulte menos seguro (Settles, 2009).

NBI. Abreviatura de inyección de haces neutros (del inglés *neutral beam injection*). Sistema de calentamiento del plasma en máquinas de fusión a través de la inyección de haces de átomos neutros de alta energía que transfieren su impulso/energía a las partículas del plasma al interactuar con las mismas (LNF, 2004a).

Pinch

Efecto *pinch*. *Pinch effect* en inglés. Astricción del plasma por el que circula una corriente provocada por la interacción de dicha corriente con el campo magnético generado por ella misma (LNF, 2004a).

Theta *pinch*. Configuración de confinamiento magnético en la que el confinamiento se produce a través de un campo magnético axial, que comprime y calenta el plasma. El campo magnético es creado por un pulso de corriente acimutal que circula por una bobina enrollada alrededor del tubo de plasma (McCracken, 2005).

Toroidal *pinch*. Configuración de confinamiento magnético cerrada en forma de toro en la cual el plasma se confina a través de un fuerte campo magnético, completamente poloidal. Dicho campo es producido por una corriente toroidal que fluye a través del plasma. Con una corriente suficientemente fuerte, el campo magnético astringe el plasma, alejándolo de las paredes del recipiente contenedor (McCracken, 2005).

Z – *pinch*. Configuración de confinamiento magnético en la cual el plasma se confina a través de un campo magnético acimutal que lo comprime, alejándolo de las paredes del recipiente contenedor. El campo magnético es producido por una corriente que fluye a lo largo del eje del plasma entre dos electrodos situados en sus extremos (McCracken, 2005).

Pozo magnético. *Magnetic well* en inglés. Concepto relacionado con el equilibrio magnetohidrodinámico y la estabilidad del plasma confinado magnéticamente en dispositivos con configuración toroidal (junto a los conceptos de *transformada rotacional* y *cizalla magnética*). El *pozo magnético* del vacío (en inglés *vacuum magnetic well*) se define como

$$\hat{W} = \frac{V}{\langle B^2 \rangle} \frac{d}{dV} \left\langle \frac{B^2}{2} \right\rangle, \text{ donde } B \text{ representa la intensidad del campo magnético y } V \text{ el volu-}$$

men contenido por una superficie magnética dada. En la fórmula los paréntesis denotan un valor promedio sobre una superficie magnética, siendo el mismo definido para una función arbitraria Q como $\langle Q \rangle \equiv \int_0^L \frac{Q}{B} dl / \int_0^L \frac{dl}{B}$, donde dl es el incremento de un segmento de arco a lo largo de una línea de fuerza del campo. Un valor positivo del *pozo magnético* favorece al confinamiento (Wakatani, 1998), (Greene, 1998).

QBC. Abreviatura del inglés, *Query-by-Committee*. Véase *consulta por comité*.

SVM. Máquina de Vectores Soporte (abreviatura de inglés, *Support Vector Machines*)

TFTR. Abreviatura de *Tokamak Fusion Test Reactor*. Dispositivo de fusión por confinamiento magnético construido en el *Princeton Plasma Physics Laboratory*, en Princeton, New Jersey, en los Estados Unidos de América. Estuvo en funcionamiento entre 1982 y 1997 (Braams, 2002), (PPPL, 2012).

Transformada rotacional. *Rotational transform* en inglés. Caracteriza el ángulo promedio en la dirección poloidal girado por las líneas del campo magnético al dar una vuelta alrededor del toro, en los dispositivos de fusión por confinamiento magnético con configuración toroidal. El referido ángulo se calcula como $\iota = \lim_{N \rightarrow \infty} \frac{\sum_{k=1}^N \iota_k}{N}$, donde N representa el número de vueltas en la dirección toroidal y ι_k el ángulo girado por las líneas entre la k -ésima y la $(k + 1)$ -ésima vuelta. La *transformada rotacional* se define como $\iota/2\pi$. (Wakatani, 1998), (LNF, 2004a), (Piel, 2010).

Wendelstein 7 – X. Dispositivo de fusión del tipo *estellator* con bobinas elaboradas con material superconductor, en construcción en la sección del *Max Planck Institut für Plasmaphysik* en Greifswald, Alemania (IPP, 2013).

CAPÍTULO 1: INTRODUCCIÓN

En este capítulo se presentan los aspectos introductorios a la Tesis. En el mismo se describen los principales aspectos que motivaron la escritura de la Tesis, sus objetivos y principales contribuciones. El capítulo termina con una descripción de la estructura de la memoria y de algunas convenciones adoptadas en su escritura.

1.1. MOTIVACIÓN

Desde finales del pasado milenio, existe una clara tendencia a incrementar el ritmo de producción y acumulación de la información. De acuerdo con un estudio realizado por la Universidad de California en Berkeley, en el año de 2002 se hizo el registro de cerca de 5 EB ($5 \cdot 10^6$ TB) de información nueva en diferentes tipos de soportes físicos, correspondiendo el 92% de la misma a soportes magnéticos, especialmente discos duros (Lyman, 2003). Las cifras mencionadas en el estudio corresponden a un incremento de cerca del 30% al año, entre los años de 1999 y 2002, casi doblándose, por tanto, la información acumulada en un periodo de 3 años.

En el dominio científico esta tendencia es bien notoria. En especialidades diversas que van desde la astronomía hasta la biología molecular, o de la geofísica a las ciencias de la atmósfe-

ra, los instrumentos científicos modernos recolectan datos con tasas de gigabytes por hora, lo que proporciona un rápido crecimiento de la distancia existente entre la capacidad de recolección de datos y la aptitud de los científicos para el análisis de los mismos (Fayyad, 1996). A título de ejemplo, un estudio realizado en el Instituto Tecnológico de Massachusetts (MIT), tomando como casos de estudio seis ramas científicas diferentes, concluye que la cantidad total de datos generados en los seis casos es de aproximadamente 41000 TB al año (Madnick, 2009). Según el mismo estudio, la cantidad de datos generada en los casos tratados es entre cinco y diez veces superior con relación a la generada cinco años atrás, siendo las razones del crecimiento atribuidas a la mejoría ocurrida en los métodos experimentales, instrumentos y recursos computacionales empleados, así como al abaratamiento de los medios de almacenamiento de datos.

Por otro lado, y al mismo tiempo, se asiste a un incremento tanto en la complejidad como en la dimensionalidad de los datos acumulados, siendo común el almacenamiento y el manejo de datos en formato de imágenes o videos de alta resolución (Hill, 2002), imágenes hiperspectrales (Li, 2011) o secuencias con información genética con miles de dígitos de longitud (Fayyad, 1996), (Bellotti, 2005).

El campo de la *fusión termonuclear* no es ajeno a esta tendencia. En los dispositivos de fusión se genera una gran cantidad de datos, principalmente formas de onda e imágenes, que son almacenados íntegramente durante e incluso más allá del ciclo de vida de los dispositivos, conservándose por tanto durante décadas (Layne, 2012). Por ejemplo, en un dispositivo como *JET*¹, el conjunto de *diagnósticos* existentes es capaz de registrar hasta 18 GB de datos en bruto en cada descarga de plasma, llegándose a alcanzar cerca de 1 TB por día de operación. El término “*diagnóstico*” se usa en fusión como sinónimo de sistema de medida. En el caso de la siguiente generación de dispositivos de fusión, se prevé una acumulación de datos muy

¹ *Joint European Torus*; dispositivo localizado en Culham, cerca de Oxford, en el Reino Unido.

superior a la actual. Por ejemplo, se espera que las bases de datos de *ITER*² almacenen petabytes³ de datos al año (Vega, 2008b), (EFDA, 2013).

La extracción e interpretación de la información en semejantes bases de datos constituye un importante desafío para la comunidad científica, resultando evidente la necesidad de implementar métodos que permitan la automatización de este proceso. Los métodos de reconocimiento de patrones y de minería de datos constituyen un instrumento valioso en este contexto, permitiendo el desarrollo de sistemas eficientes de recuperación de datos y la elaboración de modelos y teorías basadas en los propios datos. Dichos métodos han sido utilizados de forma satisfactoria en la solución de diversos problemas en el dominio de la fusión, tales como la búsqueda de semejanzas o la clasificación de formas de onda (Dormido-Canto, 2004), (Dormido-Canto, 2005), (Farias, 2006), (Vega, 2008a), la clasificación de imágenes (Vega, 2005), (Farias, 2005), (Makili, 2010), la detección de transiciones en el régimen de confinamiento del plasma (Murari, 2009), (Vega, 2009), (Rattá, 2012) o la predicción de disrupciones (Cannas, 2004), (Rattá, 2008, 2010), (López, 2012), (Vega, 2012), (Murari, 2012).

Los referidos métodos tienen como base, en la gran mayoría de los casos, algoritmos desarrollados en el campo del aprendizaje automático. El *aprendizaje automático* se relaciona con la construcción de “*sistemas que mejoran automáticamente con la experiencia*” (Mitchell, 2006), donde el concepto de “*mejoría*” hay que interpretarlo en sentido amplio como “*sistemas de mayor rendimiento*”. El concepto general de aprendizaje abarca una larga serie de procesos que hacen difícil una definición precisa del mismo. Desde el punto de vista del aprendizaje automático, se considera que una máquina aprende con relación a una tarea T, medida de rendimiento R y tipo de experiencia E, si el sistema mejora fiablemente su rendimiento R, en la ejecución de la tarea T, con base en la experiencia E (Mitchell, 2006). Es comúnmente

² *International Thermonuclear Experimental Reactor*; dispositivo experimental a gran escala, encontrándose en construcción en la localidad de Cadarache, Francia (ITER ORG, 2013).

³ 1 PB = 1000 TB.

aceptado que los problemas típicos de aprendizaje automático pueden ser subdivididos en diversos tipos fundamentales que comparten algunas características: la clasificación (o reconocimiento de patrones), la regresión y la estimación de densidad de probabilidad (Vapnik, 2000), (Cherkassky, 2007).

A pesar de los buenos resultados que obtienen los métodos de reconocimiento de patrones, no obstante, hay que mencionar que muchos de estos métodos proporcionan meras predicciones sin una estimación de la confianza asociada a las mismas. En aplicaciones reales, particularmente aquellas consideradas críticas y en las cuales se asocia un peso elevado a los errores de predicción cometidos, es de extrema importancia asociar medidas de fiabilidad a las predicciones efectuadas. Se pone por tanto de manifiesto la necesidad de proporcionar junto con cada predicción, una medida acerca de la fiabilidad de la misma. Esta necesidad constituye la principal motivación para el desarrollo de esta Tesis.

En esta Tesis se aborda el problema de la *clasificación automática con medidas de fiabilidad* para datos generados en el área de *fusión termonuclear*. En los diversos casos de estudio se emplearon datos provenientes de un dispositivo de fusión particular, el *estellarator* TJ – II (Alejaldre, 1990).

En la Tesis se describe la implementación de un conjunto de clasificadores para el análisis de datos en entornos de fusión. No obstante, las metodologías de clasificación implementadas y los clasificadores desarrollados pueden ser fácilmente empleados en cualquier ámbito científico o tecnológico.

En la misma se ha puesto un énfasis particular en la reducción de la carga computacional en el proceso de clasificación analizado, lo que se hizo en base a dos razones fundamentales:

- En los dispositivos de fusión cada vez es más necesario implementar sistemas de aprendizaje automático que puedan funcionar en tiempo real, y
- La metodología de clasificación empleada es transductiva (lo que será tratado más adelante en la Sección 4.4), existiendo la necesidad del entrenamiento de varios clasificadores en el proceso de clasificación de cada muestra, lo que implica una alta exigencia desde el punto de vista computacional.

Para lograr la referida reducción de carga computacional, se investigaron dos alternativas: (i) la implementación de variantes del método de clasificación y (ii) la reducción del tamaño de los conjuntos de entrenamiento empleados, pero manteniendo siempre la efectividad en el proceso de clasificación, recurriéndose para ello a la metodología de aprendizaje activo.

1.2. OBJETIVOS

El objetivo central de esta Tesis es *la clasificación con medidas de fiabilidad de datos procedentes de diagnósticos de fusión termonuclear*. Para el desarrollo de clasificadores con medidas de fiabilidad se recurrió a la utilización de una metodología de clasificación específica, agrupándose en una librería una serie de algoritmos desarrollados en base a la metodología propuesta, lo que permite integrarlos de manera fácil en sistemas de clasificación para diferentes datos de entrada.

De forma más concreta, se persiguen en esta Tesis una serie de objetivos específicos que permiten alcanzar el objetivo general formulado:

- El desarrollo de un conjunto de diez clasificadores basados en una metodología que incorpore medidas de fiabilidad, además de las predicciones efectuadas.

- El diseño de una aplicación que permita facilitar la realización de *experimentos*⁴ a partir de los clasificadores desarrollados.
- El diseño de una segunda aplicación que permita incorporar los clasificadores de una forma automática en el entorno experimental del TJ – II.
- La implementación de una metodología de aprendizaje activo que permita seleccionar, de los conjuntos de datos existentes, conjuntos de entrenamiento reducidos para el entrenamiento de clasificadores, pero manteniendo niveles altos de precisión y fiabilidad en la clasificación.

1.3. PRINCIPALES CONTRIBUCIONES

El trabajo desarrollado en esta Tesis ha dado lugar a diversas contribuciones, las cuales pueden ser agrupadas en tres vertientes fundamentales: el desarrollo de componentes *software*, las publicaciones en revistas científicas y técnicas y las comunicaciones en congresos. A continuación se describen las más importantes.

1.3.1. Componentes *software* desarrollados

Los componentes *software* desarrollados fueron una librería y dos aplicaciones, una para facilitar la realización de *experimentos* con los datos utilizados y otra para la inclusión de los clasificadores desarrollados de forma automática en el entorno experimental del TJ – II.

Librería de *software*. Fue escrita en Matlab. Para facilitar su reutilización, se codificó siguiendo la metodología de programación orientada a objetos, ya que se hizo uso de una librería orientada a objetos, Spider (Weston, 2006), que implementa algoritmos de aprendizaje automático y es distribuida bajo licencia pública GNU. En la librería están incluidos

⁴ Se designa por *experimento*, en este caso, la ejecución controlada de un algoritmo, con un conjunto de parámetros dados, para análisis de su efectividad. En lo adelante se destacará en cursiva el término siempre que sea utilizado con este sentido.

objetos que desarrollan diferentes clases de algoritmos: de clasificación, de aprendizaje activo y de selección automática de los parámetros de los modelos.

Aplicaciones. Estas fueron desarrolladas igualmente en Matlab. La primera permite la realización de *experimentos* de clasificación a partir de un interfaz gráfico, facilitando su ejecución y la exploración de diferentes alternativas en cuanto a las variantes del clasificador y sus parámetros asociados. La segunda permite la clasificación en tiempo real durante la operación del TJ – II de un tipo concreto de señales como se detalla más adelante. También es importante resaltar que la aplicación puede usarse en diferido. Esta segunda aplicación funciona como un servidor *software* que, una vez recibida una petición de clasificación desde un programa cliente, la efectúa y devuelve el resultado a dicho cliente, el cual tiene la posibilidad de ejecutar diferentes acciones dependiendo de la clase atribuida a la señal.

1.3.2. Publicaciones en revistas científicas y técnicas

Como producto de la actividad desarrollada en la escritura de esta Tesis, fueron publicados varios artículos en revistas especializadas. A continuación se mencionan las publicaciones relacionadas directamente con la Tesis.

- Makili, L., Vega, J., Dormido-Canto, S., “Incremental Support Vector Machines for Fast Reliable Image Recognition”, *Fusion Engineering and Design*, vol. 88, 2013, pp. 1170 – 1173
- Makili, L., Vega, J., Dormido-Canto, S., “Active Learning Using Conformal Predictors: Application to Image Classification”, *Fusion Science and Technology*, vol. 62, 2012, pp. 347 – 355

- Makili, L., Vega, J., Dormido-Canto, S., Pastor, I., Murari, A., “Computationally Efficient SVM Multi-Class Image Recognition with Confidence Measures”, *Fusion Engineering and Design*, vol. 86, 2011, pp. 1213 - 1216
- Makili, L., Vega, J., Dormido Canto, S., Pastor, I., Pereira, A., Farias, G., Portas, A., Pérez-Risco, D., Rodríguez-Fernández, M. C. y Busch, P., “Upgrade of Automatic Analysis System in the TJ-II Thomson Scattering Diagnostic: New Image Recognition Classifier and Fault Condition Detection”, *Fusion Engineering and Design*, vol. 85, 2009, pp. 415 – 418
- Makili, L., Dormido-Canto, S. y Vega, J., “Clasificador de Imágenes del Diagnóstico Scattering Thomson del TJ – II Basado en Template Matching”, *Revista de Ingeniería Electrónica, Automática y Comunicaciones*, vol. 29 (1), 2008, pp. 14 – 19

1.3.3. Participaciones en congresos

A continuación se mencionan las diferentes participaciones en congresos:

- Makili, L., Vega, J., Dormido-Canto, S., “Incremental Support Vector Machines for Fast Reliable Image Recognition”. 27th Symposium on Fusion Technology. 24th September – 28th September 2012. Liège (Belgium). (<http://sciconf.org/soft2012/ip/topic/d/session/p2/paper/62>)
- Makili, L., Vega, J., Dormido-Canto, S., “Active Learning Using Conformal Predictors: Application to Image Classification”. 7th Workshop on Fusion Data Processing Validation and Analysis. March 26th-28th, 2012. Frascati, Roma (Italy) (http://www.fusione.enea.it/EVENTS/eventfiles/validation7-2012/DATA/Agenda-Abstracts/DATA_SESSIONS_AND_ABSTRACTS/DATA_DRIVEN_MODELS_AND_PREDICTIONS/).

- Makili, L., Vega, J., Dormido-Canto, S., Pastor, I., Murari, A., “Computationally Efficient SVM Multi-class Image Recognition with Confidence Measures”. 26th Symposium on Fusion Technology. 27th September – 1st October 2010. Porto (Portugal). (http://portal.ipfn.ist.utl.pt/soft2010_abstractbook/pdf/P3-016.pdf).
- Vega, J., Murari, A., Rattá, G. A., González, S., Pereira, A., Castro, R., Portas, A., Pastor, I., Dormido-Canto, S., Dormido, R., Duro, N., Sánchez, J., Farias, G., Ramírez, J. M., Makili, L., Vargas, H., Pajares, G., Santos, M., Ruiz, M., Barrera, E., López, J. M., Arcas, G. De, “An Overview About Recent Developments on Advanced Data Analysis Techniques in Fusion”. 6th Workshop on Fusion Data Processing, Validation and Analysis. January 25th-27th, 2010. Madrid (Spain) (<http://fudaqs2.ciemat.es/V6/main.htm>).
- Makili, L., Vega, J., Dormido-Canto, S., Pastor, I., Pereira, A., Portas, A., Pérez-Risco, D., Rodríguez-Fernández, M. C., Busch, P., “Upgrade of the Automatic Analysis System in the TJ-II Thomson Scattering Diagnostic: New Image Recognition Classifier and Fault Condition Detection”. 7th IAEA Technical Meeting on Control, Data Acquisition and Remote Participation for Fusion Research. 15 – 19 June 2009. Aix-en-Provence (France) (<http://www-fusion-magnetique.cea.fr/tmiaea2009/index.html>).
- Makili, L., Dormido-Canto, S., Vega, J., “Clasificador de Imágenes del Diagnóstico Scattering Thomson del TJ – II Basado en Template Matching”. 9^{no} Simposio Internacional de Automatización – XIII Convención y Feria Internacional Informática 2009, La Habana (Cuba), 2009

1.4. ESTRUCTURA DE LA MEMORIA

La memoria de la Tesis se estructura en siete capítulos. En tres de ellos se analiza el marco teórico de los aspectos más relevantes tratados en la Tesis, como son los casos del dominio de

aplicación (en este caso la fusión termonuclear), la estimación de la confianza en aprendizaje automático y la metodología de predicción a utilizar. En otro capítulo, se explican las aplicaciones específicas de la metodología seleccionada a los datos de fusión termonuclear y en los dos últimos se presentan las conclusiones y líneas de trabajo derivadas de la Tesis así como las referencias bibliográficas. A continuación se describen brevemente los diferentes capítulos de la memoria.

En el Capítulo 2 se hace una contextualización del dominio de aplicación, presentando los conceptos fundamentales acerca de la fusión termonuclear y de los dispositivos utilizados para la investigación en este dominio.

El Capítulo 3 se dedica a presentar una revisión contextual acerca de los diferentes esfuerzos encaminados a la estimación de la confianza en aprendizaje automático existentes en la literatura, haciendo mención especial a los principales resultados teóricos y prácticos en el campo.

En el Capítulo 4 se describen los elementos fundamentales relacionados con la metodología de predicción adoptada en el desarrollo de la Tesis.

El Capítulo 5 tiene un carácter eminentemente práctico. En este capítulo se describen las aplicaciones de la metodología seleccionada en el ámbito de la fusión termonuclear. El mismo se encuentra subdividido en dos partes, presentándose en la primera los aspectos relacionados con la aplicación de la metodología seleccionada al problema de la clasificación de señales y en la segunda los aspectos relativos a la selección de conjuntos de entrenamiento basada en el aprendizaje activo.

El Capítulo 6 resume las principales conclusiones derivadas del trabajo realizado en la Tesis, así como los posibles trabajos futuros en base a las líneas de investigación seguidas durante el desarrollo de la misma y el Capítulo 7 presenta la relación de referencias bibliográficas.

Finalmente se aportan una serie de anexos con la inclusión de material diverso:

- En el Anexo A se describe el conjunto de datos utilizado en la parte práctica de la Tesis.
- En el Anexo B se presenta un ejemplo detallado de utilización del método de clasificación conformal con función *hash* con un conjunto de datos concreto.
- El Anexo C contiene un conjunto de tablas y gráficos adicionales que permiten la complementación de la presentación de los resultados descritos en el capítulo 5.
- En el Anexo D se describen los principales objetos incluidos en la librería de *software* desarrollada.
- En el Anexo E se presenta una guía de ayuda para migrar la librería Spider (Weston, 2006) desde un entorno de desarrollo de 32 bits hacia otro de 64 bits.

1.5. CONVENCIONES A LA HORA DE ESCRIBIR LA TESIS

En esta Tesis converge información proveniente de diferentes áreas del conocimiento, lo que conlleva en algunos casos cierta dualidad en el sentido atribuido a determinados términos técnicos que son utilizados en las diferentes áreas con significados que no son exactamente coincidentes. Por otro lado, en la literatura referenciada en la Tesis predomina el inglés en la gran mayoría de los casos, situación que provoca alguna dificultad a la hora de efectuar la traducción de los términos científicos y técnicos utilizados, siendo en muchos casos difícil encontrar términos o expresiones equivalentes en el idioma español que sean universalmente aceptados.

Para facilitar la lectura de la Tesis, en el caso de algunos términos aceptados con mayor frecuencia como neologismos o de otros de índole muy técnica, para los cuales es muy difícil encontrar un equivalente en español, se prefirió mantener los mismos en el idioma en que se

encuentran en las correspondientes fuentes bibliográficas de origen (el inglés). En la escritura de los referidos términos a lo largo del texto se utilizó la *letra cursiva*.

Existen otros términos, también técnicos, que se presentan en el texto de la memoria de forma traducida. En muchos casos, para ayudar a precisar su sentido, junto a los equivalentes en español utilizados se añaden los términos o expresiones utilizados en la literatura de origen, es decir, en inglés.

Por otro lado, existe un conjunto de términos, expresiones y siglas que son reunidos en un glosario, en el cual son debidamente definidos. En el caso de estos, los términos y expresiones son destacados en el texto en *letra cursiva* siempre que sean utilizados con el sentido en que son definidos en el glosario.

CAPÍTULO 2: INTRODUCCIÓN A LA FUSIÓN TERMONUCLEAR

En este capítulo se hace una introducción al campo de la fusión termonuclear. En el mismo se aborda el potencial de la fusión termonuclear para la producción de energía y los conceptos fundamentales sobre fusión termonuclear y física del plasma.

2.1. LA FUSIÓN TERMONUCLEAR EN EL CONTEXTO GLOBAL DE LA PRODUCCIÓN DE ENERGÍA

Los elevados índices de consumo de energía constituyen una característica de las sociedades industrializadas, estando tanto el modo de vida como la gran mayoría de las condiciones de bienestar relacionados en alta medida con los índices de producción y consumo de energía, en conceptos de electricidad, movilidad y usos estacionarios (principalmente calentamiento en edificios y utilización en procesos industriales). En las últimas décadas se asistió a un rápido crecimiento del consumo anual de energía, estando los números correspondientes al referido consumo dominados por los índices alcanzados en los países industrializados.

En un futuro a largo plazo se prevé una continuidad en el crecimiento de la demanda de energía, siendo previsible el incremento de las necesidades energéticas en un orden que varía

entre una vez y media y tres veces hasta el año 2050 o entre dos y cinco veces hasta 2100 (Grubler, 1999). Esta tendencia está relacionada, por un lado, con el crecimiento de la población mundial, que se prevé que alcance cerca de 10,1 mil millones de personas en 2050. El grueso del crecimiento de la población mundial está ocurriendo en los países en desarrollo y existe en la actualidad una tendencia a la estacionalidad en la demanda de energía en los países industrializados, debido al aumento de la eficiencia en el consumo. Por otro lado, en la medida en que se alcance cierto nivel de progreso y satisfacción de las necesidades al nivel de los países en desarrollo, la demanda de energía tenderá a crecer, de acuerdo a la histórica ley que relaciona la energía y el crecimiento económico: un porcentaje en el crecimiento económico conlleva aproximadamente al crecimiento de un porcentaje en la demanda de energía (Bekkeheien, 1999), (USEIA, 2011).

A pesar de la gran incertidumbre con relación a qué fuentes primarias de energía serían predominantes en los distintos escenarios de crecimiento que se pueden analizar, la esperada demanda exigiría en todos los casos formas de energía más flexibles, más convenientes y limpias (Grubler, 1999).

De acuerdo a un informe de la *US Energy Information Administration* (2011), los combustibles líquidos⁵, con predominio notable del petróleo y sus derivados, el carbón y el gas constituyen las principales fuentes primarias de energía en el actual contexto del consumo global, sumando en conjunto cerca del 85% de la energía comercializable consumida en 2008. El 15% restante corresponde a las fuentes nucleares (5%) y otras fuentes de energía (10%), estando aquí incluidas las fuentes renovables, tales como hidroeléctrica, eólica, solar, geotérmica, etc., véase la Figura 1.

⁵ Incluye a combustibles líquidos derivados del petróleo y no derivados del petróleo, tales como etanol y biodiesel, carbón licuado y gas licuado. Se incluye también coque de petróleo, que es sólido, gas natural líquido, óleo crudo consumido como combustible e hidrógeno líquido.

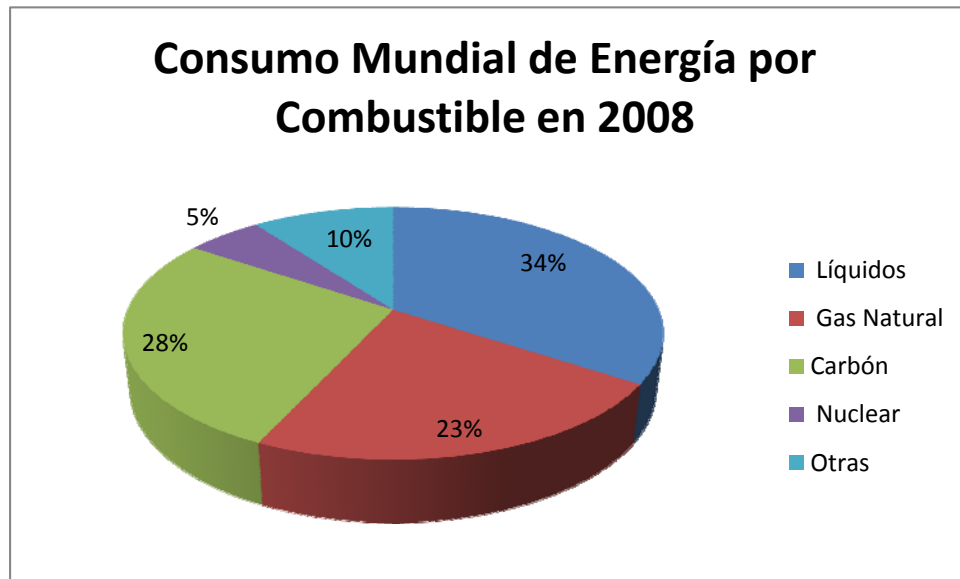


Figura 1: Fuentes primarias de energía comercializable consumida en 2008
(fuente: http://www.eia.gov/forecasts/ieo/ieo_tables.cfm)

Sin embargo, dos grandes fuentes de incertidumbre condicionan el papel jugado por los combustibles fósiles en el futuro. La primera tiene que ver con el carácter limitado de estos recursos, por lo que su explotación intensiva a los ritmos actuales o superiores conduciría inevitablemente a una situación de escasez. La segunda está relacionada con los efectos ambientales derivados de su explotación y consumo, existiendo cada vez una mayor preocupación con relación a la emisión de gases y residuos sólidos asociados al consumo de este tipo de combustibles.

En el extremo opuesto de las preocupaciones ambientales se tienen las fuentes de energía renovables (hidroeléctrica, eólica, solar, geotérmica, etc.). En el mismo contexto, estas constituyen apenas un 10% de la energía global consumida, con claro predominio de la fuente hidroeléctrica sobre las demás, con un 85% de la energía producida dentro de las energías renovables en 2008. A pesar de las propiedades positivas en términos medioambientales y de seguridad, la mayoría de las tecnologías renovables, excepto la hidroeléctrica, no son de momento competitivas desde el punto de vista económico con respecto a los combustibles fósiles, es decir, su explotación es relativamente cara. Su predominio futuro depende en alta me-

dida de su abaratamiento debido al posterior desarrollo tecnológico y a las medidas políticas y ambientales a tomar por los distintos gobiernos en el sentido de incentivar la actividad de investigación en torno a las mismas y su explotación (USEIA, 2011).

La vía nuclear constituye una importante fuente primaria. La fisión nuclear consiste en la división de núcleos de elementos pesados acompañada de la liberación de energía. La energía generada por esta fuente constituye, de acuerdo al mismo informe (USEIA, 2011), el 5% de la energía consumida globalmente. Sin embargo, existe una creciente preocupación a nivel de la opinión pública en cuanto a la seguridad en la explotación de las centrales nucleares y del manejo de sus desechos (residuos radiactivos), estando claramente enraizados en las mentes de la sociedad eventos como las explosiones de las bombas atómicas sobre Hiroshima y Nagasaki a finales de la II Guerra Mundial, la bomba de hidrógeno en 1952, y los accidentes ocurridos en las centrales nucleares de Chernóbil en 1986 o, más recientemente, de Fukushima, en 2011.

Una alternativa puede ser la fusión nuclear, basándose esta en la unión de núcleos atómicos ligeros. De acuerdo a McCracken y Stott (2005), una vez demostrada su viabilidad científica, la fusión en términos de seguridad presenta varias ventajas sobre la fisión:

- No existe la posibilidad de que un dispositivo de fusión explote o se quede fuera de control. A diferencia de los dispositivos de fisión que contienen una cantidad de combustible nuclear significativa que es capaz de mantenerla en funcionamiento durante varios años, un dispositivo de fusión contiene apenas una pequeña cantidad de combustible, que es suficiente para mantener la reacción durante unos segundos, siendo por tanto necesario un suministro continuo de combustible.
- El ciclo de fusión no produce sustancias radiactivas semejantes a las producidas en el ciclo de fisión, no existiendo la necesidad de transportar y manipular sustancias ra-

diactivas. El residuo producido en este caso es gas helio el cual no es ni tóxico ni radiactivo. El tritio utilizado como combustible es radiactivo, pero es producido *in situ* en la planta e inmediatamente consumido en la misma.

- La estructura del dispositivo de fusión puede verse afectada desde el punto de vista radiactivo debido al bombardeo de los neutrones emitidos durante las reacciones, siendo necesarias las debidas precauciones en caso de realizar reparaciones o sustituciones. En este aspecto los dispositivos de fusión y fisión son similares; sin embargo el tiempo de vida de estos desechos estructurales en la fusión es mucho menor que el de los residuos de los combustibles de la fisión nuclear.

Por otro lado, desde el punto de vista económico, proyecciones hechas por diferentes grupos muestran que los costes de la electricidad producida a través de la fusión nuclear podrían ser competitivos al estar disponible toda la tecnología necesaria (McCracken, 2005).

De esta manera, se puede tener en perspectiva un papel importante a jugar por la fusión nuclear en respuesta a las demandas energéticas del futuro. En las siguientes secciones se analizan algunos aspectos relativos a la física subyacente en la fusión termonuclear y a la tecnología relativa a su explotación de forma controlada.

2.2. FUSIÓN TERMONUCLEAR Y FÍSICA DEL PLASMA

A principios del siglo XX, los avances a nivel de la física permitieron dar respuesta a una cuestión, que desde hace muchas generaciones desconcertaba a la humanidad: ¿cuál es el origen de la energía irradiada por el Sol y las estrellas? Ya por aquel entonces se estableció que esa energía se debía a procesos de fusión nuclear que ocurren en el interior de los referidos cuerpos celestes y para tal conclusión concurrieron fundamentalmente tres avances científicos, (i) la deducción de la equivalencia entre masa y energía por parte de Albert Einstein, (ii) la invención del espectrógrafo de masa por Francis Aston, el cual permitió la medición precisa

de las masas atómicas de diferentes elementos químicos y (iii) la proposición por parte de Arthur Eddington y otros de que en el Sol y las estrellas se podía producir la conversión de masa en energía a través de la combinación de átomos ligeros. Este cuadro finalmente se completó, propiciándose un entendimiento claro de la física de la fusión nuclear, con el desarrollo de la mecánica cuántica a finales de los años 20 (Braams, 2002), (McCracken, 2005).

La posibilidad de convertir la energía de enlace nuclear en calor de forma controlada en las condiciones de la Tierra abre una vía con enormes potenciales para la producción de energía. En este caso, las reacciones exotérmicas ocurren cuando el producto final posee mayor energía de enlace por nucleón que los núcleos reactivos. En la Figura 2 se representa la energía de enlace por nucleón en función del número de nucleones en el núcleo atómico. En la misma se puede observar un rápido crecimiento de la energía de enlace con el aumento del número de nucleones, sobre todo en el caso de los átomos ligeros, por lo que la fusión de los referidos átomos ocurre con una significativa liberación de energía.

La mayor parte de la investigación para la obtención de reacciones de fusión ha girado alrededor de las reacciones del deuterio (^2D), entre sus propias partículas, con el tritio (^3T) o con el isótopo 3 del helio (^3He). Las referidas reacciones son presentadas a continuación (Piel, 2010):

- $^2\text{D} + ^2\text{D} \rightarrow ^3\text{T} + \text{p} + 4.0 \text{ MeV}$
- $^2\text{D} + ^2\text{D} \rightarrow ^3\text{He} + \text{n} + 3.3 \text{ MeV}$
- $^2\text{D} + ^3\text{T} \rightarrow ^4\text{He} + \text{n} + 17.6 \text{ MeV}$
- $^2\text{D} + ^3\text{He} \rightarrow ^4\text{He} + \text{p} + 18.3 \text{ MeV}$

De entre ellas la más atractiva en términos energéticos para una reacción de fusión controlada es la que ocurre entre el deuterio y el tritio, conocida como ciclo D – T, por presentar una mayor sección eficaz a menores energías cinéticas de las partículas reactantes y producir una

elevada cantidad de energía. En la Figura 3 se representa la sección eficaz de las diferentes reacciones en función de la energía del centro de masa de las partículas, siendo la misma construida a partir de los datos publicados por Bosch y Hale (1992). La sección eficaz es de gran importancia en el contexto de la fusión nuclear y expresa la probabilidad de que tenga lugar la reacción de fusión para un par de partículas, presentando la misma una alta dependencia de la energía cinética de las partículas en interacción (Bosch, 1992), (McCracken, 2005), (Piel, 2010).

Varios autores, por ejemplo Sheffield (1994), Braams y Stott (2002) y McCracken y Stott (2005), hacen referencia a que el tritio no se encuentra de forma natural en la Tierra por el hecho de ser radiactivo, siendo, por tanto, necesario producirlo a través del bombardeo de isótopos de litio (${}^6\text{Li}$ y ${}^7\text{Li}$) con neutrones,

- ${}^6\text{Li} + n = {}^4\text{He} + \text{T} + 4.80 \text{ MeV},$
- ${}^7\text{Li} + n = {}^4\text{He} + \text{T} + n - 2.47 \text{ MeV}.$

De este modo, los combustibles básicos para mantener un ciclo D – T controlado serían el deuterio y el litio, siendo ambos relativamente baratos, abundantes y fácilmente asequibles. Por un lado, cerca de 150 ppm⁶ del hidrógeno natural es deuterio, siendo este extraído del agua de manera relativamente fácil, y, por otro lado, el litio está presente en una proporción de 0,17 ppm en el agua de los océanos (Braams, 2002).

La probabilidad de que ocurran reacciones de fusión nuclear es considerable cuando las partículas en interacción poseen energías cinéticas tales que les permitan vencer la repulsión *coulombiana* entre los núcleos cargados, lo que implica el calentamiento del combustible nuclear hasta valores elevados de temperatura (del orden de decenas o centenas de kiloelectron-voltios). Por eso, las reacciones se denominan también de "*fusión termonuclear*". En ausencia

⁶ Partes por millón.

de aclaraciones adicionales, en las referencias posteriores a las mismas serán utilizados los términos “nuclear” y “termonuclear” indistintamente.

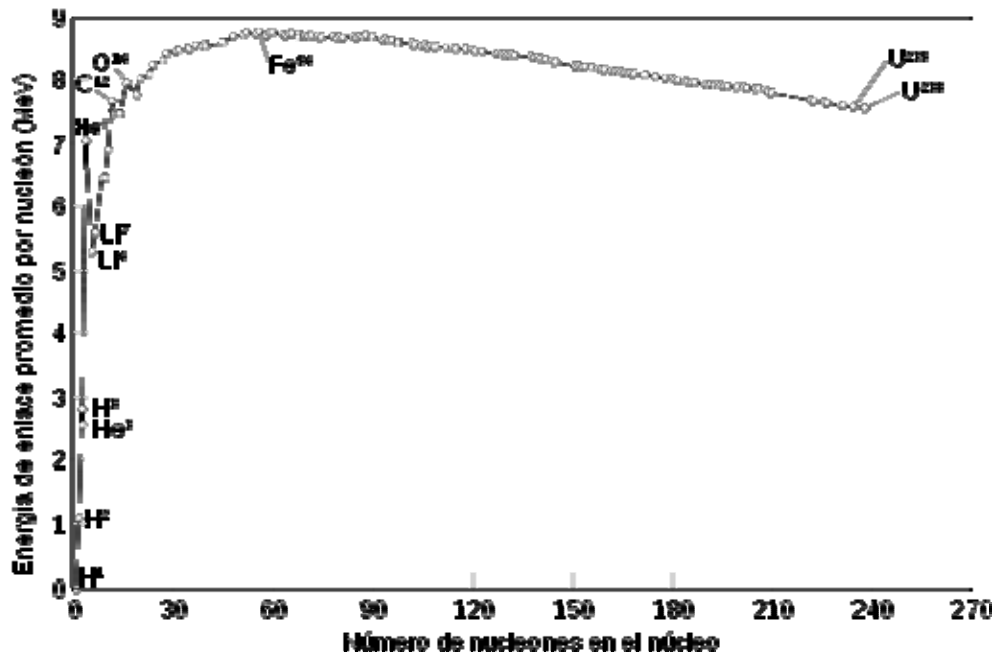


Figura 2: Energía de enlace por nucleón en función de la cantidad de nucleones en el núcleo (fuente: <http://es.wikipedia.org/wiki/Helio>)

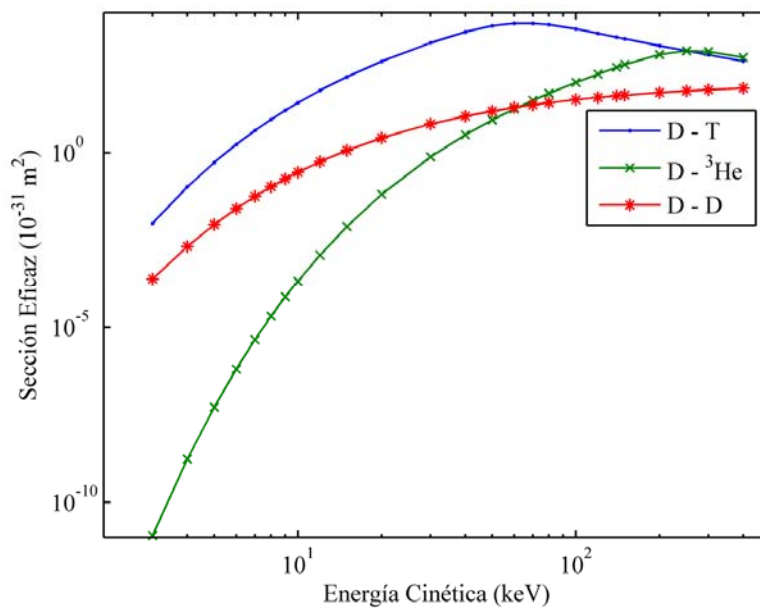


Figura 3: Secciones eficaces en función de la energía cinética para las reacciones D - T, D - ³He y D - D. La curva para la reacción D - D corresponde a la reacción ${}^2\text{D} + {}^2\text{D} \rightarrow {}^3\text{T} + \text{p} + 4.0 \text{ MeV}$

En las condiciones referidas, los materiales combustibles se encuentran en un estado físico particular denominado plasma. Como “*plasma*” se designa a un gas completa o parcialmente ionizado de manera tal que su dinámica es dominada por fuerzas electromagnéticas (Boyd, 2003), (Piel, 2010). El plasma es identificado con el cuarto estado de la materia, lo que hace alusión, por un lado, a los cuatro elementos considerados en la filosofía griega, Tierra (sólido), Agua (líquido), Aire (gaseoso) y Fuego (Boyd, 2003), (Piel, 2010). Por otro lado, desde el punto de vista fenomenológico, esa identificación se justifica por el hecho de que la división de los átomos neutros en electrones e iones está asociada a una nueva barrera de energía, la energía de ionización (Piel, 2010).

El estado de plasma posee diversas propiedades que lo diferencian de un gas neutro. A diferencia del gas neutro, en el que las partículas interactúan apenas durante las colisiones, las partículas en el plasma interactúan con una gran cantidad de otras partículas debido al predominio de las fuerzas electrostáticas de Coulomb, que son fuerzas de largo alcance. Por esto, la dinámica del plasma presenta efectos colectivos, lo que significa que el resultado macroscópico de la reacción a un estímulo externo es la respuesta cooperativa de varias partículas del plasma, siendo algunos ejemplos de este comportamiento colectivo el bloqueo mutuo de las partículas del plasma o los procesos de onda generados en el mismo (Boyd, 2003), (Piel, 2010).

En esta memoria las referencias al plasma son hechas teniendo en mente el plasma caliente producido por el hombre en la investigación de la fusión nuclear controlada. Sin embargo, se puede encontrar la materia en estado de plasma en otras condiciones, como por ejemplo el plasma natural que ocurre a nivel del sistema solar o el plasma frío de las descargas eléctricas. Un recorrido a través de la historia de la física del plasma y las diferentes condiciones en que se puede encontrar o manejar la materia en estado de plasma puede ser visto en (Piel, 2010).

Para que ocurran las reacciones de fusión nuclear es necesario que los núcleos atómicos se acerquen unos a otros a distancias comparables a su propio tamaño, ya que solamente a distancias de esa magnitud las fuerzas atractivas de carácter nuclear son suficientemente fuertes para que puedan superar las fuerzas *coulombianas*, de carácter eléctrico, que tienden a separar los referidos núcleos (McCracken, 2005). Para que ese acercamiento ocurra es necesario que las partículas del plasma sean aceleradas hasta que alcancen valores de energía muy altos. A las correspondientes temperaturas, se presenta el problema de cómo mantener las partículas del plasma juntas por un período de tiempo suficientemente largo para que las reacciones ocurran, sin pérdidas significativas de energía. Debido a que a las referidas temperaturas no es posible utilizar cualquier recipiente convencional para albergar dicho proceso, aparece el problema designado como confinamiento del plasma.

De acuerdo a Braams y Stott (2002) y a McCracken y Stott (2005), a nivel del Sol y de las estrellas el combustible nuclear permanece confinado, de forma natural, por la fuerza de gravedad, ocurriendo lo que se denomina el confinamiento gravitatorio. Las fuerzas de gravedad son débiles, en comparación con las fuerzas de origen nuclear y son capaces de confinar el plasma caliente solamente cuando su masa es extremadamente grande, como es el caso del Sol y de las estrellas, pero no en el caso de un plasma de fusión nuclear controlada que se produce en la Tierra.

En el caso de la fusión termonuclear controlada, existen dos alternativas para el confinamiento del plasma (McCracken, 2005), (Piel, 2010). La primera, denominada *fusión por confinamiento magnético* (FCM), utiliza un campo magnético para crear una barrera entre las partículas cargadas del plasma y la pared de la cámara que lo contiene. El campo magnético se utiliza para conducir el movimiento de las partículas cargadas, evitando su contacto con las paredes sólidas de la cámara y permitiendo así su confinamiento. La segunda alternativa, desarrollada inicialmente por John Nuckolls en 1957, consiste en comprimir y calentar el com-

bustible nuclear de forma tan rápida que la fusión ocurre antes de que el mismo pueda expandirse y tocar las paredes. En este caso, el plasma es confinado por un corto intervalo de tiempo, del orden de los nanosegundos, por su propia inercia, designándose por eso esta alternativa *fusión por confinamiento inercial* (FCI).

En el establecimiento de las reacciones de fusión se necesita, por un lado, una cantidad significativa de energía para calentar el combustible nuclear hasta las temperaturas requeridas y, por otro lado, el plasma caliente pierde energía de diversas formas. De este modo, la determinación de las condiciones para que las reacciones de fusión sean energéticamente rentables es una cuestión esencial, condiciones que fueron determinadas por primera vez por John Lawson en los años 50, estableciendo que es necesario mantener el producto entre la densidad del plasma (n) y el tiempo de confinamiento de la energía del plasma (τ_E) por encima de un valor determinado (Boyd, 2003), (McCracken, 2005).

Los cálculos actuales se basan en la definición del punto en el que la reacción de fusión se vuelve auto-sostenible, denominado *ignición* del plasma. McCracken y Stott (2005) muestran cómo se puede calcular la condición de ignición para una reacción de fusión D – T confinada magnéticamente. Se calcula igualando el calentamiento del plasma proporcionado por las partículas α liberadas en la reacción con la rapidez con que el plasma pierde energía (cálculos similares para el caso de la fusión por confinamiento inercial son presentados en la Sección 7.1 de la misma fuente). La referida condición se expresa de manera semejante al criterio introducido por Lawson, $n \cdot \tau_E > 1,7 \times 10^{20} \text{ m}^{-3} \text{ s}$.

En el caso de temperaturas que se encuentran entre 10 y 20 keV, por las características de la dependencia de la sección eficaz y otros parámetros con respecto a la temperatura, la condición de ignición puede ser escrita de una forma ligeramente diferente, incluyendo la tempe-

ratura, $n.T.\tau_E > 3 \times 10^{21} \text{ m}^{-3} \text{ keVs}$, siendo este triple producto el parámetro de eficiencia utilizado para comparar los experimentos de fusión (McCracken, 2005).

Los datos utilizados en esta memoria son en su totalidad relacionados con la fusión por confinamiento magnético. A continuación se hace un análisis de las características básicas de la fusión por confinamiento magnético.

2.3. FUSIÓN POR CONFINAMIENTO MAGNÉTICO

Los pasos iniciales más significativos en la investigación alrededor de la fusión termonuclear por confinamiento magnético se dieron en el periodo que siguió a la II Guerra Mundial, en países como el Reino Unido, los Estados Unidos de América o la entonces Unión Soviética, en programas rodeados de alto secretismo. Los primeros pasos para la liberalización y el intercambio de información acerca del tema se dieron con la realización de la conferencia “Átomos para la Paz” en Ginebra en 1955, mientras que la primera presentación de trabajos de investigación en el área tuvo lugar durante la segunda conferencia del género, también realizada en Ginebra en 1958 (Braams, 2002), (McCracken, 2005).

A pesar de la falta de intercambio de información existente inicialmente, las ideas básicas que regían las diferentes líneas de investigación eran muy similares. Las diferentes configuraciones del campo magnético que habían sido probadas para el confinamiento pueden, en primera instancia, ser agrupadas en dos grandes categorías, considerando el hecho de ser el campo magnético abierto en los extremos, tales como los *Z-pinch*, *theta pinch* o las *máquinas en espejo magnético*, o cerradas en forma de toro, como es el caso de los *toroidal pinch*, *estellars* o *tokamaks* (McCracken, 2005).

En las configuraciones abiertas ocurrían enormes pérdidas de plasma en los extremos y, a pesar del esfuerzo para su contención, las mismas fueron desapareciendo del panorama de la

fusión, girando la investigación actual alrededor de dos configuraciones toroidales, *tokamak* y *estellarator*.

2.3.1. Tokamak

La configuración *tokamak* fue desarrollada en Moscú, en el Instituto Kurchatov, y su designación es un acrónimo de las palabras rusas *toroidalnaya kamera magnitnaya katushka*, que pueden ser traducidas como cámara toroidal con bobinas magnéticas (Braams, 2002), (McCracken, 2005).

Varios autores, tales como Braams y Stott (2002), McCracken y Stott (2005) y Piel (2010), hacen una descripción de los elementos fundamentales de su estructura. De acuerdo a los mismos, en esta configuración el confinamiento se produce a través de un campo magnético estructurado en dos componentes, una toroidal y otra poloidal, siendo la componente toroidal típicamente 10 veces más fuerte que la componente poloidal (véase la Figura 4). La componente toroidal es generada por una bobina enrollada alrededor del toro mientras que la componente poloidal es generada por una corriente eléctrica que fluye a través del plasma, la cuál es inducida por la acción de un transformador central.

Las dos componentes del campo magnético se combinan de forma que se produzca una *transformada rotacional* del campo magnético, es decir, las líneas del campo resultante mientras giran en la dirección toroidal se tuercen alrededor del eje magnético en forma de hélice (Piel, 2010). La componente poloidal del campo determina el confinamiento del plasma mientras que el principal efecto de la componente toroidal es proporcionar estabilidad al plasma.

La mayor parte de la investigación desarrollada hasta hoy en el campo de la fusión por confinamiento magnético ha sido llevada a cabo utilizándose dispositivos de este tipo, siendo mencionables entre otros el *TFTR* en los Estados Unidos de América (PPPL, 2012), *JET* en el Reino Unido (EFDA, 2013) y *JT - 60U* en Japón (JAEA, s.f.). Por otro lado, esta tecnología

constituye la base de la construcción de un futuro dispositivo experimental de fusión, *ITER* (abreviatura del inglés *International Thermonuclear Experimental Reactor*), proyectado con una dimensión física cerca de tres veces superior a las del *JET* o del *JT-60U* y que estará ubicado en Cadarache, Francia (McCracken, 2005), (ITER Org, 2013).

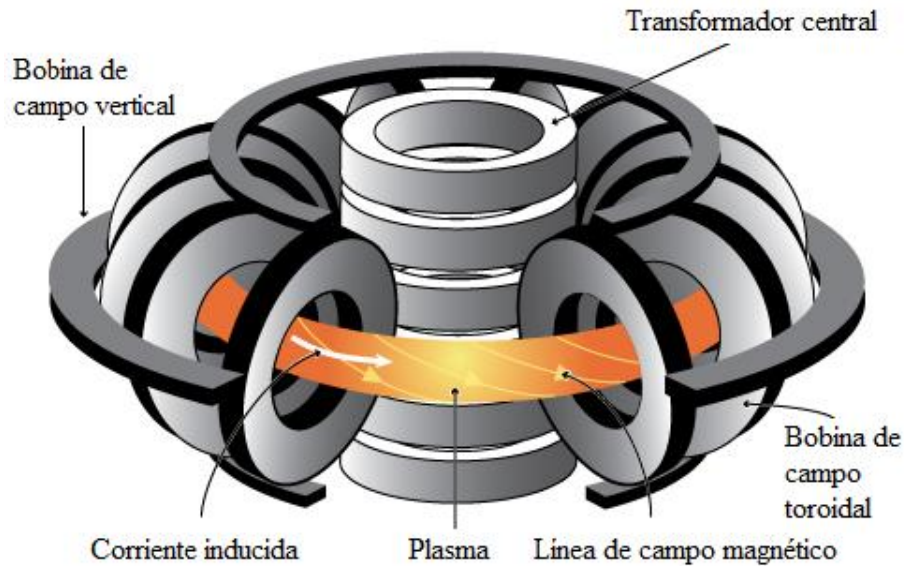


Figura 4: Vista esquemática de un *tokamak* (adaptada de http://www.generalfusion.com/magnetic_fusion.html)

2.3.2. Estellarator

El concepto de *estellarator* fue propuesto por Lyman Spitzer en 1951, en la Universidad de Princeton, y su designación se deriva de la palabra latina *stella*, que significa estrella. A semejanza del *tokamak*, el plasma se confina a través de un campo magnético compuesto por dos componentes, toroidal y poloidal, pero en este caso la *transformada rotacional* es producida completamente por corrientes externas, es decir, sin la inducción de corrientes en el plasma. La misma se logra envolviendo adicionalmente al toro con un par de bobinas en forma de

hélice, transportando corriente en sentidos contrarios, produciendo las corrientes en estos la componente poloidal del campo magnético (Braams, 2002), (McCracken, 2005), (Piel, 2010).

Los *estellarator* poseen la misma capacidad de estabilización del plasma que los *tokamak* y, por utilizar para el confinamiento corrientes que pueden ser controladas externamente, poseen un gran potencial para el funcionamiento de forma continua. Además, están exentos de disrupciones, que son fenómenos indeseables que ocurren en los *tokamak* y que consisten en la pérdida repentina de confinamiento del plasma. Sin embargo, el diseño y construcción de su sistema de bobinas adicional exige una precisión elevada y es costoso.

No obstante, a pesar del predominio actual de los dispositivos del tipo *tokamak*, el potencial de la tecnología *estellarator* para la fusión por confinamiento magnético sigue siendo reconocido y en la actualidad existen varios dispositivos de este tipo en funcionamiento o en construcción, como por ejemplo el *H – I* en Australia (ANU, 2013), el *Wendelstein 7 – X* en Alemania (IPP, 2013), el *LHD* en Japón (NIFS, 2013) o el TJ – II en Madrid (España), cuya descripción se hace en la siguiente Sección.

2.4. EL ESTELLARATOR TJ – II

El TJ – II es un *estellarator* del tipo *heliac* flexible (Harris, 1985) de tamaño medio, con 4 períodos, localizado en el *Centro de Investigaciones Energéticas Medioambientales y Tecnológicas* (CIEMAT), en Madrid. Los estudios acerca de su configuración fueron llevados a cabo a través de una colaboración entre físicos e ingenieros del CIEMAT, del *Oak Ridge National Laboratory* (ORNL, EE.UU.) y del *Institut fur PlasmaPhysik* (IPP, Garching, Alemania) (Hender, 1987).

Como todo dispositivo del tipo *estellarator* con eje magnético helicoidal, el TJ – II posee la capacidad de confinar plasmas con altos valores de *beta* (superiores a 10%) (Harris, 1985). Su diseño le dota de un alto índice de flexibilidad permitiendo, por un lado, una amplia varia-

ción de la *transformada rotacional* (entre 0,96 y 2,5) y también, y de forma independiente, de la *cizalla magnética* (entre -1 y 10%) y la profundidad del *pozo magnético* (entre 0 y 6%). Dicha flexibilidad permite, por otro lado, la variación de la posición, forma y dimensiones del plasma al cambiar de una configuración a otra, variando el radio medio del plasma entre 0,2 y 0,25 m (Alejaldre, 1990).

Su programa científico está centrado en el estudio de fenómenos de transporte en plasmas de baja colisionalidad, de los límites operacionales en plasmas con altos valores de *beta* y de optimización del confinamiento y su relación con el campo eléctrico radial, comenzando su operación en el año de 1997 (Alejaldre, 1999).

El TJ – II, la cámara de vacío, bobinas y la estructura que la soporta, tiene un diámetro de cinco metros, una altura de dos metros sobre la base de la plataforma experimental y un peso de sesenta toneladas (Rattá, 2012). Una imagen de sus instalaciones se puede ver en la Figura 5.

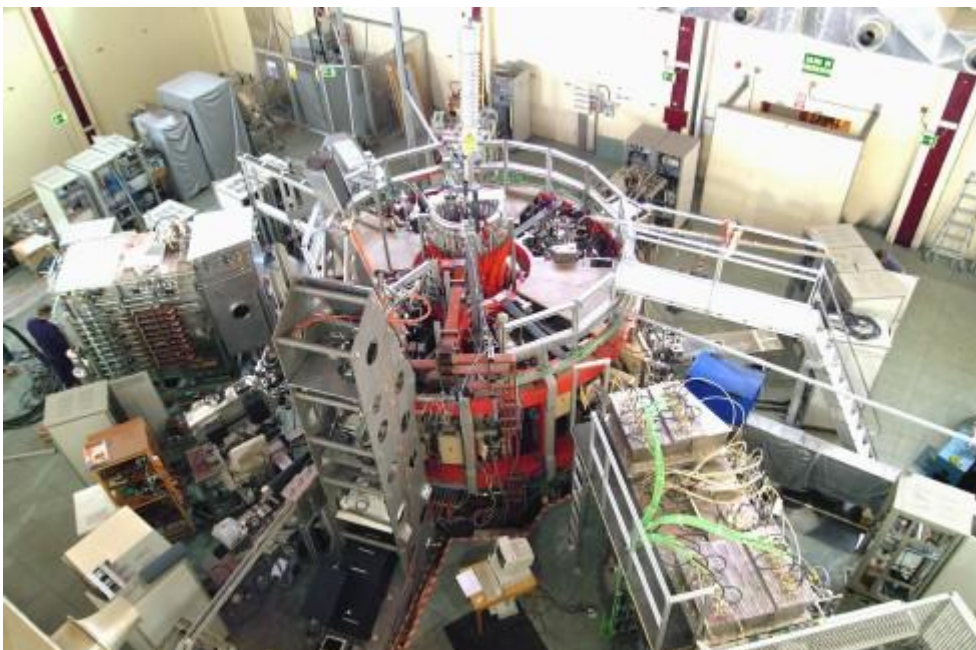


Figura 5: Instalaciones del TJ - II en el CIEMAT (fuente: <http://www-fusion.ciemat.es>)

2.4.1. Confinamiento

La trampa magnética para el confinamiento del plasma en el TJ – II es generada por un complejo sistema de bobinas, cuyo esquema se puede ver en la Figura 6. El campo magnético toroidal es creado por un conjunto de 32 bobinas cuyos centros siguen una hélice toroidal con un radio mayor de 1,5 m, un radio menor de 0,28 m y un paso que sigue a la ley $\theta = -4\Phi$, siendo θ y Φ los ángulos poloidal y toroidal, respectivamente. El núcleo del sistema, que proporciona la particularidad de flexibilidad que caracteriza el dispositivo, está constituido por dos elementos, una bobina circular coincidente con el radio mayor de la hélice toroidal y dos bobinas helicoidales enrolladas alrededor del conductor central. El sistema se completa con un conjunto de 2 bobinas de campo vertical, que controlan la posición horizontal del eje magnético (Hender, 1987), (Alejaldre, 1990). La acción combinada del sistema descrito genera un campo magnético, con un valor nominal de aproximadamente 1 T, cuyas superficies de flujo tienen la sección transversal en forma de judía y se dispone helicoidalmente, a semejanza de las bobinas del campo toroidal.

Existen además un conjunto de 4 bobinas óhmicas, cuyo objetivo es cancelar a posibles falsas corrientes toroidales, y otro de 4 bobinas radiales, utilizadas con la intención de compensar los campos parásitos. La duración nominal de la parte constante de la corriente para todas las bobinas es de 0,5 s, con una frecuencia de repetición nominal de un pulso en cada cinco minutos. Las mismas son refrigeradas directamente por agua que circula a través de orificios longitudinales existentes en los conductores y poseen una estructura reforzada para evitar deformaciones mecánicas (LNF, 2004b).

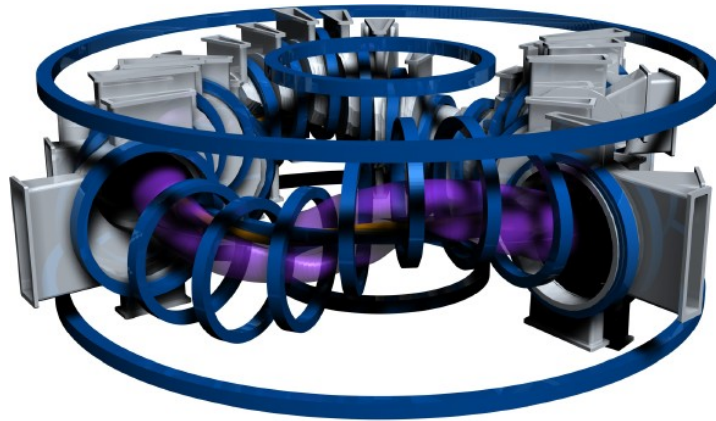


Figura 6: Vista esquemática del sistema de bobinas del TJ – II (fuente: <http://www-fusion.ciemat.es>)

El campo magnético creado por el sistema descrito confina el movimiento de las partículas del plasma en el interior de una cámara de vacío, construida en acero inoxidable con un formato helicoidal, semejante al del conductor central y las bobinas de campo toroidal, localizada en el interior de estas. El conductor central es acoplado a la superficie exterior de la cámara, existiendo para tal una ranura helicoidal cóncava en la misma. La cámara es bombeada por un sistema de cuatro bombas turbomoleculares independientes, las cuales permiten establecer en su interior una presión base inferior a 1×10^{-8} mbar (Botija, 1991, 1993).

2.4.2. Calentamiento

Para el calentamiento del plasma en el TJ – II se utilizan dos sistemas diferentes.

El primero de ellos consiste en el calentamiento a la frecuencia de resonancia electrón-ciclotrón (*ECRH*, del inglés *Electron Cyclotron Resonance Heating*) que se basa en la absorción por los electrones del plasma, por un fenómeno de resonancia, de la energía de un haz de microondas, cuya frecuencia coincide con su frecuencia ciclotrónica, o sus múltiplos (Thumm, 1995), (LNF, 2004a).

El mismo está constituido por dos *girotrones*, con 500 kW de potencia cada uno, sintonizados a la frecuencia de 53,2 GHz, que constituye el segundo armónico de la frecuencia de

resonancia ciclotrónica de los electrones del plasma en el TJ – II, y con un pulso de 1 s de duración (Tribaldos, 1998). La energía de las microondas se inyecta en el plasma a través de dos líneas de transmisión cuasi-ópticas que permiten un coeficiente de transmisión superior a 0,85, localizadas en dos posiciones toroidalmente simétricas. Para la conducción del haz de ondas de los *girotrones* hacia la cámara de vacío, cada línea utiliza un sistema de 10 espejos. Entre ellos hay uno que está localizado en el interior de la cámara de vacío, el cual posee relativa movilidad, tanto en el plano toroidal como poloidal, permitiendo la variación del ángulo de inyección del haz de ondas, garantizando una elevada flexibilidad (Fernández, 2000a, 2000b).

El segundo sistema de calentamiento se basa en la inyección de haces neutros (*NBI*, del inglés *Neutral Beam Injection*). En este caso, haces de átomos neutros de hidrógeno altamente energéticos se inyectan en el plasma precalentado, ionizándose los mismos al penetrar en el plasma, siendo, por consiguiente, atrapados por el campo magnético, calentando al plasma al colisionar con sus partículas menos energéticas.

El sistema *NBI* del TJ – II se compone de dos inyectores de haces neutros, realizando estos la inyección de las partículas en la dirección tangencial a las líneas del campo magnético toroidal, en sentidos contrapuestos. Cada uno de ellos es capaz de producir pulsos de hasta 300 ms, obteniéndose haces de partículas aceleradas hasta 40 keV y una corriente de aceleración de 100 A en la fuente de los iones, lo que permite aportar al plasma 1,2 MW de calentamiento adicional por inyector (Fuentes, 2005), (LNF, 2012).

2.4.3. *Diagnósticos* y sistema de adquisición de datos

Para el estudio de los fenómenos físicos que ocurren en el plasma, el TJ – II posee una variada gama de instrumentos de medida especialmente desarrollados para utilización en entornos de

fusión y que se denominan *diagnósticos*. El TJ – II dispone en total de un conjunto de 96 puertos de acceso.

En (Sánchez, 1998) se presenta un resumen de las diferentes técnicas de diagnóstico disponibles en el TJ – II. En el mismo artículo los *diagnósticos* son agrupados en diferentes tipos, tales como,

- *Diagnósticos* magnéticos,
- *Diagnóstico* de esparcimiento Thomson,
- *Diagnósticos* ópticos y espectroscópicos,
- *Diagnósticos* de rayos X y bolométricos,
- *Diagnósticos* de infrarrojos y microondas,
- *Diagnósticos* para la interacción plasma – pared y
- *Diagnósticos* de partículas y sondas.

Para la adquisición y manejo de toda la información generada en la operación del TJ – II, se diseñó un sistema de adquisición de datos que permite la coordinación de las acciones entre diversos sistemas experimentales dedicados a la captura y almacenamiento de datos. En general, los componentes del sistema se pueden agrupar en dos clases, por un lado, un conjunto de recursos de *hardware* que permiten, la captura de los datos, el almacenamiento y la conectividad y, por otro lado, un conjunto de recursos de *software* que permiten la minimización de las necesidades en términos de almacenamiento de los datos y proporcionan herramientas para el acceso y manejo de los mismos así como la programación de los recursos de *hardware* (Vega, 1999).

En los actuales dispositivos de fusión de mediano o gran tamaño es común que las necesidades de almacenamiento de datos asciendan hasta más de 100 MB por descarga, por lo que el

mantenimiento de un dispositivo de fusión podría exigir el almacenamiento de decenas o centenas de gigabytes mensuales (Sánchez, 1998), (Vega, 1999). Para la reducción de las referidas necesidades fue desarrollada una técnica de compresión de datos, la cual permite que sean alcanzadas tasas de compresión superiores al 80%, sin distorsión de las señales y con un esfuerzo computacional tolerable (Vega, 1996, 2007).

2.5. CONCLUSIONES

En este capítulo se hizo una introducción al campo de la fusión termonuclear. En el mismo se abordó el potencial de la fusión para la producción de energía y los conceptos fundamentales sobre fusión termonuclear y física del plasma. Por otro lado, se hizo una introducción a la fusión termonuclear por confinamiento magnético y se describieron los principales tipos de dispositivos utilizados en la investigación en el área, *tokamak* y *estellarator*. Por último, se describió un dispositivo de fusión en particular, el TJ – II.

CAPÍTULO 3: ESTIMACIÓN DE LA CONFIANZA EN EL APRENDIZAJE AUTOMÁTICO

En este capítulo se estudian y analizan algunos aspectos fundamentales relacionados con la estimación de la confianza de las predicciones en el aprendizaje automático. En la primera parte del mismo se hace una revisión bibliográfica bastante exhaustiva de los principales métodos para la estimación de la confianza, así como de las formas de representación y de evaluación de la confianza estimada, y en la segunda se abordan las principales teorías que establecen las bases sobre las cuales se desarrollaron la mayor parte de los métodos utilizados para la estimación de la confianza.

3.1. PLANTEAMIENTO Y PROBLEMÁTICA DE LA ESTIMACIÓN DE LA CONFIANZA

En las últimas décadas se ha asistido al desarrollo y a la publicación de un gran número de aplicaciones relevantes y exitosas de reconocimiento de patrones en diferentes áreas del conocimiento. Esto tuvo que ver en parte con el desarrollo teórico en el área del aprendizaje automático, lo que proporcionó la introducción de una serie de algoritmos que han traído resultados prácticos relevantes.

Sin embargo, a pesar de los apreciables resultados prácticos asociados a los referidos métodos, en muchas circunstancias existe la necesidad de ir un poco más allá de las simples predicciones puntuales usualmente ofrecidas como resultados, estimándose adicionalmente el nivel de confianza que se puede asociar a las predicciones que se realizan.

Por la propia naturaleza intrínseca de los problemas reales tratados al aplicar los referidos algoritmos, existe siempre un grado de incertidumbre alrededor de las soluciones presentadas por los mismos. De acuerdo a lo expuesto por varios autores, dicha incertidumbre se puede deber a diferentes factores, véase por ejemplo (Papadopoulos, 2001), (Shrestha, 2008) o (Seeger, 2009):

Los datos: la medición de procesos físicos es siempre propensa a imperfecciones derivadas de las limitaciones de los medios de medición empleados. Es frecuente la existencia de ruido en las medidas, en algunos casos puede suceder que la calidad de los datos sea baja o que, incluso, falten datos;

Los modelos: el modelado matemático de los problemas del mundo real tiene siempre un carácter aproximado, realizándose un conjunto de abstracciones y simplificaciones con el objetivo de hacer al problema manejable;

Los parámetros: en general, no es posible efectuar la identificación exacta de las variables existentes en los modelos a partir de un conjunto finito de datos. Habitualmente se utilizan algoritmos numéricos que aportan soluciones aproximadas, como por ejemplo los de gradiente descendiente, que en muchos casos no alcanzan los extremos globales deseados, sino extremos locales.

Diversos formalismos matemáticos fueron desarrollados con el propósito de caracterizar las incertidumbres. Cherkassky y Mulier (2007) mencionan como principales formalismos utilizados para esa finalidad en el ámbito del aprendizaje predictivo, por un lado, a la teoría de

probabilidades, siendo posible en este caso dos interpretaciones distintas del concepto de probabilidad, la “*frecuentista*” y la “*subjetivista*” o *bayesiana*, y, por otro lado, la lógica *fuzzy*. En los últimos años el estudio de la caracterización de la incertidumbre y del razonamiento impreciso ha sido objeto de elevado interés, lo que ha dado lugar a la formulación de varias teorías nuevas con este fin, como son los casos de la teoría de evidencia, la teoría de posibilidad y la teoría de incertidumbre de Boading Liu. Breves introducciones a las mismas pueden ser vistas en los artículos de Shafer (1990) o Kohlas y Monney (1994), en el caso de la primera, de Dubois y Prade (1998), en el caso de la segunda, o en el manual de Liu (2013), en el caso de la última.

La información acerca de la confianza asociada a las predicciones efectuadas es esencial en la toma de decisión en muchos campos. Límites de generalización (Vapnik, 2000), (Cherkassky, 2007), incertidumbre predictiva (Quiñonero-Candela, 2006), (Shrestha, 2006), (Todini, 2009), teoría de la predicción (Langford, 2005a), estimación de la fiabilidad (Bosnic', 2008), estimación de la confianza (Papadopoulos, 2001), son distintos términos que se pueden encontrar en la literatura, relacionada a aplicaciones del aprendizaje automático en diferentes campos del conocimiento, para la designación de esta problemática. En todos los casos se trata de dar respuesta a la cuestión de en qué medida se puede confiar o estar seguros de las predicciones hechas por un algoritmo. En esta memoria se utilizará el término “*estimación de la confianza*” para referirse a dicha cuestión.

A continuación se presentan diversas aproximaciones existentes en la literatura para la estimación de la confianza, la representación de la confianza estimada y la evaluación de las predicciones hechas por los algoritmos de aprendizaje automático.

3.1.1. Métodos para la estimación de la confianza

En la literatura se pueden encontrar una gran cantidad de métodos propuestos para la estimación de la confianza. Establecer una clasificación para los mismos es una tarea que presenta cierta complejidad debido a la multiplicidad de variantes y enfoques que aparecen en las proposiciones existentes. De todas maneras, esto se puede conseguir en alguna medida si se consideran diferentes perspectivas como criterio base para la clasificación. Así, se pueden considerar como bases para la clasificación (a) la metodología utilizada en el proceso de estimación, (b) el hecho de la estimación de la confianza ser dependiente o no de los patrones de entrada al modelo o (c) la generalidad del método de estimación.

Considerando *la metodología utilizada*, se pueden distinguir cuatro categorías fundamentales de métodos (Shrestha, 2006, 2008):

Probabilísticos: en este caso, la variable desconocida, que constituye el motivo de interés, es tratada como una variable aleatoria y las predicciones son hechas en base a distribuciones de probabilidades. En (Quiñonero-Candela, 2006) se mencionan diversos métodos probabilísticos para la estimación de la confianza. En el referido artículo se describe la idea general y los resultados de la comparación de distintos métodos probabilísticos con problemas de clasificación y regresión, utilizándose los mismos conjuntos de datos.

Estadísticos: estos estiman la confianza en las predicciones a través del análisis de los errores derivados de la aplicación de un modelo a los conjuntos de datos históricos observados. La confianza se estima con intervalos de confianza o intervalos de predicción (Heskes, 1997), (Shrestha, 2006). Los referidos intervalos se abordan en la sección siguiente.

Basados en técnicas de simulación: en estos, son utilizadas técnicas de simulación y remuestreo, conocidas como métodos de Monte Carlo, para la estimación de la confianza en las predicciones efectuadas por un modelo. Véanse, por ejemplo, en los trabajos de Kuc-

zera y Parent (1998), Hassan, Bekhit y Chapman (2009) y Zhang y otros (2012) descripciones de la aplicación de esta clase de métodos en los campos de la hidrología y la geotecnia. En estos trabajos la incertidumbre de los parámetros estructurales de los modelos utilizados se calcula mediante el empleo de técnicas de simulación y su impacto en las predicciones del modelo es cuantificada en el marco del paradigma *bayesiano*.

Basados en la teoría *Fuzzy*: constituyen una vía no probabilística de estimación de la confianza y se basa en la aplicación de conceptos acerca de imprecisión, ambigüedad o funciones de afiliación, tal como se definen en la teoría *fuzzy* (Zadeh, 1965), a la hora de modelar la incertidumbre presente en los parámetros de interés. En (Schulz, 1997), (Maskey, 2004) y (Güneralp, 2007) se presentan ejemplos de la aplicación de esta metodología para modelar problemas de toma de decisión relacionados a flujos de agua, tratamiento de precipitaciones o restauración del suelo.

Considerando *la dependencia o no de los patrones de entrada*, en las clasificaciones mencionadas en (Quiñonero-Candela, 2006) y (Bosnic', 2008), se tiene:

Estimación global: en este caso, se efectúa la estimación de un error de generalización global, el cual se utiliza como indicador de la confianza estimada para todas las futuras predicciones, independientemente de las especificidades de los vectores de entrada. El error de generalización puede ser estimado empíricamente o a través de la utilización de límites de generalización teóricos, como por ejemplo en los casos de los métodos publicados por Kohavi (1995a), Langford (2005a) o Arlot y Celisse (2010).

Estimación individual para cada vector de entrada: la confianza se estima de forma individual para cada patrón de entrada, proporcionando información local acerca de la fiabilidad de la predicción específica para cada futuro ejemplo. Se puede considerar en general la estimación hecha en este caso como superior relativamente al caso anterior, siendo esta,

por tanto, una característica deseable al efectuar la estimación de la confianza. Existen numerosas variantes de aplicación de este enfoque, como es el caso del *bagging* (Breiman, 1994), *probing* (Langford, 2005b) o el método de conversión de las salidas de clasificadores *SVM* en probabilidades a posteriori de Platt (1999).

De acuerdo a *su generalidad* se puede tener (Bosnic', 2008):

Métodos específicos para determinados modelos predictivos: constituyen extensiones de los diferentes modelos existentes para clasificación o regresión, permitiendo asociar las salidas con las correspondientes medidas de confianza. Se puede incluir en este grupo, a título de ejemplo, el método descrito por Heskes (1997) para la determinación de intervalos de predicción o el anteriormente mencionado método de Platt (1999).

Métodos independientes de los modelos predictivos: en estos, los modelos son tratados como parámetros (caja negra), siendo, por tanto, métodos generales que pueden ser utilizados con cualquier modelo predictivo. A efectos de la estimación recurren a la utilización de parámetros estándar característicos del marco de aprendizaje supervisado. En este grupo se pueden incluir los métodos similares a los descritos por Kohavi (1995a), Langford (2005a) y Arlot y Celisse (2010), el *probing* (Langford, 2005b), ya mencionado anteriormente, o el método propuesto por Ole Halck (2002).

3.1.2. Formas de representación de la confianza

A semejanza de lo que ocurre con los métodos para la estimación de la confianza, también se pueden encontrar distintas alternativas en cuanto a la forma de presentar la confianza estimada. Siguiendo el análisis de Langford (2008) y Balasubramanian (2010), se consideran cuatro alternativas fundamentales:

Probabilidades: en este caso, se considera como confianza en una predicción realizada la probabilidad de que dicho evento ocurra. La confianza en una predicción es mayor cuanto

mayor sea la probabilidad de que el evento ocurra. Esta es una variante muy utilizada y se puede encontrar su aplicación, por ejemplo, en los casos de los métodos mencionados en (Quiñonero-Candela, 2006).

Intervalos de confianza e intervalos de predicción: un intervalo de confianza es una estimación de un parámetro, o cualquier otra característica, de una población calculada a partir de una muestra aleatoria. El mismo es dado a través de un conjunto de valores que, con un grado de confianza determinado, debe contener el valor verdadero del parámetro de interés. La intuición alrededor de un intervalo de confianza para un parámetro desconocido θ es que si se calculan diferentes intervalos a partir de varias muestras aleatorias independientes, a largo plazo, una proporción determinada de los intervalos calculados debe contener el valor verdadero de θ . Dicha proporción depende del nivel de confianza considerado (Hahn, 1991). Un intervalo de predicción es una estimación en la forma de un conjunto de valores que debe contener, con un grado de confianza determinado, la siguiente observación (o cualquier otra predeterminada) seleccionada aleatoriamente a partir de una población (Hahn, 1991). Los intervalos de confianza y de predicción estiman de forma semejante la confianza en una predicción. La diferencia entre los mismos reside en el hecho de que los intervalos de confianza definen la precisión de una estimación con relación a un parámetro no observable de una población, mientras que los intervalos de predicción definen la precisión de una estimación con relación a una observación futura del valor de una variable (Heskes, 1997). En el mencionado artículo de Heskes (1997), así como en los de Shrestha y Solomatine (2006, 2008) se pueden encontrar descripciones de métodos para la estimación de intervalos de confianza y predicción aplicados a datos sintéticos en el primer caso y a datos reales provenientes del campo de la hidrología en los demás casos.

Intervalos de confianza bayesianos: constituyen el equivalente a los intervalos de confianza en el paradigma *bayesiano*, y son también denominados como *intervalos creíbles*. Se uti-

lizan con mucha frecuencia en aplicaciones de reconocimiento de patrones y desde el punto de vista práctico su utilización no difiere de la que se hace con los intervalos de confianza mencionados en el punto anterior. Más adelante se hará una breve descripción de la filosofía subyacente al paradigma *bayesiano* y, por tanto, a la determinación y utilización de los intervalos de confianza correspondientes. En la literatura se pueden encontrar una gran cantidad de aplicaciones donde la confianza se estima utilizando el marco *bayesiano*, véase por ejemplo, (Kuczera, 1998), (Hassan, 2009) o (Zhang, 2012).

Intervalos *gamesman*: este enfoque, que está siendo bastante popular, fue introducido por Vovk, Gammerman y Shafer (2005) y se deriva de la teoría de juegos. En este caso, las salidas se dan a través de conjuntos, intervalos o regiones de predicción que poseen la particularidad de contener al valor verdadero de la salida en la mayor parte de las ocasiones, siendo la proporción fijada por el usuario. En este enfoque se encuadran los predictores conformales que constituyen el principal objeto de interés de la investigación presentada en esta Tesis y que serán presentados más adelante, en el Capítulo 4.

3.1.3. Evaluación de la calidad de las predicciones

De igual forma, se está realizando un esfuerzo considerable en el sentido de la determinación de métodos que permitan evaluar el rendimiento de los algoritmos de aprendizaje automático. El “*rendimiento*” de un modelo predictivo caracteriza la calidad de las predicciones hechas por el mismo, poniendo de manifiesto en qué medida el referido modelo se aproxima al modelo verdadero desconocido. Para tal fin, se utilizan diferentes métodos siendo los mismos designados genéricamente como “*medidas de rendimiento*” (*performance metrics* en inglés), en algunos casos (Caruana, 2004), o “*funciones de pérdida*” (*loss functions* en inglés), en otros (Quiñonero-Candela, 2006).

En este caso también existen diferentes enfoques que suelen ser utilizados en distintas circunstancias. Caruana y Niculescu-Mizil (2004) clasifican en tres categorías las medidas de rendimiento: (a) medidas de umbral, (b) medidas de ordenamiento/graduación y (c) medidas probabilísticas. Los diferentes tipos de medida ponen de manifiesto distintas particularidades de los modelos de aprendizaje empleados y diversos estudios muestran que los algoritmos que presentan buenos rendimientos de acuerdo a unas medidas no lo hacen necesariamente cuando se utilizan otras, idea que es aplicable de forma particular a los casos de los algoritmos y medidas de rendimiento utilizados para estimaciones puntuales simples y los que consideran adicionalmente la confianza en las estimaciones (Caruana, 2004), (Huang, 2006), (Skalak, 2007).

Sin pretender ser exhaustivo en cuanto a la diversidad de métodos propuestos en la literatura, a continuación se presentan algunos que son comúnmente utilizados, resumidos de acuerdo a lo presentado por Blatz y otros (2004), Caruana y Niculescu-Mizil (2004) y Quiñero-Candela y otros (2006):

Error medio de clasificación: designado “*tasa de error*” o “*pérdida 0/1*” en muchas publicaciones, es el más clásico e intuitivo de los métodos y probablemente el más utilizado en el marco de la clasificación. Se define como la proporción de predicciones incorrectas hechas por un clasificador en relación al tamaño del conjunto de test. En el caso de que las salidas del clasificador sean valores continuos, se define un umbral, τ , y todos los valores por encima del umbral son predichos como positivos y los inferiores al umbral como negativos. Si se considera una función de decisión $f(\mathbf{x})$ y otra función $\delta(x, y)$ (siendo esta igual a 1 si sus argumentos son iguales y a 0, en caso contrario), el error medio de clasificación se puede definir como

$$L = \frac{1}{n} \sum_{i=1}^n (1 - \delta(f(\mathbf{x}_i), c_i)).$$

Alternativamente, se utiliza con frecuencia su valor complementario, $A = 1 - L$, designado “*tasa de acierto*” o “*precisión*” (en inglés *accuracy*).

Error cuadrático medio normalizado: es la medida de rendimiento clásica en el caso de problemas de regresión. Se define como,

$$L = \frac{1}{n} \sum_{i=1}^n \frac{(t_i - m_i)^2}{\text{var}(t)},$$

siendo m_i la media de la distribución predictiva $p(y_i | \mathbf{x}_i)$, t_i el objetivo de la predicción para cada vector de test y $\text{var}(t)$ la varianza de la muestra de los objetivos de los vectores de test.

Negativo del logaritmo de la probabilidad: es utilizado para la evaluación de la estimación de probabilidades. Se define como,

$$L = -\frac{1}{n} \sum_{i=1}^n \log p(c_i | \mathbf{x}_i).$$

De acuerdo a Quiñonero-Candela y otros (2006), este método penaliza tanto las predicciones sobreconfidentes como las subconfidentes, penalizando probabilidades predictivas iguales a cero o a uno. También se la conoce como “*negativo de la entropía cruzada*”.

En el caso de la regresión se utiliza una variante del mismo,

$$L = -\frac{1}{n} \sum_{i=1}^n \log p(t_i | \mathbf{x}_i).$$

Área bajo la curva ROC: el área bajo la curva ROC (AUC, del inglés *Area Under the ROC Curve*) es una medida de rendimiento común, utilizada con clasificadores capaces de presentar como salida un valor numérico cuya magnitud expresa el nivel de creencia en el hecho de que un punto pertenece a una clase y no a otra. La curva ROC (abreviatura del inglés *Receiver Operating Characteristic*), es un gráfico de los valores de sensibilidad vs.

(1 – especificidad) para todos los posibles valores del umbral de selección, siendo la “*sensibilidad*” definida a través de la fracción de verdaderos positivos predichos como positivos y la “*especificidad*” a través de la fracción de verdaderos negativos predichos como negativos. El área bajo la curva ROC se utiliza como una estadística acumulativa y se correlaciona con el error medio de clasificación en el sentido de que si para un clasificador el área bajo la curva es sistemáticamente mayor que para otro, entonces el error medio de clasificación para el primero es menor que en el caso del segundo. Otras propiedades importantes relacionadas con el AUC se encuentran en Hanley y McNeil (1982) o en Bradley (1997).

Pérdida LIFT: la pérdida LIFT es una medida acumulativa, definida, de manera similar al caso de la AUC, a partir del área bajo la curva de pérdida LIFT. Su valor es mínimo cuando dicha área es máxima y el área bajo la curva es igual a 1 para el predictor aleatorio medio. La pérdida LIFT se define como el área perdida por el predictor a ser evaluado con relación al predictor ideal, normalizada por el área perdida por el predictor aleatorio medio con relación al predictor ideal,

$$L = \frac{A_I - A}{A_I - 1}.$$

Detalles acerca de su cálculo pueden ser encontrados en (Caruana, 2004) o en (Quiñonero-Candela, 2006).

3.2. PRINCIPALES TEORÍAS PARA LA ESTIMACIÓN DE LA CONFIANZA

En general, se puede considerar que de momento no existe una manera obvia de determinar buenas estimaciones de la confianza en el campo del reconocimiento de patrones, o del aprendizaje automático, y existe una aparente falta de consenso en cuanto a las formas de evaluar

las estimaciones (Quiñonero-Candela, 2006). Existe un considerable esfuerzo en el sentido del desarrollo de métodos para la estimación de la confianza asociada a las predicciones, lo cual se materializa en la multiplicidad de propuestas existentes en la literatura. No obstante la dispersión de enfoques, en varios trabajos se hace referencia a que, desde el punto de vista teórico, las principales contribuciones están relacionadas a la teoría del aprendizaje estadístico y al aprendizaje bayesiano, constituyendo estas, por tanto, las principales inspiraciones de la gran mayoría de las propuestas presentadas (Papadopoulos, 2002a, 2008). Por otro lado, en la práctica existe una serie de métodos empíricos que suelen ser utilizados con resultados relativamente satisfactorios (Vovk, 2005), siendo todos en mayor o menor medida derivados del más simple de ellos, la estimación *hold-out*.

Cada una de estas líneas metodológicas posee sus ventajas y limitaciones. A continuación se analizan los elementos fundamentales de cada una, destacándose las respectivas limitaciones. Se presentan también algunos elementos acerca de la predicción conformal.

3.2.1. Teoría del aprendizaje estadístico

La teoría del aprendizaje estadístico fue introducida en los trabajos de Vapnik y Chervonenkis, véase por ejemplo (Vapnik, 2000) o (Cherkassky, 2007) para una introducción, y posteriormente, y de manera independiente, por Valiant (1984).

A diferencia de otros paradigmas estadísticos clásicos, el enfoque conceptual utilizado en la teoría del aprendizaje estadístico es el de buscar una aproximación a un sistema desconocido en lugar de buscar una identificación precisa del mismo (Cherkassky, 2007). En este caso, el objetivo es, dado un conjunto de entrenamiento, generar una función, denominada hipótesis o regla de predicción, que permita predecir la etiqueta correspondiente a un futuro objeto de test con alta probabilidad. Dado un parámetro de precisión ε y otro de confianza δ , la regla de predicción es generada de manera tal que la tasa de error asociada a su aplicación a futuros

objetos tenga como límite máximo ε con un mínimo de probabilidad de $1 - \delta$, es decir, la regla es “*probablemente aproximadamente correcta*” (PAC), y de aquí la designación del paradigma como aprendizaje PAC (Floyd, 1995).

De acuerdo a la teoría, si un determinado algoritmo puede aprender un concepto dado, para cualquier valor del parámetro de precisión ε y del parámetro de confianza δ , existe un valor fijo $n = n(\varepsilon, \delta)$ para la dimensión del conjunto de entrenamiento tal que es posible generar una regla de predicción probablemente aproximadamente correcta, lo que significa que con el aumento del tamaño del conjunto de entrenamiento se puede alcanzar un error arbitrariamente pequeño con una probabilidad arbitrariamente alta (Littlestone, 1986), (Vovk, 2005). De este modo, la teoría permite definir un límite máximo para la probabilidad de error ε asociada a la aplicación de una regla de predicción dada con respecto a un nivel de confianza $1 - \delta$.

Sin embargo, se pueden asociar diferentes restricciones a la utilización de los referidos límites teóricos. Varios trabajos mencionan que los límites derivados tanto de la ley uniforme de los grandes números de Vapnik y Chervonenkis como de la teoría del aprendizaje PAC son demasiado imprecisos para servir en la práctica de referencia de la confianza que se puede tener en las predicciones realizadas utilizando los actuales conjuntos de datos (Nouretdinov, 2001), (Papadopoulos, 2002a, 2008), (Vovk, 2005), aunque, sin embargo, se han obtenido buenos resultados al aplicar en la práctica los algoritmos derivados de las referidas teorías. A título de demostración, en dos de los mencionados trabajos, (Nouretdinov, 2001) y (Vovk, 2005), se presenta un ejemplo de aplicación de uno de los límites teóricos más precisos, el de Littlestone y Warmuth (Littlestone, 1986), (Floyd, 1995), a la identificación de dígitos de la base de datos del *US Postal Service* (USPS). En dicho ejemplo se puede ver cómo el límite para la probabilidad de error a cometer por una o varias máquinas de vectores soporte de un conjunto de diez están muy lejos de tener utilidad práctica.

Por otro lado, en (Nouretdinov, 2001) y (Papadopoulos, 2008) se destacan otros dos importantes problemas relacionados con la definición de límites de generalización teóricos. Uno de ellos tiene que ver con la utilización de constantes no especificadas y la imposición de condiciones a la distribución de probabilidades subyacente que son imposibles de verificar considerando solamente el conjunto de datos disponible. El otro está relacionado con el hecho de que los límites definidos por la teoría son para el error de forma global y no para ejemplos de test individuales.

3.2.2. Aprendizaje *bayesiano*

El aprendizaje *bayesiano* constituye una importante alternativa en las áreas de reconocimiento de patrones y aprendizaje automático, permitiendo ir más allá de la simple predicción, asociando a las mismas valores probabilísticos que estiman la confianza asociada a cada predicción realizada. Existe una gran cantidad de literatura que aborda esta metodología desde distintos puntos de vista, pudiéndose ver introducciones a la misma con una perspectiva aplicada a las técnicas de predicción utilizadas en reconocimiento de patrones, con enlaces para literatura más fundamentalista, en tutoriales y notas de clase, como son los casos de (Seeger, 2009) y (Arnborg, 2011), y libros de texto sobre reconocimiento de patrones tales como (Bishop, 1995), (Duda, 2001) y (Theodoridis, 2003).

En este caso, el problema de la toma de decisión se formula en términos probabilísticos. La base del método consiste en determinar la probabilidad a posteriori de que un patrón pertenezca a una clase determinada, a partir de nuestra representación previa de la posibilidad de que el patrón esté asociado a una clase – dada a través de una probabilidad a priori – y conocida una observación concreta del patrón. Para ello se utiliza el teorema de Bayes

$$P(C_k | \mathbf{x}_i) = \frac{P(\mathbf{x}_i | C_k)P(C_k)}{P(\mathbf{x}_i)},$$

siendo $P(C_k | \mathbf{x}_i)$ la probabilidad a posteriori, es decir, la probabilidad de que la clase es C_k después de observado el valor del patrón \mathbf{x}_i , $P(C_k)$ es la probabilidad a priori y $P(\mathbf{x}_i | C_k)$ la probabilidad condicional de \mathbf{x}_i con relación a la clase C_k que describe la distribución de los patrones \mathbf{x}_i en cada una de las clases, es decir, la verosimilitud de C_k con respecto a \mathbf{x}_i . El denominador en la fórmula es un factor de normalización, llamado evidencia, que asegura que la suma de las probabilidades a posteriori sea igual a uno, y se puede expresar en términos de las probabilidades a priori y las probabilidades condicionales relacionadas a cada una de las clases, $P(\mathbf{x}_i) = \sum_k P(\mathbf{x}_i | C_k)P(C_k)$.

La probabilidad a posteriori permite la toma de decisiones óptimas acerca de la clasificación de nuevos datos. Atribuyendo al nuevo vector de características la clase con la mayor probabilidad a posteriori se minimiza la probabilidad de error al realizar la clasificación (Bishop, 1995). Más aún, la misma permite cuantificar la incertidumbre sobrante una vez observados los datos, a través de un valor probabilístico o de la determinación de regiones en el *espacio de entradas* en las cuales los patrones pertenecen a una clase con alta confianza (Seeger, 2009).

El paradigma *bayesiano* requiere la especificación de probabilidades a priori para todas las variables, siendo, por tanto, los resultados obtenidos para las probabilidades a posteriori subjetivos, debido a que dependen de las probabilidades a priori utilizadas (Seeger, 2009). La existencia de conocimiento a priori acerca de la distribución que genera los datos es una hipótesis fuerte, difícil de cumplir para la gran mayoría de los conjuntos de datos reales, en cuyos casos es necesario, con frecuencia, utilizar una probabilidad a priori seleccionada de forma arbitraria, siendo los niveles de confianza en las predicciones realizadas incorrectos en el caso en que son asumidas probabilidades a priori incorrectas.

Este hecho se considera como la principal restricción existente al aplicar el paradigma a conjuntos de datos reales, véanse las siguientes publicaciones (Melluish, 2001), (Duda, 2001), (Papadopoulos, 2002a, 2008), (Vovk, 2005), (Balasubramanian, 2010). A título de ejemplo, Melluish y otros (2001) y Vovk, Gammerman y Shafer (2005) analizan los rendimientos resultantes de la aplicación del método de Bayes a problemas de clasificación y regresión, en el primer caso, y regresión en el segundo. Allí se muestra que en los casos de utilización de probabilidades a priori incorrectas o de utilización de conjuntos de test reales, utilizados frecuentemente como referencia en el estudio del rendimiento de algoritmos de aprendizaje automático, los valores de confianza resultantes de la aplicación de los algoritmos *bayesianos* son inexactos.

3.2.3. Estimación *hold-out*

La estimación *hold-out*, también conocida como “*validación simple*” o “*validación con conjunto de test*”, permite la determinación de la confianza asociada a la aplicación de una regla de predicción, estimando la probabilidad de error asociada a la regla y el correspondiente intervalo de confianza a partir de la observación empírica del error cometido al aplicar la regla sobre un conjunto de test.

En varios artículos se puede ver una introducción a la idea general del método, por ejemplo (Kohavi, 1995a), (Langford, 2005a), (Shah, 2008), (Arlot, 2010). El método se basa en la división de los datos en dos subconjuntos mutuamente excluyentes, denominados conjunto de entrenamiento y conjunto de test, u *hold-out*, utilizándose el primero para el entrenamiento de la regla de predicción y el segundo para la evaluación de su rendimiento.

Los valores de confianza estimados a través de este método son generales y aplicables a cualquier predicción independiente sobre un conjunto de test, independientemente de las especificidades del clasificador utilizado (Shah, 2008). A diferencia del aprendizaje *bayesiano*,

la estimación *hold-out* no hace ninguna suposición acerca de la distribución que genera los datos, basándose únicamente en la hipótesis de que los datos son independientes y están idénticamente distribuidos (iid). Esta hipótesis se utiliza en varios escenarios del aprendizaje automático, donde el conjunto de test es representativo del conjunto de entrenamiento, siendo por tanto generados por la misma distribución de probabilidades. Los límites determinados utilizando este método son relativamente buenos, en comparación con los límites teóricos establecidos por la teoría del aprendizaje estadístico. Vovk, Gammerman y Shafer (2005) presentan un ejemplo de la adecuación de los resultados obtenidos al aplicar el método al conjunto de datos del USPS.

Sin embargo, se pueden señalar varias limitaciones al método. Kohavi (1995a) y Vovk, Gammerman y Shafer (2005) señalan que el método de *hold-out* es un estimador pesimista y que hace una utilización ineficiente de los datos. Existe un ligero problema de compromiso a la hora de decidir la cantidad de ejemplos a asignar a cada subconjunto. Esto se debe a que, tanto en la tarea de entrenamiento de la regla de predicción como en la de test, los resultados son tanto mejores cuanto mayor sea la cantidad de datos en el subconjunto. Al utilizar solo una fracción de los datos para entrenamiento de la regla de predicción, usualmente no se obtiene una regla tan buena que la que se obtiene cuando se utiliza la totalidad de los datos para este fin, mientras que, por otro lado, cuanto menor es la cantidad de ejemplos en el conjunto de test, mayor será la anchura del intervalo de confianza determinado. Así, al utilizar el método siempre se tiene implícito el sacrificio de una parte de los datos, ya sea en la fase de entrenamiento o en la de test.

Kohavi (1995a) señala también que al efectuar la subdivisión de un conjunto de datos de origen, en subconjuntos de entrenamiento y de test, se viola la hipótesis de independencia entre los datos que deben integrar cada uno de los subconjuntos. Si por ejemplo una clase resulta sobrerrepresentada en uno de los subconjuntos, la misma estará subrepresentada en el

otro. Esto puede conducir en algunas situaciones a estimaciones incorrectas de la confianza, presentándose en el mismo artículo un ejemplo de esta última situación.

Por otro lado, a semejanza de lo señalado en el caso de los límites previstos por la teoría del aprendizaje estadístico, el método de *hold-out* determina un único valor para la probabilidad de error que es aplicado a todos los nuevos ejemplos, independientemente de lo difícil de predecir que estos sean (Vovk, 2005).

3.2.4. Predicción conformal

La predicción conformal (Vovk, 2005) es una teoría que permite la asociación de medidas de fiabilidad a las predicciones efectuadas por los algoritmos utilizados en el aprendizaje automático. De acuerdo a la teoría, en el proceso de predicción, las etiquetas deben ser atribuidas a los nuevos objetos de tal manera que estos se hagan en cierta medida semejantes a los ejemplos ya observados anteriormente. Las predicciones se hacen en forma de conjuntos, intervalos o regiones de predicción que deben contener al valor verdadero de la etiqueta en la mayor parte de las ocasiones, siendo esta proporción fijada por el usuario.

En los trabajos derivados de la elaboración de esta Tesis se adoptó la predicción conformal como metodología para la estimación de la confianza y, por tanto, se dedica el próximo capítulo a su descripción más detallada.

3.3. CONCLUSIONES

En este capítulo se analizó el problema de la estimación de la confianza de las predicciones hechas por los algoritmos de aprendizaje automático. En el mismo se hizo una revisión de los principales métodos para la estimación de la confianza publicados y, por otro lado, se abordaron las principales teorías que establecen las bases sobre las cuales se elaboran los referidos métodos.

CAPÍTULO 4: PREDICTORES CONFORMALES

En este capítulo se presentan los fundamentos teóricos básicos de los predictores conformales, su definición general y la metodología de su utilización para la solución de problemas de clasificación con confianza y credibilidad.

4.1. INTRODUCCIÓN

Los predictores conformales fueron desarrollados por Vovk, Gammerman y Shafer y tienen sus bases en la aplicación al aprendizaje automático de principios de aleatoriedad algorítmica, de contraste de hipótesis y de inferencia *transductiva*. Se pueden encontrar introducciones a la correspondiente teoría en diversos artículos, por ejemplo (Vovk, 1999),(Gammerman, 2007) y(Shafer, 2008), centrándose cada uno de ellos en una perspectiva diferente: desde su relación con los principios de aleatoriedad algorítmica, desde una perspectiva de aprendizaje automático, de forma directa, o desde la perspectiva de sus propiedades estadísticas. Un estudio más completo y profundo de la teoría se encuentra en el libro de los mismos autores (Vovk, 2005), donde aparecen las demostraciones de todos los teoremas subyacentes a la misma.

A semejanza de algunos otros paradigmas populares en aprendizaje automático, la teoría se desarrolla bajo la hipótesis de que las secuencias de objetos y las correspondientes etique-

tas son escogidas aleatoriamente a partir de una misma, pero desconocida, distribución de probabilidades, es decir, los objetos son independientes y están idénticamente distribuidos (iid), lo que se designa como hipótesis de aleatoriedad (en inglés *randomness hypothesis*). Sin embargo los resultados teóricos relacionados con la predicción conformal son válidos bajo la hipótesis más débil de que los ejemplos son probabilísticamente intercambiables (Vovk, 2005), (Shafer, 2008).

El concepto de intercambiabilidad se puede definir en base a la utilización de permutaciones: las variables z_1, \dots, z_n son intercambiables si para toda permutación τ de los enteros $1, \dots, n$, las variables w_1, \dots, w_n , donde $w_i = z_{\tau(i)}$, tienen la misma distribución de probabilidad conjunta que z_1, \dots, z_n (Shafer, 2008). En otras palabras, la intercambiabilidad significa que los distintos $n!$ posibles ordenamientos de las variables z_1, \dots, z_n son igualmente probables. Las variables independientes e idénticamente distribuidas son intercambiables pero lo contrario no siempre se cumple. La intercambiabilidad implica que las variables tienen la misma distribución, pero las variables intercambiables no son necesariamente independientes.

En su corta historia, desde el establecimiento de sus bases, la predicción conformal viene siendo utilizada con la finalidad de determinar la incertidumbre asociada a las predicciones proporcionadas por las técnicas tradicionales de clasificación y regresión, en el marco del aprendizaje supervisado, complementando las referidas predicciones con medidas de su fiabilidad.

En este sentido, esta metodología ha sido utilizada en combinación con diferentes técnicas de aprendizaje automático, como son los casos de k vecinos más cercanos (Duda, 2001), (Witten, 2005), centroide más cercano (también conocido como clasificador *Rocchio*) (Manning, 2008), redes neuronales (Hilera, 1995), regresión de arista (en inglés *ridge regression*) (Hastie, 2009), Máquinas de Vectores Soporte (*SVM*, abreviatura del inglés, *Support*

Vector Machines) (Vapnik, 2000), (Cherkassky, 2007) o la lógica *fuzzy* (Zadeh, 1965). Los últimos años se caracterizan por una elevada actividad sobre el tema, hecho que se consuma en la considerable cantidad de artículos publicados acerca del mismo. En los referidos artículos se presentan diversas aplicaciones relevantes y exitosas de la predicción conformal, proponiéndose soluciones para diversos problemas reales que involucran datos provenientes de diferentes y muy diversificadas áreas, como por ejemplo, datos biológicos, farmacológicos y médicos, datos relacionados al tráfico de red de ordenadores e Internet o a la monitorización de la polución ambiental, datos recolectados por sensores de diferentes tipos o, como en el presente caso, datos procedentes de dispositivos de fusión termonuclear.

Por otro lado, la aplicabilidad de este paradigma ha sido extendida más allá de los tradicionales problemas de clasificación y regresión, encontrados en el marco del aprendizaje supervisado, llegando a ser aplicado a la solución de otras clases de problemas, como son los casos del aprendizaje activo, la detección de anomalías, la selección de modelos o de características o la detección de cambios en secuencias de datos.

Un amplio resumen de las aplicaciones y extensiones del paradigma, con referencias a una serie de artículos relevantes publicados, puede ser visto en el tutorial (Balasubramanian, 2011). Por otro lado, se pueden encontrar otros ejemplos de artículos recientemente publicados en la edición especial dedicada a la predicción conformal de la revista *Progress in Artificial Intelligence* (volumen 1, fascículo 3), publicada en Septiembre de 2012, o en las memorias del *Primer Workshop sobre Predictores Conformales y sus Aplicaciones (COPA 2012)* (IFIP, 2012).

A continuación se realiza un estudio de la definición formal de predictores conformales, de su utilización en el ámbito de los problemas de clasificación, así como de las propiedades más importantes asociadas a los mismos.

4.2. TEORÍA ALGORÍTMICA DE ALEATORIEDAD Y TESTS ESTADÍSTICOS

La más profunda intuición subyacente de los predictores conformales tiene que ver con la idea de que es posible relacionar el problema de la estimación de la confianza de las predicciones con el problema de la definición de secuencias aleatorias, haciéndose corresponder un mayor nivel de confianza a las predicciones relacionadas con aquellas secuencias que presentan mayor índice de aleatoriedad. Tradicionalmente la teoría algorítmica de la aleatoriedad trata con secuencias dadas en forma de cadenas binarias. Vovk, Gammerman y Saunders (1999) hicieron la extensión de la aplicación de la teoría a secuencias formadas por ejemplos de entrenamiento clasificados y de test con sus clasificaciones provisionales (usuales en aprendizaje automático), derivando la posibilidad de su aplicación práctica a partir de la utilización de la noción de aleatoriedad introducida por Kolmogorov, que tiene la particularidad de ser aplicable a secuencias finitas y proporcionar niveles de aleatoriedad.

La noción algorítmica de aleatoriedad formaliza la noción intuitiva de tipicidad: un objeto $\omega \in \Omega$ es considerado típico de una distribución de probabilidad P sobre Ω si no hay razones para sorpresas al ser informados de que ω fue tomado aleatoriamente a partir de P (Vovk, 2005). En la teoría se utiliza la noción de deficiencia algorítmica aleatoria para la formalización del grado de desviación de la tipicidad.

De este modo, la estimación de la confianza está estrechamente relacionada al test de la aleatoriedad, o a la determinación de la deficiencia algorítmica, de secuencias aleatorias. El test de aleatoriedad definido por Martin-Lof permite establecer una vinculación entre la aleatoriedad y los contrastes de hipótesis comunes en estadística. Sea $z = (z_1, \dots, z_n)$ una secuencia de elementos de un espacio de objetos Z y $\mathcal{P} = \mathcal{P}_1, \mathcal{P}_2, \dots$ una secuencia de modelos estadísticos, tales que para todo $n = 1, 2, \dots$, \mathcal{P}_n es un conjunto de distribuciones de probabilidad

des en Z^n (Z^n representa el conjunto de todas las secuencias de elementos de Z con longitud n). De acuerdo a Vovk, Gammerman y Saunders (1999), una función $t : Z^* \rightarrow \overline{N}$ (donde \overline{N} es el conjunto $N = \{0, 1, \dots\}$ de los enteros no negativos extendido por la inclusión del infinito ∞ y Z^* el conjunto de todas las secuencias finitas de elementos de Z) es un test de aleatoriedad de Martin-Lof si:

1. para todo $n \in N$, $m \in N$ y todo $P \in \mathbb{P}_n$,

$$P\{z \in Z^n : t(z) \geq m\} \leq 2^{-m};$$

2. t es inferior semicomputable, en el sentido de que existe una *secuencia computable de funciones computables* $t_i : Z^* \rightarrow N, i = 1, 2, \dots$, tal que $t(z) = \sup_i t_i(z)$ para todo $z \in Z^*$.

Mediante un simple cambio de escala, la definición anterior puede ser reformulada de manera que sea equivalente a la definición de p -valor⁷ utilizada en los contrastes de hipótesis estadísticos. En este caso, la interpretación que se hace de la misma es que si para una secuencia determinada z se observa un valor bajo para $t(z)$, por ejemplo $t(z) \leq 1\%$, entonces los objetos en la secuencia no fueron generados de forma independiente a partir de la misma distribución de probabilidades o sucedió un evento raro (Vovk, 1999).

⁷ En los contrastes de hipótesis estadísticos se define p -valor como la probabilidad de obtención de un resultado igual o más extremo que el observado, asumiendo que la hipótesis nula es verdadera (GOODMAN, 1999).

4.3. ESCENARIOS DE APRENDIZAJE *ONLINE* Y *OFFLINE*

La teoría de los predictores conformales fue desarrollada en un escenario de aprendizaje *online* (Vovk, 2005). En este escenario, se asume que los ejemplos son presentados uno a uno. Primero, el predictor observa el objeto \mathbf{x}_1 y predice su etiqueta \hat{y}_1 ; a continuación, recibe la etiqueta verdadera y_1 , proporcionada por un “maestro”, seguidamente observa el segundo objeto \mathbf{x}_2 , y predice la etiqueta correspondiente. Este proceso sigue hasta que se realiza la predicción de las etiquetas de todos los objetos.

En el n -ésimo paso, el predictor tendrá observado los ejemplos anteriores $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_{n-1}, y_{n-1})$ y observará el nuevo objeto \mathbf{x}_n para realizar la predicción de su etiqueta. Al final de cada paso existe nueva información disponible y así la calidad de las predicciones va mejorando a medida que se incrementa la cantidad de pasos.

El escenario de aprendizaje *online* implica que la etiqueta verdadera de un objeto sea proporcionada inmediatamente después de la predicción. Esto hace que su implementación práctica sea cuestionable. En la mayoría de las aplicaciones reales cualquier forma de realimentación es costosa o simplemente no está disponible (Vanderlooy, 2006), y en función de esto se pueden considerar relajamientos al *escenario online*, como los descritos por Vovk, Gammernan y Shafer (2005, 106 : 110).

Los predictores conformales pueden ser utilizados en un escenario de aprendizaje *offline*, también conocido como *aprendizaje en lotes* (*batch learning* en inglés). En este, dado un conjunto de entrenamiento $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, el objetivo es predecir las etiquetas de los objetos dados en el conjunto de test $\mathbf{x}_{n+1}, \dots, \mathbf{x}_{n+k}$. No existe realimentación de las etiquetas verdaderas después de cada predicción y, por tanto, no existe nueva información para reentrenar al predictor.

En este escenario de aprendizaje, la propiedad de validez de los predictores conformales, que será vista más adelante en la Sección 4.7, sigue siendo válida. Desde el punto de vista teórico la garantía de validez en este escenario disminuye con respecto al *escenario online* (Vovk, 2005), pero existe una fuerte evidencia empírica de que los predictores conformales son válidos en el mismo (Vanderlooy, 2007).

4.4. TRANSDUCCIÓN E INDUCCIÓN

Además de *online*, el entorno de desarrollo de la teoría de la predicción conformal es *transductivo*, hecho que la hace diferir de la mayoría de los métodos de predicción utilizados en el aprendizaje automático, que emplean un marco inferencial *inductivo* en el proceso de aprendizaje (Vovk, 2005).

En el aprendizaje *inductivo*, el razonamiento se lleva a cabo en dos pasos, que corresponden a los dos tipos clásicos de inferencia. En el primero, se razona en el sentido del caso particular hacia el general, construyéndose una regla de predicción, más o menos general, a partir de los datos de entrenamiento disponibles, siendo este el paso *inductivo*. En el segundo, al estar disponible un nuevo objeto, se deriva una predicción a partir de la regla general. Este constituye el paso *deductivo*, en el cual se razona en el sentido del caso general hacia el particular (Vovk, 2005), (Cherkassky, 2007).

La *transducción* es una forma de inferencia en la cual el razonamiento se hace de particular hacia particular (Vapnik, 2000). Al utilizar este tipo de inferencia, la predicción de las etiquetas para cada nuevo objeto se hace directamente, a partir de los ejemplos observados anteriormente, sin la construcción de una regla de predicción general (Bosnic', 2008). En este caso se implementa, de acuerdo a Vapnik (2000), el principio que establece que al solucionar un problema se debe evitar resolver, como paso intermedio, un problema más general, siempre que la cantidad de información disponible sea limitada.

La diferencia entre los escenarios de aprendizaje *transductivo* e *inductivo* se puede analizar desde diferentes perspectivas, conceptual, matemática o computacional.

Desde el *punto de vista conceptual*, la transducción es más simple que la inducción. La inferencia *inductiva* tiene como objetivo la estimación de una función (o modelo) para todos los valores posibles del *espacio de entradas*, mientras que en la *transducción* la inferencia se implementa de forma que se estiman los valores de la función (o modelo) solamente para un conjunto discreto de puntos, lo que constituye un problema relativamente más simple (Cherkassky, 2007).

Cualquier método *inductivo* crea una regla de predicción que puede ser aplicada a diversos objetos nuevos y el objetivo es que al emplear la regla se tenga una alta probabilidad de realizar predicciones con elevada precisión. Desde el *punto de vista matemático*, esto implica el control de dos parámetros, la precisión deseada para la regla (ϵ) y la probabilidad de alcanzar dicha precisión (δ). Por tener un objetivo más modesto, los métodos *transductivos* implican el control de un único parámetro, la probabilidad del error a ser tolerado (ϵ) al aplicar el método (Vovk, 2005)

Desde el *punto de vista computacional*, en la práctica los métodos *transductivos* están asociados a un mayor coste. En los métodos *inductivos* la mayor parte del coste computacional está asociado a la creación de la regla de predicción (paso designado “*entrenamiento*”) mientras que la aplicación de la regla sobre nuevos objetos (paso designado “*test*”) es relativamente simple y tiene un coste computacional despreciable. El hecho de que la regla de predicción se crea una única vez, siendo después empleada reiteradas veces en la predicción de objetos nuevos, reduce la exigencia computacional de esta clase de métodos. En el caso de los métodos *transductivos*, no existe una separación clara entre las fases de entrenamiento y test, siendo la regla de predicción creada de forma implícita en el acto de predicción. De este modo, la exigencia computacional es más alta por ser necesario crear una regla implícita al pre-

decir cada objeto nuevo, en el caso en que los objetos se predicen de forma secuencial, uno a uno (Vovk, 2005), o formular la inferencia como un problema combinatorio, en el caso en que se realiza la predicción simultánea para un conjunto de objetos nuevos (Cherkassky, 2007).

4.5. DEFINICIÓN GENERAL DE PREDICTOR CONFORMAL

La implementación práctica de un test de Martin-Lof no es viable desde el punto de vista computacional. En función de esto, para trasladar toda la teoría a la práctica es necesaria la definición de una aproximación al referido test que sea factible desde el punto de vista computacional, lo que se hace con la introducción de la noción de medidas de no conformidad. Pero antes de pasar a definir las medidas de no conformidad, y consecuentemente los predictores conformales, se va a tratar de formalizar en términos de aprendizaje automático el problema de la predicción con confianza.

Sea una secuencia de ejemplos $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, siendo cada ejemplo (\mathbf{x}_i, y_i) , $i = 1, \dots, n$, constituido por un objeto \mathbf{x}_i y la correspondiente etiqueta y_i . Los objetos son elementos de un espacio medible X , denominado espacio de objetos, y las etiquetas de un espacio medible Y , denominado espacio de etiquetas. En una notación más compacta, z_i es utilizado para (\mathbf{x}_i, y_i) y $Z := X \times Y$ es definido como el espacio de ejemplos. El problema consiste en predecir la etiqueta y_{n+1} de un nuevo objeto \mathbf{x}_{n+1} y asociar a esta predicción alguna medida de su fiabilidad.

Se denomina “*medida de no conformidad*” (en inglés *nonconformity measure*), a una función que atribuye a cada secuencia de ejemplos $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ una secuencia de valores numéricos $\alpha_1, \dots, \alpha_n$, denominados “*valores de no conformidad*” (en inglés *nonconformity score*), que indica cuán diferente es el correspondiente ejemplo con relación a los demás, de tal forma que el intercambio de cualesquiera de los ejemplos (\mathbf{x}_i, y_i) y (\mathbf{x}_j, y_j) conlleva al

intercambio de los correspondientes valores de no conformidad α_i y α_j , permaneciendo sin alteraciones los demás valores (Gammerman, 2007).

Sin embargo, el simple examen de un valor de no conformidad α_i , por sí solo, no nos permite determinar cuán inusual es el correspondiente ejemplo z_i . Para ello es necesario comparar el referido valor con los demás α_i 's y esto se puede hacer empleándose los p -valores.

El p -valor para un ejemplo z_n , que se calcula como la razón

$$p_y = \frac{|\{i = 1, 2, \dots, n : \alpha_i \geq \alpha_n\}|}{n},$$

es un valor numérico entre $1/n$ y 1 y corresponde a la fracción de los ejemplos en la secuencia que son al menos tan no conformes como z_n . Cuanto más cerca esté su valor a 1, más típico es el ejemplo en la secuencia completa (Vovk, 2005).

Alternativamente, para el refuerzo de la propiedad de validez de los predictores conformales (véase la Sección 4.7) su cálculo puede ser redefinido como

$$p_y = \frac{|\{i = 1, 2, \dots, n : \alpha_i > \alpha_n\}| + \tau_n |\{i = 1, 2, \dots, n : \alpha_i = \alpha_n\}|}{n},$$

donde $\tau_n \in [0, 1]$ es un valor generado aleatoriamente a partir de la distribución uniforme, siendo esta variante denominada p -valor suavizado (en inglés, *smoothed p-value*) (Vovk, 2005).

Como *predictor conformal* se define una función tal que dada una secuencia de datos $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, un nuevo objeto \mathbf{x}_{n+1} y un nivel de significancia $\varepsilon \in (0, 1)$ (cuyo complementario $1 - \varepsilon$ es denominado nivel de confianza), proporciona a la salida un subconjunto $\Gamma(\mathbf{x}_1, y_1, \dots, \mathbf{x}_n, y_n, \mathbf{x}_{n+1})$ compuesto por todas las etiquetas $y \in Y$ tales que $p_y > \varepsilon$, siendo p_y

los p -valores definidos anteriormente (Vovk, 2005), (Gammerman, 2007). De igual manera se puede definir un *predictor conformal suavizado*, empleándose en este caso la variante de los p -valores suavizados (Vovk, 2005).

A diferencia de los predictores simples, usualmente utilizados en aprendizaje automático, que proporcionan una única etiqueta $y \in Y$, los predictores conformales proporcionan como salida un conjunto de etiquetas $y \in Y$ asociadas a un nivel de confianza determinado, siendo el tamaño del conjunto de etiquetas dependiente del referido nivel de confianza.

Sin embargo, esta forma de proporcionar las salidas de un predictor conformal no es única. En el caso de problemas de clasificación, las salidas del predictor pueden ser dadas a través de una etiqueta única $y \in Y$, tal como se procede habitualmente, acompañada de dos valores que resumen el alcance de los posibles conjuntos predictivos, la “*confianza*”, dada por $\sup\{1 - \varepsilon : |\Gamma^\varepsilon| \leq 1\}$ y la “*credibilidad*”, $\inf\{\varepsilon : |\Gamma^\varepsilon| = 0\}$ (Vovk, 1999), (Saunders, 1999), (Vovk, 2005). En adelante, esta será la forma de representación adoptada.

4.6. CLASIFICACIÓN CON CONFIANZA Y CREDIBILIDAD

Considerando el problema de la clasificación, que constituye el eje central del trabajo presentado en esta Tesis, y una perspectiva práctica, las ideas anteriores pueden ser unificadas mediante la formulación de un algoritmo. Asumiendo que se dispone de un recurso para el cálculo de los p -valores, siguiendo la línea establecida en (Saunders, 1999) y (Vovk, 1999), la clasificación, considerando el caso binario, se llevaría a cabo según el Algoritmo 1.

En el referido algoritmo, los valores de confianza y credibilidad sirven para estimar la calidad de la predicción. La confianza está relacionada con la probabilidad de que ninguna de las etiquetas excluidas sea la correcta. Un alto valor de confianza significa que todas las etiquetas rechazadas son improbables. Por otro lado, la credibilidad expresa la calidad de los

datos que han servido de base a la toma de decisión. Una credibilidad baja significa que el objeto de test es no conforme en todos los escenarios de clasificación posibles y, por tanto, o el conjunto de entrenamiento no es aleatorio o el objeto de test no es representativo del conjunto de entrenamiento (Saunders, 1999), (Vanderlooy, 2006).

Algoritmo 1: clasificación con predictores conformales

Entradas:

$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$	secuencia de entrenamiento
\mathbf{x}_{n+1}	objeto de test
\mathbf{Y}	espacio de etiquetas

Salidas:

Y	etiqueta predicha
Cf	confianza
Cr	credibilidad

get \mathbf{x}_{n+1} **get** $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ **for each** clase $y \in \mathbf{Y}$ **do**Formar secuencia extendida $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n), (\mathbf{x}_{n+1}, y)$ Calcular p -valor para nuevo ejemplo (\mathbf{x}_{n+1}, y) , $p_y = \frac{|\{i = 1, 2, \dots, n+1 : \alpha_i \geq \alpha_{n+1}\}|}{n+1}$ **end for** $P = \text{sort_desc}(p)$ $Y = y \Big|_{p_y = \max(p)}$ $Cr = P_1$ $Cf = 1 - P_2$

4.7. PROPIEDADES DE LOS PREDICTORES CONFORMALES

Los predictores conformales poseen un conjunto de propiedades importantes. A continuación se realiza un resumen de dichas propiedades, de acuerdo a lo descrito por Vovk, Gammernan y Shafer (2005).

4.7. Propiedades de los predictores conformales

Conjuntos de predicción anidados: los predictores conformales proporcionan como salidas conjuntos de predicción para cada nivel de confianza y estos son anidados. Esto significa que si se consideran diferentes niveles de confianza $1 - \varepsilon$ y para cada uno de ellos se especifica el correspondiente conjunto de predicción $\Gamma^\varepsilon \subseteq \mathcal{Y}$, estos estarán relacionados de tal manera que $\Gamma^{\varepsilon_1} \subseteq \Gamma^{\varepsilon_2} \subseteq \Gamma^{\varepsilon_3}$ si $\varepsilon_1 \geq \varepsilon_2 \geq \varepsilon_3$.

Validez: los predictores conformales son siempre válidos. Por validez se entiende que la frecuencia de errores cometidos por el predictor no excede ε para todo el nivel de confianza $1 - \varepsilon$. Esta, también se denomina como propiedad de calibración.

Eficiencia: una vez que los predictores conformales son automáticamente válidos, esta propiedad pasa a ser el objetivo más inmediato al implementarlos. La eficiencia significa que los conjuntos de predicción proporcionados a la salida deben ser lo más pequeños posible, siendo preferibles los conjuntos con una única etiqueta. Al ser posible la implementación de diferentes predictores válidos para un mismo problema, el hecho de que un predictor sea más o menos eficiente depende de los detalles del entorno, es decir, de las características de la distribución de probabilidades que genera los datos y de las medidas de no conformidad utilizadas.

Condicionabilidad: al determinar el conjunto de predicción correspondiente a un objeto dado, los predictores conformales consideran no solamente las particularidades inherentes al conjunto de entrenamiento utilizado sino también las características particulares del objeto a predecir, es decir, la predicción es condicionada por el propio objeto.

Flexibilidad: es posible convertir cualquier algoritmo de aprendizaje automático en un predictor conformal que, en este caso, hereda su rendimiento predictivo, pasando esto por la definición de una medida de no conformidad adecuada. El algoritmo a transformarse es designado como *algoritmo base* (en inglés *underlying algorithm*).

4.8. MÁQUINAS DE VECTORES SOPORTE Y VALORES DE NO CONFORMIDAD

En el Algoritmo 1 se puede observar que en la práctica el elemento clave para la clasificación con predictores conformales, y que particulariza las diferentes implementaciones, es el cálculo de los valores de no conformidad, es decir, la definición de una medida de no conformidad.

Se pueden utilizar diferentes alternativas para la creación de medidas de no conformidad, siendo posible convertir la mayoría de los métodos modernos de aprendizaje automático en predictores conformales. En esta Tesis se utilizaron las Máquinas de Vectores Soporte (*SVM*, abreviatura del inglés *Support Vector Machines*) (Vapnik, 2000), (Cherkassky, 2007) como *algoritmo base*. A continuación se analizan los elementos básicos relativos a las mismas y como utilizarlas para el cálculo de los valores de no conformidad.

Las máquinas *SVM* constituyen una técnica de aprendizaje universal basada en la teoría del aprendizaje estadístico. La idea fundamental subyacente a las mismas es, dado un conjunto de n ejemplos de entrenamiento

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n), \mathbf{x} \in R^d, y \in \{+1; -1\}$$

determinar un hiperplano $D(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b$ que permita alcanzar la mejor separación entre dos clases, con tal de que el riesgo empírico sea minimizado y el margen maximizado para los vectores correctamente clasificados. En la Figura 7 se muestra una representación esquemática de la solución de un problema de clasificación con una máquina *SVM*, para el caso de un conjunto de datos no separable.

Este es un problema de optimización cuadrática. El mismo se resuelve determinando \mathbf{w} y b tales que sea minimizada la función

$$C \sum_{i=1}^n \xi_i + \frac{1}{2} \|\mathbf{w}\|^2$$

impuestas las restricciones

$$y_i[(\mathbf{w} \cdot \mathbf{x}_i) + b] \geq 1 - \xi_i, \quad i = 1, \dots, n$$

y dado un valor suficientemente alto (fijo) del parámetro de regularización C .

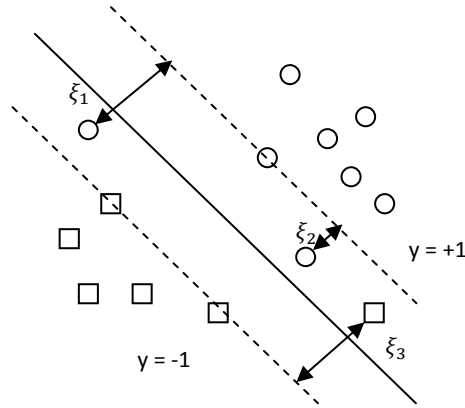


Figura 7: Clasificación binaria con SVM; datos no separables

En espacios de elevada dimensionalidad, para su solución el problema se traslada a la correspondiente forma dual. Esta puede ser formulada como sigue: determinar los parámetros $\alpha_i, i = 1, \dots, n$, que maximizan al funcional

$$L(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j),$$

impuestas las restricciones

$$\sum_{i=1}^n y_i \alpha_i = 0$$

$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, n,$$

y dados los vectores de entrenamiento $(\mathbf{x}_i, y_i), i = 1, \dots, n$ y el parámetro de regularización C , siendo $K(\mathbf{x}_i, \mathbf{x}_j)$ el *kernel* de producto escalar (Vapnik, 2000), (Cherkassky, 2007).

El *kernel* $K(\mathbf{x}_i, \mathbf{x}_j)$ es una función que representa las funciones de base utilizadas para transformar el *espacio de entradas* original en un espacio de altas dimensiones, designado

espacio de características. Su utilización permite, por un lado, la construcción de superficies de decisión no lineales en el *espacio de entradas*, una vez que los hiperplanos óptimos determinados en el *espacio de características* corresponden a superficies de decisión no lineales en el *espacio de entradas*. Por otro lado, la utilización del *kernel* soluciona el problema del cálculo de productos escalares entre vectores en el *espacio de características* (de elevadas dimensiones), siendo la evaluación de los referidos productos realizada de forma indirecta a través de la evaluación de la función *kernel* en el *espacio de entradas* (Cherkassky, 2007).

La calidad de la clasificación con máquinas *SVM* depende en alta medida de la función *kernel* empleada. En este trabajo fueron utilizadas tres funciones *kernel* distintas: lineal, polinómica y funciones de base radial (RBF), cuyas definiciones se pueden ver en la Tabla 1.

Como solución al problema se obtiene una función de la forma

$$D(\mathbf{x}) = \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b, \text{ donde los coeficientes } \alpha_i \text{ son las soluciones del problema dual y}$$

b es dado por la fórmula $b = y_s - \sum_{i=1}^n \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_s)$, siendo (\mathbf{x}_s, y_s) los vectores soporte.

Los multiplicadores de Lagrange, α_i , obtenidos al solucionar el problema dual son ideales para utilizarse como valores de no conformidad (Vovk, 2005). Cada coeficiente α_i dice cuán extraño es el correspondiente ejemplo (\mathbf{x}_i, y_i) con relación a los demás ejemplos en el conjunto de entrenamiento:

Kernel	Definición
Lineal	$K(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$
Polinómico de grado d	$K(\mathbf{x}, \mathbf{x}') = (\langle \mathbf{x}, \mathbf{x}' \rangle + 1)^d$
RBF	$K(\mathbf{x}, \mathbf{x}') = \exp\left\{-\frac{\ \mathbf{x} - \mathbf{x}'\ ^2}{2\sigma^2}\right\}$

Tabla 1: Funciones *kernel* empleadas

- Si $\alpha_i = 0$, el ejemplo es típico;
- Si $\alpha_i > 0$, el ejemplo es un vector soporte; el mismo es atípico y el valor de α_i puede ser interpretado como una medida de su no conformidad.

Originalmente el método *SVM* resuelve problemas binarios. Existen dos métodos estándar que permiten reducir un problema multiclase al caso binario: “*uno-contra-el-resto*” y “*uno-contra-uno*”. Supóngase que se tiene definida una medida de no conformidad razonable A para la clasificación binaria, asumiendo que el espacio de etiquetas en el problema binario es $\{0, 1\}$. La utilización de las medidas de no conformidad en un problema multiclase, de acuerdo a los métodos mencionados anteriormente, se llevaría a cabo como se explica a continuación (Vovk, 2005), (Nouretdinov, 2010).

En el método *uno-contra-el-resto*, se utiliza como medida de no conformidad

$$A((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), (\mathbf{x}, y)) = \lambda A((\mathbf{x}_1, \mathbf{I}_{y_1=y}), \dots, (\mathbf{x}_l, \mathbf{I}_{y_l=y}), (\mathbf{x}, 1)) + \frac{1-\lambda}{|\mathbf{Y}|-1} \sum_{y' \neq y} A((\mathbf{x}_1, \mathbf{I}_{y_1=y'}), \dots, (\mathbf{x}_l, \mathbf{I}_{y_l=y'}), (\mathbf{x}, 0)); \quad (1)$$

en este caso, $\lambda \in [0, 1]$ es un parámetro constante e \mathbf{I} la función indicadora, la cual dado un

conjunto B se define como $\mathbf{I}_B(x) = \begin{cases} 1 & \text{si } x \in B \\ 0 & \text{si } x \notin B \end{cases}$.

En el método *uno-contra-uno*, la medida de no conformidad sería,

$$A((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), (\mathbf{x}, y)) = \frac{1}{|\mathbf{Y}|-1} \sum_{y' \neq y} A(B_{y, y'}, (\mathbf{x}, 1)); \quad (2)$$

en este caso, $B_{y, y'}$ es un conjunto de entrenamiento reducido, utilizado para resolver los problemas de clasificación binarios auxiliares, que se determina de la siguiente manera:

- Tomar del conjunto de entrenamiento todos los (\mathbf{x}_i, y_i) tales que $y_i \in \{y, y'\}$;

- Substituir cada (\mathbf{x}_i, y) por $(\mathbf{x}_i, 1)$;
- Substituir cada (\mathbf{x}_i, y') por $(\mathbf{x}_i, 0)$.
- Para los ejemplos (\mathbf{x}_j, y_j) tales que $y_j \notin \{y, y'\}$, se utiliza $\alpha_j = 0$ como los correspondientes valores de no conformidad.

4.9. LIMITACIONES DE LOS PREDICTORES CONFORMALES

En la Sección 4.7 se mencionaron algunas propiedades deseables de los predictores conformales. Sin embargo, en el marco de la predicción conformal hay que tener en cuenta algunas limitaciones. A continuación se hace una breve descripción de las dos principales limitaciones, de acuerdo a lo expuesto por Vovk, Gammerman y Shafer (2005).

Falta de condicionalidad de la validez: no obstante por el hecho de que los predictores conformales sean condicionales, no son automáticamente condicionalmente válidos, es decir, no se cumple la propiedad de validez para cada clase o categoría de objetos en particular. Suponiendo que se tiene un conjunto de datos en el cual existen dos categorías de objetos, “fáciles” (de predecir) y “difíciles” (de predecir), la propiedad de validez no se alcanza automáticamente para las dos categorías de objetos por separado, como sería de esperar. Por ejemplo, para un predictor con un nivel de confianza del 95% la tasa de error puede ser significativamente superior al 5% para los objetos difíciles e inferior al 5% para los fáciles, permaneciendo cerca del 5% si se considera la totalidad de los objetos simultáneamente. La validez condicional es una propiedad deseable, particularmente en los casos de clasificación asimétrica, donde el coste del error es función de la clase real a la que pertenece el objeto erróneamente clasificado.

Ineficiencia computacional: la predicción conformal es relativamente ineficiente desde el punto de vista computacional, una vez que es necesario recalcular los valores de no con-

formidad para cada uno de los objetos a predecir. Esto se agudiza de forma particular en el caso de la utilización de clasificadores *SVM* como *algoritmo base*, existiendo la necesidad de entrenar varias máquinas para la predicción de cada objeto. A medida que la cantidad de ejemplos en el conjunto de entrenamiento va creciendo, el problema con máquinas *SVM* se hace menos tratable desde el punto de vista computacional.

La primera limitación se soluciona mediante la introducción de una variante denominada *predictores conformales Mondrianos* (Vovk, 2005), los cuales permiten reforzar la propiedad de validez de la predicción conformal. Si se consideran varias categorías de ejemplos, donde cada categoría puede corresponder a diferentes etiquetas o a diferentes tipos de objetos, los predictores conformales Mondrianos garantizan la validez intracategoría. Esto significa que la propiedad de validez se cumple no solamente al considerarse la predicción de todos los objetos en general, sino que se cumple automáticamente para cada categoría en particular. Su estudio se encuentra fuera de los objetivos de esta Tesis.

En contrapartida para resolver la segunda de las limitaciones apuntadas, se han propuesto diversas variantes de clasificadores conformales. En las siguientes secciones se hace una descripción de algunas de esas variantes a las cuáles se recurrió en los marcos del trabajo de investigación subyacente a esta Tesis.

4.10. VARIANTES DE CLASIFICADORES CONFORMALES

En el Algoritmo 1 (Sección 4.6) se puede ver que el punto con una mayor carga computacional reside en el paso donde se calculan los valores de no conformidad y los p -valores para los distintos objetos de test y, por tanto, cualquier tentativa de mejoría de la eficiencia computacional del algoritmo pasaría por mejorar la eficiencia del proceso de cálculo en el referido paso.

En esta sección se hace la descripción de algunas variantes de clasificadores conformales cuyo principal objetivo es la mejoría de la eficiencia computacional con relación a la versión estándar. En la descripción de las mismas se considerará un marco típico de clasificación tal como el considerado en la Sección 4.5, y que aquí se vuelve a presentar por una cuestión de claridad, es decir, dado un conjunto de entrenamiento compuesto por n ejemplos $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$, donde $\mathbf{x}_i \in R^d$ constituyen los objetos e $y_i \in \{1, 2, \dots, k\}$ las correspondientes etiquetas, el objetivo es la clasificación de un nuevo objeto \mathbf{x}_{n+1} en una de las k clases y la complementación de la clasificación con medidas de fiabilidad.

4.10.1. Clasificación con función *hash*

Este método fue introducido por Saunders, Gammerman y Vovk (2000). Para efectos de la clasificación se considera una función *hash* $f_h : R^d \rightarrow \{1, 2, \dots, h\}$. Dado un ejemplo de entrenamiento (\mathbf{x}_i, y_i) , la función *hash* devuelve un valor entero en el rango $\{1, 2, \dots, h\}$ de forma tal que permite la división del conjunto de entrenamiento en varias partes aproximadamente iguales. Para ello, desde el punto de vista matemático, la función debe ser definida de tal forma que sea “pseudo-aleatoria”. A semejanza de lo descrito por Saunders, Gammerman y Vovk (2000), la función utilizada en esta Tesis calcula el módulo h de la suma de los valores

de las características de un objeto más uno, $f_h(\mathbf{x}) = [\text{round}(\sum_{l=1}^d x_l) \bmod h] + 1$.

Esta función permite dividir el conjunto de entrenamiento en varios problemas binarios, de forma semejante a lo que se hace al emplear la estrategia multiclase *uno-contra-el-resto* y con la misma se pueden crear $k * h$ subconjuntos del conjunto de entrenamiento. Para cada posible salida j de la función *hash* se forma un subconjunto S_{cj} que incluye, en primer lugar, todos los ejemplos que pertenecen a la clase c a ser considerada y que no devuelven un valor

igual a j al aplicarse sobre los mismos la función f_h y, en segundo lugar, todos los ejemplos que pertenecen a las demás clases, es decir,

$$S_{cj} = \{(\mathbf{x}_i, 1) : y_i = c \wedge f_h(\mathbf{x}_i) \neq j\} \cup \{(\mathbf{x}_i, -1) : y_i \neq c\}.$$

Para cada subconjunto S_{cj} es posible entrenar un clasificador *SVM*, obteniéndose una función de decisión de la forma $F_{cj}(\mathbf{x}) = \sum_{i:(\mathbf{x}_i, y_i) \in S_{cj}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x})$, tal como se describió en la Sección

4.8. La función $F_{cj}(\mathbf{x})$ determina la distancia del vector \mathbf{x} con relación al hiperplano de separación y $\text{signo}(F_{cj}(\mathbf{x}))$ se toma como regla para su clasificación como perteneciente a la clase c o no.

Para cada subconjunto S_{cj} se hace la extracción, desde el conjunto de entrenamiento original, de un correspondiente subconjunto de trabajo

$$W_c = \{\mathbf{x}_i : f_h(\mathbf{x}_i) = j, y_i = c, i = 1, \dots, n\},$$

el cual incluye todos los ejemplos del conjunto de entrenamiento que son miembros de la clase c a ser considerada y devuelven un valor igual a j cuando la función *hash* es aplicada sobre ellos. Nótese que ninguno de los ejemplos incluidos en el subconjunto de trabajo W_c está presente en el correspondiente subconjunto S_{cj} utilizado para el entrenamiento de la máquina *SVM*.

Para efectos de clasificación de un nuevo ejemplo \mathbf{x}_{n+1} , la función $F_{cj}(\mathbf{x})$ se utiliza en el cálculo de las distancias, d_i , con relación al hiperplano para cada elemento del subconjunto de trabajo y para el nuevo ejemplo. Estas distancias se utilizan como valores de no conformidad, siendo tanto mayor la no conformidad cuanto menor sea la misma. La distancia calculada a través de la función $F_{cj}(\mathbf{x})$ puede ser negativa y, por tanto, el más inusual de los objetos

sería el más distante del hiperplano de separación, en el caso de que existan objetos en el lado negativo del hiperplano, o el más cercano al hiperplano, si todos los objetos se encuentran en el lado positivo. Una vez definido el valor de no conformidad, se calculan los p -valores,

$$p_c = \frac{\#\{i : d_i \leq d_{new}\}}{|W_c| + 1}.$$

Resumiendo, como procedimiento práctico para la clasificación de un nuevo ejemplo, véase el Algoritmo 2. En el Anexo B se presenta un ejemplo de utilización del referido algoritmo con un conjunto de datos concreto. En el ejemplo se hace énfasis en el procedimiento utilizado para la formación de los subconjuntos de trabajo y de entrenamiento, a partir de los datos de entrenamiento originales, y la subsiguiente formulación de los problemas binarios auxiliares.

La mejoría esperada en la eficiencia computacional se debe esencialmente al hecho de que en este caso para cada problema binario se resuelve únicamente un problema de optimización para cada objeto de test, a diferencia de los dos problemas resueltos en la versión estándar, lo que puede ser significativo en el caso de problemas multiclase.

En contrapartida, en este método se efectúa la subdivisión de los datos en subconjuntos de entrenamiento y de trabajo lo que conlleva un problema de equilibrio entre la obtención de una mejor función de decisión, que ocurre al atribuir una mayor cantidad de datos al subconjunto de entrenamiento, y la mejoría de la calidad de los p -valores, cuando se asigna una mayor cantidad de datos al subconjunto de trabajo. Este problema de equilibrio es significativamente importante cuando la cantidad de datos disponible para el entrenamiento es limitada y va disminuyendo a medida que va creciendo el conjunto de datos disponible.

Algoritmo 2: clasificación conformal con función *hash***Entradas:**

$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$	secuencia de entrenamiento
\mathbf{x}_{n+1}	objeto de test
\mathbf{Y}	espacio de etiquetas
h	partes a dividir el conjunto de entrenamiento

Salidas:

Y	etiqueta predicha
Cf	confianza
Cr	credibilidad

get \mathbf{x}_{n+1}

$$j = f_h(\mathbf{x}_{n+1})$$

for each clase $c \in \mathbf{Y}$ **do**

$$W_c = \{\mathbf{x} : f_h(\mathbf{x}_i) = j, y_i = c, i = 1, \dots, n\}$$

$$S_{cj} = \{(\mathbf{x}_i, 1) : y_i = c, f_h(\mathbf{x}_i) \neq j\} \cup \{(\mathbf{x}_i, -1) : y_i \neq j\}$$

Determinar la función de decisión $F_{cj}(\mathbf{x})$, entrenando *SVM* con S_{cj} **for each** $\mathbf{x} \in W_c$ **do**

$$\text{Calcular distancias } d_i = \{F_{cj}(\mathbf{x})\} \cup \{F_{cj}(\mathbf{x}_{n+1})\}$$

end for

$$d_{n+1} = F_{cj}(\mathbf{x}_{n+1})$$

Calcular *p*-valor para nuevo ejemplo, $p_c = \frac{\#\{i : d_i \leq d_{n+1}\}}{|W_c| + 1}$ **end for**

$$P = \text{sort_desc}(p)$$

$$Y = c \Big|_{p_y = \max(p)}$$

$$Cr = P_1$$

$$Cf = 1 - P_2$$

4.10.2. Cálculo de valores de no conformidad con *SVM* incremental

En la versión estándar del clasificador, descrito en la Sección 4.6, varias máquinas *SVM* son entrenadas para la clasificación de un único objeto de test y existe la necesidad de reentrenar las máquinas a medida que se tienen nuevos objetos para clasificación. Entrenar las máquinas

SVM de raíz para cada nuevo objeto de test conlleva una significativa sobrecarga computacional, por lo que una solución adecuada para la mejora de la eficiencia computacional podría ser la realización del entrenamiento en forma incremental (Ho, 2004).

Diferentes enfoques del entrenamiento incremental de máquinas *SVM* han sido propuestos para tratar de abordar problemas relacionados con el aprendizaje a partir de conjuntos de datos muy grandes o con el aprendizaje activo. Algunos de esos enfoques utilizan la propiedad de que los vectores soporte condensan toda la información (contenida en el conjunto de entrenamiento) necesaria para la definición de la función de decisión para la reducción del tamaño del referido conjunto a medida que se acumulan nuevos ejemplos, lográndose soluciones aproximadas al problema del entrenamiento incremental (Syed, 1999a), (Rüping, 2001). Otros utilizan aproximaciones no estándar para la solución del problema de optimización subyacente y no calculan en ese proceso los multiplicadores de Lagrange, los cuales se han utilizado en esta Tesis como valores de no conformidad (Fung, 2002), o utilizan una optimización estocástica con gradiente descendiente en un espacio de Hilbert (Kivinen, 2004). Una solución exacta al problema del entrenamiento incremental de máquinas *SVM* cuando se añade un ejemplo al conjunto de entrenamiento (o se elimina) fue introducida por Cawenberghs y Poggio (2001), siendo generalizada más tarde por Diehl y Cawenberghs (2003). Otra solución similar fue la propuesta por Shilton y otros (2005).

En esta Tesis, como *algoritmo base*, se consideran dos enfoques distintos para el entrenamiento incremental de máquinas *SVM*, el enfoque de Syed, Liu y Sung (Syed, 1999a) y el de Cawenberghs y Poggio (Cawenberghs, 2001). A continuación se describen ambos enfoques.

A. Enfoque de Syed, Liu y Sung (SVMi_SLS)

Se basa en el hecho de que el algoritmo *SVM* conserva la información esencial acerca de la función de decisión en forma de un conjunto de puntos, los vectores soporte, que se guardan

para clasificaciones futuras. En general, solamente una pequeña parte de los datos de entrenamiento se convierten en vectores soporte, lo que significa que el algoritmo puede preservar las propiedades del espacio de datos de una forma concisa y que entrenar una máquina *SVM* utilizando solamente los vectores soporte proporciona la misma función de decisión que se obtiene cuando se entrena con el conjunto de datos de entrenamiento completo.

De esa manera, si se entrena una máquina *SVM* en varios pasos, manteniendo los vectores soporte de los pasos anteriores y se les añade el conjunto de datos a ser utilizado para el entrenamiento del paso siguiente, se obtiene un efecto incremental y al final del proceso se obtiene una función de decisión similar a la que se obtendría si se hubiese entrenado la máquina *SVM* con el conjunto completo de datos en un único paso.

Se ha demostrado que las máquinas *SVM* entrenadas incrementalmente de esta forma son muy similares a sus homólogas no incrementales y que traslaciones en el concepto representado (en inglés *concept drifts*), son manejadas de forma eficiente con tal de que los lotes de datos utilizados para el entrenamiento incremental sean suficientemente buenos, es decir, que sus propiedades estadísticas no difieran significativamente de las propiedades del conjunto de datos completo (Syed, 1999b), (Rüping, 2001). Esto último queda garantizado por las hipótesis hechas a priori en el caso de la clasificación conformal (que los datos son generados por una distribución iid o al menos intercambiable).

B. Enfoque de Cawenberghs y Poggio (SVMi_CP)

Implementa un método exacto y secuencial para el entrenamiento de máquinas *SVM*. La clave reside en la conservación de las condiciones de Karush-Kuhn-Tucker (KKT) para todos los ejemplos previamente vistos mientras se realiza “*adiabáticamente*” la adición de nuevos puntos de datos a la solución (Cawenberghs, 2001), (Diehl, 2003).

Las condiciones KKT definen de forma única la solución del problema dual. Estas se formulan como (Diehl, 2003):

$$g_i = \frac{\partial W}{\partial \alpha_i} = \sum_{j=1}^N Q_{ij} \alpha_j + y_i b - 1 \begin{cases} > 0 & \alpha_i = 0 \\ = 0 & 0 \leq \alpha_i \leq C \\ < 0 & \alpha_i = C \end{cases}$$

$$h = \frac{\partial W}{\partial b} = \sum_{j=1}^N y_j \alpha_j \equiv 0.$$

Las derivadas parciales g_i sirven de base para la división del conjunto de entrenamiento en tres categorías:

- El subconjunto S de los *vectores soporte de margen* localizados estrictamente en el margen ($g_i = 0$);
- El subconjunto E de los *vectores soporte de error*, que violan al margen ($g_i < 0$) y
- El subconjunto R de *vectores de reserva*, que exceden al margen ($g_i > 0$).

A medida que se incrementa con nuevos ejemplos una solución existente, el objetivo es la conservación de las condiciones KKT para todos los datos previamente vistos, lo que se alcanza mediante la variación de los coeficientes de los vectores de margen en respuesta a la perturbación impuesta por los nuevos coeficientes incrementados (Diehl, 2003). Los nuevos ejemplos pueden ser directamente asignados al subconjunto R, si $g_i > 0$, o eventualmente pasar a ser vectores soporte de margen o de error a través de una secuencia de pasos “*adiabáticos*”, siendo posible el cambio del estado de los elementos de las distintas categorías durante este proceso.

Detalles del método, como incrementos adiabáticos, determinación de coeficientes de sensibilidad y actualización del jacobiano inverso, pueden ser vistos en (Diehl, 2003).

C. Algoritmo de clasificación

Si se dispone de un recurso para el entrenamiento de máquinas *SVM* de forma incremental, se puede realizar la clasificación conformal empleando un algoritmo muy parecido al presentado en la Sección 4.6. El mismo se describe en el Algoritmo 3.

En el caso de referido algoritmo, la disminución de la carga computacional del entrenamiento de las máquinas *SVM* en el momento de la clasificación de nuevos objetos constituye la clave para mejorar la eficiencia computacional del método.

Algoritmo 3: clasificación con predictores conformales empleando *SVM* incremental

Entradas:

$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$	secuencia de entrenamiento
\mathbf{x}_{n+1}	objeto de test
\mathbf{Y}	espacio de etiquetas

Salidas:

Y	etiqueta predicha
Cf	confianza
Cr	credibilidad

get $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$

Entrenar *SVM* con secuencia $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ y guardar solución

get \mathbf{x}_{n+1}

for each clase $y \in \mathbf{Y}$ **do**

Incrementar ejemplo (\mathbf{x}_{n+1}, y) a las soluciones previamente guardadas

Calcular p -valor para nuevo ejemplo (\mathbf{x}_{n+1}, y) , $p_y = \frac{|\{i = 1, 2, \dots, n+1 : \alpha_i \geq \alpha_{n+1}\}|}{n+1}$

end for

$P = \text{sort_desc}(p)$

$Y = y \Big|_{p_y = \max(p)}$

$Cr = P_1$

$Cf = 1 - P_2$

En contrapartida, también se pueden apuntar algunas desventajas de esta estrategia. En el caso de utilizar el enfoque de Syed, Liu y Sung (Syed, 1999a) las soluciones que se obtienen en el entrenamiento de las máquinas *SVM* son aproximadas, lo que teóricamente puede afectar el rendimiento predictivo del método con relación a la utilización del método *SVM* estándar. Las soluciones proporcionadas por el enfoque de Cawenberghs y Poggio (Cawenberghs, 2001) son exactas y, por tanto, no se ven afectadas por el inconveniente anteriormente mencionado. La principal limitación en este caso tiene que ver con sus requerimientos en términos de memoria, debido principalmente a la necesidad de almacenar el jacobiano inverso en memoria, lo que limita su utilización en el caso de conjuntos de datos masivos (Cawenberghs, 2001), (Shilton, 2005).

4.10.3. Predictor conformal inductivo

Los predictores conformales inductivos (ICP, del inglés *inductive conformal predictor*) son una modificación de los predictores conformales que, típicamente, sacrifican parte de la fiabilidad predictiva con el objetivo de mejorar la eficiencia computacional. Fueron introducidos por Papadopoulos (2002a, 2002b) en el escenario de aprendizaje *offline* y por Vovk (2002) en el escenario de aprendizaje *online*. A continuación se presta atención a la implementación *offline*. Para los detalles de la implementación *online* véase Vovk (2002) o Vovk, Gammerman y Shafer (2005, cap. 4).

Los predictores conformales inductivos dividen el conjunto de datos de entrenamiento $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$ en dos subconjuntos menores:

- el conjunto de entrenamiento propiamente dicho $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)$, siendo $m < n$ y
- el conjunto de calibración $(\mathbf{x}_{m+1}, y_{m+1}), \dots, (\mathbf{x}_n, y_n)$.

El conjunto de entrenamiento propiamente dicho se utiliza para entrenar el *algoritmo base*, derivándose una regla de decisión inductiva general $F_{[(x_1, y_1), \dots, (x_m, y_m)]}(\mathbf{x})$. Esta regla se utiliza para el cálculo de los valores de no conformidad correspondientes a los ejemplos del conjunto de calibración y a los nuevos objetos, \mathbf{x}_l , $l = n + 1, n + 2, \dots$, los cuales, a su vez, se utilizan para el cálculo de los p -valores, a partir de la siguiente expresión:

$$p_y = \frac{|\{i = m + 1, \dots, n, l : \alpha_i \leq \alpha_l\}|}{n - m + 1}.$$

En este caso particular, como valores de no conformidad fueron utilizadas las distancias de los ejemplos del conjunto de calibración y de los nuevos objetos con relación al hiperplano definido por la regla inductiva F , a semejanza de lo hecho en la Sección 4.10.1.

En general se puede resumir la idea para la clasificación binaria tal como se presenta en el Algoritmo 4.

Algoritmo 4: clasificación con predictores conformales inductivos

Entradas:

$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)$	secuencia de entrenamiento
\mathbf{x}_{n+1}	objeto de test
\mathbf{Y}	espacio de etiquetas
m	tamaño del conjunto de entrenamiento propio

Salidas:

Y	etiqueta predicha
Cf	confianza
Cr	credibilidad

Formar conjunto de entrenamiento propiamente dicho P con m ejemplos y de calibración C con $n - m$ ejemplos

Entrenar SVM , con el conjunto P , creando regla de decisión $F(\mathbf{x})$

for each $\mathbf{x} \in C$ **do**

 Calcular valores de no conformidad $\alpha_i = |F(\mathbf{x})|$

end for

get \mathbf{x}_{n+1}

for each clase $y \in \mathbf{Y}$ **do**

 Calcular valor de no conformidad para nuevo ejemplo (\mathbf{x}_{n+1}, y) , $\alpha_{n+1} = F(\mathbf{x}_{n+1})$

 Calcular p -valor $p_y = \frac{|\{i = m + 1, \dots, n, n + 1 : \alpha_i \leq \alpha_{n+1}\}|}{n - m + 1}$

end for

$P = \text{sort_desc}(p)$

$Y = y \Big|_{p_y = \max(p)}$

$Cr = P_1$

$Cf = 1 - P_2$

4.11. CONCLUSIONES

En este capítulo se realizó la presentación de los aspectos fundamentales relacionados con los predictores conformales. En el mismo se presentaron los fundamentos teóricos básicos de los predictores conformales, su definición general y la metodología de su utilización para la solución de problemas de clasificación con confianza y credibilidad. También se presentaron las máquinas *SVM* como clasificadores base y cómo determinar los valores de no conformidad a partir de estas. Por último, se presentó un conjunto de variantes de predictores conformales desarrollados con el objetivo de mejorar la eficiencia computacional del método de la clasificación conformal.

CAPÍTULO 5: APLICACIONES A LOS DATOS DE FUSIÓN TERMONUCLEAR

En esta sección se presentan dos aplicaciones de la predicción conformal a problemas existentes en el ámbito de la fusión termonuclear. El primero de los problemas consiste en la clasificación con medidas de fiabilidad de las imágenes de un *diagnóstico* y el segundo en la selección, mediante aprendizaje activo, de las muestras a emplear en el entrenamiento de un clasificador conformal con el objetivo de mejorar su respuesta temporal. Recuérdese que la predicción conformal es una metodología *transductiva* y, como tal, tiene asociada un elevado coste computacional. Para más detalles véase la Sección 4.4.

5.1. SISTEMA DE CLASIFICACIÓN DE IMÁGENES

En esta sección se aborda el problema de la clasificación de las imágenes del *Diagnóstico* de Esparcimiento Thomson (TSD, abreviatura del inglés *Thomson Scattering Diagnostic*) del TJ – II.

5.1.1. Introducción

El sistema de adquisición de datos de un dispositivo de fusión procesa una cantidad significativa de datos por lo que resulta importante la integración en el mismo de herramientas *software* de aplicación que faciliten el proceso de análisis, tanto en la perspectiva del cumplimiento de su programa científico como en la perspectiva de la operación y calibración del dispositivo.

En el caso del TJ – II, cada componente del sistema de adquisición de datos tiene la posibilidad de operar de dos modos diferentes, es decir, en correspondencia con *software* ejecutado en un computador central, por un lado, o de forma independiente, por otro lado (Vega, 1999). En particular, el *Diagnóstico* de Esparcimiento Thomson (TSD, abreviatura del inglés *Thomson Scattering Diagnostic*) del TJ – II posee en su funcionamiento un alto nivel de automatización e integración con el entorno experimental del TJ – II. Su programa de control está integrado con el Sistema de Distribución de Eventos Asíncrono (AEDS, abreviatura del inglés *Asynchronous Event Distribution System*) del TJ – II, con la finalidad de sincronizar su operación con la producción de las descargas (Vega, 2004). Por otro lado, fue desarrollado un mecanismo específico para la ejecución automática de procesos de análisis después de cada disparo de su láser, siendo uno de los elementos centrales del referido mecanismo un clasificador de imágenes (Vega, 2005).

En (Barth, 1999) y (Herranz, 2003) se hace una descripción del sistema TSD del TJ – II. De acuerdo a los mencionados artículos, este tipo de sistema se encuentra entre los más potentes *diagnósticos* para la determinación de los perfiles de temperatura y densidad electrónica del plasma. Su funcionamiento se basa en el análisis espectral de la luz dispersada por el plasma, extrayéndose de aquí la información necesaria para obtener los datos de temperatura y densidad. El TSD del TJ – II tiene la capacidad de efectuar mediciones de temperatura y densidad electrónica en 160 posiciones espaciales diferentes, a lo largo de un segmento de 360 mm de láser, con una resolución espacial media de 2,2 mm. El espectro de la luz disper-

sada es separado, en cada posición, en 90 canales de longitud de onda con un ancho de 2,1 nm cada uno, siendo el rango espectral del espectrómetro utilizado desde 580 hasta 810 nm.

De acuerdo a las mismas fuentes (Barth, 1999), (Herranz, 2003), en el TSD la luz emitida por un pulso láser se dispersa por los electrones libres del plasma y se direcciona a un espectrómetro para su detección y análisis. Cuando la luz incidente posee una frecuencia ω suficientemente baja, es decir, la energía del fotón ($\hbar\omega$) es mucho menor que la energía del electrón en reposo ($m_e c^2$), este efecto es conocido como “*esparcimiento Thomson*” (en inglés *Thomson scattering*) (Boyd, 2003), (Froula, 2011). La imagen de la luz detectada es al final conducida a un sistema de dos cámaras ICCD (abreviatura del inglés *Intensified Charge – Coupled Device*) que hacen su registro. Estas funcionan de forma sincronizada: una se encarga del registro de la imagen del plasma mientras que la otra registra la imagen de una mezcla de las luces del plasma y del esparcimiento Thomson. El objetivo primario de la utilización de dos cámaras es evitar la contaminación de la señal a analizar por la luz del plasma, lo que se puede conseguir a través de la resta de las imágenes captadas por cada una de ellas.

Las imágenes del plasma captadas en el TSD corresponden a distintos regímenes de medida del *diagnóstico*. Dichas imágenes pueden agruparse en cinco clases distintas: fondo de la cámara ICCD, plasma durante calentamiento *ECRH*, plasma en el que se ha alcanzado la densidad de corte durante calentamiento *ECRH*, plasma durante calentamiento *NBI* y luz parásita sin plasma (véase el Anexo A para mayores detalles).

Un clasificador automático de imágenes, basado en *SVM* (véase la Sección 4.8 para una introducción a las máquinas *SVM*) ha estado operativo durante años en el TSD del TJ – II con buenos resultados en términos de tasas de acierto (Vega, 2005), (Makili, 2010). Sin embargo, las máquinas *SVM*, así como la gran mayoría de las técnicas usualmente empleadas en el campo del aprendizaje automático, proporcionan simples predicciones sin una estimación de la fiabilidad asociada a las mismas.

En contrapartida, los predictores conformales permiten cuantificar las predicciones efectuadas con medidas de fiabilidad. En (Vega, 2010) se presenta el primer intento de mejoría del clasificador de imágenes del TSD mediante la utilización de esta técnica, utilizándose el método de k vecinos más cercanos (Duda, 2001), (Witten, 2005) como *algoritmo base* de la predicción conformal.

Considerando los buenos resultados de las máquinas *SVM* en la clasificación de las imágenes del TSD, resulta interesante su utilización como base de la clasificación conformal que se pretende con las imágenes del TSD del TJ – II. En esta sección, se presenta un conjunto de clasificadores conformales que para su desarrollo utilizan máquinas *SVM* como *algoritmo base*. También se muestra la serie de *experimentos*⁸ realizados para su validación. Finalmente, se describe la actualización de una herramienta *software* que permite la integración de los clasificadores de una forma automática en el entorno experimental del TJ – II.

5.1.2. Experimentos y resultados

En esta sección se hace la presentación de los resultados obtenidos al emplear los clasificadores desarrollados en la clasificación de las imágenes del TSD del TJ – II.

A continuación se hace una descripción general de los *experimentos* realizados y la discusión de los respectivos resultados.

A. Implementación de los clasificadores

Los clasificadores fueron desarrollados en el lenguaje Matlab. Para la codificación de las diferentes rutinas se utilizó la librería Spider (Weston, 2006). En todos los casos se utilizó LIBSVM (Chang, 2011) como librería *SVM* para clasificación que, junto con otras, se encuentra integrada en la librería Spider.

⁸ Recuérdese que por *experimento* se designa, en este caso, la ejecución controlada de un algoritmo, con un conjunto de parámetros dados, para análisis de su efectividad.

Para la realización de los *experimentos* se desarrolló una interfaz gráfica, véase la Figura 8. Esta permite a los usuarios una selección fácil de las diferentes opciones a utilizar en los *experimentos*, tales como la variante del clasificador a utilizar y los correspondientes parámetros, la función *kernel* y los parámetros relacionados con la optimización de *SVM*, las posibles opciones de preprocesamiento y la ejecución de los diferentes comandos para la realización y análisis de los resultados de los *experimentos*.

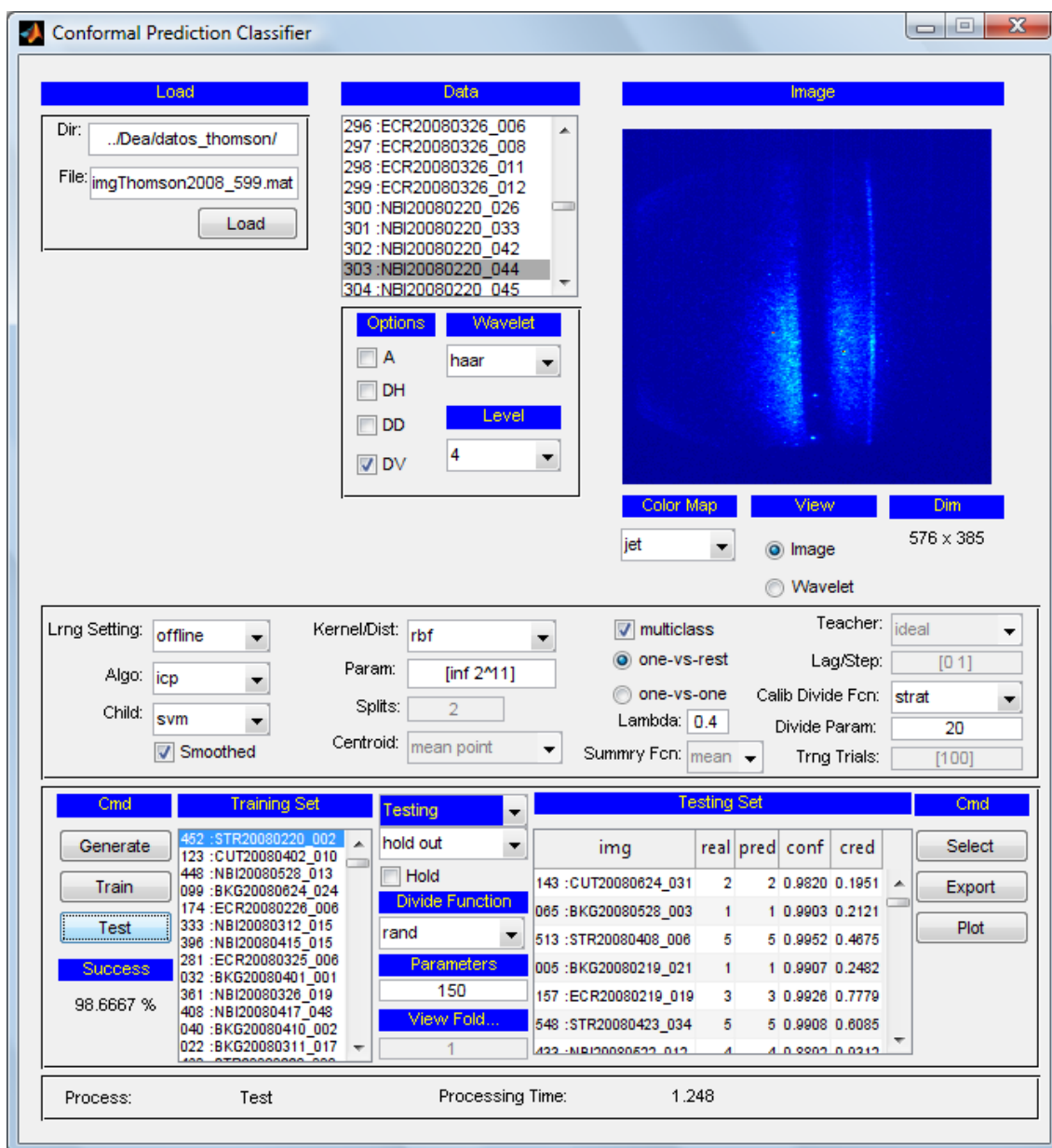


Figura 8: Interfaz gráfica para realización de los *experimentos*

B. Metodología

En los *experimentos* realizados se empleó un total de 1149 imágenes correspondientes a las cinco clases del TSD, véase el Anexo A para mayores detalles acerca del conjunto de datos. Cada imagen se representa por una matriz de 576 x 385 píxeles, es decir, un total de 221.760 posibles atributos. Para evitar problemas de dimensionalidad en el proceso de clasificación, cada imagen se representa a través de un vector de características dado por la transformada *wavelet* de Haar (Mallat, 1999) de la imagen original. Debido al hecho de que la descomposición por la transformada *wavelet* es multiescala, las imágenes pueden ser caracterizadas por un conjunto de coeficientes de aproximación y tres conjuntos de coeficientes de detalle (horizontal, vertical y diagonal). Los coeficientes de aproximación representan la información sobre la imagen a gran escala, conteniendo la mayor parte de la energía de la imagen. Por el contrario, los valores de los coeficientes de detalle son próximos a cero, pero la información que representan puede ser relevante en determinados contextos.

En el caso del TSD, se utilizaron como vectores de características los coeficientes de detalle vertical a los niveles 4 y 5 de descomposición, de forma separada. En trabajos previos se ha demostrado que estas descomposiciones poseen buena capacidad de discriminación para este tipo de señales (Farias, 2004). Las mismas permiten reducir el número de atributos a 36 x 25 (0,41% del original) y 18 x 13 (0,11%), respectivamente.

En los *experimentos* realizados se utilizaron diferentes variantes del clasificador conformal las cuales se diferencian en cuanto a la manera de calcular los valores de no conformidad. Las referidas variantes se diferencian teniendo en cuenta diferentes aspectos:

La forma de inferencia. En algunos casos los clasificadores utilizan únicamente la inferencia *transductiva* mientras que en otros la inferencia *transductiva* se utiliza de forma combinada con la *inductiva*, véase la Sección 4.4.

Las máquinas SVM. Como *algoritmo base* se utilizaron diferentes versiones de máquinas SVM, la versión estándar, por un lado, y otras dos versiones entrenadas incrementalmente, por otro lado, véanse las Secciones 4.8 y 4.10.2.

La extensión multiclase. Fueron utilizadas dos alternativas en este caso, las extensiones *uno-contra-el-resto* y *uno-contra-uno*, véase la Sección 4.8.

Estos aspectos se combinan de diferentes formas, implementándose un total de diez métodos; en la Tabla 2 se presentan los referidos métodos. En la descripción de los resultados, se hace referencia en algunas ocasiones a grupos de métodos, siendo los mismos formados en base a la forma utilizada al realizar la inferencia. En este caso se consideran tres grupos: los transductivos estándar (CP; métodos 1 – 6), los transductivos con función *hash* (*Hashed CP*; métodos 7 y 8) y los predictores conformales inductivos (ICP; métodos 9 y 10).

En implementaciones anteriores, las imágenes del TSD del TJ – II suelen ser clasificadas empleándose un escenario de aprendizaje *offline*, véase la Sección 4.3. Siendo fieles a esta tendencia, en los *experimentos* descritos todos los clasificadores fueron entrenados utilizándose el referido escenario en lugar del escenario de aprendizaje *online* utilizado en el desarrollo teórico y en muchas aplicaciones prácticas de la predicción conformal.

La calidad de los clasificadores basados en la utilización de máquinas SVM depende en elevada medida de la función *kernel* empleada. En este trabajo se utilizaron tres *kernels* distintos: lineal, polinómico y funciones de base radial (RBF), cuyas definiciones se pueden ver en la Sección 4.8, Tabla 1. Por otro lado, la referida calidad también depende de los parámetros empleados en el proceso de optimización de las máquinas SVM.

En función de esto, como paso previo a la clasificación, se efectuó un proceso de búsqueda de los valores adecuados de los parámetros de regularización (C) y del *kernel* (el parámetro de control de la anchura del kernel RBF σ o el grado del kernel polinómico d). A tal fin, se

empleó la búsqueda discretizada (en inglés *grid search*) (Hsu, 2010). En este método, el espacio de búsqueda es discretizado en forma de malla y se prueba la adecuación del algoritmo en los puntos correspondientes a los nudos de la malla. La secuencia de valores de los parámetros utilizada varía de forma exponencial, según una escala diádica. Por ejemplo, para el *kernel* RBF se intentan varios pares (C, σ) , y las secuencias de valores correspondientes podrían ser $C = 2^{-3}, 2^{-2}, \dots, 2^5$ y $\sigma = 2^{-2}, 2^{-1}, \dots, 2^7$.

Método	Designación	Descripción
1	CP (SVM Uno-Contra-Resto)	Clasificador conformal <i>transductivo</i> con SVM estándar como <i>algoritmo base</i> y extensión multiclase <i>uno-contra-el-Resto</i>
2	CP (SVM Uno-Contra-Uno)	Clasificador conformal <i>transductivo</i> con SVM estándar como <i>algoritmo base</i> y extensión multiclase <i>uno-contra-uno</i>
3	CP (SVMi_SLS Uno-Contra-Resto)	Clasificador conformal <i>transductivo</i> con SVM incremental en la variante de Syed, Liu y Sung como <i>algoritmo base</i> y extensión multiclase <i>uno-contra-el-resto</i>
4	CP (SVMi_SLS Uno-Contra-Uno)	Clasificador conformal <i>transductivo</i> con SVM incremental en la variante de Syed, Liu y Sung como <i>algoritmo base</i> y extensión multiclase <i>uno-contra-uno</i>
5	CP (SVMi_CP Uno-Contra-Resto)	Clasificador conformal <i>transductivo</i> con SVM incremental en la variante de Cawenberghs y Poggio como <i>algoritmo base</i> y extensión multiclase <i>uno-contra-el-resto</i>
6	CP (SVMi_CP Uno-Contra-Uno)	Clasificador conformal <i>transductivo</i> con SVM incremental en la variante de Cawenberghs y Poggio como <i>algoritmo base</i> y extensión multiclase <i>uno-contra-uno</i>
7	Hashed CP (Partes = 2)	Clasificador conformal <i>transductivo</i> con función <i>hash</i> , SVM estándar como <i>algoritmo base</i> y división del conjunto de entrenamiento en dos partes
8	Hashed CP (Partes = 4)	Clasificador conformal <i>transductivo</i> con función <i>hash</i> , SVM estándar como <i>algoritmo base</i> y división del conjunto de entrenamiento en cuatro partes
9	ICP (Uno-Contra-Resto)	Clasificador conformal <i>inductivo</i> , con SVM estándar como <i>algoritmo base</i> y extensión multiclase <i>uno-contra-el-resto</i>
10	ICP (Uno-Contra-Uno)	Clasificador conformal <i>inductivo</i> , con SVM estándar como <i>algoritmo base</i> y extensión multiclase <i>uno-contra-uno</i>

Tabla 2: Variantes de clasificadores utilizados en los *experimentos* realizados. Los clasificadores se subdividen en tres grupos, los *transductivos* estándar (en fondo blanco), los *transductivos* con función *hash* (en fondo gris claro) y los *predictores* conformales *inductivos* (en fondo gris oscuro)

Por otro lado, se realizó la búsqueda de los valores adecuados del parámetro del algoritmo de extensión multiclase *uno-contra-el-resto* (λ) (véase la ecuación 1 en la Sección 4.8) en los casos en que esto proceda. Para tal se utilizó igualmente la búsqueda discretizada, variando la secuencia de valores del parámetro linealmente en el espacio de búsqueda.

Como parámetros adecuados fueron seleccionados los que presentaron el menor error de clasificación, empleándose como estimador del error la validación cruzada en 5 partes (Kohavi, 1995a), (Hsu, 2010) sobre un subconjunto de 600 imágenes seleccionadas aleatoriamente de los datos disponibles. En la validación cruzada, el conjunto de datos es subdividido en k partes (*folds* en inglés) mutuamente excluyentes, con tamaños aproximadamente iguales. Secuencialmente, cada parte es probada utilizando el clasificador entrenado con las $k - 1$ partes restantes. De esta manera, cada elemento del conjunto de entrenamiento se prueba una vez y la precisión de la validación cruzada es dada por el porcentaje de los datos correctamente clasificados. En las tablas 3, 4, 5 y 6 se presentan el espacio de búsqueda empleado en cada caso y los valores adecuados que fueron seleccionados.

En la evaluación de la calidad de los diferentes métodos fueron considerados diferentes indicadores. A continuación se hace la descripción de los mismos.

Precisión. Dada por la tasa de acierto del clasificador. Indica el porcentaje de aciertos de la predicción efectuada por un clasificador respecto a la clasificación verdadera de cada uno de los objetos de test.

Valores de confianza y credibilidad. Constituyen las medidas de fiabilidad asociadas al método de la clasificación conformal. A cada objeto clasificado se le asocia un valor de confianza y otro de credibilidad, expresando estos cuán fiable es la referida predicción, véase la Sección 4.6. Como primera medida de la “bondad” de un clasificador al analizar su calidad, se ha considerado el valor medio de la confianza asociada a cada una de sus

predicciones. Cuanto mayor sea este valor, mayor será la calidad del clasificador. Por otro lado, se considera como segunda medida de “bondad” el valor medio de la credibilidad de los vectores clasificados de forma incorrecta. Para la mayoría de las predicciones incorrectas se espera un bajo valor de credibilidad (Saunders, 1999, 2000) por lo que su valor medio debe ser en principio bajo. Cuanto más bajo el valor, mejor el clasificador.

Tiempo de clasificación. Corresponde al tiempo de CPU empleado por un clasificador en el proceso de clasificación de una única imagen.

Los *experimentos* se desarrollaron empleándose un ordenador con procesador Intel Core i7, con 4 núcleos a 2,67 GHz y 8,0 GB de memoria RAM.

Kernel	C	σ, d	λ
Lineal	$[2^{-1}, 2^0, \dots, 2^{10}, \text{inf}]$		$[0, 0,1, \dots, 1]$
Polinómico	$[2^{-1}, 2^0, \dots, 2^3, \text{inf}]$	$[2^{-1}, 2^0, \dots, 2^3]$	$[0, 0,1, \dots, 1]$
RBF	$[2^5, 2^6, \dots, 2^{11}, \text{inf}]$	$[2^6, 2^7, \dots, 2^{15}]$	$[0, 0,1, \dots, 1]$

Tabla 3: Espacio de búsqueda de los valores adecuados de los parámetros a emplear en la clasificación

Método	Nivel 4		Nivel 5	
	C	λ	C	λ
1	2^0	0,3	2^0	0,3
2	2^0		2^0	
3	2^0	0	2^0	0
4	2^0		2^0	
5	2^0	0,3	2^0	0,4
6	2^0		2^0	
7	2^0		2^0	
8	2^0		2^0	
9	2^0	0,6	2^0	0,6
10	2^0		2^0	

Tabla 4: Valores adecuados de los parámetros empleados en la clasificación; *kernel* lineal

Método	Nivel 4			Nivel 5		
	C	d	λ	C	d	λ
1	2^0	2^0	0,1	2^0	2^0	0,1
2	2^0	2^0		2^0	2^0	
3	2^0	2^0	0,1	2^0	2^0	0
4	2^0	2^0		2^0	2^0	
5	2^0	2^0	0,3	2^0	2^0	0,2
6	2^0	2^0		2^0	2^0	
7	2^0	2^1		2^0	2^1	
8	2^0	2^1		2^0	2^1	
9	2^0	2^0	0,3	2^0	2^0	0,6
10	2^0	2^0		2^0	2^0	

Tabla 5: Valores adecuados de los parámetros empleados en la clasificación; *kernel* polinómico

Método	Nivel 4			Nivel 5		
	C	σ	λ	C	σ	λ
1	∞	2^{11}	0,4	∞	2^{11}	0,4
2	∞	2^{11}		∞	2^{11}	
3	∞	2^{11}	0,4	∞	2^{10}	0,1
4	2^{12}	2^{14}		2^{12}	2^{14}	
5	∞	2^{11}	0,3	∞	2^{12}	0,2
6	2^{12}	2^{14}		2^{12}	2^{14}	
7	2^9	2^9		2^9	2^9	
8	2^9	2^9		2^9	2^9	
9	2^9	2^{10}	0,6	2^9	2^{10}	0,8
10	2^9	2^{10}		2^9	2^{10}	

Tabla 6: Valores adecuados de los parámetros empleados en la clasificación; *kernel* RBF

C. Resultados

Se realizaron varios *experimentos* encaminados a analizar distintos atributos de los clasificadores: la precisión, los valores de confianza y credibilidad y la eficiencia temporal. A continuación se presentan los resultados de los referidos *experimentos*.

Precisión de la clasificación

En este caso como indicador para la evaluación de la calidad de los distintos clasificadores se empleó la tasa de acierto y como estimador de la misma se utilizó la validación cruzada en 10 partes (Kohavi, 1995a). En la validación cruzada, las imágenes disponibles son divididas en 10 partes aproximadamente iguales, siendo cada parte probada con un clasificador entrenado con las 9 partes restantes, es decir, cerca de 115 imágenes son incluidas en el subconjunto de test y cerca de 1034 en el subconjunto de entrenamiento.

En la clasificación con ICP es necesario dividir los datos de entrenamiento en dos partes, un conjunto de entrenamiento propiamente dicho y un conjunto de calibración, véase la Sección 4.10.3. En este caso, en cada parte de la validación cruzada se incluyó en el conjunto de calibración un total de 200 imágenes, extraídas de forma aleatoria de los datos de entrenamiento, utilizándose las restantes imágenes para el entrenamiento propiamente dicho del clasificador. De esta manera, de las aproximadamente 1034 imágenes del subconjunto de entrenamiento, 200 son utilizadas para calibración y 834 para el entrenamiento propiamente dicho.

En las tablas 7 y 8 se presentan los valores de la tasa de acierto (Acierto) y los correspondientes *errores estándar* (Error Estd) obtenidos para cada método, empleándose los niveles 4 y 5 de descomposición en la transformada *wavelet*, respectivamente, en el preprocesamiento de los datos.

En las tablas se puede ver que en general las tasas de acierto alcanzadas son altas, a excepción de las correspondientes al método ICP (Uno-Contra-Uno) cuyas tasas de acierto son

claramente inferiores a las demás, al ser utilizado con los *kernels* lineal y polinómico. Las tasas más elevadas, tanto en el caso del nivel 4 como del nivel 5 de descomposición, corresponden al uso del *kernel* RBF y las correspondientes a los *kernels* lineal y polinómico son muy similares entre sí.

Las tasas de acierto son ligeramente más elevadas en el caso de los métodos transductivos estándar, excepto cuando son comparadas con las correspondientes al método ICP (Uno-Contra-Resto). Esta tendencia es clara en el caso del nivel 4 de descomposición y un poco más difusa en el caso del nivel 5.

En términos globales los valores alcanzados utilizándose el nivel 4 de descomposición tienden a ser superiores a los alcanzados con el nivel 5, aunque no existe gran diferencia en este sentido, siendo estos últimos igualmente altos.

La utilización del escenario de aprendizaje *offline* posibilita la comparación de los resultados de la predicción conformal, a través de la correspondiente tasa de acierto. A título de comparación, las tablas 9 y 10 presentan los resultados de la clasificación del mismo conjunto de datos empleando *SVM* estándar, es decir, sin medidas de fiabilidad. En las mismas se puede ver que los valores de las tasas de acierto obtenidas son muy próximos a los obtenidos en el caso de la predicción conformal. Sin embargo, es importante enfatizar que una mayor tasa de acierto en los clasificadores "*tradicionales*" no significa en absoluto que proporcionen mejores resultados, ya que no dan una medida de la calidad de la predicción.

Métodos	Lineal		Polinómico		RBF	
	Acierto (%)	Error Estd	Acierto (%)	Error Estd	Acierto (%)	Error Estd
1	96,3432	0,5168	96,7803	0,3437	97,7384	0,4335
2	96,4325	0,4569	97,5637	0,4813	97,2159	0,5149
3	95,9115	0,5186	95,9115	0,5189	97,6514	0,7221
4	95,5629	0,3525	95,5629	0,3525	97,2990	0,5604
5	96,2601	0,4494	96,2601	0,4494	97,1289	0,4305
6	97,1304	0,4109	97,0412	0,3225	97,3898	0,4847
7	90,4249	0,7694	92,1663	0,6747	95,2982	0,7377
8	90,6850	0,9740	91,2067	1,0119	93,2105	0,6597
9	96,6041	0,5733	96,3432	0,4279	97,8261	0,6115
10	82,7704	1,9883	82,7704	1,9883	95,3883	0,8199

Tabla 7: Tasa de acierto correspondiente a una validación cruzada en 10 partes; nivel 4 de descomposición

Métodos	Lineal		Polinómico		RBF	
	Acierto (%)	Error Estd	Acierto (%)	Error Estd	Acierto (%)	Error Estd
1	94,7773	0,7676	95,0389	0,6618	97,3021	0,4386
2	95,8223	0,4639	95,6499	0,7327	95,3005	0,8010
3	94,3410	0,6792	94,3410	0,6792	97,4752	0,4198
4	93,2960	0,9988	93,2960	0,9988	96,1693	0,5693
5	94,3425	0,5543	94,3417	0,7259	96,9558	0,5222
6	95,5622	0,7149	95,9108	0,7101	97,3013	0,3303
7	89,5507	0,8318	92,0813	0,6621	96,1701	0,3938
8	90,6850	0,8836	92,7780	0,8293	93,8185	0,4435
9	95,9962	0,3716	95,9962	0,3716	96,8665	0,5971
10	83,6400	2,0757	83,6400	2,0757	94,7788	0,6074

Tabla 8: Tasa de acierto correspondiente a una validación cruzada en 10 partes; nivel 5 de descomposición

Kernel	Nivel 4		Nivel 5	
	Acierto (%)	Error Estd	Acierto (%)	Error Estd
Lineal ($C = 2^0$)	97,1281	0,0049	95,8207	0,0045
Poli ($C = 2^0, d = 2^0$)	97,1281	0,0049	95,8207	0,0045
RBF ($C = 2^6, \sigma = 2^{11}$)	98,0870	0,0072	97,4737	0,0054

Tabla 9: Tasa de acierto correspondiente a una validación cruzada en 10 partes; SVM estándar, uno-contra-el-resto

Kernel	Nivel 4		Nivel 5	
	Acierto (%)	Error Estd	Acierto (%)	Error Estd
Lineal ($C = 2^{-6}$)	97,4760	0,0056	97,2151	0,0061
Poli ($C = 2^{-6}$, $d = 2^0$)	97,4760	0,0056	97,2151	0,0061
RBF ($C = 2^6$, $\sigma = 2^{11}$)	98,3463	0,0046	97,7368	0,0054

Tabla 10: Tasa de acierto correspondiente a una validación cruzada en 10 partes; *SVM* estándar *uno-contra-uno*

Valores de confianza y credibilidad

En esta sección se consideran los valores de confianza y credibilidad generados por los distintos clasificadores en las validaciones cruzadas en 10 partes analizadas anteriormente. En este proceso, cada una de las imágenes disponibles se incluye una única vez en el conjunto de test, y, por tanto, se efectúa el análisis de 1149 valores de confianza y credibilidad.

En las tablas 11 – 16 se presenta un resumen de los valores de confianza asociados a la totalidad de las predicciones efectuadas, así como de los de credibilidad para las predicciones incorrectas. Tablas semejantes con los mismos datos agrupados por métodos se pueden ver en el anexo C.

En las referidas tablas se puede ver que los valores medios de la confianza son altos en los casos de todos los métodos transductivos estándar, independientemente de la variante de máquina *SVM* utilizada como *algoritmo base*, de la función *kernel* y del nivel de descomposición utilizado en el preprocesamiento de los datos, siendo los valores en todos los casos superiores a 0,93. Se puede observar también que para este grupo de métodos, los valores medios de confianza generados por las implementaciones *uno-contra-uno* son siempre superiores a los generados por las implementaciones *uno-contra-el-resto* correspondientes a la utilización de la misma variante de máquina *SVM* como *algoritmo base*, la misma función *kernel* y el mismo nivel de descomposición. Los valores medios más elevados corresponden a los métodos 6 y 2, es decir, CP (*SVMi_CP Uno-Contra-Uno*) y CP (*SVM Uno-Contra-Uno*) respectivamente, siendo ligeramente mayor en el primer caso.

Los valores medios de confianza para los métodos transductivos con función *hash* son igualmente altos, encontrándose los mismos por encima de 0,91 en todos los casos. En los resultados correspondientes a este grupo de métodos, se puede observar que los citados valores de confianza son superiores en la implementación con la división de los datos en dos partes respecto a la implementación con la división en cuatro partes. Esto sucede para todas las combinaciones de función *kernel* y nivel de descomposición, a excepción de una (*kernel* lineal, nivel 5 de descomposición). Comparando con los valores obtenidos en los métodos transductivos estándar, se puede observar que en este caso los valores obtenidos son aproximados a los correspondientes a las implementaciones *uno-contra-el-resto* y visiblemente inferiores a los correspondientes a las versiones *uno-contra-uno*.

En lo que se refiere a los ICP, los valores medios de confianza obtenidos son altos cuando se usa un *kernel* RBF, siendo comparables a los más elevados en los casos mencionados anteriormente. En los casos de utilización de otras funciones *kernel* (lineal y polinómica) los referidos valores son relativamente más bajos, llegándose a observar una degradación visible de los mismos en los casos del método 10, es decir, ICP (Uno-Contra-Uno), en los dos niveles de descomposición, siendo los valores medios de confianza del orden del 70%. Con el método 9, es decir, ICP (Uno-Contra-Resto), la degradación se observa al ser utilizado con el *kernel* polinómico y el nivel 4 de descomposición. Se puede observar también una tendencia en el sentido de que los valores medios de confianza son más altos en el caso de la implementación *uno-contra-el-resto* respecto a la implementación *uno-contra-uno* correspondiente.

Con relación a los valores medios de la credibilidad para las predicciones incorrectas, en las mismas tablas se puede observar que en el caso de los métodos transductivos estándar existe una clara diferenciación entre los valores obtenidos para las implementaciones *uno-contra-el-resto* y *uno-contra-uno*, siendo los valores visiblemente inferiores, y por tanto mejores, en el caso de esta última. Para cada una de las referidas implementaciones, la utilización

de diferentes variantes de máquina *SVM* como *algoritmo base* no conduce a una tendencia definida en cuanto a los mejores valores de credibilidad. El mérito es repartido entre las tres variantes de *algoritmo base* (*SVM* estándar, *SVMi_SLS* y *SVMi_CP*), en dependencia de la función *kernel* y del nivel de descomposición utilizado.

Los valores de credibilidad obtenidos en el caso de los métodos transductivos con función *hash* son muy similares para las dos implementaciones probadas (división en dos y cuatro partes), siendo en ambos casos los valores de credibilidad comparables con los obtenidos al utilizar las implementaciones *uno-contra-el-resto* de los métodos transductivos estándar.

A los ICP corresponde el peor comportamiento en cuanto a los valores de credibilidad en general. Los valores obtenidos con los mismos son competitivos con relación a los obtenidos con los métodos mencionados anteriormente únicamente en el caso de la implementación *uno-contra-el-resto*, cuando se utiliza con el *kernel* RBF. Este es el único grupo para el cual se pueden observar valores medios de credibilidad superiores a 0,3, llegando a alcanzar valores en el orden del 70% en el caso de la implementación *uno-contra-uno*.

Métodos	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
1	0,9668	0,9997	0,6647	0,2127	0,9647	0,0021
2	0,9825	0,9995	0,8657	0,1071	0,9892	0,0024
3	0,9320	0,9996	0,8088	0,1777	0,7031	0,0866
4	0,9706	0,9997	0,8799	0,1235	0,9517	0,0026
5	0,9673	0,9998	0,6554	0,2282	0,7720	0,0198
6	0,9828	0,9996	0,8681	0,1163	0,7672	0,0025
7	0,9324	0,9960	0,2083	0,2740	0,9725	0,0121
8	0,9185	0,9936	0,2454	0,2997	0,9877	0,0173
9	0,9175	0,9988	0,1205	0,2944	0,8986	0,0140
10	0,7763	0,9989	0,0022	0,7183	0,9999	0,0041

Tabla 11: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; *kernel* lineal, nivel 4 de descomposición

Métodos	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
1	0,9720	0,9998	0,7760	0,1557	0,9782	0,0042
2	0,9805	0,9994	0,9236	0,1264	0,8005	0,0023
3	0,9368	0,9985	0,0046	0,2851	0,9816	0,0040
4	0,9659	0,9996	0,9226	0,0910	0,7024	0,0031
5	0,9737	0,9994	0,7572	0,1427	0,9605	0,0036
6	0,9802	0,9996	0,9222	0,1443	0,8707	0,0028
7	0,9163	0,9958	0,0608	0,2696	0,9953	0,0192
8	0,9201	0,9917	0,0653	0,2557	0,9891	0,0286
9	0,9400	0,9990	0,1940	0,3241	0,9286	0,0229
10	0,7346	0,9985	0,0223	0,5295	0,9987	0,0045

Tabla 12: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; *kernel* lineal, nivel 5 de descomposición

Métodos	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
1	0,9555	0,9986	0,6574	0,2450	0,9647	0,0024
2	0,9826	0,9995	0,8595	0,1190	0,9093	0,0029
3	0,9320	0,9996	0,8088	0,1777	0,7031	0,0866
4	0,9706	0,9997	0,8799	0,1235	0,9517	0,0026
5	0,9673	0,9998	0,6554	0,2282	0,7720	0,0198
6	0,9826	0,9996	0,8557	0,1264	0,6158	0,0028
7	0,9447	0,9971	0,4817	0,1784	0,9942	0,0125
8	0,9264	0,9942	0,5773	0,2091	0,9877	0,0192
9	0,7375	0,9984	0,0697	0,3721	0,9136	0,0031
10	0,7763	0,9989	0,0022	0,7183	0,9999	0,0041

Tabla 13: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; *kernel* polinómico, nivel 4 de descomposición

Métodos	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
1	0,9649	0,9988	0,7576	0,1815	0,9782	0,0018
2	0,9801	0,9995	0,9222	0,1415	0,9494	0,0026
3	0,9368	0,9985	0,0046	0,2851	0,9816	0,0040
4	0,9659	0,9996	0,9226	0,0910	0,7024	0,0031
5	0,9692	0,9994	0,7630	0,1585	0,9605	0,0026
6	0,9801	0,9993	0,9259	0,1075	0,9402	0,0044
7	0,9496	0,9953	0,0248	0,1854	0,9936	0,0240
8	0,9380	0,9919	0,0438	0,1689	0,9855	0,0291
9	0,9400	0,9990	0,1940	0,3241	0,9286	0,0229
10	0,7346	0,9985	0,0223	0,5295	0,9987	0,0045

Tabla 14: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; *kernel* polinómico, nivel 5 de descomposición

Métodos	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
1	0,9414	0,9995	0,8408	0,1637	0,7280	0,0299
2	0,9645	0,9997	0,8555	0,0747	0,5397	0,0029
3	0,9389	0,9997	0,8279	0,1647	0,7173	0,0304
4	0,9894	0,9999	0,8881	0,0647	0,6628	0,0005
5	0,9334	0,9997	0,2590	0,2770	0,9690	0,0114
6	0,9962	0,9998	0,8727	0,0344	0,1110	0,0023
7	0,9679	0,9977	0,8503	0,1267	0,8879	0,0228
8	0,9455	0,9952	0,7544	0,1155	0,8489	0,0265
9	0,9921	0,9989	0,8235	0,1628	0,4965	0,0043
10	0,9777	0,9989	0,3817	0,3107	0,8581	0,0041

Tabla 15: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; *kernel* RBF, nivel 4 de descomposición

Métodos	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
1	0,9475	0,9985	0,2534	0,2054	0,9815	0,0057
2	0,9769	0,9995	0,9095	0,0994	0,9153	0,0026
3	0,9432	0,9990	0,8825	0,1317	0,9121	0,0048
4	0,9871	0,9998	0,9382	0,0904	0,7699	0,0019
5	0,9760	0,9995	0,2824	0,2329	0,8724	0,0130
6	0,9971	0,9996	0,9266	0,0319	0,1351	0,0021
7	0,9632	0,9968	0,7478	0,0942	0,5324	0,0217
8	0,9415	0,9935	0,7698	0,1007	0,4397	0,0156
9	0,9918	0,9991	0,9347	0,0907	0,4272	0,0040
10	0,9679	0,9991	0,2902	0,3645	0,9645	0,0075

Tabla 16: Valores de confianza de todas las predicciones y de credibilidad de las predicciones incorrectas; *kernel* RBF, nivel 5 de descomposición

En las figuras 9 – 14 se representan histogramas que muestran la cantidad de imágenes predichas de forma incorrecta cuyos valores de credibilidad se encuentran en diferentes rangos para las diferentes combinaciones de función *kernel* y nivel de descomposición. Histogramas semejantes organizados por métodos y las correspondientes tablas de distribución de frecuencias pueden ser vistos en el anexo C.

En las mismas se puede observar que en la mayoría de los casos la frecuencia de las imágenes es más elevada en los intervalos correspondientes a los valores de credibilidad más bajos, observándose una caída abrupta al final de los tres o cuatro primeros intervalos, aunque persisten siempre valores no nulos, pero bajos, para las frecuencias correspondientes a los valores de credibilidad más elevados.

Esta tendencia es más acentuada en las implementaciones *uno-contra-uno* de los métodos transductivos estándar, observándose para los mismos la caída en los valores de frecuencia a partir del segundo intervalo de credibilidad, lo que significa que estos presentan mejor calidad respecto a los demás métodos, en cuanto a este indicador. Por otro lado, la principal desviación con respecto a este comportamiento se observa en el caso del ICP (Uno-Contra-Uno), particularmente en los casos del uso de los *kernels* lineal y polinómico. Se observa en este

caso que las frecuencias correspondientes a los valores de credibilidad más elevados tienden a ser superiores a las frecuencias correspondientes a los valores de credibilidad más bajos.

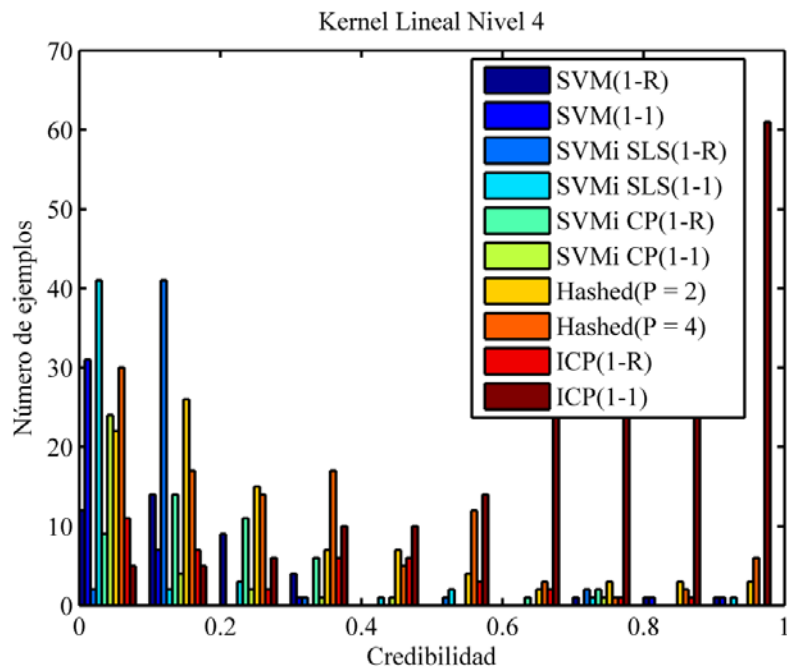


Figura 9: Valores de credibilidad para las imágenes predichas de forma incorrecta; *kernel* lineal, nivel 4 de descomposición

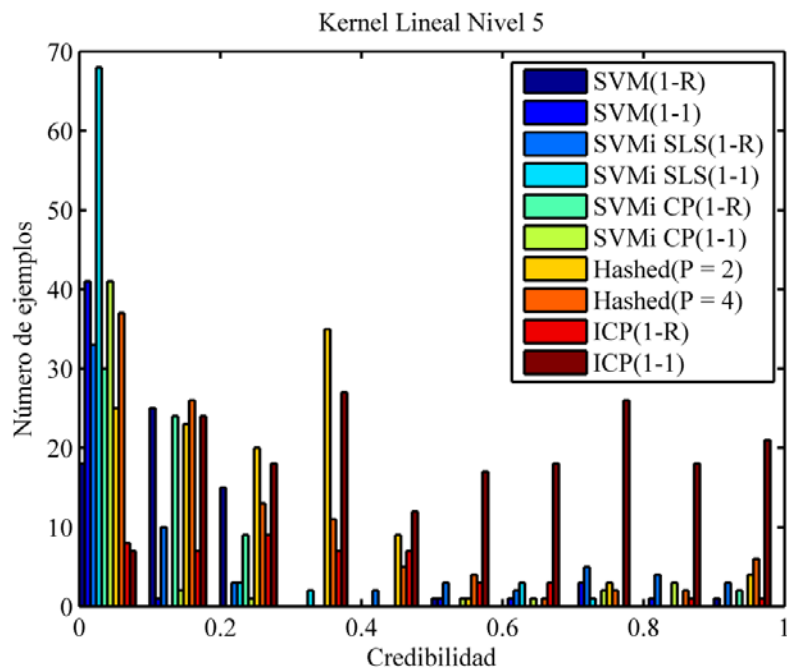


Figura 10: Valores de credibilidad para las imágenes predichas de forma incorrecta; *kernel* lineal, nivel 5 de descomposición

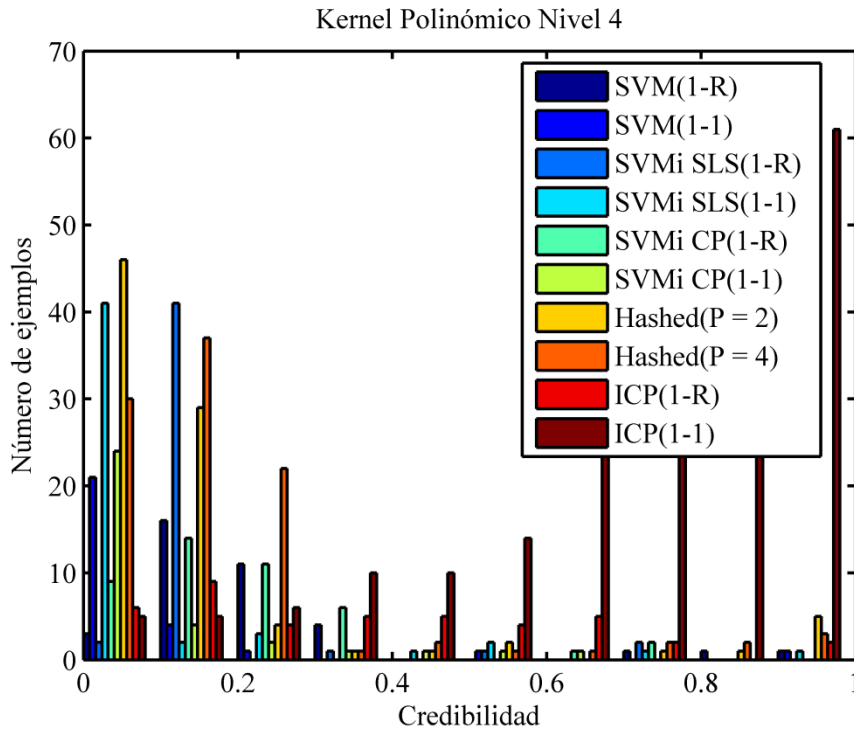


Figura 11: Valores de credibilidad para las imágenes predichas de forma incorrecta; *kernel* polinómico, nivel 4 de descomposición

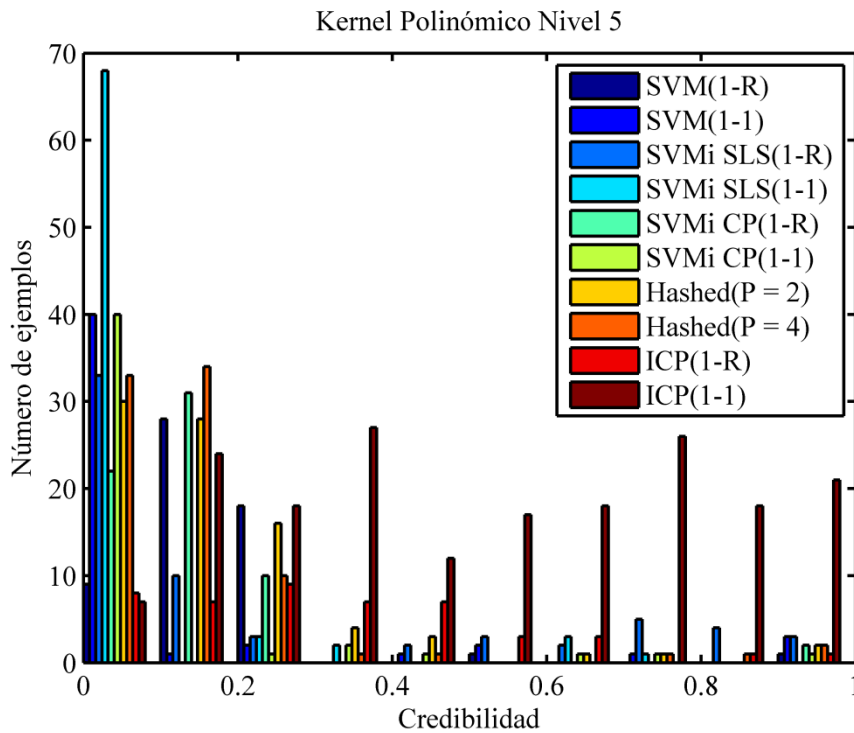


Figura 12: Valores de credibilidad para las imágenes predichas de forma incorrecta; *kernel* polinómico, nivel 5 de descomposición

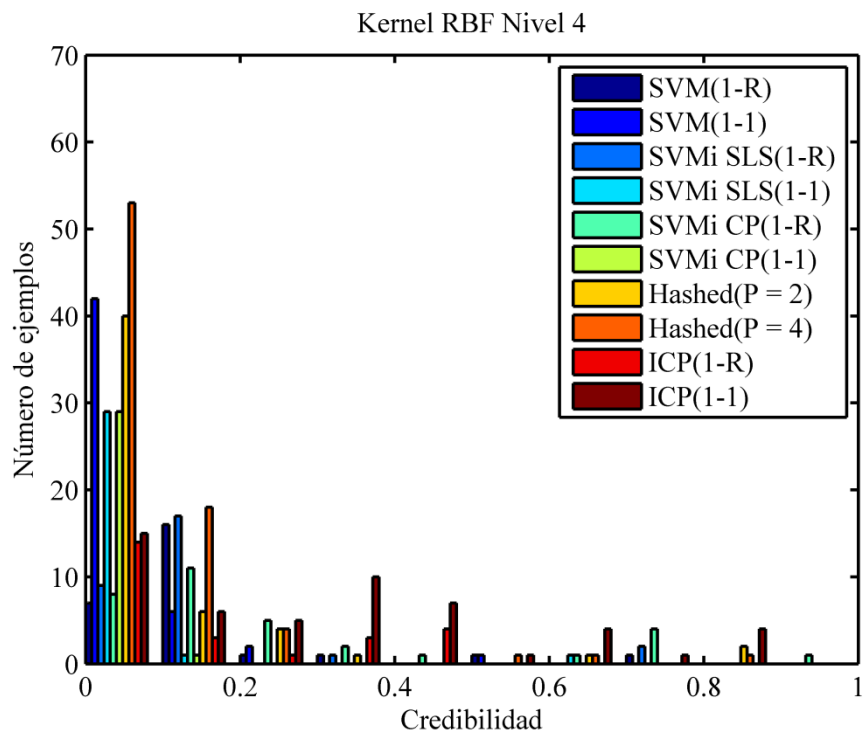


Figura 13: Valores de credibilidad para las imágenes predichas de forma incorrecta; *kernel* RBF, nivel 4 de descomposición

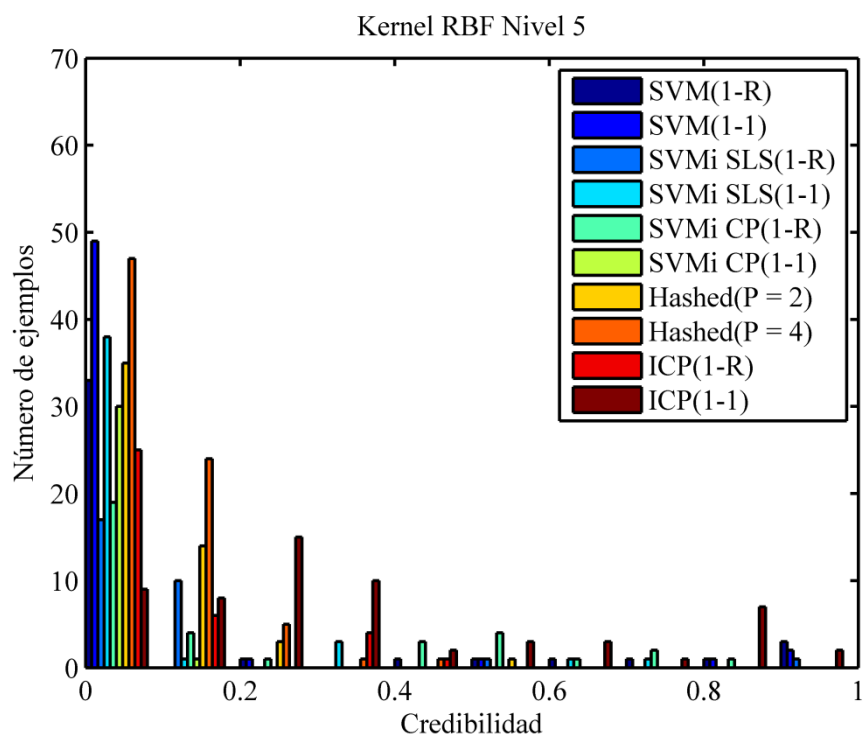


Figura 14: Valores de credibilidad para las imágenes predichas de forma incorrecta; *kernel* RBF, nivel 5 de descomposición

Comparación de la eficiencia temporal

Se desarrolló un conjunto de *experimentos* con el objetivo de comparar la eficiencia temporal entre los diferentes métodos probados y se eligió como medida de eficiencia el tiempo de CPU (en segundos) empleado por un clasificador para la clasificación de una única imagen.

Con tal fin, los diferentes clasificadores fueron entrenados utilizándose los mismos valores para los parámetros de regularización y del *kernel* ($C = \infty$ y $\sigma = 2^{11}$, para el *kernel* RBF, $C = 2^0$, para el *kernel* lineal, y $C = 2^0$ y $d = 2^0$, para el *kernel* polinómico) y conjuntos de entrenamiento de diferentes tamaños, seleccionados de forma aleatoria y balanceada. Cada clasificador fue entrenado un total de 10 veces para cada tamaño del conjunto de entrenamiento, siendo los valores representados en las figuras 15 – 20 los valores medios de los tiempos de clasificación obtenidos. Las correspondientes tablas pueden ser vistas en el Anexo C.

En las referidas figuras se puede observar que los ICP son claramente superiores a los demás métodos para todas las combinaciones de funciones *kernel* y niveles de descomposición. Los tiempos de CPU que se obtienen oscilan entre 1 y 4 ms.

Por otro lado, en lo que respecta a la utilización del nivel 4 de descomposición, se puede observar en las figuras que el tiempo de procesamiento para el CP (SVMi_SLS), en sus dos implementaciones, permanece aproximadamente constante, independientemente de la función *kernel* utilizada (con valores de tiempo que se encuentran entre 0,3 y 0,7 s, aproximadamente, en los diversos casos), siendo la versión *uno-contra-el-resto* ligeramente más rápida que la *uno-contra-uno*. Los referidos métodos tienden a superar a los demás a medida que va creciendo el conjunto de entrenamiento, pudiéndose poner un claro umbral a partir de las 300 – 400 muestras en el conjunto de entrenamiento.

En los casos de los demás métodos, los valores del tiempo de procesamiento tienden a crecer según aumenta el tamaño del conjunto de entrenamiento, siendo visiblemente peor el resultado correspondiente a la versión *uno-contra-el-resto* del CP (*SVM*).

En las figuras correspondientes al nivel 5 de descomposición se puede observar una mejora en los tiempos de CPU de todos los clasificadores respecto al nivel 4 de descomposición, lo cual es lógico teniendo en cuenta la menor dimensionalidad de los vectores de características. En las mencionadas figuras se puede observar las mismas tendencias de cuasi-constancia y crecimiento de los tiempos de procesamiento para los CP (*SVMi_SLS*) y los demás métodos, respectivamente, pero en este caso la variante *uno-contra-el-resto* de CP (*SVMi_SLS*) supera a los demás métodos solamente a partir de las 700 – 800 muestras en el conjunto de entrenamiento. Para conjuntos de entrenamiento menores que el mencionado se pueden destacar las dos variantes de *Hashed CP* cuyos tiempos de procesamiento son inferiores a los correspondientes a los demás métodos. En este caso, los peores resultados corresponden a las dos variantes de CP (*SVMi_CP*).

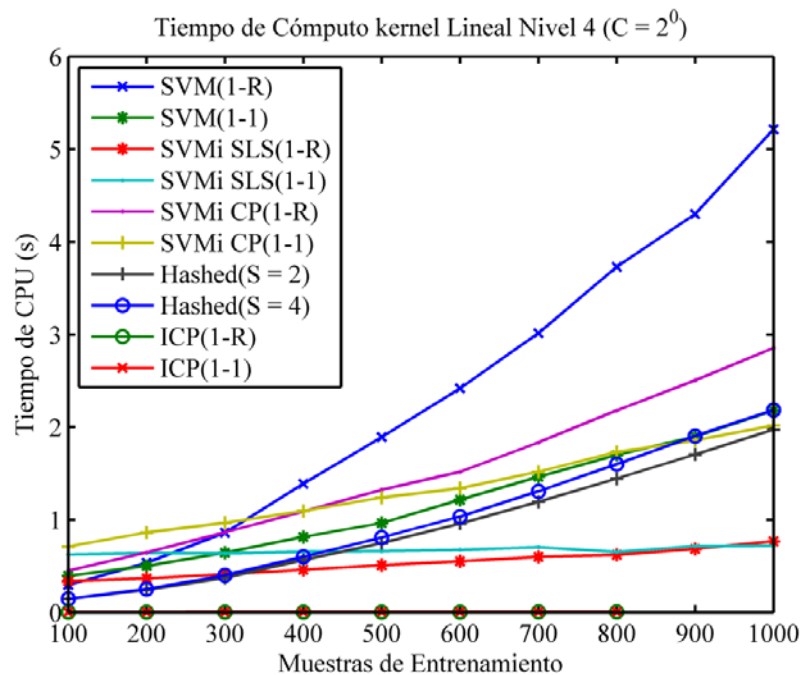


Figura 15: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* lineal, nivel 4 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador

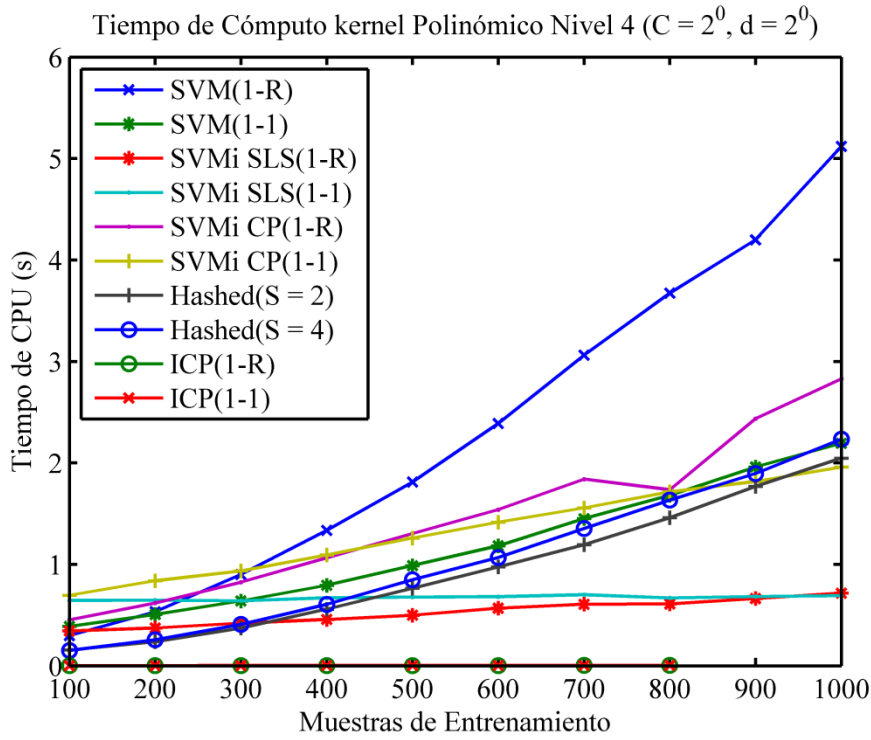


Figura 16: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* polinómico, nivel 4 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador

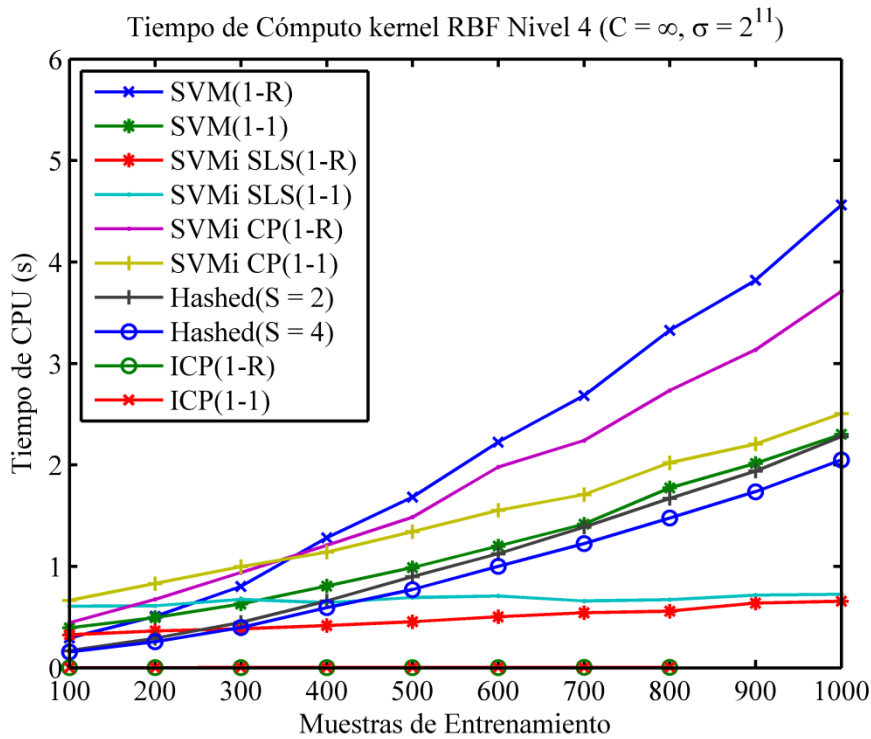


Figura 17: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* RBF, nivel 4 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador

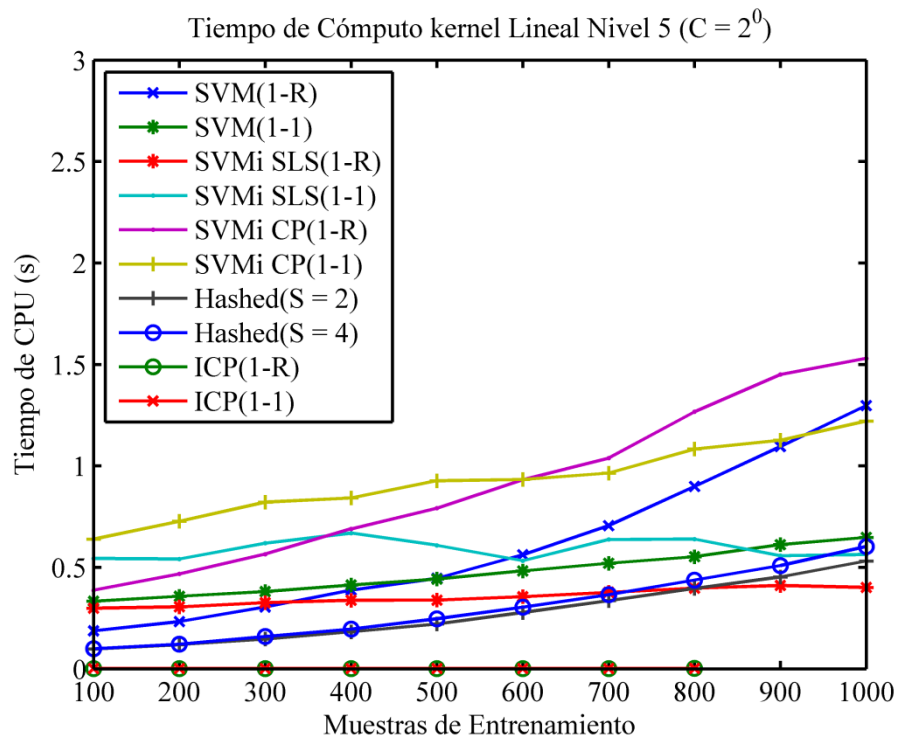


Figura 18: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* lineal, nivel 5 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador

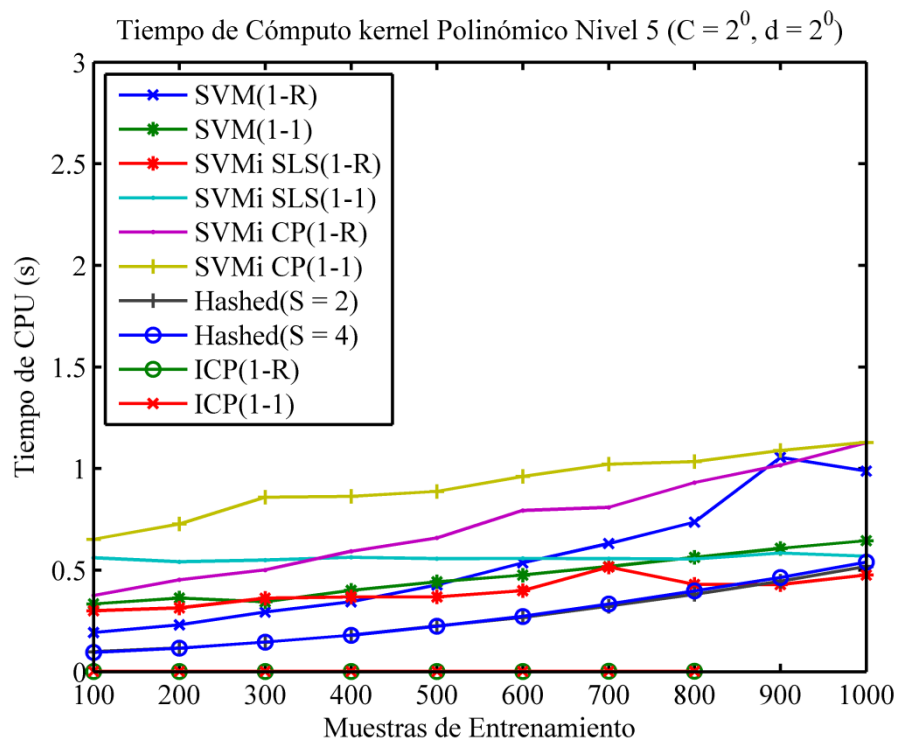


Figura 19: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* polinómico, nivel 5 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador

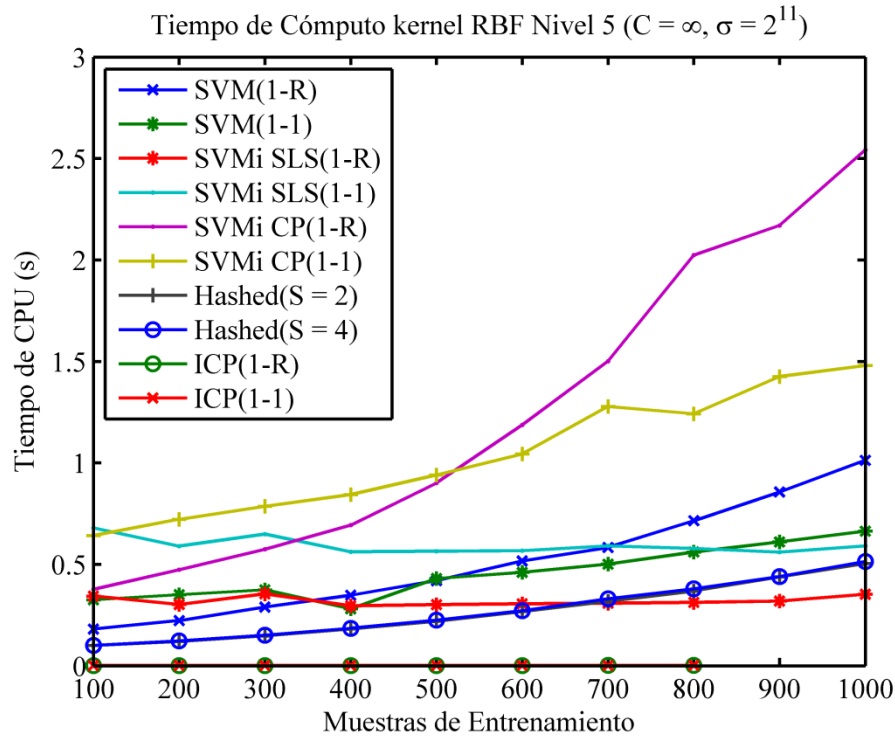


Figura 20: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* RBF, nivel 5 de descomposición. Los valores representados constituyen la media obtenida en 10 ejecuciones de cada clasificador

5.1.3. Implementación práctica

Una de las aplicaciones prácticas de los clasificadores aquí presentados fue una nueva versión de la aplicación que permite la clasificación de las imágenes del TSD (Vega, 2005). Esta aplicación permite el análisis automático de las imágenes, incluyendo los valores de confianza y credibilidad, de forma sincronizada con la producción de las descargas.

La referida aplicación consiste en un *software* servidor, desarrollado en Matlab, el cual efectúa la recepción de las imágenes enviadas por un *software* cliente, llama a una rutina de clasificación y devuelve el resultado de dicha clasificación al cliente. Este proceso puede llevarse a cabo tanto en tiempo real, durante la producción de descargas en el TJ – II, como en diferido. Dicho de otra manera, existen dos tipos de aplicaciones cliente, una de tiempo real sincronizada con la operación del TSD y otra en la que las aplicaciones pueden ejecutarse en cualquier momento con datos almacenados en la base de datos del *diagnóstico*. En realidad hay dos aplicaciones clientes de este último tipo. En la primera de ellas se pueden enviar al

clasificador imágenes almacenadas en la base de datos del TSD. En la segunda, el experto puede poner manualmente la clase de una imagen en la que se haya equivocado el clasificador.

Por otra parte, el mencionado *software* servidor efectúa el mantenimiento de una base de datos, implementada en Microsoft Access, en la cual se registran todos los datos relativos a las imágenes procesadas. En las figuras 21 y 22 se presentan, respectivamente, la interfaz gráfica del servidor y un diagrama esquemático del ciclo de automatización del TJ – II con el servidor.

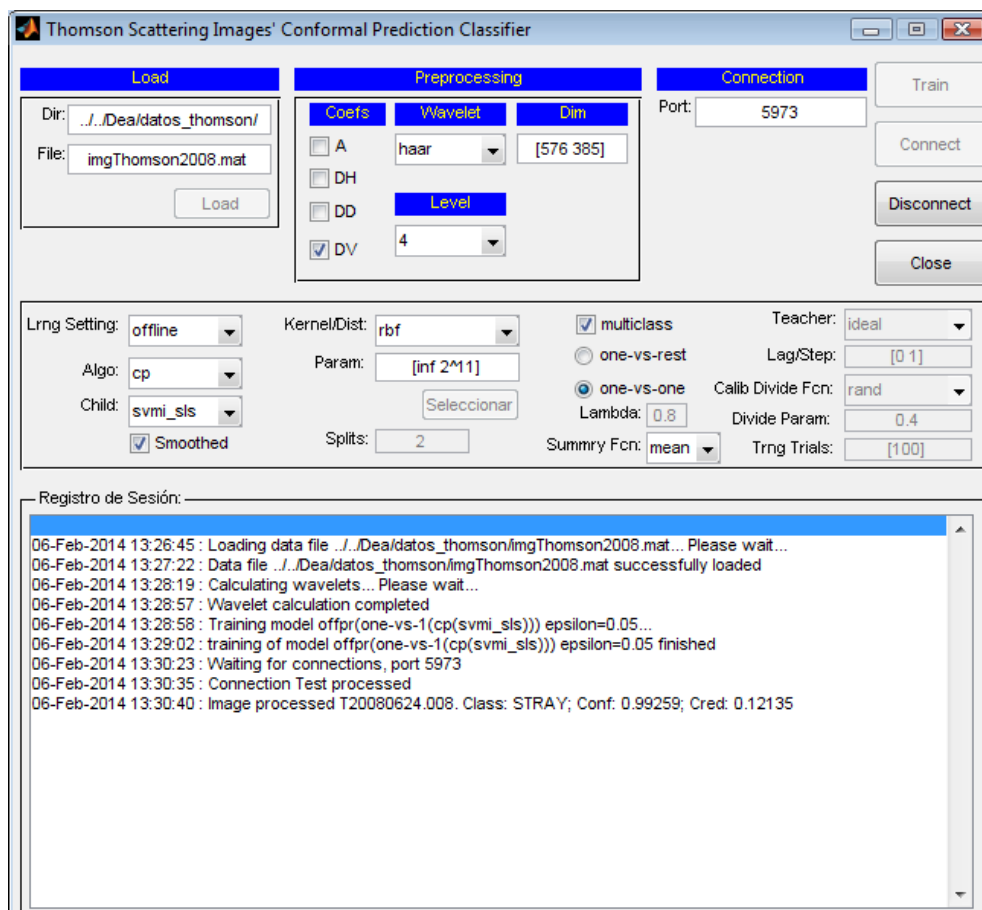


Figura 21: Interfaz gráfica del servidor de clasificación del TSD

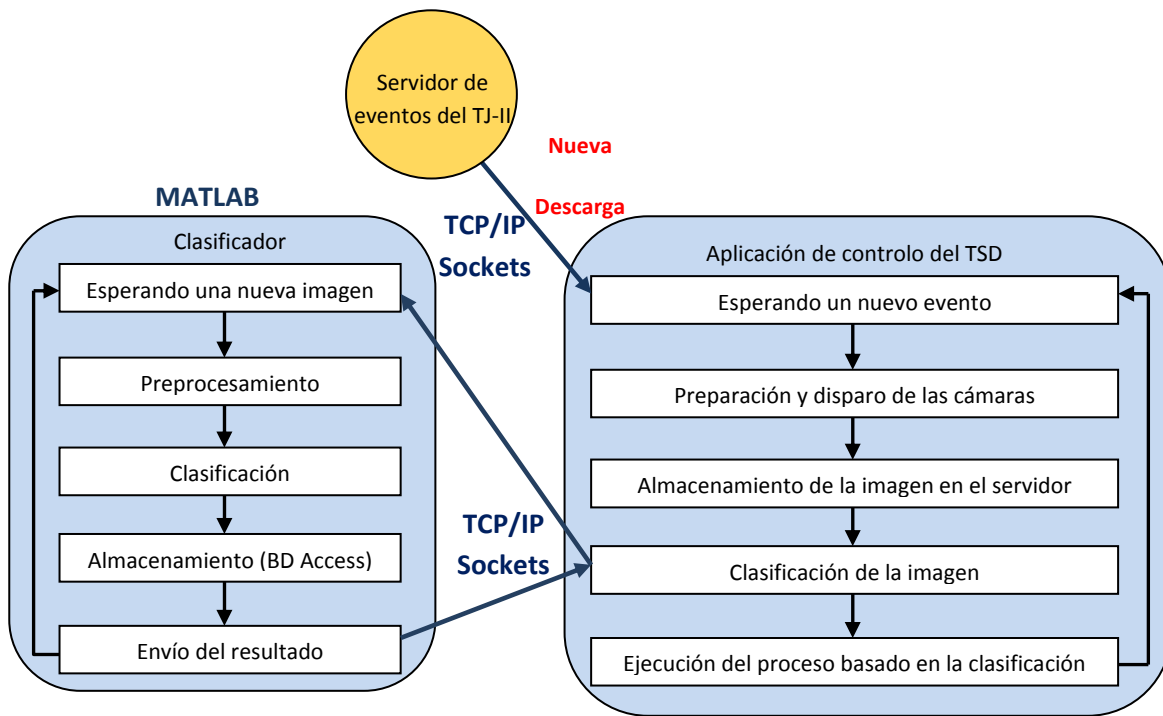


Figura 22: Ciclo de automatización del TSD

La comunicación entre los programas cliente y servidor se procesa en base a un protocolo diseñado específicamente por el equipo de adquisición de datos del TJ – II. En el referido protocolo se encuentran definidos por el momento tres tipos de conexión. Para cada clase de conexión, el protocolo especifica la secuencia de intercambio de datos entre el cliente y el servidor, para que los mismos puedan operar de forma sincronizada (figuras 23 – 25). A continuación se presenta una breve descripción de cada uno de los tipos de conexiones.

Conexión “Test”. En esta conexión, el cliente intenta comunicar con el servidor para saber si este se encuentra activo.

Conexión “Procesar Imagen”. Se envía una imagen al servidor, este la clasifica y devuelve el resultado.

Conexión “Clasificación de Usuario”. Permite al usuario (mediante la oportuna aplicación cliente) efectuar la rectificación de la clasificación efectuada por el servidor a una imagen.

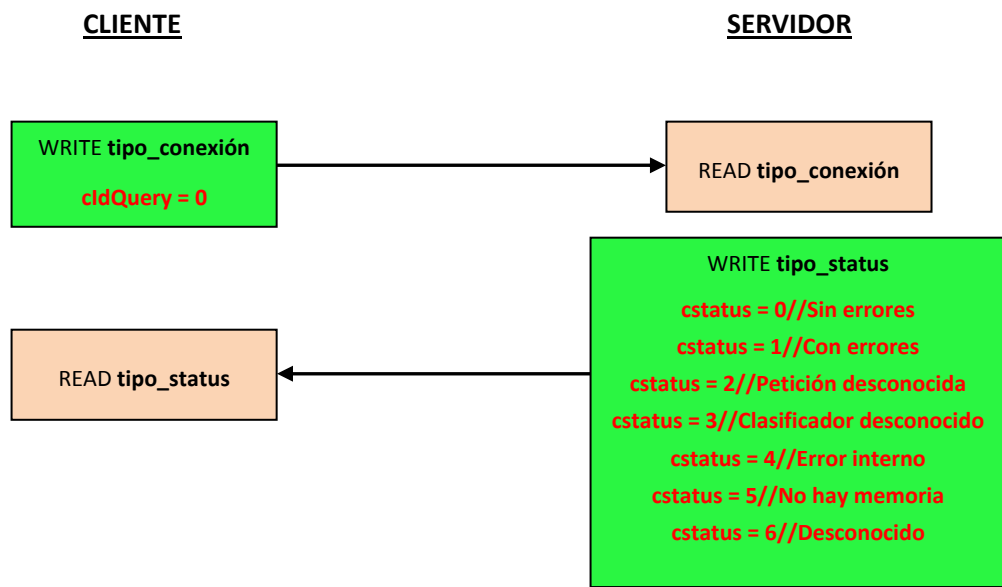


Figura 23: Protocolo de comunicación: conexión “test”

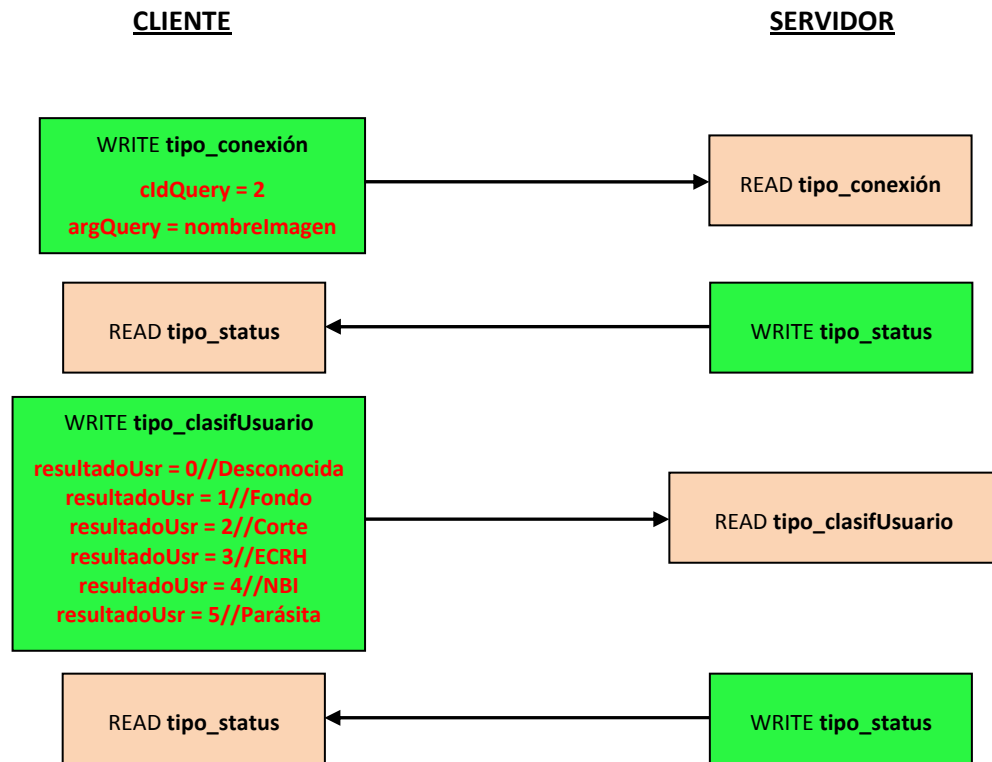


Figura 25: Protocolo de comunicación: conexión “clasificación de usuario”

Para la comunicación cliente/servidor se recurrió a la utilización de *sockets* TCP/IP (Hall, 2012). La habilitación del servidor para la comunicación con *sockets* en el entorno Matlab se hizo en base a la interfaz proporcionada por este para el manejo de librerías dinámicas (MathWorks, 2009). Con tal fin, se elaboró una librería, escrita en lenguaje C, donde se definen las estructuras de datos intercambiadas entre el cliente y el servidor así como el conjunto de funciones que permiten gestionar las llamadas a las rutinas de la librería *winsock2* (Microsoft, 2012) a través de la interfaz proporcionada por Matlab.

5.1.4. Conclusiones

En esta sección se discutió un conjunto de clasificadores conformales así como una aplicación que permite la incorporación de los mismos de forma automática en el entorno experimental del TJ – II.

En la discusión de los clasificadores se describió el conjunto de *experimentos* realizados para valorar la precisión de cada uno de ellos, la calidad de los valores de confianza y credibilidad y las respectivas eficiencias temporales. A tal efecto se utilizó un conjunto de imágenes del TSD del TJ – II, preprocesadas a través de la transformada *wavelet* de Haar a los niveles 4 y 5 de descomposición, y tres funciones *kernel* diferentes: lineal, polinómica y RBF.

La calidad de la clasificación es alta en la mayor parte de los casos probados, donde la palabra “*calidad*” hay que entenderla en términos de tasa de acierto, confianza y credibilidad. La tasa de acierto es ligeramente inferior a la que se obtiene con clasificadores *SVM* estándar (sin medidas de fiabilidad), pero esto no tiene gran significado ya que los clasificadores *SVM* estándar no proporcionan medidas de la fiabilidad de cada predicción. La calidad de los clasificadores conformales desarrollados viene determinada por una alta tasa de acierto junto con elevados valores de confianza de las predicciones efectuadas y la baja credibilidad de las predicciones falladas.

Las tasas de acierto más elevadas, en el orden del 97%, corresponden a los métodos transductivos estándar y al método ICP (Uno-Contra-Resto).

Los valores medios de confianza obtenidos son elevados en el caso de todos los métodos transductivos estándar, particularmente en el caso de las implementaciones *uno-contra-uno* (que llegan a alcanzar valores del orden del 99%). En el caso de los métodos transductivos con función *hash*, los valores medios de confianza también son altos, encontrándose al mismo nivel de los correspondientes a las implementaciones *uno-contra-el-resto* de los métodos transductivos estándar, siendo, por tanto, ligeramente inferiores a los correspondientes a las implementaciones *uno-contra-uno*. Con respecto a los predictores conformales inductivos, los valores medios de confianza asociados a sus predicciones son altos, comparables con los valores más altos obtenidos en los casos de los métodos transductivos estándar siempre que se

utilice un *kernel* RBF. Son claramente inferiores cuando se utilizan los *kernels* lineal y polinómico.

En cuanto a los valores medios de la credibilidad para las predicciones incorrectas, los mejores datos corresponden igualmente a las implementaciones *uno-contra-uno* de los métodos transductivos estándar. Los peores resultados se obtienen con los ICP, siendo estos clasificadores competitivos solamente en el caso de la implementación *uno-contra-el-resto*.

En general los valores de confianza generados por un clasificador conformal tienden a crecer con el aumento de la cantidad de datos utilizados en el proceso de predicción (también disminuye progresivamente la dispersión de sus valores) hasta alcanzar una fase de saturación. En la Figura 26 se muestra una curva típica de evolución de la confianza, correspondiendo la misma a un clasificador con un *algoritmo base* entrenado incrementalmente, CP (SVMi_SLS uno-contra-uno), en un escenario de aprendizaje *online*. En el caso de los *experimentos* presentados, la validación cruzada utiliza como datos de entrenamiento cerca de 1034 imágenes (9/10 de los datos disponibles) por lo que se espera para cada predicción efectuada un alto valor de la confianza, ya sea en la región de saturación o cerca de esta. Este hecho queda confirmado por los valores medios presentados, los cuales se encuentran por encima del 90% en la mayoría de los casos.

En estas circunstancias, la calidad de la predicción tiene como principal indicador el valor de credibilidad asociado, dependiendo estos valores de en qué medida cada objeto de test está suficientemente representado por el conjunto de entrenamiento. Por consiguiente, los referidos valores de credibilidad pueden ser utilizados como criterio para el filtraje de las predicciones de baja calidad, siendo importante para tal que los errores de predicción y las bajas credibilidades estén correlacionados, lo que ocurre con la mayoría de los clasificadores presentados en esta Tesis.

Un factor de peso que condiciona la aplicación práctica de la metodología de la clasificación conformal suele ser su relativa ineficiencia temporal. En esta Tesis se han probado diversos métodos con el objetivo de mejorar la eficiencia temporal del método de predicción conformal, utilizándose diferentes enfoques tales como el uso de máquinas *SVM* entrenadas incrementalmente (SVMi_SLS y SVMi_CP) como *algoritmo base* en un marco de clasificación transductiva, la división de los datos de entrenamiento utilizando una función *hash* (Hashed CP) y la combinación de la transducción y la inducción en el proceso de clasificación (ICP).

Los resultados obtenidos muestran que la metodología ICP supera en gran medida a los demás métodos. Los tiempos de CPU obtenidos son inferiores a los de los demás métodos en varios órdenes de magnitud.

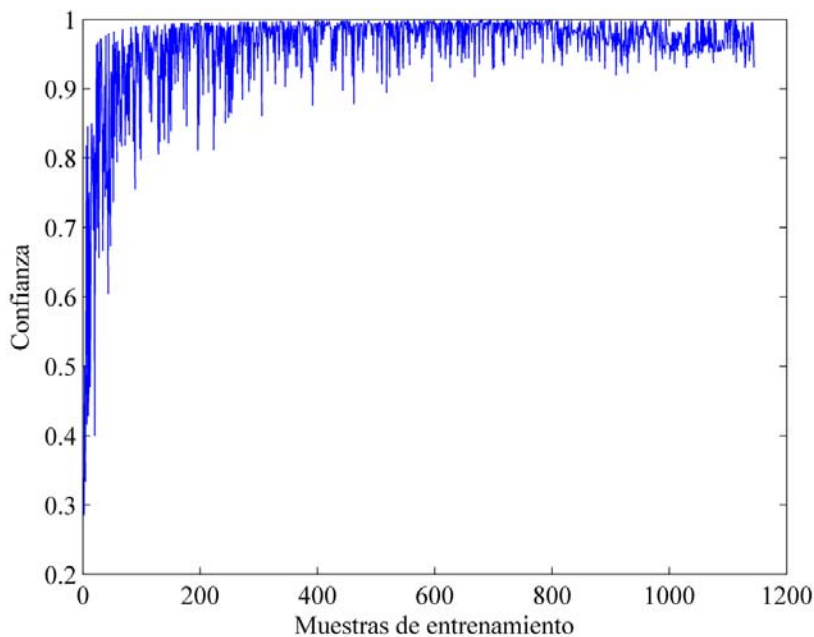


Figura 26: Evolución de la confianza con la cantidad de muestras de entrenamiento; CP (SVMi_SLS uno-contra-uno) entrenado en un *escenario online*

En lo que respecta a los restantes métodos probados, hay que enfatizar que la utilización de *SVM* entrenado incrementalmente en el enfoque de Syed, Liu y Sung (SVMi_SLS) tiene gran potencial para mejorar la eficiencia temporal en el caso de grandes conjuntos de entre-

namiento. En el caso del conjunto de datos utilizado este potencial es más notable al hacer uso del nivel 4 de descomposición en el preprocesamiento, con el cual los resultados obtenidos con esta variante del clasificador son claramente mejores que los correspondientes a los demás a partir de las 300 – 400 muestras en el conjunto de entrenamiento. Por otro lado, se puede destacar que los métodos transductivos con función *hash* (Hashed CP) proporcionan mejores resultados que los demás siempre que se utilice el nivel 5 de descomposición y conjuntos de entrenamiento pequeños y medianos. Se pudo ver que los resultados obtenidos son mejores para el caso de conjuntos de entrenamiento con menos de 600 muestras.

El efecto esperado de reducción del tiempo de procesamiento, respecto a la variante estándar de predicción conformal, no es observado en el caso de utilizar máquinas *SVM* entrenadas incrementalmente en el enfoque de Cawenberghs y Poggio (SVMi_CP). Esto puede deberse al hecho de que la eficiencia del método depende de la estrategia de almacenamiento y del cálculo de matrices para la determinación de los coeficientes de sensibilidad y actualización del jacobiano inverso para cada problema binario auxiliar. La implementación de este enfoque de entrenamiento incremental en un lenguaje de programación a un nivel más bajo podría resultar beneficioso con respecto a la eficiencia temporal del método (Laskov, 2006).

En general, se puede concluir que la metodología empleada posee un elevado potencial para la clasificación con confianza y credibilidad de las imágenes del TSD. En la operación del TJ - II se estima como requisito de tiempo para la clasificación en tiempo real de las imágenes del TSD unos pocos segundos, una vez las descargas se producen separadas por intervalos de tiempo de cerca de 10 minutos.

5.2. SELECCIÓN ACTIVA DE CONJUNTOS DE ENTRENAMIENTO

En esta sección se aborda el problema de la reducción del tamaño de los conjuntos de entrenamiento a través del aprendizaje activo.

5.2.1. Introducción

El TJ – II, a semejanza de otros dispositivos de fusión, posee almacenados una gran cantidad de datos de varios tipos, en muchos casos debidamente etiquetados, que han sido acumulados durante varias campañas experimentales. En función de esta situación, surgen varias preguntas al tratar de definir una estrategia para la implementación y entrenamiento de un clasificador, por ejemplo, ¿cuál es la cantidad de ejemplos más conveniente para el entrenamiento de un clasificador?, o, de entre los ejemplos disponibles ¿cuáles poseen mayor poder descriptivo y son por tanto preferibles para el entrenamiento del clasificador?

En particular, la clasificación con predictores conformales es reconocidamente un proceso intensivo desde el punto de vista computacional. La utilización de máquinas *SVM* como *algoritmo base* para el referido método implica el entrenamiento de varias máquinas para la clasificación de un único objeto y el entrenamiento de máquinas *SVM* se hace cada vez menos tratable a medida que la cantidad de datos en el conjunto de entrenamiento va creciendo. Esto hace emerger un problema importante que consiste en tratar de seleccionar un conjunto de datos de entrenamiento que sea mínimo en cuanto a su tamaño pero suficientemente bueno en cuanto a los resultados de la clasificación.

Semejante problema se puede abordar desde la perspectiva del aprendizaje activo. El aprendizaje activo suele ser utilizado en diversos problemas de aprendizaje supervisado, generalmente con el propósito de minimizar el coste del proceso de etiquetado de los datos en casos donde los datos no etiquetados abundan o pueden ser obtenidos fácilmente pero el proceso

de etiquetado es difícil o caro. Existen muchos ejemplos de estas aplicaciones a varios problemas reales tales como el procesamiento de lenguaje natural (Dagan, 1995), (Thompson, 1999), (Olsson, 2009a), la clasificación de texto (Lewis, 1994a, 1994b), (Schohn, 2000), (Tong, 2002), la recuperación de imágenes (Tong, 2001), el análisis de seguridad en sistemas de fornecimiento de energía (Cohn, 1994) o el aprendizaje de funciones de gradación (*ranking functions* en inglés) para recuperación de información (Yu, 2005), por ejemplo.

Recientemente ha crecido el interés por la utilización de la *inferencia transductiva* (en donde se puede incluir la predicción conformal) en problemas de reconocimiento de patrones. Ho y Wechsler (2003, 2008) hicieron la extensión del uso de la *transducción* para el dominio del aprendizaje activo, introduciendo el enfoque conocido como “*consulta por transducción*” (QBT, abreviatura del inglés *Query-by-Transduction*). En dicho enfoque, la selección de los objetos para la asignación de las etiquetas se realiza con base en los p -valores determinados a través de la utilización de la predicción conformal. Este enfoque fue ampliado posteriormente por Balasubramanian, Chakraborty y Panchanathan (2009) introduciendo la denominada “*consulta por transducción generalizada*” (GQBT, abreviatura del inglés *Generalized Query-by-Transduction*).

En esta Tesis, se propone la utilización del aprendizaje activo para la selección (a efectos de clasificación) de un conjunto de entrenamiento reducido pero efectivo de entre un conjunto de datos debidamente etiquetado y además relativamente grande. La “*efectividad*” del referido conjunto reducido se debe entender en el sentido de que el mismo debe proporcionar altas tasas de acierto con predicciones fiables, utilizando para el entrenamiento un conjunto de datos con menos ejemplos que los conjuntos de entrenamiento originales utilizados con métodos no activos. Para ello, se va a recurrir a la metodología de la predicción conformal de acuerdo al enfoque QBT.

A continuación se explican algunos elementos acerca del aprendizaje activo y su respectiva implementación, los *experimentos*⁹ realizados y las conclusiones más relevantes acerca de su aplicación.

5.2.2. Aprendizaje activo

Los problemas de clasificación se abordan habitualmente mediante un enfoque de muestreo aleatorio, es decir, los objetos (o muestras) utilizados en el aprendizaje son seleccionados aleatoriamente a partir del conjunto de objetos disponibles. Este es algunas veces denominado “*aprendizaje a partir de ejemplos*” o también “*aprendizaje pasivo*” (Cohn, 1994), (Olsson, 2009a).

El “*aprendizaje activo*” constituye un paradigma alternativo. Settles (2009) y Olsson (2009a) presentan introducciones al campo del aprendizaje activo. En el primer caso se hace un estudio en general de la literatura relevante en el campo y el segundo se circunscribe al dominio del procesamiento del lenguaje natural.

A diferencia del aprendizaje pasivo, en el activo los algoritmos de aprendizaje poseen cierto control sobre las entradas utilizadas para el entrenamiento, en vez de seleccionarlas de forma aleatoria (Cohn, 1994). Por consiguiente, el algoritmo juega, en este caso, un papel importante en la definición de qué muestras del conjunto de datos deben ser utilizadas en el aprendizaje. La hipótesis clave consiste en que si los datos a ser incluidos en el conjunto de entrenamiento son seleccionados de manera apropiada, el algoritmo de aprendizaje puede alcanzar mejores rendimientos con menos datos, rendimientos que con frecuencia son mejores que los alcanzados cuando se utilizan todos los datos disponibles para el entrenamiento (Schohn, 2000), (Ho, 2008).

⁹ Recuérdese que por *experimento* se designa, en este caso, la ejecución controlada de un algoritmo, con un conjunto de parámetros dados, para análisis de su efectividad

Para la implementación del aprendizaje activo, el algoritmo de aprendizaje debe tener la capacidad de consultar un “*experto*”, por ejemplo un especialista humano, cuya función es etiquetar los objetos, solicitando el etiquetado de las muestras de datos que parecen más informativas para el proceso de aprendizaje. En general, en cualquier implementación del aprendizaje activo, las mencionadas “*consultas*” al experto consisten en objetos que no estén todavía etiquetados.

De acuerdo a Settles (2009), en la literatura se consideran tres escenarios de aprendizaje activo fundamentales: (i) síntesis de *consultas* de pertenencia, (ii) muestreo selectivo en *stream*, también designado como secuencial u *online*, y (iii) muestreo basado en *pool*.

El primer escenario constituye uno de los primeros que fueron investigados en el dominio del aprendizaje activo (Angluin, 1987, 1988). En el mismo, el propio algoritmo de aprendizaje hace la síntesis artificial de las *consultas* y solicita sus etiquetas al experto. El algoritmo construye las *consultas* alterando los valores de los atributos de los objetos de tal forma que la respuesta a las *consultas* sea lo más informativa posible. La síntesis de las *consultas* es efectiva en varios tipos de problemas. Settles (2009) menciona un ejemplo de aplicación de este escenario en el diseño de experimentos biológicos ejecutados de forma autónoma por un robot científico. En el referido ejemplo, la utilización de la estrategia de aprendizaje activo resultó en ganancias significativas en cuanto al coste de los materiales experimentales respecto a otras estrategias de diseño, como son los casos de la ejecución sistemática del experimento menos costoso o la selección aleatoria de los experimentos a ser ejecutados.

Sin embargo, el uso de este escenario puede conducir a dificultades en los casos en que el experto sea un especialista humano y se generen objetos ininteligibles para el mismo. Settles (2009) y Olsson (2009a) hacen referencia a algunos ejemplos relativos a la referida dificultad en dominios como el reconocimiento de escritura manual o la categorización de textos.

En los otros dos escenarios alternativos, muestreo selectivo en *stream* y muestreo basado en *pool*, el desarrollo del aprendizaje se realiza utilizándose instancias de los objetos existentes en un *espacio de entradas*. Esto se basa en la suposición de que la obtención de un objeto no etiquetado a partir de la distribución de entrada es relativamente barata con respecto a etiquetar dicho objeto. Todos los objetos utilizados por el algoritmo en el aprendizaje provienen de datos no etiquetados existentes. Estos son previamente muestreados a partir de una distribución de entrada, filtrados en base a un criterio determinado, etiquetándose y utilizándose para el entrenamiento solamente las instancias más informativas para el aprendizaje (Cohn, 1994), (Lewis, 1994b).

La principal diferencia entre los muestreos basados en *stream* y en *pool* consiste en que en el primer caso se exploran los datos no etiquetados secuencialmente y las decisiones sobre etiquetar o no las muestras son tomadas individualmente, mientras que en el segundo caso, se evalúa y ordena toda la colección de datos no etiquetados antes de la selección de las mejores muestras para etiquetado (Settles, 2009).

En todos los escenarios de aprendizaje activo se reserva un papel importante a los métodos utilizados para evaluar qué objetos no etiquetados son más informativos, denominados "*estrategias de consulta*" (en inglés *query strategies*). Fueron introducidas diferentes alternativas con esta finalidad tales como "*muestreo por incertidumbre*" (en inglés *uncertainty sampling*) (Lewis, 1994a, 1994b), "*consulta por comité*" (*QBC*, abreviatura del inglés *Query-by-Committee*) (Seung, 1992), (Dagan, 1995), (Freund, 1997) o los "*métodos de ponderación de densidad*" (en inglés *density weighted methods*) (McCallum, 1998), (Fujii, 1998), entre otras. En (Settles, 2009) se puede encontrar un resumen extensivo acerca de las varias estrategias propuestas.

En esta Tesis se ha adoptado como estrategia de *consulta* el "*muestreo por incertidumbre*" en un escenario basado en *pool* (Lewis, 1994b). Esta estrategia consiste en un proceso

iterativo de etiquetado manual de los objetos, entrenamiento de un clasificador a partir de los mismos y utilización del clasificador para la selección de nuevos objetos cuya pertenencia a la clase predicha no es clara. La implementación de semejante estrategia es viable en los casos en que los clasificadores utilizados no solamente toman decisiones de clasificación sino que también estiman el nivel de certeza de las mismas, como, por ejemplo, el caso de los clasificadores probabilísticos o, por supuesto, de los predictores conformales.

Los predictores conformales proporcionan información acerca de la fiabilidad de las predicciones de una forma natural, asociando a cada predicción un valor de confianza y otro de credibilidad, véase la Sección 4.6. Esa información (o equivalentemente los p -valores determinados en el proceso de predicción), puede ser utilizada como base de la decisión en el proceso de *muestreo por incertidumbre* de distintas maneras, sirviendo así para la definición de diferentes criterios de *consulta*. La alternativa más obvia consiste en utilizar llanamente el valor de credibilidad, o de confianza, asociado a cada predicción y, este caso, se tendría como criterio de *consulta*,

- $I(\mathbf{x}) = cred = p_i$ o
- $I(\mathbf{x}) = conf = 1 - p_j$,

donde p_i y p_j son respectivamente los mayores p -valores (ordenados de mayor a menor) obtenidos en el proceso de clasificación conformal. En la medida en que los referidos valores se acercan a cero más incierta es la predicción y, por tanto, más informativa es la muestra.

Semejante criterio considera solamente la información contenida en el mayor o en el segundo p -valor, ignorando la información contenida en todos los demás. Alternativamente, Ho y Weschler (2003, 2008) introdujeron una medida de calidad de la información, desarrollando el enfoque designado por los mismos como "*consulta por transducción*" (QBT, abreviatura del inglés *Query-by-Transduction*). Esta medida de calidad ha sido utilizada tanto en el esce-

nario de aprendizaje activo basado en *pool* como en *stream*, habiéndose usado como *algoritmo base* en la predicción conformal el método de k vecinos más cercanos (Duda, 2001), (Witten, 2005) y *SVM* (véase la Sección 4.8 para una introducción), respectivamente.

La referida medida se basa en el uso conjunto de la información relativa a la credibilidad y la confianza de las predicciones y se define como

$$I(\mathbf{x}) = cred - (1 - conf) = |p_i - p_j|,$$

donde p_i y p_j son nuevamente los dos mayores p -valores computados en el proceso de predicción, implementándose de esa manera una medida en forma de intervalo, en el sentido utilizado por Abe y Mamitsuka (1998) o por Melville y Mooney (2004), es decir, una medida que considera la información contenida en la diferencia entre dos p -valores en lugar de la contenida en el valor individual de cualquier de ellos.

Este enfoque fue ampliado posteriormente por Balasubramanian, Chakraborty y Panchanathan (2009) a través de la denominada “consulta por transducción generalizada” (GQBT, abreviatura del inglés *Generalized Query-by-Transduction*), habiendo sido aplicada la misma en un escenario basado en *stream*, utilizándose máquinas *SVM* como *algoritmo base* en la predicción conformal.

En este caso, en la definición del criterio de calidad de la información, se utilizan todos los p -valores determinados en la predicción conformal, o tantos como sean requeridos. Para tal fin, se define una matriz C cuadrada y simétrica de orden igual al número de clases existentes y que contiene los valores absolutos de las diferencias entre cada par de p -valores obtenidos en el proceso de clasificación,

$$C_{kl} = |p_k - p_l|.$$

Como medida de calidad de la información se utiliza el mayor de los valores propios de la matriz C , $I(\mathbf{x}) = \eta(C)$, el cual es directamente proporcional al mayor de los elementos de la matriz, es decir, a la mayor diferencia entre los pares de p -valores. Cuanto menor sea el valor de la medida así definida, menor es la diferencia entre los pares de p -valores y, por tanto, mayor es la incertidumbre asociada a la predicción.

Hasta aquí se han mencionado diversos criterios de calidad de la información, pero la estrategia de *consulta* se basa en la utilización de uno u otro criterio de forma individual. La utilización simultánea de varios criterios puede resultar en una plusvalía para el proceso de aprendizaje activo. A continuación se hace referencia a este aspecto.

A. Combinación de múltiples criterios

En el proceso de modelado de la incertidumbre, la búsqueda del mejor modelo en términos absolutos resulta ser una tarea infructífera una vez que el modelo verdadero es generalmente desconocido y los distintos modelos empleados representan aproximaciones al verdadero desde diferentes perspectivas. Sin embargo, generalmente, a través de la combinación de varios criterios es posible alcanzar mejores resultados que en el caso de la utilización de un único criterio. Esta idea suele ser aplicada en varios dominios, inclusive el del aprendizaje activo, véase por ejemplo (Shen, 2004), (Hsu, 2006), (Kapetanios, 2006), (Pilatowska, 2009) o (Balasubramanian, 2010).

En trabajos anteriores, tales como (Ho, 2003, 2008) o (Balasubramanian, 2009), la bondad de las estrategias de aprendizaje activo era determinada a través de las tasas de acierto alcanzadas por los clasificadores. En esta Tesis, con el aprendizaje activo se pretende seleccionar el conjunto de entrenamiento para un clasificador conformal y, por tanto, en el análisis de los resultados se consideran no solamente las tasas de acierto sino también las medidas de fiabilidad asociadas a las predicciones.

Para establecer un compromiso entre la mejoría de la tasa de acierto y de las medidas de fiabilidad, en esta Tesis se recurrió a un enfoque multicriterio para la toma de la decisión sobre el etiquetado o no de una muestra determinada, siendo posible la implementación del referido enfoque desde dos perspectivas distintas.

En la primera perspectiva, la implementación se lleva a cabo combinando dos de las medidas de incertidumbre anteriormente mencionadas. A tal fin, las mismas pueden ser combinadas de dos maneras diferentes, por “*marcador*” (en inglés *score*) o por “*rango*” (en inglés *rank*) (Hsu, 2006). En la combinación por marcador se utilizan los propios valores de las medidas de incertidumbre mientras que en la combinación por rango se hace el ordenamiento, en orden creciente, de los valores de cada medida y en la combinación se utiliza el posicionamiento correspondiente a cada entrada en la lista ordenada.

En ambos casos la combinación se realiza a través de una suma ponderada:

$$I(\mathbf{x}) = \mu I_1(\mathbf{x}) + (1 - \mu) I_2(\mathbf{x}),$$

donde $\mu \in [0,1]$ es un parámetro definido por el usuario que permite balancear el peso relativo de los criterios implicados en la suma.

Al implementar la estrategia de aprendizaje activo haciendo uso solamente de los criterios de información hasta ahora descritos, surgen dos cuestiones importantes. La primera cuestión tiene que ver con el hecho de que la construcción activa de un clasificador en base a criterios de incertidumbre, u otros como *QBC* por ejemplo, no ocurre de forma óptima, sino que suele suceder con frecuencia el etiquetado de muestras que resultan ser *outliers* (Roy, 2001), (Settles, 2008). El conocimiento de las etiquetas de estas muestras, al no ser representativas de los demás objetos que forman parte del conjunto de datos no etiquetado, poco aporta a la mejoría del clasificador. La segunda cuestión consiste en que la adición de un lote de nuevas muestras al conjunto de entrenamiento, siendo estas las k mejores de acuerdo a un criterio de

información, no conduce necesariamente a una mejoría en la calidad del clasificador mayor que la obtenida en el caso de añadir una única muestra (Brinker, 2003), (Xu, 2007). Esto se debe al hecho de que esta estrategia de selección no considera la posible superposición del contenido informativo de las muestras seleccionadas, incluyéndose con frecuencia en el lote varias muestras que son muy semejantes o incluso iguales entre sí.

Esto desemboca en la segunda perspectiva de la implementación del enfoque multicriterio, la cual se materializa considerando los conceptos de "*representatividad*" y "*diversidad*". En el proceso de aprendizaje activo, al seleccionar las nuevas muestras a incluir en el conjunto de entrenamiento, es importante que ellas sean no solamente informativas sino también representativas y diversificadas.

El criterio de representatividad trata de hacer uso del modelo de la distribución de los datos existentes. Una muestra se considera "*representativa*" cuando la misma proviene de una región densamente poblada del *espacio de entradas*, siendo, por tanto, similar a una cantidad relativamente grande de otras muestras.

En la literatura se pueden encontrar varios ejemplos de utilización de la información de densidad para la mejoría del aprendizaje activo, en base a diversas estrategias, como es el caso de su estimación a través de la distancia media (o la semejanza media) existente entre las muestras a ser seleccionadas y las demás muestras del *pool* o de la selección de muestras representativas de *clusters* existentes en el *espacio de entradas*, evitándose la etiquetación repetida de muestras pertenecientes a un mismo *cluster*. Algunos ejemplos pueden ser vistos en (McCallum, 1998), (Nguyen, 2004), (Shen, 2004), (Xu, 2007) o (Settles, 2008).

El concepto de diversidad pretende maximizar la aportación en términos informativos de un lote de muestras al ser seleccionado para *consulta*, garantizando que las muestras incluidas

en el lote sean lo más distintas posibles unas de las otras, reduciendo, por tanto, la redundancia en el contenido de información de las mismas.

Análogamente al caso anterior, se pueden encontrar en la literatura diferentes estrategias de implementación del criterio de diversidad. Algunas publicaciones consideran la construcción del lote de muestras de forma iterativa, siendo las muestras seleccionadas para inclusión en el lote en cada paso de acuerdo a diferentes criterios, como la maximización de los ángulos entre los hiperplanos inducidos por las muestras a incluir en el lote, la maximización de la distancia mínima entre la muestra candidata y las muestras ya incluidas en el lote o la definición de un umbral de semejanza, siendo incluidas en el lote únicamente las muestras cuyos niveles de semejanza respecto a las muestras ya incluidas estén por debajo del referido umbral. En otros casos se divide el conjunto de muestras candidatas en *clusters*, incluyéndose en el lote una muestra perteneciente a cada *cluster*, o se formula la selección de las muestras como un problema de optimización, seleccionándose el lote de muestras que permita con mayor eficiencia el alcance de un objetivo dado, como es el caso de la reducción de la matriz de información de Fisher o de la optimización de funciones objetivo que consideren de forma conjunta tanto la influencia de los datos etiquetados como de los no etiquetados. Ejemplos de las referidas estrategias pueden ser vistos en (Brinker, 2003), (Shen, 2004), (Hoy, 2006a, 2006b), (Xu, 2007) o (Guo, 2008).

En esta Tesis, la información de densidad se determina a partir de la división de los datos no etiquetados (que forman parte del *pool*) en un conjunto de *clusters*, utilizándose para tal el algoritmo *k*-medias (en inglés *k-means*) (Duda, 2001), (Sá, 2001). Como criterio de representatividad se considera la distancia mínima entre cada una de las muestras y el conjunto de centroides de los *clusters*, siendo una muestra más representativa cuanto más cerca se encuentre de uno de los centroides. Al calcular el índice de representatividad las distancias son nor-

malizadas en el intervalo $[0, 1]$, de forma tal que el índice de representatividad de una muestra concreta \mathbf{x} se determina a través de la expresión

$$R(\mathbf{x}) = 1 - \exp\left[-\frac{\min_i d(\mathbf{x}, c_i)}{\sigma}\right]$$

donde $d(\mathbf{x}, c_i)$ es la distancia euclídea entre el vector \mathbf{x} y el centroide c_i y σ es un factor de peso determinado empíricamente por el usuario.

En la estimación de la diversidad, también se hace uso de la distancia euclídea entre las distintas muestras. Para ello, el lote de datos se construye de forma iterativa, de manera semejante a lo descrito por Brinker (2003) o por Xu, Akella y Zhang (2007). Las muestras son seleccionadas secuencialmente, siendo en cada paso seleccionada la muestra cuya distancia mínima con relación a las muestras incluidas en el lote en los pasos anteriores es máxima, sirviendo la referida distancia como base para el cálculo del índice de diversidad para la muestra respectiva.

El índice de diversidad de una muestra concreta \mathbf{x} respecto a un lote de muestras $\{\mathbf{x}_i\}$, que es una distancia normalizada en el intervalo $[0, 1]$, se calcula en base a la expresión

$$D_{\mathbf{x}_i}(\mathbf{x}) = \exp\left[-\frac{\min_i d(\mathbf{x}, \mathbf{x}_i)}{\sigma}\right],$$

siendo $d(\mathbf{x}, \mathbf{x}_i)$ el vector de distancias euclídeas entre el vector \mathbf{x} y el conjunto de vectores $\{\mathbf{x}_i\}$ y σ un factor de peso determinado empíricamente por el usuario.

El proceso de determinación de los índices de diversidad se lleva a cabo de acuerdo al Algoritmo 5, el cual se representa esquemáticamente en la Figura 27. El algoritmo toma como entradas el conjunto de muestras candidatas, el factor de peso σ y el índice de la primera muestra a ser seleccionada (idx). En el caso de esta Tesis, la primera muestra se selecciona en

base al criterio de información utilizado, seleccionándose la muestra correspondiente al menor valor del referido criterio. Por otro lado, a la salida el algoritmo devuelve un vector con los índices de diversidad calculados para cada muestra.

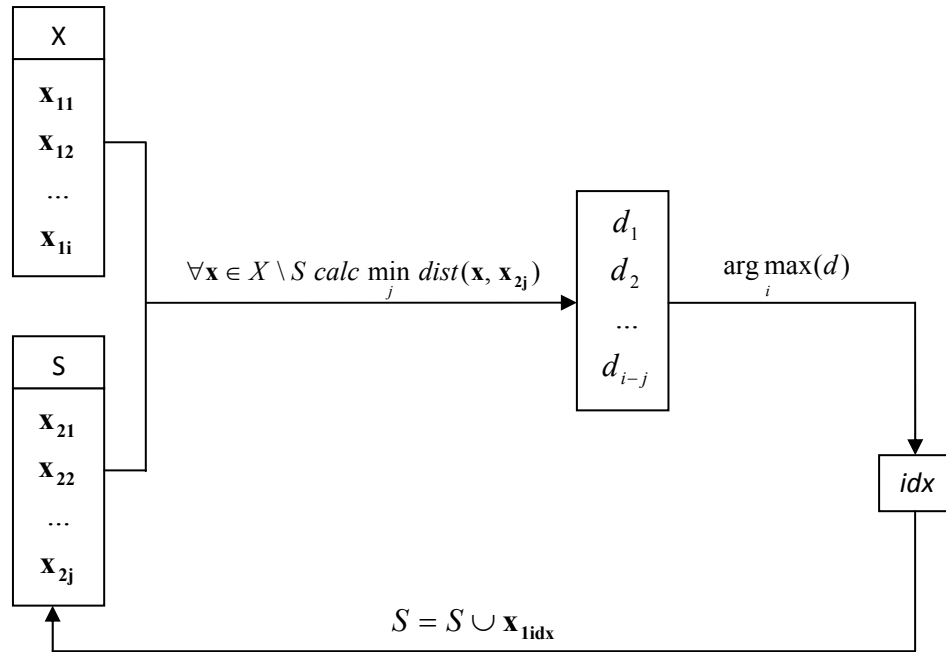


Figura 27: Representación esquemática del ciclo de determinación de la diversidad de un conjunto de datos

Algoritmo 5: determinación de la diversidad de un conjunto de datos

Entradas:

$X = \{x_1, \dots, x_n\}$ conjunto de objetos cuya diversidad se pretende determinar

σ factor de peso

idx índice de la primera muestra a ser seleccionada

Salidas:

D vector de índices de diversidad

$S = x_{idx}$

$D(idx) = 0$

repeat

$idx = \arg \max_{i: x_i \in X \setminus S} (\min_{j: x_j \in S} (d(x_i, x_j)))$

$d = \max_{i: x_i \in X \setminus S} (\min_{j: x_j \in S} (d(x_i, x_j)))$

$D(idx) = e^{-\frac{d}{\sigma}}$

$S = S \cup x_{idx}$

until $card(S) = card(X)$

B. Condición de parada

En el curso de un ciclo de aprendizaje activo, el conjunto de entrenamiento se construye de forma iterativa. Esto lleva a tener que responder una importante pregunta: ¿en qué momento se debe terminar el ciclo de aprendizaje? A pesar de que aparentemente la importancia de la condición de parada es menor que la definición de las estrategias de *consulta*, es importante y necesario definir una condición de parada.

Existen varias formas “*naturales*” de terminar un ciclo de aprendizaje activo, como por ejemplo, (i) alcanzar el conjunto de entrenamiento un tamaño determinado, (ii) agotar el *pool* de datos, dejando, por tanto, de existir en el *pool* muestras que cumplan los requisitos impuestos por el criterio de selección, o (iii) alcanzar un rendimiento preestablecido del clasificador (Li, 2006), (Zhu, 2010).

Sin embargo, la definición de un tamaño adecuado para el conjunto de entrenamiento, así como un nivel de rendimiento apropiado y alcanzable son cuestiones complejas y difíciles de decidir con antelación y de una manera general (Zhu, 2010). También ocurre con frecuencia que en el proceso de aprendizaje activo, a partir de cierto punto, el rendimiento del clasificador entrenado comienza a decaer, o evoluciona muy poco, a pesar de la existencia de posibles muestras que cumplen con los requisitos para su selección (Schohn, 2000), (Laws, 2008). Este hecho motiva la interrupción temprana del ciclo de aprendizaje.

En la literatura se consideran dos estrategias fundamentales para la definición de condiciones de parada temprana: (i) a través del análisis de la evolución del rendimiento de los clasificadores entrenados en cada iteración, como son los casos descritos en (Zhu, 2008a) y en (Laws, 2008), o (ii) a través del análisis de la evolución de la confianza global de los mencionados clasificadores, tal como se describe en (Vlachos, 2008), (Zhu, 2010) y (Ghayoomi, 2010).

Considerando el rendimiento, la manera más simple e intuitiva de definir una condición de parada tiene que ver con la utilización de un conjunto de datos separado y etiquetado para efectos de calibración. El aprendizaje debe finalizar cuando se dejen de verificar mejoras significativas en el rendimiento del clasificador al ser aplicado sobre el conjunto de calibración.

Sin embargo, surgen algunas cuestiones acerca de la utilización de semejante conjunto, como por ejemplo en cuanto a la definición de cuál debería ser el tamaño ideal para el mismo (Vlachos, 2008), (Zhu, 2010). La utilización de un conjunto demasiado pequeño puede dar lugar a estimaciones incorrectas, por el hecho de no ser suficientemente representativo de la distribución de los datos en el *pool*. Por otro lado, un conjunto demasiado grande es contraproducente en términos del esfuerzo de etiquetado (y eventualmente de cálculo) necesario, lo que es algo que va en sentido contrario al objetivo general de los trabajos existentes, que es la minimización del esfuerzo de etiquetado. Como alternativa, algunas veces se recurre a la utilización del *pool* de datos, o de un conjunto de referencia separado y no etiquetado, para la estimación del futuro error esperado, utilizándose para tal fin métodos estadísticos como son los casos descritos en (Laws, 2008) y (Zhu, 2008a).

La utilización de la confianza como base de la definición de una condición de parada no exige el etiquetado previo de los datos de referencia, siendo posible, por tanto, utilizar en la estimación de la confianza el *pool* de datos completo o conjuntos de datos separados relativamente grandes que debido a su tamaño la representatividad deja de ser un problema. Esta estrategia suele ser ampliamente utilizada, existiendo en la literatura varios trabajos en los cuales se utilizan diferentes medidas para la estimación de la confianza, como son los casos de medidas basadas en métodos estadísticos (Vlachos, 2008), (Ghayoomi, 2010), en distancias con relación al hiperplano de separación, en el caso de utilización de métodos de aprendizaje basados en margen, como *SVM*, (Schohn, 2000), (Vlachos, 2008) o heurísticas como el

nivel de concordancia entre los miembros de un comité de clasificadores (Olsson, 2009b) o el nivel de incertidumbre de las muestras candidatas a etiquetado (Zhu, 2010).

En esta Tesis se implementaron diversas variantes de condición de parada, basándose todas ellas en el análisis de la estabilidad de las predicciones hechas por los clasificadores entrenados en las diferentes iteraciones, a semejanza de lo descrito por Laws y Schutze (2008) y por Bloodgood y Vijay-Shanker (2009). Se considera que las predicciones son estables cuando determinada medida deja de evolucionar durante varias iteraciones consecutivas, o su evolución es tan pequeña que se encuentra dentro de determinados límites definidos por el usuario. Para valorar la estabilidad de las predicciones se utilizaron diferentes indicadores:

- *La tasa de error de clasificación.* Esta se calcula sobre un conjunto de datos separado utilizado para fines de calibración, siendo el conjunto de datos necesariamente etiquetado.
- *El valor medio de la confianza de las predicciones efectuadas por el clasificador.* Se utilizó como base para la determinación de la incertidumbre un clasificador conformal y los valores de confianza asociados por este a cada predicción efectuada pueden ser utilizados con esta finalidad. Se utilizan los valores relativos a la clasificación del *pool* de datos completo o de un conjunto separado de datos de calibración, que no tiene que ser necesariamente etiquetado.
- *La tasa de error de clasificación calculada sobre el lote de datos seleccionado para etiquetado* Siendo estos los datos más difíciles de clasificar en cada iteración, el error cometido al clasificarlos constituye una buena medida de la confianza del clasificador sobre los restantes datos no etiquetados (Zhu, 2010). En este caso la corrección de la clasificación es verificada en base al *feedback* proporcionado por el experto.

- *Los cambios de las clases atribuidas a los datos no etiquetados del pool en iteraciones consecutivas.* Esta estrategia, introducida por Zhu, Wang y Hovy (2008b), consiste en monitorizar las alteraciones en las etiquetas que el clasificador atribuye a los datos no etiquetados en iteraciones consecutivas. Si deja de haber alteraciones el ciclo de aprendizaje debe finalizar.

C. Algoritmo de aprendizaje activo

En la implementación concreta que se hizo del algoritmo se empleó la estrategia de *muestreo por incertidumbre* basado en *pool* (Lewis, 1994b). Por consiguiente, se evalúa de forma iterativa todas las muestras candidatas existentes en el *pool* de acuerdo a un criterio de incertidumbre y se añaden las muestras seleccionadas al actual conjunto de entrenamiento. El algoritmo considera un conjunto de datos etiquetados T para el entrenamiento inicial y un *pool* de muestras no etiquetadas U , candidatas a pertenecer al conjunto de entrenamiento. Opcionalmente se puede considerar un conjunto de calibración C , etiquetado o no, para ser utilizado en la determinación de la condición de parada. Por otro lado, el algoritmo considera dos parámetros: un valor τ que representa el umbral para el nivel de incertidumbre de las predicciones y otro valor, β , que define la cantidad máxima de muestras a ser incluidas en el conjunto de entrenamiento en cada iteración.

A medida que transcurre el aprendizaje, en cada iteración, el algoritmo toma, de las muestras cuyos índices de certeza asociados a sus predicciones se encuentran por debajo del umbral τ , las β muestras más alejadas del umbral, las añade al conjunto de entrenamiento y entrena un clasificador mejorado, prosiguiendo las iteraciones hasta que se alcanza una condición de parada. Este proceso es descrito en el Algoritmo 6.

Para detener el algoritmo se considera un conjunto de condiciones diferentes. El proceso de selección debe seguir hasta que se verifique una de las siguientes condiciones: a) el tamaño

del conjunto de entrenamiento alcanza un valor N previamente especificado; b) se agota el *pool* de datos U , lo que ocurre cuando dejan de existir en el *pool* muestras que cumplan con el requisito establecido para que sean seleccionadas; c) activación de una condición de parada temprana, descritas en la sección anterior.

Opcionalmente, si la selección activa termina sin que el conjunto de entrenamiento alcance el tamaño deseado, el algoritmo completa dicho conjunto de entrenamiento con muestras seleccionadas de forma aleatoria a partir de los restantes ejemplos, efectuándose una selección híbrida de las muestras (Ho, 2003).

Algoritmo 6: aprendizaje activo

Entradas:

T conjunto de entrenamiento inicial
 U *pool* de datos U
 C conjunto de calibración (opcional)

Salidas:

E conjunto de entrenamiento seleccionado

Entrenar sobre T un clasificador inicial

while not condición de parada

 Aplicar el actual clasificador al *pool* de datos U

 Ordenar las muestras del *pool* de acuerdo al criterio de incertidumbre considerado

 Seleccionar β muestras, cuyos valores de incertidumbre son inferiores al umbral τ y se encuentran más alejadas del referido umbral

 Solicitar al *experto* la etiquetación de las muestras y añadirlas al conjunto de entrenamiento E

 Entrenar un nuevo clasificador con el conjunto de entrenamiento expandido

 Verificar la condición de parada

end while

Dar salida al conjunto seleccionado E

5.2.3. Experimentos y resultados

A continuación se hace la presentación de los *experimentos* y los resultados obtenidos al emplear la estrategia de aprendizaje activo desarrollada con un conjunto de datos real, las imágenes del TSD del TJ – II.

A. Implementación de los algoritmos

A semejanza de las rutinas referidas en la Sección 5.1.2, en este caso los algoritmos empleados fueron desarrollados en Matlab, utilizándose nuevamente para desarrollo de los métodos de aprendizaje el *software* Spider (Weston, 2006). Para la clasificación con *SVM* se empleó la librería LIBSVM (Chang, 2011), librería que entre otras se encuentra integrada en Spider.

B. Metodología

En los *experimentos* realizados se empleó un total de 1149 imágenes, repartidas en 5 clases, véase el Anexo A, siendo utilizado como vector de características los coeficientes de detalle vertical de las correspondientes transformadas *wavelet* de Haar (Mallat, 1999), al nivel 4 de descomposición.

Estando el conjunto de datos debidamente etiquetado, los *experimentos* fueron encaminados hacia la selección de un conjunto reducido, pero efectivo, de muestras para el entrenamiento de un clasificador conformal. Con este motivo, se emplearon estrategias de aprendizaje activo basadas en diferentes criterios de *consulta*, comparándose los resultados con los correspondientes a la selección aleatoria del conjunto de entrenamiento.

En cada ejecución de los *experimentos*, el conjunto de datos disponible fue subdividido de forma aleatoria en tres partes, un conjunto de test con 200 muestras, utilizado para la construcción de las curvas de aprendizaje, un conjunto de calibración con 150 muestras, utilizado para la implementación de la condición de parada temprana, y un *pool* de datos constituido

por las demás muestras. Para implementar la condición de parada temprana se empleó la tasa de acierto del clasificador entrenado al aplicarle como datos de entrada el conjunto de datos de calibración. La selección de esta estrategia está motivada por el hecho de que la evaluación de la tasa de acierto es muy simple en este caso, ya que los datos disponibles se encuentran debidamente etiquetados y su cantidad es relativamente grande.

El aprendizaje activo se inicia con 5 imágenes, una de cada clase, seleccionadas de manera aleatoria, y en cada iteración se selecciona y se añade al conjunto de entrenamiento hasta un total de 15 imágenes, las más informativas, teniendo como premisa el hecho de que el valor del criterio de *consulta* para las mismas debe encontrarse por debajo de un valor umbral fijado empíricamente. Este valor fue fijado inicialmente en 0,2 pero tiene la posibilidad de auto adaptarse en el transcurrir del aprendizaje hasta un valor máximo de 0,5. El conjunto de entrenamiento es seleccionado de forma híbrida (Ho, 2003), es decir, la selección activa se lleva a cabo hasta que se activa la condición de parada y, a continuación, las imágenes son seleccionadas de forma aleatoria hasta que se alcance un total de 700 imágenes en el conjunto de entrenamiento.

Los resultados se muestran en la sección siguiente a través de las correspondientes curvas de aprendizaje, utilizándose para su construcción los resultados alcanzados al clasificar el conjunto de test. A medida que el conjunto de entrenamiento crece, el clasificador es sucesivamente entrenado y utilizado para la clasificación del conjunto de test.

Para tener la certeza de que los resultados alcanzados no dependen de factores como la inicialización del algoritmo, la división hecha de los datos o el orden en que las muestras son observadas por el algoritmo, el proceso de aprendizaje fue repetido 10 veces para cada caso, siendo los resultados presentados en las curvas de aprendizaje las medias y las correspondientes desviaciones estándar.

En todos los *experimentos* se empleó como *algoritmo base* la variante de *SVM* entrenado incrementalmente según el enfoque de Syed, Liu y Sung (SVMi_SLS), en su implementación *uno-contra-el-resto*. Se empleó un *kernel* RBF, con $\sigma = 2^{11}$, y los valores de $C = \infty$ y $\lambda = 0,4$, como parámetros de regularización y del algoritmo *uno-contra-el-resto*, respectivamente.

Resultados

En las figuras 28, 29 y 30 se representan las curvas de aprendizaje activo correspondientes a tres criterios de *consulta* distintos, QBT, GQBT y confianza. Como base de comparación se muestra también la curva correspondiente al aprendizaje a través de la selección aleatoria del conjunto de entrenamiento. Las citadas curvas presentan la evolución de la tasa de acierto (Figura 28), la confianza (Figura 29) y la credibilidad (Figura 30) a lo largo del proceso de aprendizaje, es decir, su comportamiento a medida que el clasificador va siendo entrenado cada vez con una mayor cantidad de datos.

A pesar del hecho de que la confianza y la credibilidad son valores asociados a cada predicción en particular y no responden a una medida relativa al conjunto de las predicciones efectuadas, en el caso de la confianza existe un patrón de evolución típico a medida que aumenta el tamaño del conjunto de entrenamiento; la confianza tiende a crecer monótonamente, con una dispersión cada vez menor de sus valores, hasta alcanzar un punto de saturación. Por esta razón, es posible la utilización de sus valores medios para tener una idea acerca de la evolución general de la confianza en cada caso. En lo que respecta a los valores de credibilidad, sus valores medios son menos informativos, existiendo siempre una gran dispersión alrededor de la media, independientemente del tamaño del conjunto de entrenamiento. No obstante, siempre es posible observar una tendencia al crecimiento de su valor medio a medida que aumenta el tamaño del conjunto de entrenamiento, siendo las referidas medias más elevadas en unos casos que en otros.

En la Figura 28 se puede observar que la utilización de estrategias de aprendizaje activo, en el caso de QBT y GQBT, mejora la evolución de la tasa de acierto del clasificador respecto a la selección aleatoria, permitiendo la obtención de picos más elevados de acierto con menos datos. Tales picos se alcanzan alrededor de las 200 muestras en el conjunto de entrenamiento. En cuanto a la utilización de la confianza como criterio de *consulta*, no se verifica una mejoría en la evolución de la tasa de acierto en comparación con la selección aleatoria, quedándose inclusive su curva por debajo en las etapas más tempranas del aprendizaje.

Por otro lado, en la Figura 29 se puede ver que, a diferencia de lo que ocurre con la tasa de acierto, la evolución de la confianza es más lenta al emplear los criterios QBT y GQBT como base de las estrategias de aprendizaje activo, en comparación con la evolución correspondiente a la selección aleatoria del conjunto de entrenamiento. En el caso de la utilización de la confianza como criterio de *consulta*, se verifica una evolución más rápida de los valores de confianza de las predicciones, relativamente a la selección aleatoria del conjunto de entrenamiento, alcanzándose el pico más elevado alrededor de las 250 muestras.

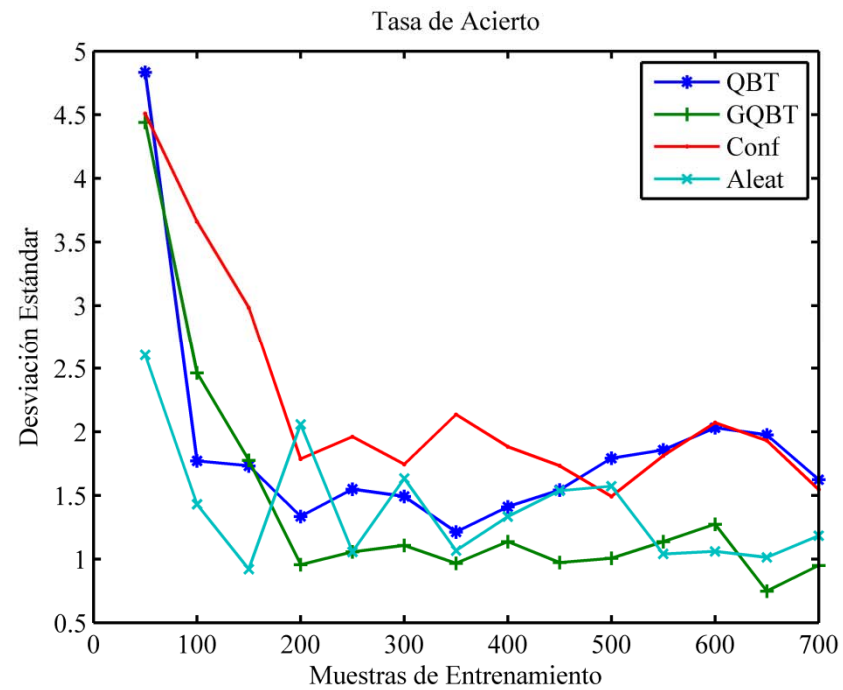
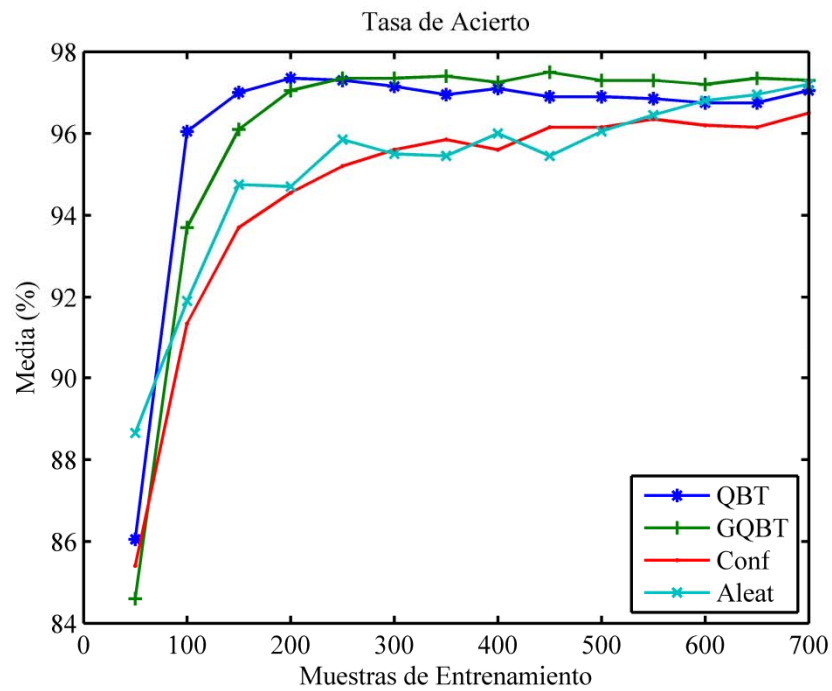


Figura 28: Tasa de acierto media y desviación estándar de la tasa de acierto en función del número de muestras de entrenamiento (después de 10 ejecuciones)

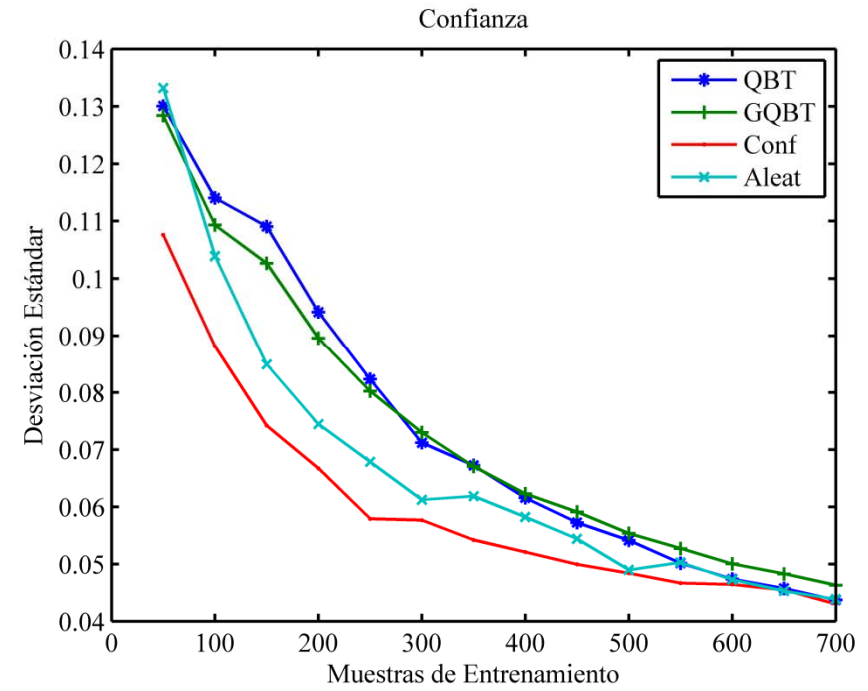
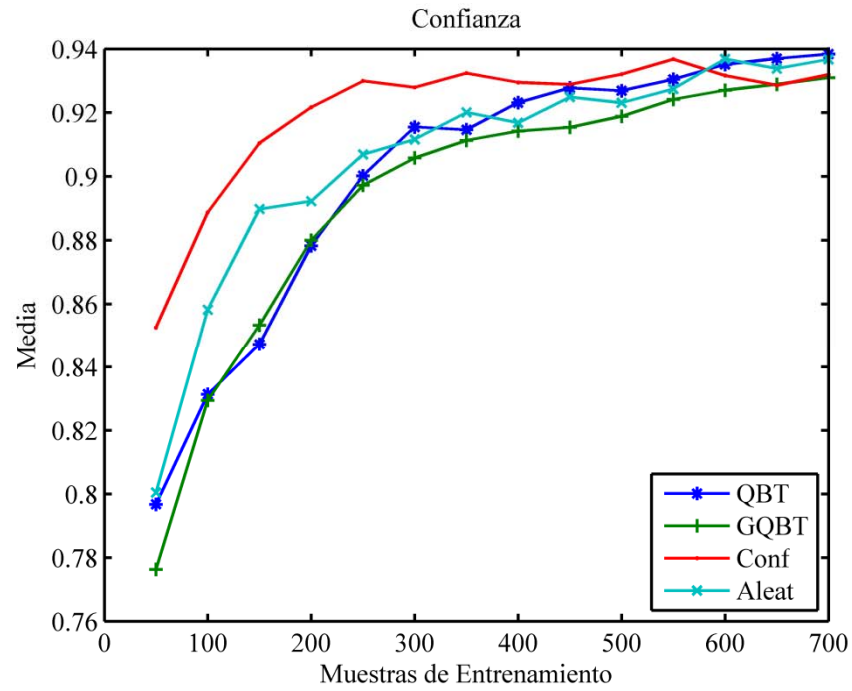


Figura 29: Confianza media y desviación estándar de la confianza en función del número de muestras de entrenamiento (después de 10 ejecuciones)

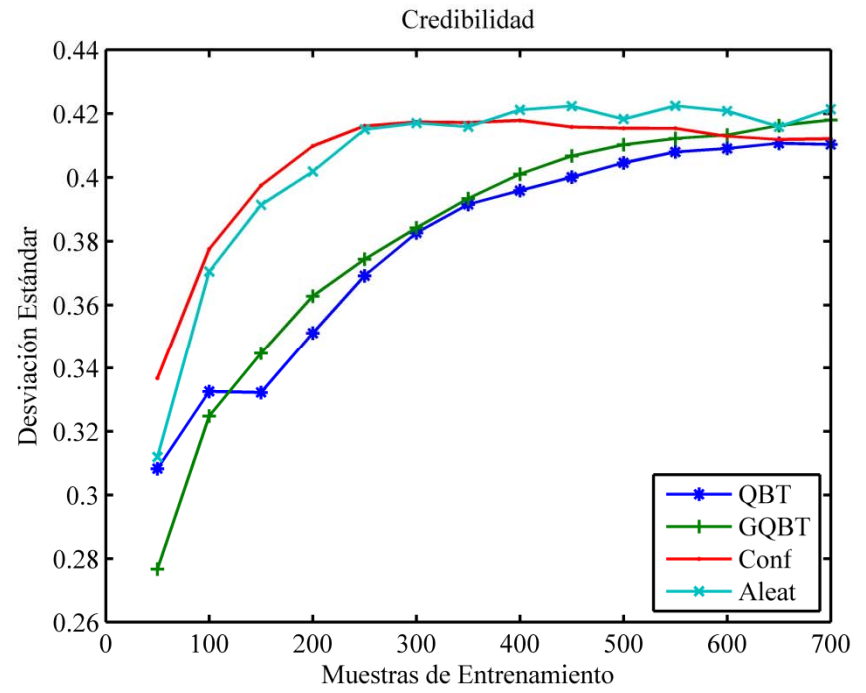
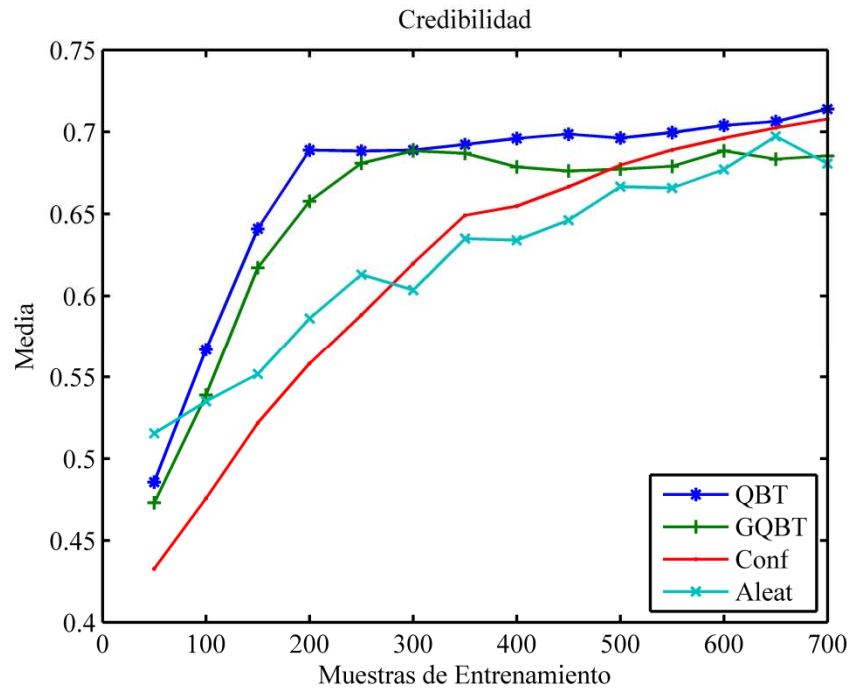


Figura 30: Credibilidad media y desviación estándar de la credibilidad en función del número de muestras de entrenamiento (después de 10 ejecuciones)

Por lo expuesto en los párrafos anteriores, se observa un ligero problema de equilibrio entre la mejoría de la evolución de la tasa de acierto y de la confianza al ser empleados los diferentes criterios de *consulta*. Es decir, para los criterios que proporcionan una mejoría en la evolución de la tasa de acierto con relación a la selección aleatoria, no se observan mejoras, o se observa incluso una degradación, en la evolución de la confianza y viceversa.

Para balancear las evoluciones de la tasa de acierto y de la confianza de las predicciones, se probó el enfoque multicriterio en la definición de la incertidumbre de las predicciones. Se efectuó la combinación por rango (Hsu, 2006) de dos criterios de información distintos, QBT y confianza, a través de su suma ponderada,

$$I(\mathbf{x}) = \mu I_1(\mathbf{x}) + (1 - \mu) I_2(\mathbf{x}),$$

correspondiendo en este caso $I_1(\mathbf{x})$ a QBT e $I_2(\mathbf{x})$ a la confianza.

En las figuras 31, 32 y 33 se representan las curvas de aprendizaje obtenidas para dos combinaciones diferentes, correspondientes a $\mu = 0,5$ y $\mu = 0,78$, y se hace la comparación de las mismas con las curvas correspondientes a la selección aleatoria del conjunto de entrenamiento y a la utilización de un único criterio de forma aislada, el QBT.

En las figuras se comprueba que la utilización combinada de los criterios de incertidumbre permite alcanzar, en las etapas más tempranas del aprendizaje, una mejoría en la evolución de la tasa de acierto sin una degradación considerable de la evolución de la confianza respecto a la selección aleatoria del conjunto de entrenamiento o el efecto contrario en dependencia del peso atribuido a cada criterio al efectuar la suma.

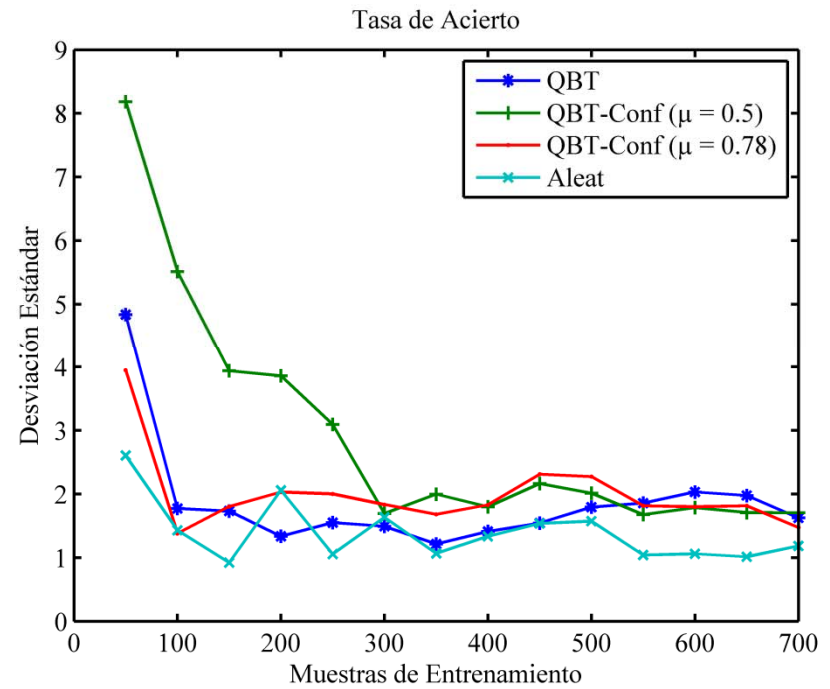
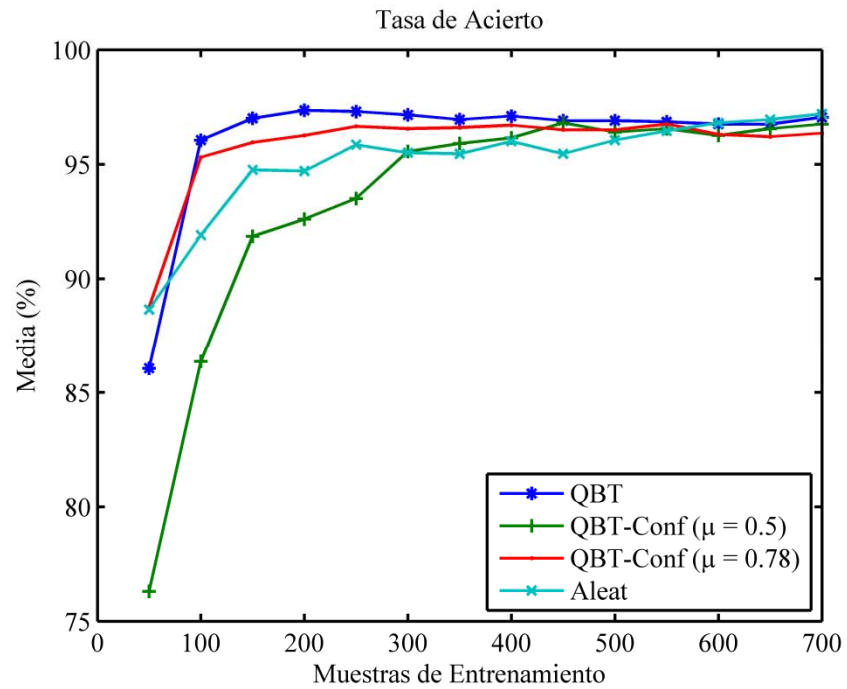


Figura 31: Tasa de acierto media y desviación estándar de la tasa de acierto en función del número de muestras de entrenamiento (después de 10 ejecuciones)

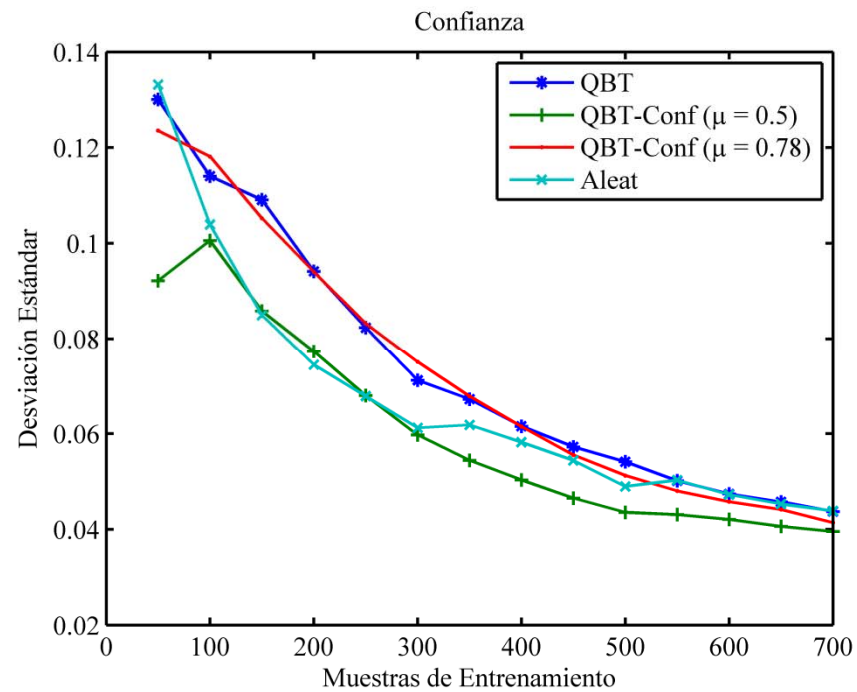
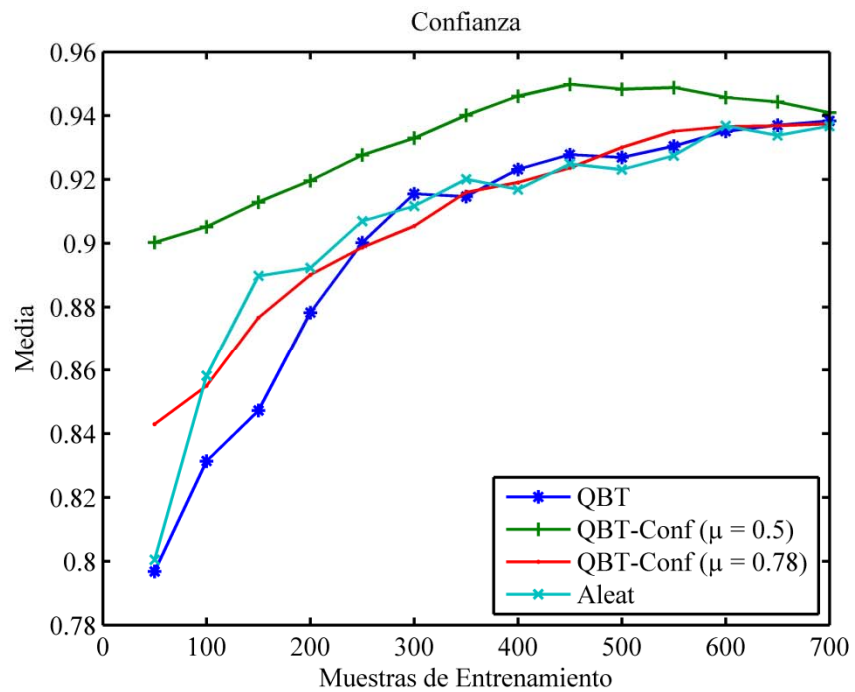


Figura 32: Confianza media y desviación estándar de la confianza en función del número de muestras de entrenamiento (después de 10 ejecuciones)

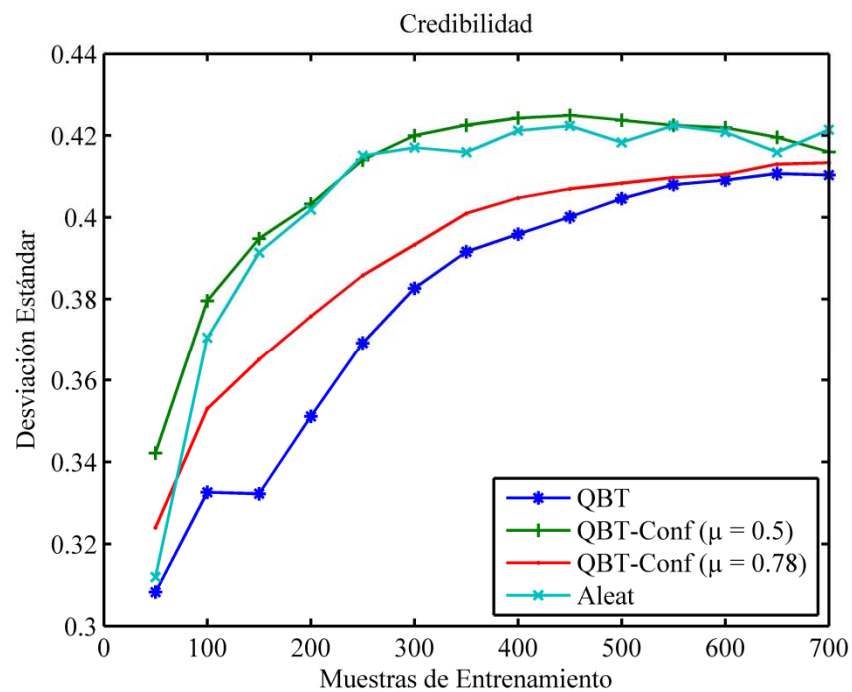
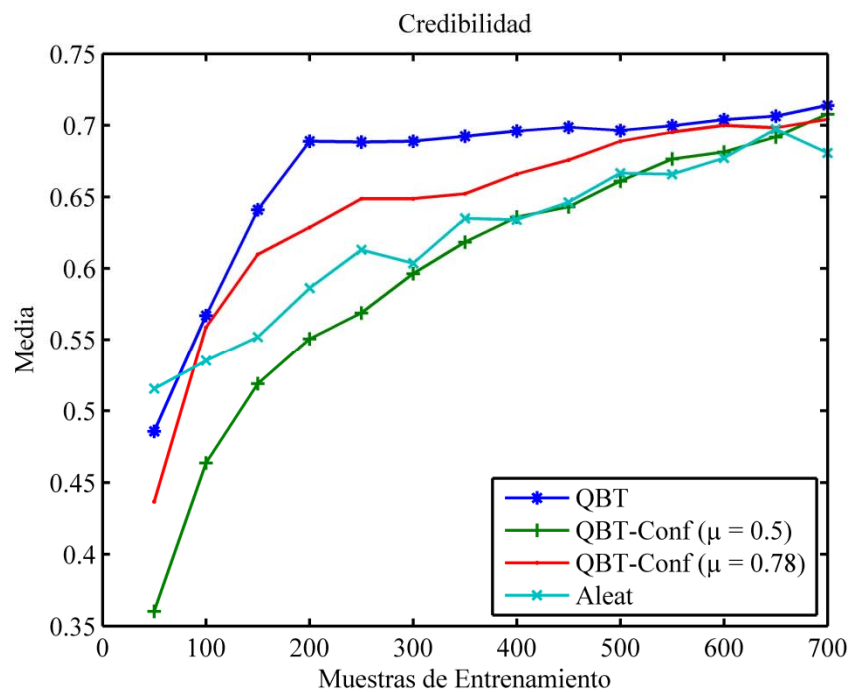


Figura 33: Credibilidad media y desviación estándar de la credibilidad en función del número de muestras de entrenamiento (después de 10 ejecuciones)

También se verificó el efecto que tendría la utilización de los criterios de diversidad y representatividad en el proceso de aprendizaje. Con este objetivo, los referidos criterios fueron utilizados de manera combinada con el criterio de información QBT, siendo en este caso los criterios combinados por marcador (Hsu, 2006) a través de una suma semejante a la empleada en el caso anterior, correspondiendo igualmente $I_1(\mathbf{x})$ a QBT e $I_2(\mathbf{x})$ al segundo criterio, diversidad o representatividad según el caso. En el cálculo de los índices de diversidad y representatividad de las muestras fueron utilizados como factores de peso $\sigma = 500$ y $\sigma = 2000$, respectivamente, valores estos determinados empíricamente.

En las figuras 34, 35 y 36 se representan las curvas de aprendizaje correspondientes a la combinación *QBT + diversidad* para diferentes valores del factor de peso μ , siendo representadas igualmente curvas similares para la combinación *QBT + representatividad* en las figuras 37, 38 y 39.

Respecto a la inclusión del criterio de diversidad, en las curvas correspondientes se observa que su utilización permite igualmente la introducción de un efecto de balance entre la mejora de la evolución de la tasa de acierto y de la confianza. En las curvas se muestra que en todos los casos se alcanzan picos de la tasa de acierto entre las 150 y las 200 muestras, siendo los referidos picos superiores a los alcanzados al seleccionar aleatoriamente el conjunto de entrenamiento para la misma cantidad de muestras, aunque algo inferiores a los alcanzados al ser utilizado el criterio QBT de forma aislada. En contrapartida, se observan valores de confianza más elevados respecto a los correspondientes al criterio QBT, siendo en los casos de algunos valores de μ semejantes o superiores a los correspondientes a la selección aleatoria del conjunto de entrenamiento.

En cuanto a la inclusión del criterio de representatividad no se aprecia ningún efecto destacable de mejora tanto en la evolución de la tasa de acierto como de la confianza, siendo en

ambos casos la evolución de las curvas de aprendizaje inferior a la obtenida utilizando el criterio QBT de forma aislada.

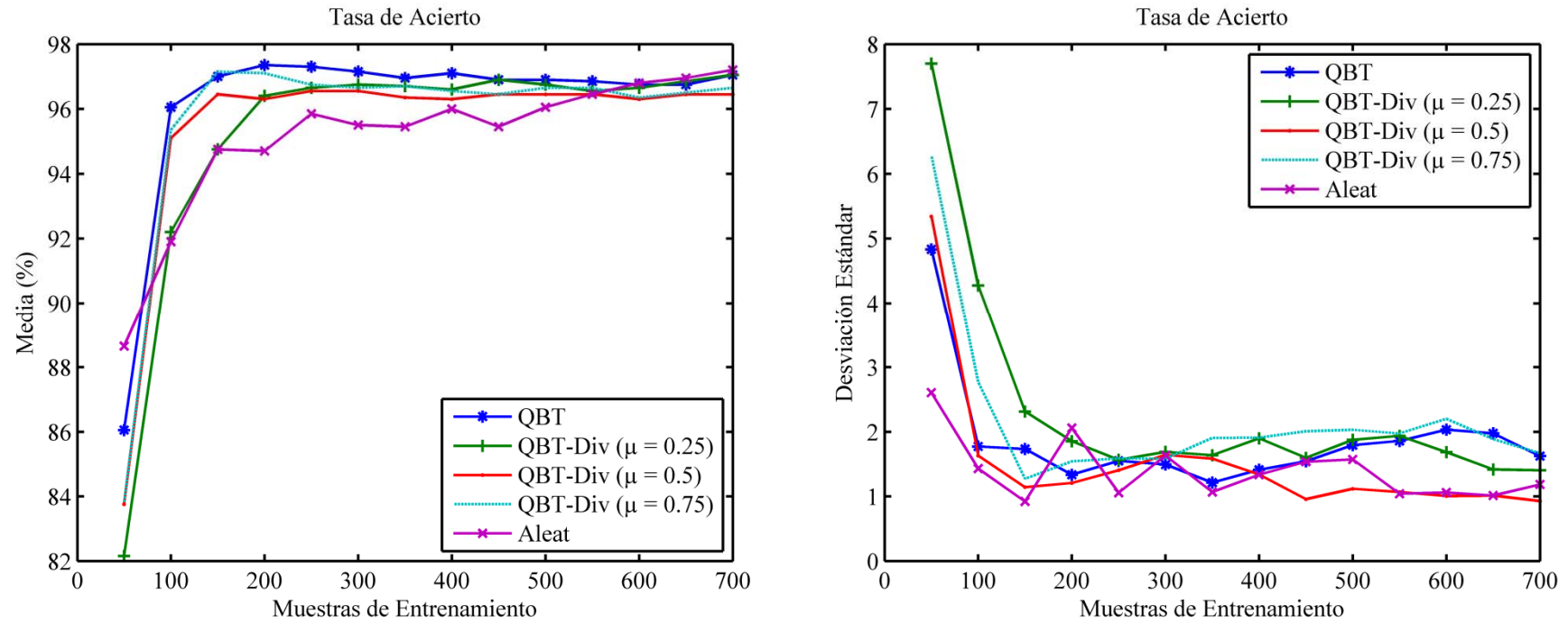


Figura 34: Tasa de acierto media y desviación estándar de la tasa de acierto en función del número de muestras de entrenamiento (después de 10 ejecuciones)

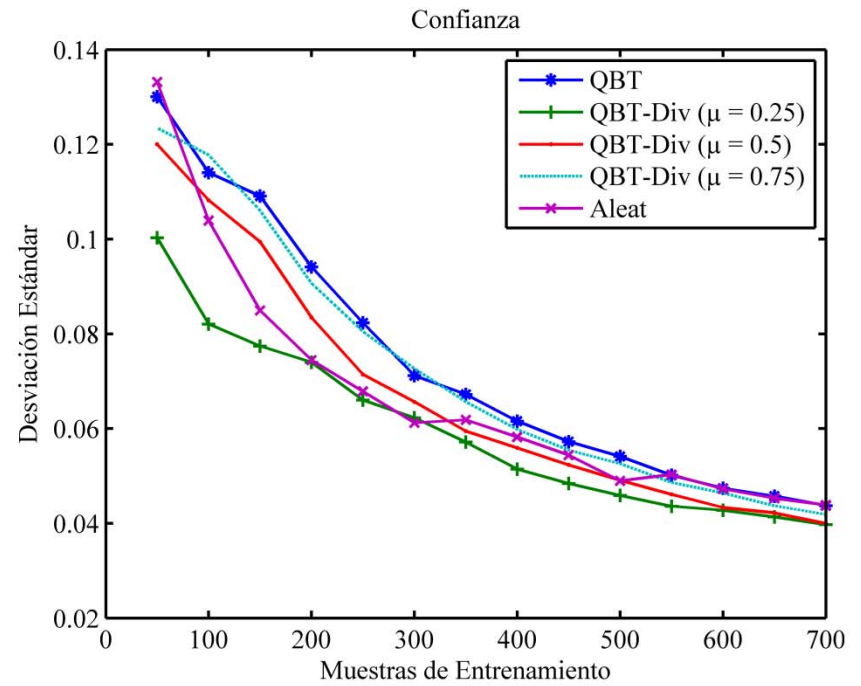
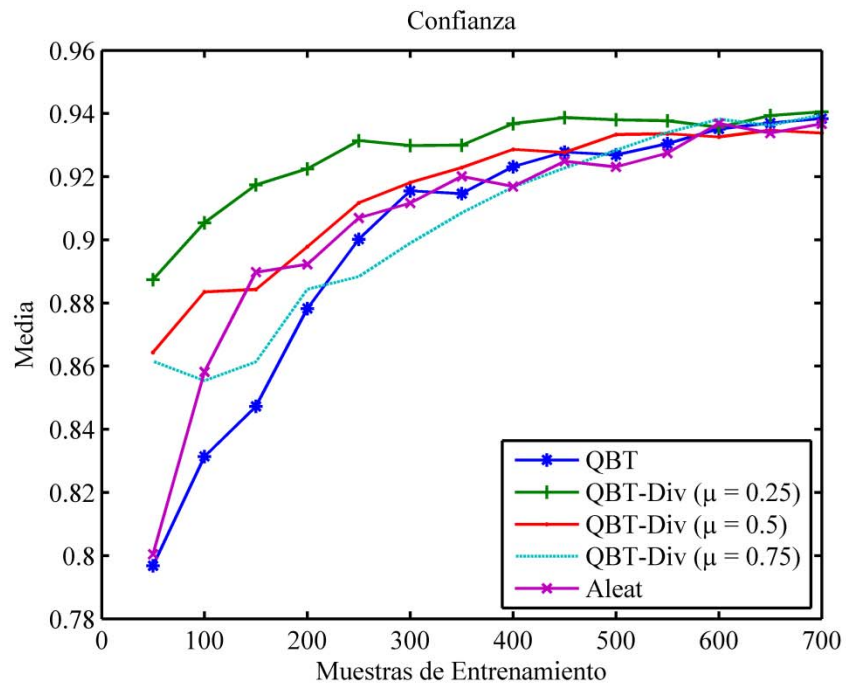


Figura 35: Confianza media y desviación estándar de la confianza en función del número de muestras de entrenamiento (después de 10 ejecuciones)

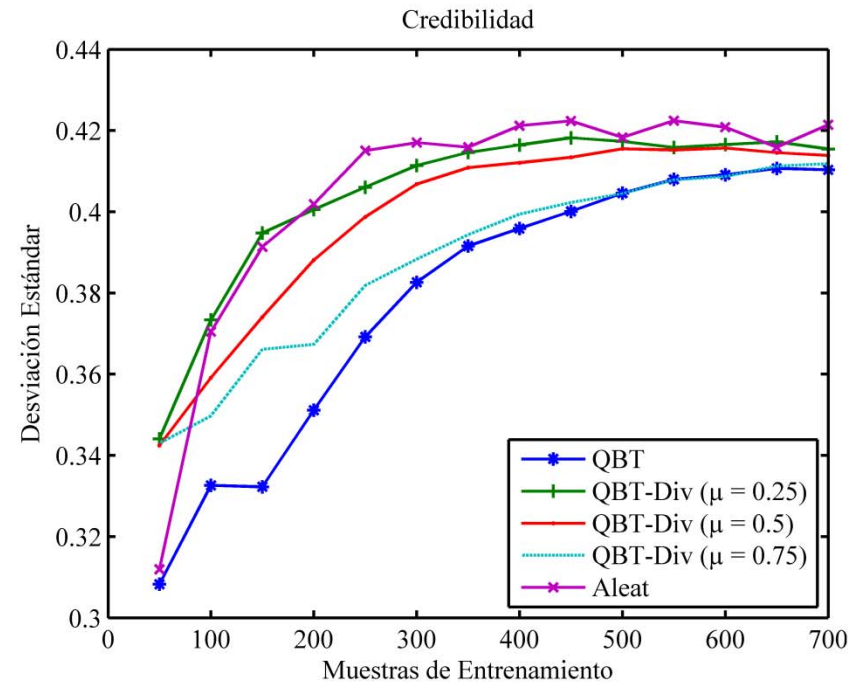
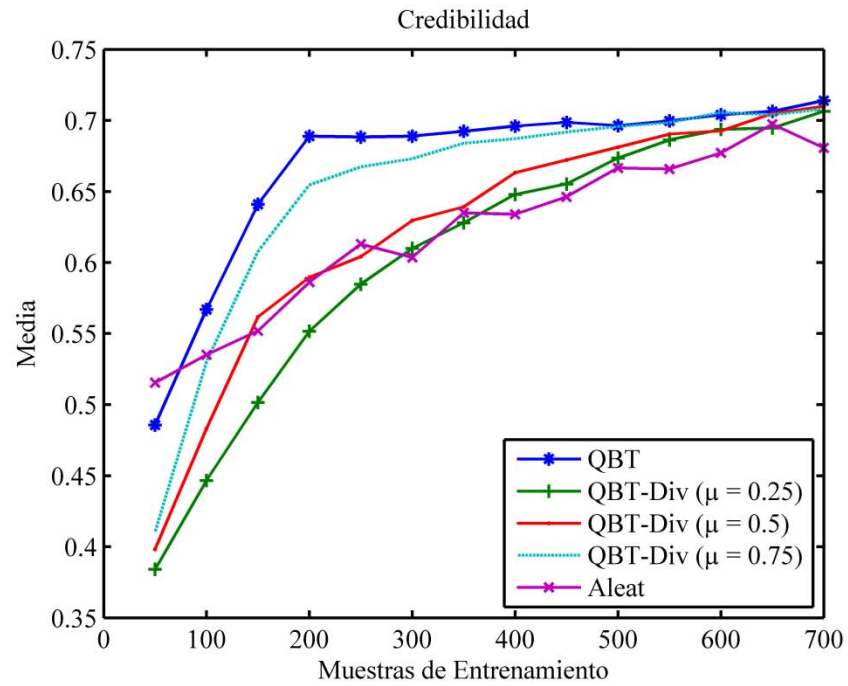


Figura 36: Credibilidad media y desviación estándar de la credibilidad en función del número de muestras de entrenamiento (después de 10 ejecuciones)

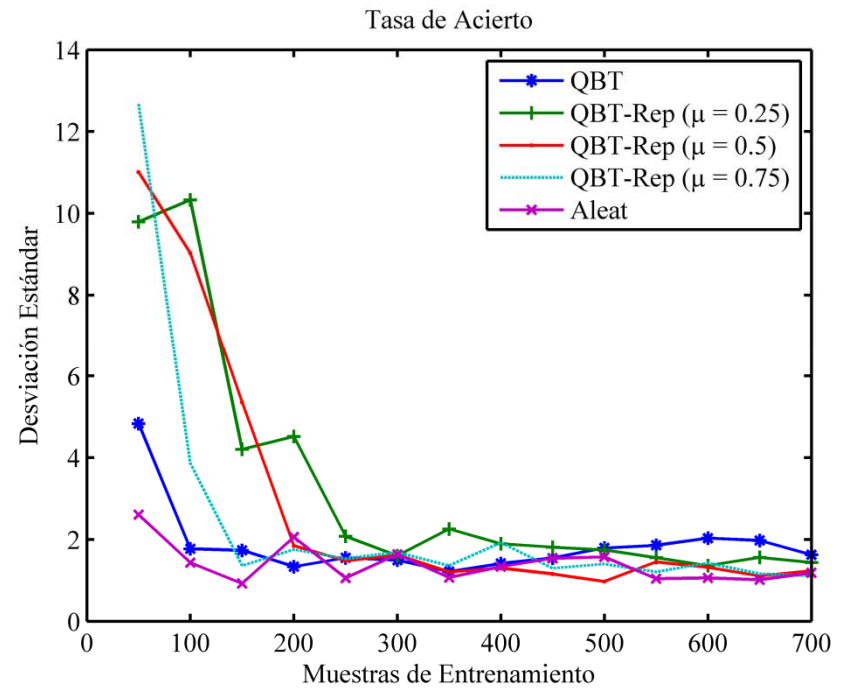
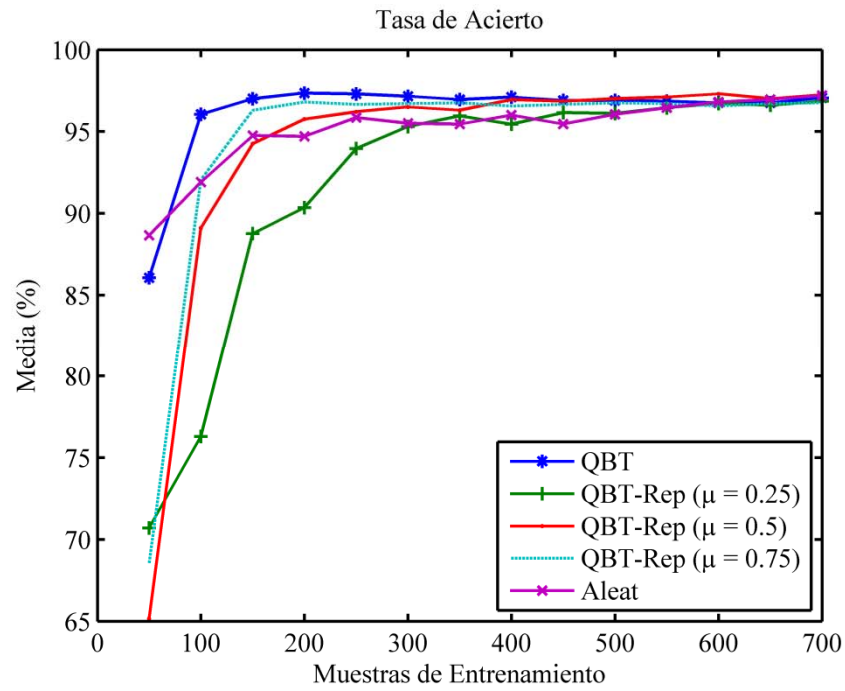


Figura 37: Tasa de acierto media y desviación estándar de la tasa de acierto en función del número de muestras de entrenamiento (después de 10 ejecuciones)

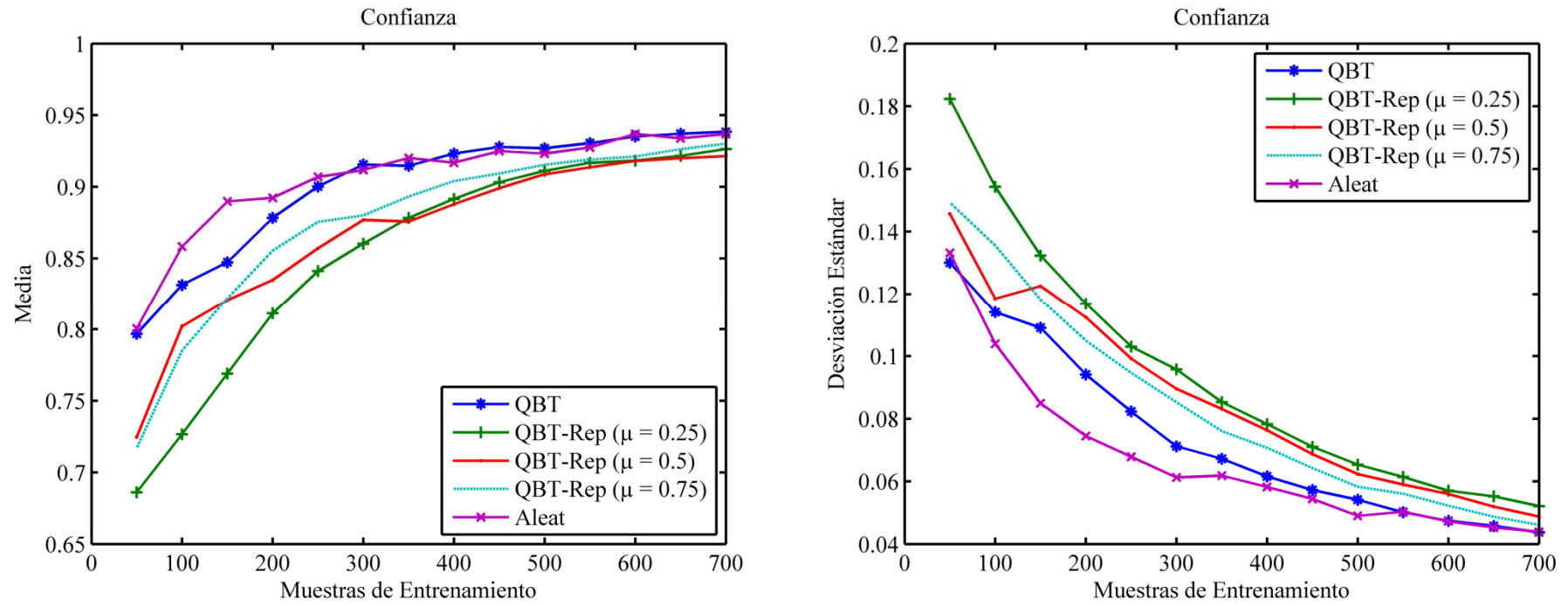


Figura 38: Confianza media y desviación estándar de la confianza en función del número de muestras de entrenamiento (después de 10 ejecuciones)

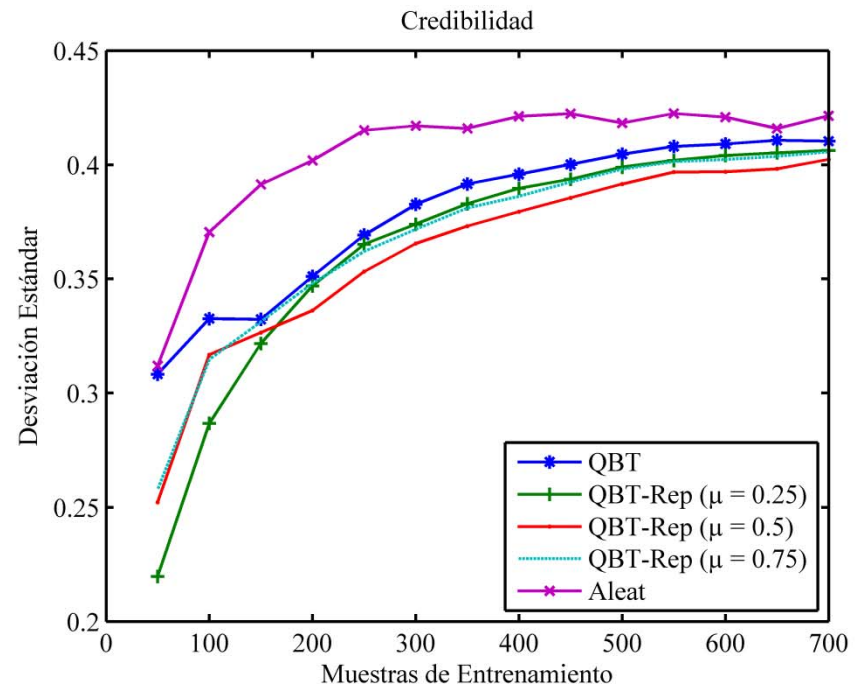
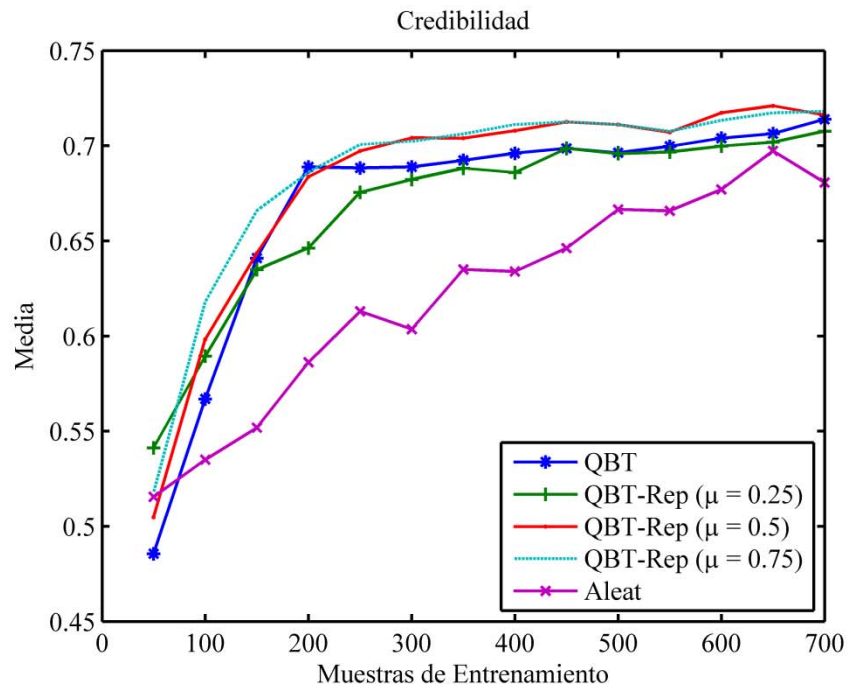


Figura 39: Credibilidad media y desviación estándar de la credibilidad en función del número de muestras de entrenamiento (después de 10 ejecuciones)

5.2.4. Conclusiones

En esta sección se hizo la descripción de una estrategia de aprendizaje activo con la cual se pretende mejorar el aporte informativo de las muestras seleccionadas para su inclusión en el conjunto de entrenamiento de un clasificador conformal. De esta manera, se reduce la dimensión del referido conjunto, pero se mantienen sus prestaciones, lo que proporciona una reducción de la carga computacional del proceso de clasificación conformal.

A través de los *experimentos* presentados, realizados sobre un conjunto de imágenes del TSD del TJ – II, se observó que la mencionada estrategia constituye un instrumento válido en la materialización del objetivo buscado. Se pudo observar que la utilización de estrategias de aprendizaje activo permite que el clasificador alcance con menos de 200 imágenes de entrenamiento similares o superiores tasas de acierto a las alcanzadas con aproximadamente 700 imágenes de entrenamiento en el caso de la selección aleatoria de dicho conjunto.

Por otro lado, se observó un ligero problema de equilibrio entre el efecto de mejoría de la evolución de la tasa de acierto, por un lado, y de la confianza del clasificador, por otro, al utilizar diferentes criterios de *consulta* como base de la selección del conjunto de entrenamiento. A través de la combinación de múltiples criterios, sea combinando diferentes criterios de información o por la combinación entre un criterio de información y el de diversidad, es posible aminorar el referido problema, lográndose una mejoría en la evolución de la tasa de acierto sin una degradación visible de la evolución de la confianza y viceversa.

CAPÍTULO 6: CONCLUSIONES Y TRABAJO FUTURO

En este capítulo se presentan las principales conclusiones extraídas del trabajo desarrollado así como algunas líneas de trabajo futuras derivadas del trabajo presentado en la Tesis.

6.1. CONCLUSIONES

La asociación de medidas de fiabilidad a las predicciones proporcionadas por los algoritmos de aprendizaje automático constituye una temática relativamente activa, observándose en los últimos años una gran cantidad de publicaciones relacionadas. Esto surge a consecuencia de los niveles de éxito alcanzados por las aplicaciones de aprendizaje automático, reconocimiento de patrones o minería de datos, derivándose del mismo en forma implícita como siguiente paso natural la determinación del nivel de fiabilidad de las predicciones efectuadas.

Se puede encontrar en la literatura un gran número de métodos prácticos formulados persiguiendo este objetivo así como algunas teorías importantes que permiten encuadrar varios de los referidos métodos en un marco bien determinado.

En esta Tesis, esta problemática fue abordada desde el punto de vista de su aplicabilidad a los datos generados en los estudios realizados en el área de la fusión termonuclear, siendo adoptado como marco teórico la novedosa teoría de la predicción conformal introducida por

Vovk, Gammerman y Schaffer (2005). La Tesis estuvo circunscrita a la solución de problemas de clasificación formulados en la referida área, tomándose como caso de estudio las imágenes del *Diagnóstico de Esparcimiento Thomson* (TSD, abreviatura del inglés, *Thomson Scattering Diagnostic*) del TJ – II. En el Anexo A se puede ver una descripción de los referidos datos.

La adopción de un escenario de aprendizaje *offline*, en la Tesis, permite la complementación de los resultados de la clasificación con medidas de fiabilidad siguiendo los protocolos habitualmente utilizados para la clasificación de señales en el área. Esto permite, por un lado, una fácil comparación, en cuanto a los rendimientos alcanzados, con los clasificadores tradicionalmente utilizados empleándose las medidas habituales y, por otro lado, cierta continuidad en la actualización de los algoritmos de clasificación empleados, es decir, la introducción de los nuevos clasificadores no implica una alteración radical en los casos de los sistemas de clasificación hasta el momento utilizados.

Considerando la reconocida carga computacional asociada al método de la predicción conformal (al tratarse de un método *transductivo*) y la frecuente necesidad de realización de tareas de clasificación en tiempo real en los dispositivos de fusión termonuclear, en la Tesis se puso particular énfasis en la necesidad de reducir la carga computacional del proceso de clasificación. Esto se trató de lograr siguiendo dos aproximaciones distintas, la utilización de variantes del método de clasificación, por un lado, y la reducción del tamaño de los conjuntos de entrenamiento utilizados con recurso a estrategias de aprendizaje activo, por otro.

El trabajo desarrollado permitió compactar en una librería una serie de algoritmos de clasificación basados en la predicción conformal, permitiendo estos la fácil implementación de clasificadores para las señales generadas en los estudios realizados en el campo de la fusión termonuclear, como es el caso de la aplicación desarrollada para la clasificación de imágenes del TSD.

Los niveles de precisión alcanzados en los *experimentos* de clasificación desarrollados son similares a los obtenidos al utilizar los métodos que constituyen el estado del arte en materia de clasificación, mientras que fue posible observar que las medidas de fiabilidad asociadas a las predicciones en los referidos *experimentos* pueden constituir una ayuda importante en el proceso de toma de decisión a partir de las referidas predicciones.

Por otro lado, se observó que los algoritmos implementados poseen un alto potencial de utilización en tiempo real. Con una selección adecuada a cada problema, es posible gestionar el equilibrio entre las ganancias en términos de tiempo de procesamiento y las pérdidas en cuanto a la calidad de las medidas de fiabilidad, relativamente a la utilización del procedimiento estándar de clasificación conformal.

Una situación muy común en los entornos dedicados al estudio de la fusión termonuclear es la existencia en las bases de datos con una gran cantidad de señales, debidamente etiquetadas, recolectadas para el estudio de diferentes procesos. Esta situación convierte la selección de las señales más adecuadas en un desafío importante en el proceso de desarrollo de clasificadores para las mismas.

La estrategia de aprendizaje activo implementada permite dar respuesta a esta cuestión en buena medida, posibilitando con una cantidad reducida de muestras de entrenamiento construir clasificadores cuyos rendimientos alcanzan o superan los correspondientes a una cantidad mucho mayor de muestras de entrenamiento, cuando se seleccionan de manera aleatoria. En los *experimentos* desarrollados se observó que la referida estrategia permite no solamente alcanzar el objetivo habitualmente formulado en los problemas de aprendizaje activo descritos en la literatura, es decir, el de la mejoría de la evolución de la precisión del clasificador con el aumento del tamaño del conjunto de entrenamiento, sino también alcanzar otro objetivo importante, el de la mejoría de la evolución de la confianza asociada a las predicciones con el crecimiento del conjunto de entrenamiento. El balance entre los dos objetivos mencionados se

logró en el trabajo mediante la combinación de múltiples criterios al definir la estrategia de *consulta* empleada.

De este modo, fue posible observar que con la utilización de una estrategia de aprendizaje activo se puede alcanzar un doble objetivo, por un lado, la mejoría de la calidad de la clasificación con una cantidad reducida de datos y, por otro, la reducción de la carga computacional al utilizar una cantidad de datos significativamente menor en el proceso de clasificación.

6.2. TRABAJO FUTURO

La predicción conformal posee un enorme potencial de aplicación en diversos tipos de problemas relacionados al análisis de datos generados en los dispositivos de fusión termonuclear. De los trabajos derivados de la elaboración de esta Tesis, varias líneas de investigación emergen, pasando a constituir potenciales direcciones para trabajos futuros. A continuación se hace una breve presentación de las más relevantes.

Aplicación a la detección de la transición LH. La transición entre los regímenes de bajo y alto confinamiento del plasma (designada “*transición LH*”) es un fenómeno de gran interés en el estudio de la fusión termonuclear y su detección ha sido automatizada con éxito empleándose algoritmos de clasificación. En el diseño y entrenamiento de los referidos clasificadores se emplean conjuntos de datos con tamaños considerables, por ejemplo en el caso del *JET* existe un conjunto de datos con más de 100 mil vectores de entrenamiento. Por consiguiente, resulta interesante la aplicación del entrenamiento selectivo mediante técnicas de aprendizaje activo, encontrándose ya abierta una línea de investigación con esa finalidad. La cantidad de datos envuelta en este caso no permite la ejecución de los algoritmos en un ordenador personal, como se hizo en los casos descritos en esta Tesis, por lo que se encuentra en fase de codificación una versión del algoritmo presentado en la Tesis

para ejecución en el *cluster* de supercomputación del CIEMAT, explotando recursos de computación paralela.

Aplicación a la predicción de interrupciones. Las interrupciones son fenómenos indeseables que ocurren en los dispositivos del tipo *tokamak*. La predicción de su ocurrencia, con el objetivo de mitigar sus efectos, constituye una línea de trabajo de elevado interés. La asociación de medidas de fiabilidad a la predicción de interrupciones, así como el entrenamiento activo de los clasificadores utilizados con esa finalidad, constituyen motivos de interés para trabajos futuros, estando abierta una línea de investigación en este sentido.

Aplicación a problemas de regresión. En esta Tesis, las aplicaciones presentadas se circunscriben exclusivamente al marco de la clasificación de señales. Sin embargo, en el análisis de los datos de fusión termonuclear se presentan diversos problemas de regresión, constituyendo, por tanto, una línea de aplicación de la teoría de predicción conformal con elevado potencial para trabajos futuros.

Fusión de información. En los dispositivos de fusión termonuclear existe una gran cantidad de *diagnósticos* de diferentes tipos, siendo las señales adquiridas por los mismos utilizadas en el análisis de distintos fenómenos de interés. Una línea importante que se puede adoptar en el futuro es la utilización de la predicción conformal para la fusión de la información adquirida a través de diferentes *diagnósticos*, implementándose lo que se podría considerar una estrategia de predicción multidiagnóstico.

Empleo del escenario de clasificación *online*. En la gran mayoría de los sistemas de clasificación implementados en los dispositivos de fusión se recurre al escenario de aprendizaje *offline*, o aprendizaje en lotes o *batch*, paradigma que es mantenido en el caso de los clasificadores presentados en esta Tesis. Sin embargo, en los ambientes de fusión se pueden presentar frecuentemente circunstancias en las cuales es de una gran importancia la utili-

zación de un escenario de aprendizaje *online*, como por ejemplo al realizarse actualizaciones del *hardware* de los dispositivos de fusión o de los *diagnósticos* o al entrar en funcionamiento nuevos dispositivos de fusión. Consecuentemente, resulta interesante, como línea para trabajos futuros, el estudio de la aplicabilidad de la clasificación empleando este escenario de aprendizaje, en el cual las señales son etiquetadas a medida que son adquiridas y clasificadas. La combinación del proceso de clasificación en el referido escenario de aprendizaje con una estrategia de aprendizaje activo constituye una alternativa de estudio adicional teniendo en cuenta los objetivos de optimización de la eficiencia temporal de la clasificación y minimización del esfuerzo de etiquetado.

Sistema de etiquetado. En las bases de datos de los dispositivos de fusión es común la existencia de grandes cantidades de señales etiquetadas de diversos tipos, razón que motivó la utilización en la Tesis de una estrategia de aprendizaje activo para la selección de las señales más convenientes para el entrenamiento de un clasificador determinado. Sin embargo, el etiquetado de las referidas señales posee un elevado coste en términos de tiempo, de recursos y económicos, ya que la misma es una actividad reconocidamente lenta, tediosa y que requiere la atención de personal altamente especializado. En este sentido, la estrategia de aprendizaje activo empleada en la Tesis puede ser utilizada desde otra perspectiva, comúnmente destacada en la literatura, que es el aligeramiento de la tarea de etiquetado de los datos a la hora de implementarse clasificadores para las diferentes señales, mencionada en el punto anterior, sirviendo, por tanto, como base para la implementación de un sistema amplio de etiquetado de señales. En este caso, varias cuestiones nuevas pueden ser abordadas, como por ejemplo la combinación de las estrategias de *consulta* ya empleadas con restricciones de carácter económico, considerando los costes de obtención de las etiquetas, por ejemplo en términos del tiempo requerido para su etiquetado o del nivel de especiali-

zación del experto a realizar el etiquetado, con el objetivo de reducir el coste global de entrenamiento de un clasificador.

Condiciones de parada. Al emplear estrategias de aprendizaje activo, un problema importante, y en abierto, es la definición de las condiciones de parada más adecuadas. En los *experimentos* discutidos en la Tesis se hizo uso de la más trivial de las condiciones, considerándose el hecho de la existencia de una cantidad suficiente de señales etiquetadas. En general, la selección de la condición de parada no es tan simple como fue en este caso, siendo el estudio de la correlación entre la efectividad de la misma y los tipos de señales a seleccionar en cada caso una futura línea de trabajo interesante.

CAPÍTULO 7: BIBLIOGRAFÍA

- [Abe, 1998] Naoki Abe y Hiroshi Mamitsuka, "Query Learning Strategies Using Boosting and Bagging," en *Proceedings of the Fifteenth International Conference on Machine Learning*, San Francisco, CA, US, 1998, pp. 1 - 9.
- [Alejaldre, 1990] Carlos Alejaldre y otros., "TJ - II Project: A Flexible Helic Stellarator," *Fusion Technology*, vol. 17, pp. 131 - 139, 1990.
- [Alejaldre, 1999] Carlos Alejaldre y otros., "First Plasmas in the TJ II Flexible Helic," *Plasma Physics and Controlled Fusion*, vol. 41, pp. A539 - A548, 1999.
- [Altman, 2005] Douglas G. Altman y J. Martin Bland, "Standard Deviations and Standard Errors," *British Medical Journal*, vol. 331, p. 903, Oct. 2005.
- [Angluin, 1987] Dana Angluin, "Learning Regular Sets from Queries and Counterexamples," *Information and Computation*, vol. 75, pp. 87 - 106, 1987.

- [Angluin, 1988]** Dana Angluin, "Queries and Concept Learning," *Machine Learning*, vol. 2, no. 4, pp. 319 - 342, 1988.
- [Arlot, 2010]** Sylvain Arlot y Alain Celisse, "A survey of Cross-validation Procedures for Model Selection," *Statistical Surveys*, vol. 4, pp. 40 - 79, 2010.
- [Arnborg, 2011]** Stefan Arnborg, "Statistical Methods in Applied Computer Science," KTH Royal Institute of Technology, Stockholm, Sweden, Lecture Notes 2011.
- [ANU, 2013]** Australian National University. (2013, Aug.) Australian Plasma Fusion Research Facility. [Online]. <http://h1nf.anu.edu.au/>
- [Bache, 2013]** K. Bache y M. Lichman. (2013) UCI Machine Learning Repository. [Online]. <http://archive.ics.uci.edu/ml/>
- [Balasubramanian, 2009]** Vineeth Nallure Balasubramanian, Shayok Chakraborty y Sethuraman Panchanathan, "Generalized Query by Transduction for Online Active Learning," en *IEEE 12th International Conference on Computer Vision Workshops*, Kyoto, Japan, 2009, pp. 1378 - 1385.
- [Balasubramanian, 2010]** Vineeth Nallure Balasubramanian, "Conformal Predictions in Multimedia Pattern Recognition," Arizona State University, Phoenix, AZ, USA, Doctoral Thesis 2010.
- [Balasubramanian, 2011]** Vineeth Nallure Balasubramanian, Shen-Shyang Ho, Vladimir Vovk y Sethuraman Panchanathan. (2011) Conformal Predictions for Reliable Machine Learning: Theory and Applications (Tutorial Slides). [Online].

<http://www.public.asu.edu/~vnallure/conformalpredictions/CP%20Tutorial%20Slides-AAAI2011.pdf>

- [Barth, 1999] C. J. Barth, F. J. Pijper, H. J. Meiden, J. Herranz y I. Pastor, "High-Resolution Multiposition Thomson Scattering for the TJ II Stellarator," *Review of Scientific Instruments*, vol. 70, no. 1, pp. 763 - 767, 1999.
- [Bekkeheien, 1999] Martin Bekkeheien, Øystein Håland y Reidulf Klovening, "Energy Demand Patterns Towards 2050," en *Energy: The Next Fifty Years*. Paris: OECD Publications Service, 1999, pp. 95 - 120.
- [Bellotti, 2005] Tony Bellotti, Zhiyuan Luo, Alex Gammerman, Frederick Van Delft y Vaskar Saha, "Qualified Predictions for Microarray and Proteomics Pattern Diagnostics with Confidence Machines," *International Journal of Neural Systems*, vol. 15, no. 4, pp. 247 - 258, 2005.
- [Bishop, 1995] Christopher M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, UK: Oxford University Press, 1995.
- [Blatz, 2004] John Blatz y otros., "Confidence Estimation for Machine Translation," The Center for Language and Speech Processing, John Hopkins University, Baltimore, MD, USA, Final Report Summer Workshop 2004.
- [Bloodgood, 2009] Michael Bloodgood y K. Vijay-Shanker, "A Method for Stopping Active Learning Based on Stabilizing Predictions and the Need for User-Adjustable Stopping," en *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, Stroudsburg, PA, USA, 2009, pp. 39 - 47.

- [Boozer, 1983]** A. H. Boozer y otros., "Two High-Beta Toroidal Configurations: A stellarator and a Tokamak - Torsatron Hybrid," *Plasma Physics and Controlled Nuclear Fusion Research 1982 (Proc. 9th Int. Conf. Baltimore, 1982)*, vol. 3, p. 129, 1983.
- [Bosch, 1992]** H. S. Bosch y G. M. Hale, "Improved Formulas for Fusion Cross-Sections and Thermal Reactivities," *Nuclear Fusion*, vol. 32, no. 4, pp. 611 - 631, 1992.
- [Bosnic', 2008]** Zoran Bosnic' y Igor Kononenko, "An Overview of Advances in Reliability Estimation of Individual Predictions in Machine Learning," Faculty of Computer and Information Science, University of Ljubljana, Technical Report T-03/08 2008.
- [Botija, 1991]** J. Botija y M. Blaumoser, "Vacuum Vessel Design for the TJ-II Device," en *Proceedings of the 14th IEEE/NPSS Symposium on Fusion Engineering*, San Diego, CA, USA, 1991, pp. 992 - 995.
- [Botija, 1993]** J. Botija y otros., "Vacuum Vessel, Wall Protection, Pumping System and Poloidal Limiters of the Spanish Stellarator TJ-II," en *Proceedings of the 15th IEEE/NPSS Symposium on Fusion Engineering*, Hyannis, MA, USA, 1993, pp. 529 - 532.
- [Boyd, 2003]** T. J. M. Boyd y Jeff J. Sanderson, *The Physics of Plasmas*. Cambridge, UK: Cambridge University Press, 2003.
- [Braams, 2002]** C. M. Braams y P. E. Stott, *NUCLEAR FUSION Half a Century of Magnetic Confinement Fusion Research*. Bristol, UK: IOP Publishing Ltd, 2002.

- [Bradley, 1997]** Andrew P. Bradley, "The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms," *Pattern Recognition*, vol. 30, no. 7, pp. 1145 - 1159, 1997.
- [Breiman, 1994]** Leo Breiman, "Bagging Predictors," Department of Statistics, University of California Berkeley, Berkeley, CA, USA, Technical Report N° 421 1994.
- [Brinker, 2003]** Klaus Brinker, "Incorporating Diversity in Active Learning with Support Vector Machines," en *Proceedings of the 20th International Conference on Machine Learning*, Washington, DC, USA, 2003, pp. 59 - 66.
- [Cannas, 2004]** B. Cannas, A. Fanni, E. Marongiu y P. Sonato, "Disruption Forecasting at JET Using Neural Networks," *Nuclear Fusion*, vol. 44, pp. 68 - 76, 2004.
- [Caruana, 2004]** Rich Caruana y Alexandru Niculescu-Mizil, "Data Mining in Metric Space: An Empirical Analysis of Supervised Learning Performance Criteria," en *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Seattle, WA, USA, 2004, pp. 69 - 78.
- [Cawenberghs, 2001]** Gert Cawenberghs y Tomaso Poggio, "Incremental and Decremental Support Vector Machine Learning," *Advances in Neural Information Processing Systems*, vol. 13, pp. 409 - 415, 2001.
- [Chang, 2011]** Chih-Chung Chang y Chih-Jen Lin, "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and*

Technology, vol. 2, no. 3, pp. 27:1 - 27:27, 2011, Software disponible en: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

- [Chelouah, 2000] Rachid Chelouah y Patrick Siarry, "Tabu Search Applied to Global Optimization," *European Journal of Operational Research*, vol. 123, pp. 256 - 270, 2000.
- [Cherkassky, 2007] Vladimir Cherkassky y Filip Mulier, *Learning From Data. Concepts, Theory, and Methods*, 2nd ed. New Jersey, USA: John Wiley & Sons, Inc., 2007.
- [Cohn, 1994] David Cohn, Les Atlas y Richard Ladner, "Improving Generalization with Active Learning," *Machine learning - Special Issue on Structured Connectionist Systems*, vol. 15, no. 2, pp. 201 - 221, 1994.
- [Collobert, 2001] Ronan Collobert y Samy Bengio, "SVM Torch: Support Vector Machines for Large-Scale Regression Problems," *Journal of Machine Learning Research*, vol. 1, pp. 143 - 160, 2001.
- [Cutland, 1980] Nigel Cutland, *Computability, An Introduction to Recursive Function Theory*. Cambridge, UK: Cambridge University Press, 1980.
- [Dagan, 1995] Ido Dagan y Sean P. Engelson, "Committee-Based Sampling for Training Probabilistic Classifiers," en *Proceedings of the Twelfth International Conference on Machine Learning*, San Francisco, CA, USA, 1995, pp. 150 - 157.
- [Diehl, 2003] Christopher P. Diehl y Gert Cawenberghs, "SVM Incremental Learning, Adaptation and Optimization," en *Proceedings of the*

International Joint Conference on Neural Networks, vol. 4, Portland, OR, USA, 2003, pp. 2685 - 2690, Código disponible en: <http://www.cpdiehl.org/IncrementalSVM.zip>.

- [Dietterich, 1997]** Thomas G. Dietterich, "Fundamental Experimental Research in Machine Learning," en *Probably Approximately Correct Learning. In the Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*.: AAAI/MIT Press, 1997, pp. 1101 - 1108.
- [Dormido-Canto, 2004]** Sebastián Dormido-Canto y otros., "TJ-II Wave Forms Analysis with Wavelets and Support Vector Machines," *Review of Scientific Instruments*, vol. 75, no. 10, pp. 4254 - 4257, 2004.
- [Dormido-Canto, 2005]** Sebastián Dormido-Canto, Jesús Vega, Joaquín Sánchez y Gonzalo Farias, "Information Retrieval and Classification with Wavelets and Support Vector Machines," en *Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation; Conference on Artificial Intelligence and Knowledge Engineering Applications: a Bioinspired Approach*, vol. Part II, Berlin Heidelberg, 2005, pp. 548 - 557.
- [Dubois, 1998]** Didier Dubois y Henri Prade, "Possibility Theory: Qualitative and Quantitative Aspects," en *Handbook of Defeasible Reasoning and Uncertainty Management Systems*, D. M. Gabbay y P. Smets, Eds. Dordrecht, Netherlands: Kluwer Academic Publishers, 1998, vol. 1, pp. 169 - 226.
- [Duda, 2001]** Richard O. Duda, Peter E. Hart y David G. Stork, *Pattern*

Classification, 2nd ed. New York, USA: John Wiley & Sons, Inc., 2001.

[EFDA, 2013] European Fusion Development Agreement. (2013) JET. [Online]. <http://www.efda.org/jet/>

[Everitt, 2002] B. S. Everitt, *The Cambridge Dictionary of Statistics*, 2nd ed. Cambridge, UK: Cambridge University Press, 2002.

[Farias, 2004] Gonzalo Farias, Matilde Santos, J. L. Marrón y Sebastián Dormido-Canto, "Determinación de los Parámetros de la Transformada Wavelet para la Clasificación de Señales del Diagnóstico Scattering Thomson," en *XXV Jornadas de Automática*, Ciudad Real, España, 2004.

[Farias, 2005] Gonzalo Farias, R. Dormido, M. Santos y N. Duro, "Image Classifier for the TJ - II Scattering Thomson Diagnostic: Evaluation with a Feedforward Neural Network," en *Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation; Conference on Artificial Intelligence and Knowledge Engineering Applications: a Bioinspired Approach*, vol. Part II, Berlin Heidelberg, 2005, pp. 604 - 612.

[Farias, 2006] Gonzalo Farias y otros., "Searching for Patterns in TJ-II Time Evolution Signals," *Fusion Engineering and Design*, vol. 81, pp. 1993 - 1997, 2006.

[Fayyad, 1996] Usama Fayyad, David Haussler y Paul Stolorz, "KDD for Science Data Analysis: Issues and Examples," en *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*,

Portland, Oregon, USA, 1996, pp. 50 - 56.

- [Fernández, 2000a] Ángela Fernández, W. Kasperek, Konstantin M. Likin y Romualdo Martín, "Design of the Upgraded TJ-II Quasi-Optical Transmission Line," en *Proceedings of the 25th Conference on Infrared and Millimeter Waves*, Beijing, China, 2000a, pp. 91 - 92.
- [Fernández, 2000b] Ángela Fernández y otros., "Quasi-Optical Transmission Lines for ECRH at TJ - II Stellarator," *International Journal of Infrared and Millimeter Waves*, vol. 21, no. 12, pp. 1945 - 1957, 2000b.
- [Floyd, 1995] Sally Floyd y Manfred K. Warmuth, "Sample Compression, Learnability, and the Vapnik-Chervonenkis Dimension," *Machine Learning*, vol. 21, no. 3, pp. 269 - 304, 1995.
- [Freund, 1997] Yoav Freund, H. Sebastian Seung, Eli Shamir y Naftali Tishby, "Selective Sampling Using the Query by Committee Algorithm," *Machine Learning*, vol. 28, no. 2 - 3, pp. 133 - 168, 1997.
- [Froula, 2011] Dustin H. Froula, Siegfried H. Glenzer, Neville C. Luhmann Jr y John Sheffield, *Plasma Scattering of Electromagnetic Radiation: Theory and Measurement Techniques*, 2nd ed. USA: Elsevier Academic Press, 2011.
- [Fuentes, 2005] C. Fuentes y otros., "Neutral Beam Injection Optimization at TJ-II," *Fusion Engineering and Design*, vol. 74, pp. 249 - 253, 2005.
- [Fujii, 1998] Atsushi Fujii, Kentaro Inui, Takenobu Tokunaga y Hozumi Tanaka, "Selective Sampling for Example-Based Word Sense Disambiguation,"

Computational Linguistics, vol. 24, no. 4, pp. 573 - 597, 1998.

- [Fung, 2002]** Glen Fung y Olvi L. Mangasarian, "Incremental Support Vector Machine Classification," en *Proceedings of the Second SIAM International Conference on Data Mining*. Arlington, VA, USA: Society for Industrial and Applied Mathematics, 2002, pp. 247 - 260.
- [Gammerman, 2007]** Alex Gammerman, "Hedging Predictions in Machine Learning," *The Computer Journal*, vol. 50, no. 2, pp. 151 - 163, 2007.
- [Ghayoomi, 2010]** Masood Ghayoomi, "Using Variance as a Stopping Criterion for Active Learning of Frame Assignment," en *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, Stroudsburg, PA, USA, 2010, pp. 1 - 9.
- [Goodman, 1999]** Steven N. Goodman, "Toward Evidence-Based Medical Statistics. 1: The P Value Fallacy," *Annals of Internal Medicine*, vol. 130, no. 12, pp. 995 - 1004, 1999.
- [Greene, 1998]** John M. Greene, "A Brief Review of Magnetic Wells," General Atomics, San Diego, CA, USA, Technical Report GA-A22135, 1998.
- [Grubler, 1999]** Arnulf Grubler, "Global Energy Perspectives: 2050 and Beyond," en *Energy: The Next Fifty Years*. Paris: OECD Publications Service, 1999, pp. 41 - 61.
- [Güneralp, 2007]** Burak Güneralp, George Gertner, Guillermo Mendoza y Alan Anderson, "Evaluating Probabilistic Data with a Possibilistic Criterion in Land-Restoration Decision-Making: Effects on the Precision of

- Results," *Fuzzy Sets and Systems*, vol. 158, pp. 1546 - 1560, 2007.
- [Guo, 2008]** Yuhong Guo y Dale Schuurmans, "Discriminative Batch Mode Active Learning," en *Advances in Neural Information Processing Systems 20, Proceedings of the 21st Annual Conference on Neural Information Processing Systems*, 2008.
- [Hahn, 1991]** Gerald J. Hahn y William Q. Meeker, *Statistical Intervals: A Guide for Practitioners*. New York, USA: John Wiley & Sons, Inc., 1991.
- [Halck, 2002]** Ole Martin Halck, "Using Hard Classifiers to Estimate Conditional Class Probabilities," en *Proceedings of the Thirteenth European Conference Machine Learning - Lecture Notes in Artificial Intelligence*, T. Elomaa, H. Mannila y H. Toivonen, Eds. Berlin Heidelberg: Springer - Verlag, 2002, vol. 2430, pp. 123 - 134.
- [Hall, 2012]** Brian Hall, "Beej's Guide to Network Programming. Using Internet Sockets," Tutorial 2012. [Online]. <http://beej.us/guide/bgnet/>
- [Hanley, 1982]** James A. Hanley y Barbara J. McNeil, "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve," *Radiology*, vol. 143, no. 1, pp. 29 - 36, 1982.
- [Harris, 1985]** J. H. Harris, T. C. Hender, J. L. Cantrell, B. A. Carreras y R. N. Morris, "A Flexible Helic Configuration," *Nuclear Fusion*, vol. 25, no. 5, pp. 623 - 629, 1985.
- [Hassan, 2009]** Ahmed E. Hassan, Hesham M. Bekhit y Jenny B. Chapman, "Using Markov Chain Monte Carlo to Quantify Parameter Uncertainty and Its

Effect on Predictions of a Groundwater Flow Model," *Environmental Modelling & Software*, vol. 24, pp. 749 - 763, 2009.

- [Hastie, 2009]** Trevor Hastie, Robert Tibshirani y Jerome H. Friedman, *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*, 2nd ed. New York, USA: Springer-Verlag, 2009.
- [Hender, 1987]** T. C. Hender y otros., "Studies of a Flexible Helic Configuration," Oak Ridge National Laboratory, Oak Ridge, Tennessee, USA, Technical Report TM - 10374 1987.
- [Herranz, 2003]** J. Herranz, F. Castejón, I. Pastor y K. J. McCarthy, "The Spectrometer of the High-Resolution Multiposition Thomson Scattering Diagnostic for TJ II," *Fusion Engineering and Design*, vol. 65, pp. 525 - 536, 2003.
- [Heskes, 1997]** Tom Heskes, "Practical Confidence and Prediction Intervals," en *Advances in Neural Information Processing Systems 9*, M. Mozer, M. Jordan y T. Petsche, Eds. Cambridge, MA, USA: MIT Press, 1997, pp. 176 - 182.
- [Hilera, 1995]** José R. Hilera y Víctor J. Martínez, *Redes Neuronales Artificiales. Fundamentos, Modelos y Aplicaciones*. Madrid: RA-MA, 1995.
- [Hill, 2002]** Paul R. Hill, "Wavelet Based Texture Analysis and Segmentation for Image Retrieval and Fusion," Bristol, UK, Doctoral Thesis 2002.
- [Ho, 2003]** Shen-Shyang Ho y Harry Wechsler, "Transductive Confidence Machine for Active Learning," en *Proceedings of International Joint*

- Conference on neural Networks*, vol. 2, Portland, OR, USA, 2003, pp. 1435 - 1440.
- [Ho, 2004]** Shen-Shyang Ho y Harry Wechsler, "Learning From Data Streams Via Online Transduction," en *Proceedings of 4th International Conference on Data Mining Workshop On Temporal Data Mining: Algorithms, Theory and Applications*, Sheng Ma, Tao Li y Chang-shing Perng, Eds. Brighton, UK: IEEE Computer Society Press, 2004, pp. 45 - 52.
- [Ho, 2008]** Shen-Shyang Ho y Harry Wechsler, "Query by Transduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 9, pp. 1557 - 1571, September 2008.
- [Hoi, 2006a]** Steven C. Hoi, Rong Jin y Michael R. Lyu, "Large Scale Text Categorization by Batch Mode Active Learning," en *Proceedings of the 15th International Conference on World Wide Web*, New York, NY, USA, 2006a, pp. 633 - 642.
- [Hoi, 2006b]** Steven C. Hoi, Rong Jin, Jianke Zhu y Michael R. Lyu, "Batch Mode Active Learning and Its Application to Medical Image Classification," en *23rd International Conference on Machine Learning*, Pittsburgh, PA, USA, 2006b, pp. 417 - 424.
- [Hsu, 2010]** Chih-Wei Hsu, Chih-Chung Chang y Chih-Jen Lin, "A Practical Guide to Support Vector Classification," Department of Computer Science, National Taiwan University, Taipei, Taiwan, Tutorial 2010. [Online]. <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
- [Hsu, 2006]** Frank Hsu, Yun-Sheng Chung y Bruce S. Kristal, "Combinatorial

Fusion Analysis: Methods and Practices of Combining Multiple Scoring Systems," en *Advanced Data Mining Technologies in Bioinformatics*, Hui-Huang Hsu, Ed.: Idea Group, Inc, 2006, pp. 32 - 63.

- [Huang, 2006]** Jin Huang y Charles X. Ling, "Evaluating Model Selection Abilities of Performance Measures," en *Evaluation Methods for Machine Learning - Papers from the 2006 AAI Workshop - Technical Report WS-06-06*, Chris Drummond, William Elazmeh y Nathalie Japkowicz, Eds. Menlo Park, CA, USA: AAAI Press, 2006, pp. 12 - 17.
- [IFIP, 2012]** International Federation for Information Processing, *Artificial Intelligence Applications and Innovations - AIAI 2012 International Workshops: AIAB, AIEIA, CISE, COPA, IIVC, ISQL, MHDW, and WADTMB, Halkidiki, Greece, September 27-30, 2012, Proceedings, Part II*, Lazaros S. Iliades y otros., Eds. Berlin Heidelberg, Germany: Springer, 2012, vol. 382 - IFIP Advances in Information and Communication Technology. [Online]. <http://dx.doi.org/10.1007/978-3-642-33412-2>
- [Ingber, 2011]** Lester Ingber. (2011) Lester Ingber's Archive. [Online]. <http://www.ingber.com/>
- [IPP, 2013]** Max Planck Institut für PlasmaPhysic. (2013, Julio) Projects Wendelstein 7-X. [Online]. <http://www.ipp.mpg.de/ippcms/eng/for/projekte/w7x/index.html>
- [ITER Org, 2013]** ITER Organization. (2013) ITER the way to new energy. [Online].

<http://www.iter.org>

- [JAEA, s.f.] Japan Atomic Energy Agency. (s.f.) Fusion Plasma Research. [Online].
<http://www-jt60.naka.jaea.go.jp/english/index-e.html>
- [Joachims, 1999] T. Joachims, "Making Large-Scale SVM Learning Practical," en *Advances in Kernel Methods*, B. Schölkopf, C. Burges y A. Smola, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 169 - 184.
- [Kapetanios, 2006] George Kapetanios, Vincent Labhard y Simon Price. (2006) Working Paper Series, School of Economics and Finance, Queen Mary University of London. [Online]. <http://www.econ.qmul.ac.uk/papers/doc/wp567.pdf>
- [Kirkpatrick, 1983] S. Kirkpatrick, C. D. Gelatt y M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, vol. 220, no. 4598, pp. 671 - 680, 13 1983.
- [Kivinen, 2004] Jyrki Kivinen, Alexander J. Smola y Robert C. Williamson, "Online Learning with Kernels," *IEEE Transactions on Signal Processing*, vol. 52, no. 8, pp. 2165 - 2176, 2004.
- [Kohavi, 1995a] Ron Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," en *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, vol. 2, Montréal, Québec, Canada, 1995a, pp. 1137 - 1143.
- [Kohavi, 1995b] Ron Kohavi y George H. John, "Automatic Parameter Selection by Minimizing Estimated Error," en *Machine Learning: Proceedings of*

the Twelfth International Conference, Tahoe City, California, July 9-12, 1995, Armand Prieditis y Stuart Russell, Eds. San Francisco, CA, USA: Morgan Kaufmann, 1995b.

- [Kohlas, 1994]** Jurg Kohlas y Paul-André Monney, "Theory of Evidence: A Survey of its Mathematical Foundations Applications and Computational Aspects," *ZOR- Mathematical Methods of Operations Research*, vol. 39, pp. 35 - 68, 1994.
- [Kuczera, 1998]** George Kuczera y Eric Parent, "Monte Carlo Assessment of Parameter Uncertainty in Conceptual Catchment Models: the Metropolis Algorithm," *Journal of Hydrology*, vol. 211, pp. 69 - 85, 1998.
- [Langford, 2005a]** John Langford, "Tutorial on Practical Prediction Theory for Classification," *Journal of Machine Learning Research*, vol. 6, pp. 273 - 306, 2005a.
- [Langford, 2005b]** John Langford y Bianca Zadrozny, "Estimating Class Membership Probabilities using Classifier Learners," en *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, Barbados, 2005b, pp. 198 - 205.
- [Langford, 2008]** John Langford. (2008, Febrero) Machine Learning (Theory). [Online]. <http://hunch.net/?p=317>
- [Langley, 1988]** Pat Langley, "Machine Learning as an Experimental Science," *Machine Learning*, vol. 3, pp. 5 - 8, 1988.
- [Laskov, 2006]** Pavel Laskov, Christian Gehl, Stefan Kruger y Klaus-Robert Muller,

- "Incremental Support Vector Learning: Analysis, Implementation and Applications," *Journal of Machine Learning Research*, vol. 7, pp. 1909 - 1936, 2006.
- [Laws, 2008]** Florian Laws y Hinrich Schütze, "Stopping Criteria for Active Learning of Named Entity Recognition," en *Proceedings of the 22nd International Conference on Computational Linguistics*, Stroudsburg, PA, USA, 2008, pp. 465 - 472.
- [Layne, 2012]** R. Layne, A. Capel, N. Cook, M. Wheatley y JET EFDA contributors, "Long Term Preservation of Scientific Data: Lessons from JET and Other Domains," *Fusion Engineering and Design*, vol. 87, no. 12, pp. 2209 - 2212, 2012.
- [Lewis, 1994a]** David D. Lewis y Jason Catlett, "Heterogeneous Uncertainty Sampling for Supervised Learning," en *Proceedings of the Eleventh International Conference on Machine Learning*, San Francisco, CA, USA, 1994a, pp. 148 - 156.
- [Lewis, 1994b]** David D. Lewis y William A. Gale, "A Sequential Algorithm for Training Text Classifiers," en *Proceedings of 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA, 1994b, pp. 3 - 12.
- [Li, 2011]** Jun Li, "Discriminative Image Segmentation: Applications to Hyperspectral Data," Lisboa, Doctoral Thesis 2011.
- [Li, 2006]** Mingkun Li y Ishwar K. Sethi, "Confidence-Based Active Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.

28, no. 8, pp. 1251 - 1261, 2006.

- [Littlestone, 1986] Nick Littlestone y Manfred K. Warmuth, "Relating Data Compression and Learnability," Santa Cruz, CA, USA, Technical Report 1986.
- [Liu, 2013] Boading Liu. (2013) Uncertainty Theory Online. [Online]. <http://orsc.edu.cn/online>
- [LNF, 2004a] Laboratorio Nacional de Fusión. (2004a, Agosto) Glossario de Términos de Fusión y de ITER. [Online]. http://www-fusion.ciemat.es/New_fusion/en/Fusion/glosario.shtml
- [LNF, 2004b] Laboratório Nacional de Fusión. (2004b, Agosto) The TJ - II Project: Coil System. [Online]. <http://www-fusion.ciemat.es>
- [LNF, 2012] Laboratorio Nacional de Fusión y Fusetnet. (2012, Agosto) FusionWiki. [Online]. [http://fusionwiki.ciemat.es/wiki/TJ-II:Neutral Beam Injection](http://fusionwiki.ciemat.es/wiki/TJ-II:Neutral_Beam_Injection)
- [López, 2012] J. M. López y otros., "Implementation of the Disruption Predictor APODIS in JET Real Time Network using the MARTe Framework," en *Proceedings of the 18th IEEE - NPSS Real Time Conference*, Berkeley, CA, USA, 2012, pp. 1 - 4.
- [Lyman, 2003] Peter Lyman y Hal R. Varian. (2003, Octubre) How Much Information? 2003. [Online]. <http://www.sims.berkeley.edu/how-much-info-2003>
- [Madnick, 2009] Stuart Madnick, John Norris Maguire, MacKenzie Smith y Kate Clopeck. (2009, Junio) How Much Information? Case Studies on

- Scientific Research at MIT. [Online]. http://hmi.ucsd.edu/pdf/MIT_case_studies_combination.pdf
- [**Makili, 2010**] Lázaro Makili y otros., "Upgrade of the Automatic Analysis System in the TJ-II Thomson Scattering Diagnostic: New Image Recognition Classifier and Fault Condition Detection," *Fusion Engineering and Design*, vol. 85, pp. 415 - 418, 2010.
- [**Mallat, 1999**] Stéphane Mallat, *A Wavelet Tour of Signal Processing*, 2nd ed. San Diego, CA, USA: Elsevier Academic Press, 1999.
- [**Manning, 2008**] Christopher D. Manning, Prabhakar Raghavan y Hinrich Schütze. (2008) Introduction to Information Retrieval. [Online]. <http://informationretrieval.org/>
- [**Maskey, 2004**] Shreedhar Maskey, Vincent Guinot y Roland K. Price, "Treatment of Precipitation Uncertainty in Rainfall-Runoff Modelling: A Fuzzy Set Approach," *Advances in Water Resources*, vol. 27, pp. 889 - 898, 2004.
- [**MathWorks, 2009**] The MathWorks, Inc., *Matlab External Interfaces*. Natick, MA, USA: The MathWorks, Inc., 2009. [Online]. <http://www.mathworks.com>
- [**MathWorks, 2013**] The Mathworks, Inc. (2013) MathWorks Documentation Center. [Online]. http://www.mathworks.es/es/help/matlab/matlab_external/upgrading-mex-files-to-use-64-bit-api.html
- [**McCallum, 1998**] Andrew Kachites McCallum y Kamal Nigam, "Employing EM and Pool-Based Active Learning for Text Classification," en *Proceedings of the 15th International Conference on Machine Learning*, San

Francisco, CA, USA, 1998, pp. 359 - 367.

- [**McCracken, 2005**] Garry McCracken y Peter Stott, *Fusion The Energy of the Universe*. Burlington, MA, USA: Elsevier Academic Press, 2005.
- [**Melluish, 2001**] Thomas Melluish, Craig Saunders, Ilia Nourtdinov y Vladimir Vovk, "Comparing the Bayes and typicalness frameworks," en *Proceedings of the 12th European Conference on Machine Learning - Lecture Notes in Artificial Intelligence*, Luc De Raedt y Peter Flash, Eds. Berlin Heidelberg: Springer-Verlag, 2001, vol. 2167, pp. 360 - 371.
- [**Melville, 2004**] Prem Melville y Raymond J. Mooney, "Diverse Ensembles for Active Learning," en *Proceedings of the Twenty-First International Conference on Machine Learning*, New York, NY, USA, 2004, p. 74.
- [**Microsoft, 2012**] Microsoft Corporation. (2012, Noviembre) Windows Dev Center. [Online]. <http://msdn.microsoft.com/en-us/library/windows/desktop/ms740673%28v=vs.85%29.aspx>
- [**Mitchell, 2006**] Tom M. Mitchell, "The Discipline of Machine Learning," Pittsburgh, PA, USA, Technical Report CMU-ML-06-108 2006.
- [**Murari, 2009**] Andrea Murari y otros. (2009) EFDA-JET Preprints and Conference Papers Service. [Online]. www.iop.org/Jet/fulltext/EFDP09004.pdf
- [**Murari, 2012**] Andrea Murari y otros., "Latest Developments in Data Analysis Tools for Disruption Prediction and for the Exploration of Multimachine Operational Spaces," en *Proceedings of the 24th IAEA Fusion Energy Conference*, San Diego, CA, USA, 2012, p. 253. [Online]. <http://www->

naweb.iaea.org/napc/physics/FEC/FEC2012/html/proceedings.pdf

- [**Nguyen, 2004**] Hieu T. Nguyen y Arnold Smeulders, "Active Learning Using Pre-clustering," en *Proceedings of the 21st International Conference on Machine Learning*, New York, NY, USA, 2004, p. 79.
- [**NIFS, 2013**] National Institute for Fusion Science. (2013) Large Helical Device Information. [Online]. <http://www.lhd.nifs.ac.jp/en/lhd/>
- [**Nilsson, 2010**] Nils J. Nilsson. (2010, Junio) Introduction to Machine Learning (An Early Draft of a Proposed Textbook). [Online]. <http://ai.stanford.edu/~nilsson/mlbook.html>
- [**Nouretdinov, 2001**] Ilia Nouretdinov, Vladimir Vovk, Michael Vyugin y Alex Gammerman, "Pattern Recognition and Density Estimation under the General IID Assumption," Department of Computer Science, Royal Holloway University of London, Technical Report CLRC - TR - 01 - 06 2001.
- [**Nouretdinov, 2010**] Ilia Nouretdinov, Guang Li, Alex Gammerman y Zhiyuan Luo, "Application of Conformal Predictors for Tea Classification Based on Electronic Nose," en *Artificial Intelligence Applications and Innovations*, H. Papadopoulos, A. Andreou y M. Bramer, Eds. Berlin Heidelberg: Springer - Verlag, 2010, pp. 303 – 310.
- [**Olsson, 2009a**] Fredrik Olsson, "A literature Survey of Active Machine Learning in the Context of Natural Language Processing," Swedish Institute of Computer Science, Technical Report T2009:06 2009a.

- [Olsson, 2009b]** Fredrik Olsson y Katrin Tomanek, "An Intrinsic Stopping Criterion for Committee-Based Active Learning," en *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, Stroudsburg, PA, USA, 2009b, pp. 138 - 146.
- [Papadopoulos, 2001]** G. Papadopoulos, P. J. Edwards y A. F. Murray, "Confidence Estimation Methods for Neural Networks: A Practical Comparison," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1278 - 1287, 2001.
- [Papadopoulos, 2002a]** Harris Papadopoulos, Kostas Proedru, Vladimir Vovk y Alex Gammerman, "Inductive Confidence Machines for Regression," en *Proceedings of the Thirteenth European Conference on Machine Learning*, Helsinki, Finland, 2002a, pp. 345 - 356.
- [Papadopoulos, 2002b]** Harris Papadopoulos, Vladimir Vovk y Alex Gammerman, "Qualified Predictions for Large Data Sets in the Case of Pattern Recognition," en *Proceedings of the International Conference on Machine Learning and Applications*, Las Vegas, NV, USA, 2002b, pp. 159 - 163.
- [Papadopoulos, 2008]** Harris Papadopoulos, "Inductive Conformal Prediction: Theory and Application to Neural Networks," en *Tools in Artificial Intelligence*, Paula Fritzsche, Ed.: InTech, 2008, pp. 315 - 330. [Online]. http://www.intechopen.com/books/tools_in_artificial_intelligence/inductive_conformal_prediction_theory_and_application_to_neural_networks
- [Piel, 2010]** Alexander Piel, *Plasma Physics An Introduction to Laboratory, Space,*

and Fusion Plasmas. Berlin Heidelberg: Springer - Verlag, 2010.

- [Pilatowska, 2009] Mariola Pilatowska, "Combined Forecasts Using the Akaike Weights," *Dynamic Econometric Models*, pp. 5 - 16, 2009. [Online]. http://www.dem.umk.pl/dem/archiwa/v9/01_MPilatowska_UMK.pdf
- [Platt, 1999] John C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," en *Advances in Large Margin Classifiers*, Alexander J. Smola y otros., Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 61 - 74.
- [PPPL, 2012] Princeton Plasma Physics Laboratory. (2012, Febrero) The TFTR Project at Princeton Plasma Physics Laboratory. [Online]. <http://w3.pppl.gov/tftr/>
- [Quiñonero-Candela, 2006] Joaquin Quiñonero-Candela, Carl Edward Rasmussen, Fabian Sinz, Olivier Bousquet y Bernhard Scholkopf, "Evaluating Predictive Uncertainty Challenge," en *Machine Learning Challenges: Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment - Lecture Notes on Artificial Intelligence*, Joaquin Quiñonero-Candela y otros., Eds. Berlin Heidelberg: Springer-Verlag, 2006, vol. 3944, pp. 1 - 27.
- [Rattá, 2008] G. A. Rattá, J. Vega, A. Murari, M. Johnson y JET-EFDA COntributors, "Feature Extraction for Improved Disruption Prediction Analysis at JET," *Review of Scientific Instruments*, vol. 79, no. 10, pp. 10F328 - 10F328-3, 2008.
- [Rattá, 2010] G. A. Rattá y otros., "An Advanced Disruption Predictor for JET

Tested in a Simulated Real Time Environment," *Nuclear Fusion*, vol. 50, no. 2, p. 025005 (10pp), 2010.

- [Rattá, 2012]** Giuseppe A. Rattá, "Técnicas de Minería de Datos Aplicadas a Fusión Nuclear: Predicción en Tiempo Real y Clasificación," Departamento de Informática y Automática, ETS de Ingeniería Informática - UNED, Madrid, Tesis Doctoral 2012.
- [Roy, 2001]** Nicholas Roy y Andrew Kachites McCallum, "Toward Optimal Active Learning through Sampling Estimation of Error Reduction," en *Proceedings of the 18th International Conference on Machine Learning*, San Francisco, CA, USA, 2001, pp. 441 - 448.
- [Rüping, 2001]** Stefan Rüping, "Incremental Learning With Support Vector Machines," en *Proceedings of the First IEEE International Conference on Data Mining*, San José, CA, USA, 2001, pp. 641 - 642.
- [Russel, 2003]** Stuart J. Russel y Peter Norvig, *Artificial Intelligence A Modern Approach*, 2nd ed. New Jersey, NJ, USA: Prentice Hall - Pearson Education, Inc., 2003.
- [Sá, 2001]** J. P. Marques de Sá, *Pattern Recognition. Concepts, Methods and Applications*. Berlin Heidelberg: Springer, 2001.
- [Sánchez, 1998]** Joaquin Sánchez y otros., "Diagnostic Systems for the TJ-II Flexible Helic," *Journal of Plasma and Fusion Research SERIES*, vol. 1, pp. 338 - 341, 1998.
- [Saunders, 1999]** Craig Saunders, Alex Gammerman y Vladimir Vovk, "Transduction

- with Confidence and Credibility," en *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, 1999, pp. 722 - 726.
- [Saunders, 2000]** Craig Saunders, Alex Gammerman y Vladimir Vovk, "Computationally Efficient Transductive Machines," en *Proceedings of the Eleventh International Conference on Algorithmic Learning Theory - Lecture Notes on Artificial Intelligence*, Hiroki Arimura, Sanjay Jain y Arun Sharma, Eds. Berlin Heidelberg: Springer - Verlag, 2000, vol. 1968, pp. 325 - 333.
- [Schohn, 2000]** Greg Schohn y David Cohn, "Less is More: Active Learning with Support Vector Machines," en *Proceedings of the 17th International Conference on Machine Learning*, San Francisco, CA, USA, 2000, pp. 839 - 846.
- [Schulz, 1997]** K. Schulz y B. Huwe, "Water Flow Modeling in the Unsaturated Zone with Imprecise Parameters Using a Fuzzy Approach," *Journal of Hydrology*, vol. 201, pp. 211 - 229, 1997.
- [Seeger, 2009]** Matthias Seeger, "Bayesian Modelling in Machine Learning: A Tutorial Review," Saarland University, Saarbrücken, Germany, Technical Report 2009.
- [Settles, 2008]** Burr Settles y Mark Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks," en *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, PA, USA, 2008, pp. 1070 - 1079.

- [Settles, 2009] Burr Settles, "Active Learning Literature Survey," University of Wisconsin – Madison, Madison, WI, USA, Computer Sciences Technical Report 1648 2009.
- [Seung, 1992] H. Sebastian Seung, M. Opper y H. Sompolinsky, "Query by Committee," en *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, New York, NY, USA, 1992, pp. 287 - 294.
- [Shafer, 1990] Glenn Shafer, "Perspectives on the Theory and Practice of Belief Functions," *International Journal of Approximate Reasoning*, vol. 4, no. 5 - 6, pp. 323 - 362, 1990.
- [Shafer, 2008] Glenn Shafer y Vladimir Vovk, "A Tutorial on Conformal Prediction," *Journal of Machine Learning Research*, vol. 9, pp. 371 - 421, 2008.
- [Shah, 2008] Mohak Shah, "Risk Bounds for Classifier Evaluation: Possibilities and Challenges," en *3rd Workshop on Evaluation Methods for Machine Learning in conjunction with the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.
- [Shah, 2009] Mohak Shah y Sara Shanián, "Hold-out Risk Bounds for Classifier Performance Evaluation," en *4th Workshop on Evaluation Methods for Machine Learning in conjunction with the 26th International Conference on Machine Learning (ICML 2009)*, Montréal, Québec, Canada, 2009.
- [Sheffield, 1994] John Sheffield, "The Physics of Magnetic Fusion Reactors," *Reviews of Modern Physics*, vol. 66, no. 3, pp. 1015 - 1103, 1994.

- [Shen, 2004]** Dan Shen, Jie Zhang, Jian Su, Guodong Zhou y Chew-Lim Tan, "Multi-Criteria-based Active Learning for Named Entity Recognition," en *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2004, pp. 589 - 596.
- [Shilton, 2005]** Alistair Shilton, M. Palaniswami, Daniel Ralph y Ah Chung Tsoi, "Incremental Training of Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 114 - 131, 2005.
- [Shrestha, 2006]** Durga Lal Shrestha y Dimitri P. Solomatine, "Machine Learning Approaches for Estimation of Prediction Interval for the Model Output," *Neural Networks (2006 special issue: Earth Sciences and Environmental Applications of Computational Intelligence)*, vol. 19, no. 2, pp. 225 - 235, 2006.
- [Shrestha, 2008]** Durga Lal Shrestha y Dimitri P. Solomatine, "Comparing Machine Learning Methods in Estimation of Model Uncertainty," en *Proceedings of the IEEE International Joint Conference on Neural Networks*, Hong Kong, 2008, pp. 1410 - 1416.
- [Siarry, 1997]** P. Siarry y G. Berthiau, "Fitting of Tabu Search to Optimize Functions of Continuous Variables," *International Journal for Numerical Methods in Engineering*, vol. 40, pp. 2449 - 2457, 1997.
- [Skalak, 2007]** David B. Skalak, Rich Caruana y Alexandru Niculescu-Mizil, "Classifier Loss under Metric Uncertainty," en *Proceedings of the 18th European Conference on Machine Learning*, Joost N. Kok y otros.,

Eds. Berlin Heidelberg: Springer - Verlag, 2007, pp. 310 - 322.

- [**Staelin, 2002**] Carl Staelin, "Parameter Selection for Support Vector Machines," Technical Report HPL-2002-354R1 2002.
- [**Syed, 1999a**] Nadeem Ahmed Syed, Huan Liu y Kah Kay Sung, "Incremental Learning with Support Vector Machines," en *Proceedings of Workshop on Support Vector Machines at International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999a.
- [**Syed, 1999b**] Nadeem Ahmed Syed, Huan Liu y Kah Kay Sung, "Handling Concept Drifts in Incremental Learning with Support Vector machines," en *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 1999b, pp. 317 - 321.
- [**Theodoridis, 2003**] Sergios Theodoridis y Konstantinos Koutroumbas, *Pattern Recognition*, 2nd ed. San Diego, CA, USA: Elsevier Academic Press, 2003.
- [**Thompson, 1999**] Cynthia A. Thompson, Mary Elaine Califf y Raymond J. Mooney, "Active Learning for Natural Language Parsing and Information Extraction," en *Proceedings of the 16th International Machine Learning Conference*, San Francisco, CA, USA, 1999, pp. 406 - 414.
- [**Thumm, 1995**] Manfred Thumm, "Advanced Electron Cyclotron Heating Systems for Next-Step Fusion Experiments," *Fusion Engineering and Design*, vol. 30, pp. 139 - 170, 1995.

- [**Todini, 2009**] Ezio Todini, "Predictive Uncertainty Assessment in Real Time Flood Forecasting," en *Uncertainties in Environmental Modelling and Consequences for Policy Making*, P. C. Baveye, M. Laba y J. Mysiak, Eds. Dordrecht, Netherlands: Springer Science + Business Media, Inc, 2009, pp. 205 – 228.
- [**Tong, 2001**] Simon Tong y Edward Chang, "Support Vector Machine Active Learning for Image Retrieval," en *Proceedings of the 9th ACM International Conference on Multimedia*, New York, NY, USA, 2001, pp. 107 - 118.
- [**Tong, 2002**] Simon Tong y Daphne Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *The Journal of Machine Learning Research*, vol. 2, no. 1, pp. 45 - 66, 2002.
- [**Tribaldos, 1998**] V. Tribaldos, J. A. Jiménez, J. Guasp y B. Ph. Van Milligen, "Electron Cyclotron Heating and Current Drive in the TJ - II Stellarator," *Plasma Physics and Controlled Fusion*, vol. 40, pp. 2113 - 2130, 1998.
- [**USEIA, 2011**] US Energy Information Administration, "International Energy Outlook 2011," Washington, DC, USA, 2011.
- [**Valiant, 1984**] Leslie G. Valiant, "A Theory of the Learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134 - 1142, 1984.
- [**Vanderlooy, 2006**] Stijn Vanderlooy y Ida G. Sprinkhuisen-Kuyper, "An Overview of Algorithmic Randomness and its Application to Reliable Instance Classification," Universiteit Maastricht, Maastricht, Netherlands, Technical Report MICC-IKAT 07-02 2006.

- [Vanderlooy, 2007]** Stijn Vanderlooy, Laurens van der Maaten y Ida G. Sprinkhuisen-Kuyper, "Off-Line Learning with Transductive Confidence Machines: An Empirical Evaluation," en *Proceedings of the 5th International Conference on Machine Learning and Data Mining in Pattern Recognition - Lecture Notes on Artificial Intelligence*, Petra Perner, Ed. Berlin Heidelberg: Springer - Verlag, 2007, vol. 4571, pp. 310 - 323.
- [Vapnik, 2000]** Vladimir Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 2000, vol. 2nd.
- [Vega, 1996]** Jesús Vega, C. Crémy, E. Sánchez, A. Portas y Sebastián Dormido-Canto, "Encoding Technique for High Data Compaction in Data Bases of Fusion Devices," *Review of Scientific Instruments*, vol. 67, no. 12, pp. 4154 - 4160, 1996.
- [Vega, 1999]** Jesús Vega, C. Crémy, E. Sánchez y A. Portas, "The TJ-II Data Acquisition System: An Overview," *Fusion Engineering and design*, vol. 43, pp. 309 - 319, 1999.
- [Vega, 2004]** Jesús Vega y otros., "A Distributed Synchronization System for the TJ-II Local Area Network," *Fusion Engineering and Design*, vol. 71, pp. 117 - 121, 2004.
- [Vega, 2005]** Jesús Vega y otros., "Application of Intelligent Classification Techniques to the TJ II Thomson Scattering Diagnostic," en *32nd EPS Conference on Plasma Physics and Controlled Fusion*, vol. ECA 29, C, Tarragona, 2005, pp. 2090 - 2093.
- [Vega, 2007]** Jesús Vega y otros., "Real-time Lossless Data Compression Techniques

- for Long-pulse Operation," *Fusion Engineering and Design*, vol. 82, pp. 1301 - 1307, 2007.
- [Vega, 2008a]** Jesús Vega y otros., "Data Mining Technique for Fast Retrieval of Similar Waveforms in Fusion Massive Databases," *Fusion Engineering and Design*, vol. 83, pp. 132 - 139, 2008a.
- [Vega, 2008b]** Jesús Vega, G. A. Rattá, P. Castro, A. Murari y JET EFDA contributors, "Development of Learning Systems with Data Tours Techniques for Fusion Databases," en *Computational Intelligence In Decision And Control: Proceedings of the 8th International FLINS Conference*, Da Ruan y otros., Eds. Singapore: World Scientific Publishing Co. Pte. Ltd., 2008b, pp. 103 - 108.
- [Vega, 2009]** Jesús Vega, Andrea Murari, G. Vagliasindi, G. A. Rattá y JET-EFDA Contributors, "Automated Estimation of L/H Transition Times at JET by Combining Bayesian Statistics and Support Vector Machines," *Nuclear Fusion*, vol. 49, no. 085023, 2009.
- [Vega, 2010]** Jesús Vega, Andrea Murari, Augusto Pereira, S. González y Ignacio Pastor, "Accurate and Reliable Image Classification by Using Conformal Predictors in the TJ-II Thomson Scattering," *Review of Scientific Instruments*, vol. 81, no. 10, p. 10E118, 2010.
- [Vega, 2012]** Jesús Vega y otros. (2012) Results of the JET Real-Time Disruption Predictor in the ITER-Like Wall Campaigns. [Online]. <http://www.iop.org/Jet/toc?type=C&start=EFDC120501>
- [Vereshchagin,** Nikolai Konstantinovich Vereshchagin y Alexander Shen, *Computable*

- 2003]** *Functions*. USA: American Mathematical Society, 2003.
- [Vlachos, 2008]** Andreas Vlachos, "A Stopping Criterion for Active Learning," *Journal of Computer Speech and Language*, vol. 22, no. 3, pp. 295 - 312, 2008.
- [Vovk, 1999]** Vladimir Vovk, Alex Gammerman y Craig Saunders, "Machine Learning Applications of Algorithmic Randomness," en *Proceedings of the 16th International Conference on Machine Learning*, Ivan Bratko y Saso Dzeroski, Eds. San Francisco, CA, USA: Morgan Kaufmann, 1999, pp. 444 - 453.
- [Vovk, 2002]** Vladimir Vovk, "On-Line Confidence Machines Are Well-Calibrated," en *Proceedings of the 43th Annual Symposium on Foundations of Computer Science*, Washington, DC, USA, 2002, pp. 187 - 196.
- [Vovk, 2005]** Vladimir Vovk, Alex Gammerman y Glenn Shafer, *Algorithmic Learning in a Random World*. New York, USA: Springer Science + Business Media, Inc, 2005.
- [Wakatani, 1998]** Masahiro Wakatani, *Stellarator and Heliotron Devices*. New York, USA: Oxford University Press, 1998.
- [Weston, 2006]** Jason Weston, Andre Elisseeff, Gokhan Bakir y Fabian Sinz. (2006, Julio) The Spider. [Online]. <http://people.kyb.tuebingen.mpg.de/spider/main.html>
- [Witten, 2005]** Ian H. Witten y Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. USA: Morgan Kaufmann, 2005.
- [Xu, 2007]** Zuobing Xu, Ram Akella y Yi Zhang, "Incorporating Diversity and

- Density in Active Learning for Relevance Feedback," en *Proceedings of the 29th European Conference on IR Research - Lecture Notes on Computer Science*, vol. 4425, Berlin, Heidelberg, 2007, pp. 246 - 257.
- [Yu, 2005]** Hwanjo Yu, "SVM Selective Sampling for Ranking with Application to Data Retrieval," en *Proceedings of the 11th International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2005, pp. 354 - 363.
- [Zadeh, 1965]** Lofti A. Zadeh, "Fuzzy Sets," *Information and Control*, pp. 338 - 353, 1965.
- [Zhang, 2012]** Jie Zhang, Wilson H. Tang, L. M. Zhang y H. W. Huang, "Characterising Geotechnical Model Uncertainty by Hybrid Markov Chain Monte Carlo Simulation," *Computers and Geotechnics*, vol. 43, pp. 26 - 36, 2012.
- [Zhu, 2008a]** Jingbo Zhu, Huizhen Wang y Eduard Hovy, "Learning a Stopping Criterion for Active Learning for Word Sense Disambiguation and Text Classification," en *International Joint Conference on Natural Language Processing*, 2008a.
- [Zhu, 2008b]** Jingbo Zhu, Huizhen Wang y Eduard Hovy, "Multi-Criteria-based Strategy to Stop Active Learning for Data Annotation," en *Proceedings of the 22nd International Conference on Computational Linguistics*, Stroudsburg, PA, USA, 2008b, pp. 1129 - 1136.
- [Zhu, 2010]** Jingbo Zhu, Huizhen Wang, Eduard Hovy y Matthew Ma, "Confidence-Based Stopping Criteria for Active Learning for Data

Annotation," *ACM Transactions on Speech and Language Processing*,
vol. 6, no. 3 - 3, 2010.

ANEXO A: CONJUNTO DE DATOS

Las imágenes del TSD del TJ – II son agrupadas en cinco clases diferentes, BKGND, *ECRH*, CUTOFF, *NBI* y STRAY, correspondiendo cada una a un régimen de medida distinto del *diagnóstico*. En la Tabla 17 se puede ver una breve descripción de cada uno de los patrones de imágenes.

En la Figura 40 se muestran los patrones característicos de las imágenes del TSD. En varios trabajos, como por ejemplo (Farias, 2004, 2005), se trata de hacer una descripción de la apariencia visual de los mismos. En todos existe una concordancia en cuanto a que todos los patrones, excepto BKGND, presentan al menos cuatro características distintivas fundamentales: una zona vacía en el centro, dos componentes verticales centrales y una línea fina a la derecha. Las diferencias fundamentales entre los diferentes patrones se derivan de la intensidad de la luz, elevada en las componentes centrales, en el caso de *NBI*, más atenuada, pero con la línea fina bien demarcada, en el caso de *ECRH*, con las componentes centrales concentradas en la región inferior en el caso de CUTOFF y casi nulas en el caso de STRAY.

Clase	Descripción
BKGND	Fondo de la Cámara ICCD
ECRH	Plasma durante el calentamiento ECRH
CUTOFF	Plasma después de alcanzada la densidad de corte durante el calentamiento ECRH
NBI	Plasma durante el calentamiento NBI
STRAY	Luz parásita sin plasma

Tabla 17: Clases de imágenes del TSD del TJ – II

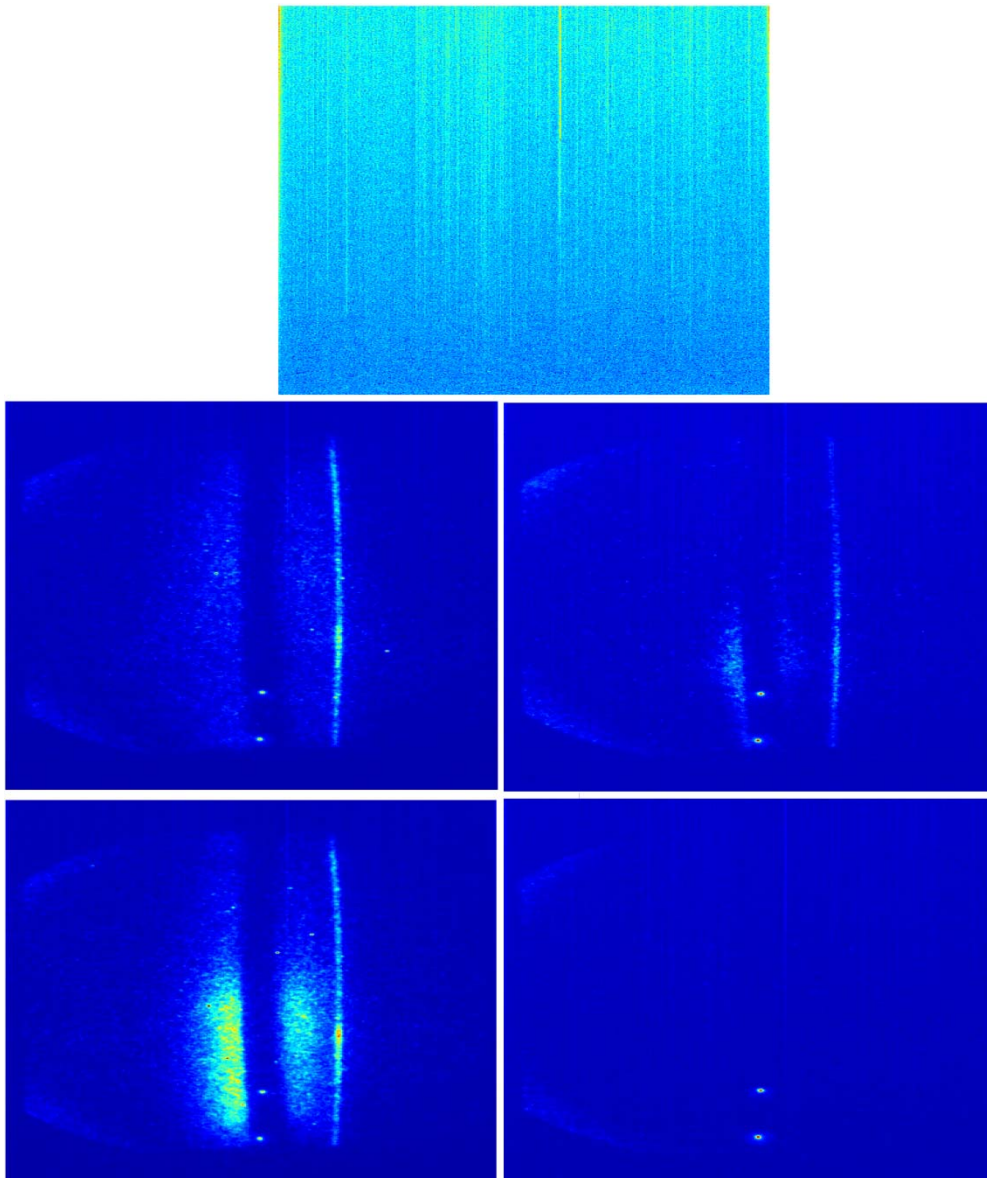


Figura 40: Patrones típicos de imágenes del TSD. De arriba a abajo y de izquierda a derecha: BKGND, ECRH, CUTOFF, NBI y STRAY

El conjunto de datos utilizado está compuesto por 1149 imágenes adquiridas en la campaña de 2008 del TJ – II, distribuidas por las diferentes clases, tal como se muestra en la Tabla 18. Cada imagen es dada a través de una matriz de 576 x 385 píxeles, es decir, un total de 221.760 píxeles. Debido a la grande cantidad de posibles atributos, para la determinación de los vectores de características se efectuó una operación de preprocesamiento de las mismas, operación que es descrita a continuación.

Clase	Cantidad
BKGND	107
ECRH	461
CUTOFF	42
NBI	334
STRAY	205
Total	1149

Tabla 18: Cantidad de imágenes por clase

A.1. PREPROCESAMIENTO

Para evitar problemas de dimensionalidad cada imagen fue representada por un vector de características constituido por su transformada *wavelet* (Mallat, 1999). Los resultados del análisis de señales con *wavelets* dependen de la selección de los parámetros a ser empleados, es decir, del tipo de *wavelet* madre y del nivel de descomposición. Por otro lado, por el hecho de ser multiescala la descomposición con transformada *wavelet*, las imágenes pueden ser caracterizadas por un conjunto de coeficientes de aproximación y tres conjuntos de coeficientes de detalles (horizontal, vertical y diagonal). Los coeficientes de aproximación representan la información sobre la imagen en grande escala, conteniendo la mayor parte de la energía de la imagen, mientras que los valores de los coeficientes de detalle son próximos a cero, pero la

información que los mismos representan puede ser relevante en determinados contextos. La referida selección depende de la experiencia con las señales utilizadas.

En este caso como vectores de características fueron utilizados los coeficientes de detalle vertical a los niveles 4 y 5 de descomposición, de forma separada. En trabajos previos (Farias, 2004) se demostró que estas descomposiciones poseen buena capacidad de discriminación para este tipo de señales. Las mismas permiten reducir el número de atributos a 36×25 (0,41 % del original) y 18×13 (0,11 %), respectivamente.

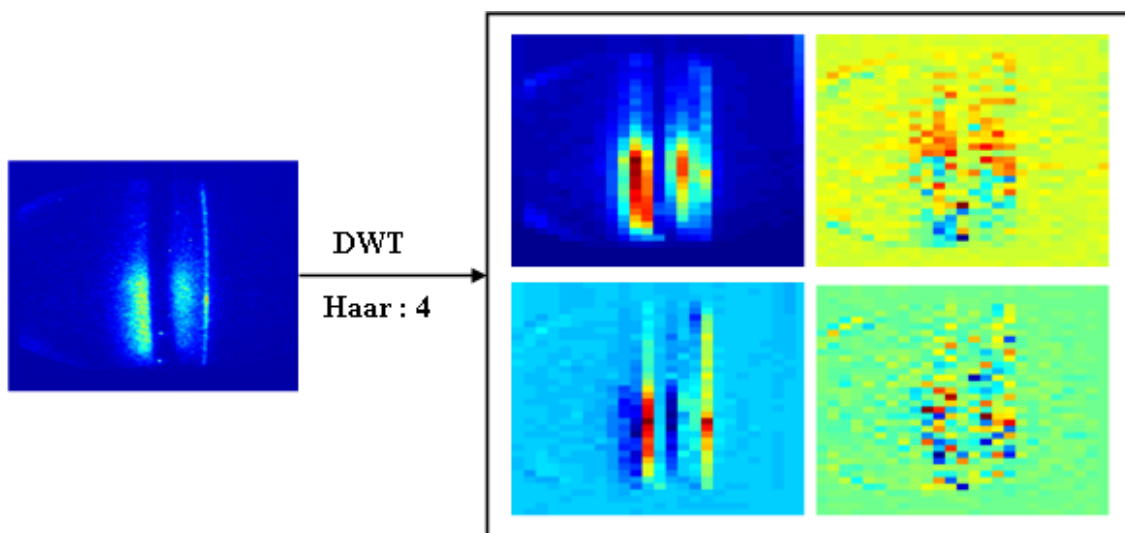


Figura 41: Transformada wavelet discreta de una imagen del TSD. A izquierda: imagen original. A derecha, de arriba a abajo y de izquierda a derecha: aproximación, detalle horizontal, detalle vertical y detalle diagonal

ANEXO B: EJEMPLO DE UTILIZACIÓN DEL MÉTODO DE CLASIFICACIÓN CON FUNCIÓN *HASH*

En este anexo se presenta un ejemplo de clasificación conformal utilizándose una función *hash*. En el mismo se ejemplifica el proceso de clasificación, enfatizándose el mecanismo de formación de los subconjuntos de trabajo y de entrenamiento, a partir de los datos de entrenamiento originales, para la formulación de los problemas binarios auxiliares. Los datos utilizados en el ejemplo fueron extraídos del conjunto de datos *wine* del repositorio de datos de la UCI (Bache, 2013). Los mismos se describen en la **Tabla 19**.

Sea un conjunto de entrenamiento $X_{ent} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_{15}, y_{15})\}$ y un vector de test $\mathbf{x}_{test} = \mathbf{x}_{16}$. El ejemplo consiste en clasificar el vector de test dividiendo el conjunto de entrenamiento en dos partes, es decir, utilizándose $h = 2$ al emplear el algoritmo. A efectos de la clasificación se considera una función *hash*, $f_h : R^{13} \rightarrow \{1, 2, \dots, h\}$, la cual dado un vector de entrada \mathbf{x}_i devuelve un valor entero en el rango $\{1, 2, \dots, h\}$. La referida función se define como

$$f_h(\mathbf{x}) = [\text{round}(\sum_{l=1}^{13} x_l) \bmod h] + 1.$$

Num	y	x
1	1	1:13,2 2:1,78 3:2,14 4:11,2 5:100 6:2,65 7:2,76 8:0,26 9:1,28 10:4,38 11:1,05 12:0,34 13:1050
2	1	1:13,16 2:2,36 3:2,67 4:18,6 5:101 6:2,8 7:3,24 8:0,3 9:2,81 10:5,68 11:1,03 12:3,17 13:1185
3	1	1:14,37 2:1,95 3:2,5 4:16,8 5:113 6:3,85 7:3,49 8:0,24 9:2,18 10:7,8 11:0,86 12:3,45 13:1480
4	1	1:13,24 2:2,59 3:2,87 4:21 5:118 6:2,8 7:2,69 8:0,39 9:1,82 10:4,32 11:1,04 12:2,93 13:735
5	1	1:14,2 2:1,76 3:2,45 4:15,2 5:112 6:3,27 7:3,39 8:0,34 9:1,97 10:6,75 11:1,05 12:2,85 13:1450
6	2	1:12,37 2:0,94 3:1,36 4:10,6 5:88 6:1,98 7:0,57 8:0,28 9:0,42 10:1,95 11:1,05 12:1,82 13:520
7	2	1:12,33 2:1,1 3:2,28 4:16 5:101 6:2,05 7:1,09 8:0,63 9:,41 10:3,27 11:1,25 12:1,67 13:680
8	2	1:12,64 2:1,36 3:2,02 4:16,8 5:100 6:2,02 7:1,41 8:0,53 9:0,62 10:5,75 11:,98 12:1,59 13:450
9	2	1:13,67 2:1,25 3:1,92 4:18 5:94 6:2,1 7:1,79 8:0,32 9:,73 10:3,8 11:1,23 12:2,46 13:630
10	2	1:12,17 2:1,45 3:2,53 4:19 5:104 6:1,89 7:1,75 8:,45 9:1,03 10:2,95 11:1,45 12:2,23 13:355
11	3	1:12,51 2:1,24 3:2,25 4:17,5 5:85 6:2 7:0,58 8:0,6 9:1,25 10:5,45 11:0,75 12:1,51 13:650
12	3	1:13,4 2:3,91 3:2,48 4:23 5:102 6:1,8 7:0,75 8:0,43 9:1,41 10:7,3 11:0,7 12:1,56 13:750
13	3	1:13,27 2:4,28 3:2,26 4:20 5:120 6:1,59 7:0,69 8:0,43 9:1,35 10:10,2 11:,59 12:1,56 13:835
14	3	1:13,17 2:2,59 3:2,37 4:20 5:120 6:1,65 7:0,68 8:0,53 9:1,46 10:9,3 11:0,6 12:1,62 13:840
15	3	1:14,13 2:4,1 3:2,74 4:24,5 5:96 6:2,05 7:0,76 8:0,56 9:1,35 10:9,2 11:0,61 12:1,6 13:560
16	1	1:14,23 2:1,71 3:2,43 4:15,6 5:127 6:2,8 7:3,06 8:0,28 9:2,29 10:5,64 11:1,04 12:3,92 13:1065

Tabla 19: Datos utilizados en el ejemplo descrito. Los mismos fueron extraídos del conjunto de datos *wine* del repositorio de la UCI (Bache, 2013)

El proceso de clasificación consiste en lo siguiente:

1. Tomar el objeto de test \mathbf{x}_{tst} y aplicar sobre el mismo la función *hash*, considerando, en

el caso de este ejemplo $h = 2$, $f_h(\mathbf{x}_{\text{tst}}) = 2$.

2. Aplicar la función *hash* sobre cada elemento del conjunto de entrenamiento X_{ent} , determinándose el vector,

$$f_h(\mathbf{x} : \mathbf{x} \in X_{ent}) = [2, 1, 1, 2, 2, 2, 2, 1, 2, 1, 2, 2, 2, 1, 1].$$

3. Atribuir provisionalmente la clase $c = 1$ al objeto de test.
4. Formar el subconjunto de trabajo W_1 . En este se incluyen todos los vectores pertenecientes a la clase provisional atribuida al vector de test ($c = 1$) y que devuelven un valor $j = 2$ cuando la función *hash* es aplicada sobre ellos (valor similar al devuelto al aplicar la función *hash* sobre el vector de test), es decir,

$$W_1 = \{\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_5\}.$$

5. Formar el subconjunto de entrenamiento S_{12} . En este se incluyen todos los vectores pertenecientes a la clase provisional $c = 1$ y que no devuelven un valor $j = 2$ al ser aplicada sobre ellos la función *hash*. A estos vectores se les atribuye la etiqueta 1. También se incluyen en este subconjunto todos los vectores pertenecientes a las demás clases (2 y 3). A estos se les atribuye la etiqueta -1.

$$S_{12} = \{(\mathbf{x}_2, 1), (\mathbf{x}_3, 1), (\mathbf{x}_6, -1), (\mathbf{x}_7, -1), (\mathbf{x}_8, -1), (\mathbf{x}_9, -1), \dots \\ \dots (\mathbf{x}_{10}, -1), (\mathbf{x}_{11}, -1), (\mathbf{x}_{12}, -1), (\mathbf{x}_{13}, -1), (\mathbf{x}_{14}, -1), (\mathbf{x}_{15}, -1)\}.$$

6. Entrenar una máquina SVM, utilizándose el subconjunto de entrenamiento S_{12} , y determinar una función de decisión $F_{12}(\mathbf{x})$.
7. Calcular las distancias al hiperplano de separación de los vectores incluidos en el subconjunto de trabajo W_1 y del vector de test \mathbf{x}_{tst} , utilizándose la función $F_{12}(\mathbf{x})$, es decir,

$$d_i = \{F_{12}(\mathbf{x}) : \mathbf{x} \in W_1\} \cup \{F_{12}(\mathbf{x}_{tst})\}.$$

8. Calcular el p -valor correspondiente, $p_1 = \frac{\#\{i : d_i \leq d_{ist}\}}{|\mathcal{W}_1| + 1}$.
9. Repetir los pasos 3 – 8 atribuyendo provisionalmente las clases 2 y 3 al vector de test.
10. Predecir la clase correspondiente al mayor de los p -valores calculados y asociar a la predicción los valores $Cr = P_1$ y $Cf = 1 - P_2$ para la credibilidad y la confianza, respectivamente. P_1 y P_2 representan al mayor y al segundo mayor p -valores.

ANEXO C: FIGURAS Y TABLAS ADICIONALES

En este anexo se presentan algunas figuras y tablas adicionales, con las cuales se pretende dar una visión complementaria de los resultados expuestos en el capítulo 5.

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9414	0,9995	0,8408	0,1637	0,7280	0,0299
RBF Nivel 5	0,9475	0,9985	0,2534	0,2054	0,9815	0,0057
Lineal Nivel 4	0,9668	0,9997	0,6647	0,2127	0,9647	0,0021
Lineal Nivel 5	0,9720	0,9998	0,7760	0,1557	0,9782	0,0042
Polinómico Nivel 4	0,9555	0,9986	0,6574	0,2450	0,9647	0,0024
Polinómico Nivel 5	0,9649	0,9988	0,7576	0,1815	0,9782	0,0018

Tabla 20: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (SVM Uno-Contra-Resto)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9645	0,9997	0,8555	0,0747	0,5397	0,0029
RBF Nivel 5	0,9769	0,9995	0,9095	0,0994	0,9153	0,0026
Lineal Nivel 4	0,9825	0,9995	0,8657	0,1071	0,9892	0,0024
Lineal Nivel 5	0,9805	0,9994	0,9236	0,1264	0,8005	0,0023
Polinómico Nivel 4	0,9826	0,9995	0,8595	0,1190	0,9093	0,0029
Polinómico Nivel 5	0,9801	0,9995	0,9222	0,1415	0,9494	0,0026

Tabla 21: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (SVM Uno-Contra-Uno)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9389	0,9997	0,8279	0,1647	0,7173	0,0304
RBF Nivel 5	0,9432	0,9990	0,8825	0,1317	0,9121	0,0048
Lineal Nivel 4	0,9320	0,9996	0,8088	0,1777	0,7031	0,0866
Lineal Nivel 5	0,9368	0,9985	0,0046	0,2851	0,9816	0,0040
Polinómico Nivel 4	0,9320	0,9996	0,8088	0,1777	0,7031	0,0866
Polinómico Nivel 5	0,9368	0,9985	0,0046	0,2851	0,9816	0,0040

Tabla 22: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (SVMi_SLS Uno-Contra-Resto)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9894	0,9999	0,8881	0,0647	0,6628	0,0005
RBF Nivel 5	0,9871	0,9998	0,9382	0,0904	0,7699	0,0019
Lineal Nivel 4	0,9706	0,9997	0,8799	0,1235	0,9517	0,0026
Lineal Nivel 5	0,9659	0,9996	0,9226	0,0910	0,7024	0,0031
Polinómico Nivel 4	0,9706	0,9997	0,8799	0,1235	0,9517	0,0026
Polinómico Nivel 5	0,9659	0,9996	0,9226	0,0910	0,7024	0,0031

Tabla 23: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (SVMi_SLS Uno-Contra-Uno)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9334	0,9997	0,2590	0,2770	0,9690	0,0114
RBF Nivel 5	0,9760	0,9995	0,2824	0,2329	0,8724	0,0130
Lineal Nivel 4	0,9673	0,9998	0,6554	0,2282	0,7720	0,0198
Lineal Nivel 5	0,9737	0,9994	0,7572	0,1427	0,9605	0,0036
Polinómico Nivel 4	0,9673	0,9998	0,6554	0,2282	0,7720	0,0198
Polinómico Nivel 5	0,9692	0,9994	0,7630	0,1585	0,9605	0,0026

Tabla 24: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (SVMi_CP Uno-Contra-Resto)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9962	0,9998	0,8727	0,0344	0,1110	0,0023
RBF Nivel 5	0,9971	0,9996	0,9266	0,0319	0,1351	0,0021
Lineal Nivel 4	0,9828	0,9996	0,8681	0,1163	0,7672	0,0025
Lineal Nivel 5	0,9802	0,9996	0,9222	0,1443	0,8707	0,0028
Polinómico Nivel 4	0,9826	0,9996	0,8557	0,1264	0,6158	0,0028
Polinómico Nivel 5	0,9801	0,9993	0,9259	0,1075	0,9402	0,0044

Tabla 25: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; CP (SVMi_CP Uno-Contra-Uno)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9679	0,9977	0,8503	0,1267	0,8879	0,0228
RBF Nivel 5	0,9632	0,9968	0,7478	0,0942	0,5324	0,0217
Lineal Nivel 4	0,9324	0,9960	0,2083	0,2740	0,9725	0,0121
Lineal Nivel 5	0,9163	0,9958	0,0608	0,2696	0,9953	0,0192
Polinómico Nivel 4	0,9447	0,9971	0,4817	0,1784	0,9942	0,0125
Polinómico Nivel 5	0,9496	0,9953	0,0248	0,1854	0,9936	0,0240

Tabla 26: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; Hashed CP (partes = 2)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9455	0,9952	0,7544	0,1155	0,8489	0,0265
RBF Nivel 5	0,9415	0,9935	0,7698	0,1007	0,4397	0,0156
Lineal Nivel 4	0,9185	0,9936	0,2454	0,2997	0,9877	0,0173
Lineal Nivel 5	0,9201	0,9917	0,0653	0,2557	0,9891	0,0286
Polinómico Nivel 4	0,9264	0,9942	0,5773	0,2091	0,9877	0,0192
Polinómico Nivel 5	0,9380	0,9919	0,0438	0,1689	0,9855	0,0291

Tabla 27: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; Hashed CP (partes = 4)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9921	0,9989	0,8235	0,1628	0,4965	0,0043
RBF Nivel 5	0,9918	0,9991	0,9347	0,0907	0,4272	0,0040
Lineal Nivel 4	0,9175	0,9988	0,1205	0,2944	0,8986	0,0140
Lineal Nivel 5	0,9400	0,9990	0,1940	0,3241	0,9286	0,0229
Polinómico Nivel 4	0,7375	0,9984	0,0697	0,3721	0,9136	0,0031
Polinómico Nivel 5	0,9400	0,9990	0,1940	0,3241	0,9286	0,0229

Tabla 28: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; ICP (Uno-Contra-Resto)

Kernel/Nivel	Confianza			Credibilidad		
	Med	Max	Min	Med	Max	Min
RBF Nivel 4	0,9777	0,9989	0,3817	0,3107	0,8581	0,0041
RBF Nivel 5	0,9679	0,9991	0,2902	0,3645	0,9645	0,0075
Lineal Nivel 4	0,7763	0,9989	0,0022	0,7183	0,9999	0,0041
Lineal Nivel 4	0,7346	0,9985	0,0223	0,5295	0,9987	0,0045
Polinómico Nivel 4	0,7763	0,9989	0,0022	0,7183	0,9999	0,0041
Polinómico Nivel 5	0,7346	0,9985	0,0223	0,5295	0,9987	0,0045

Tabla 29: Valores de confianza de las predicciones y de credibilidad de las predicciones incorrectas; ICP (Uno-Contra-Uno)

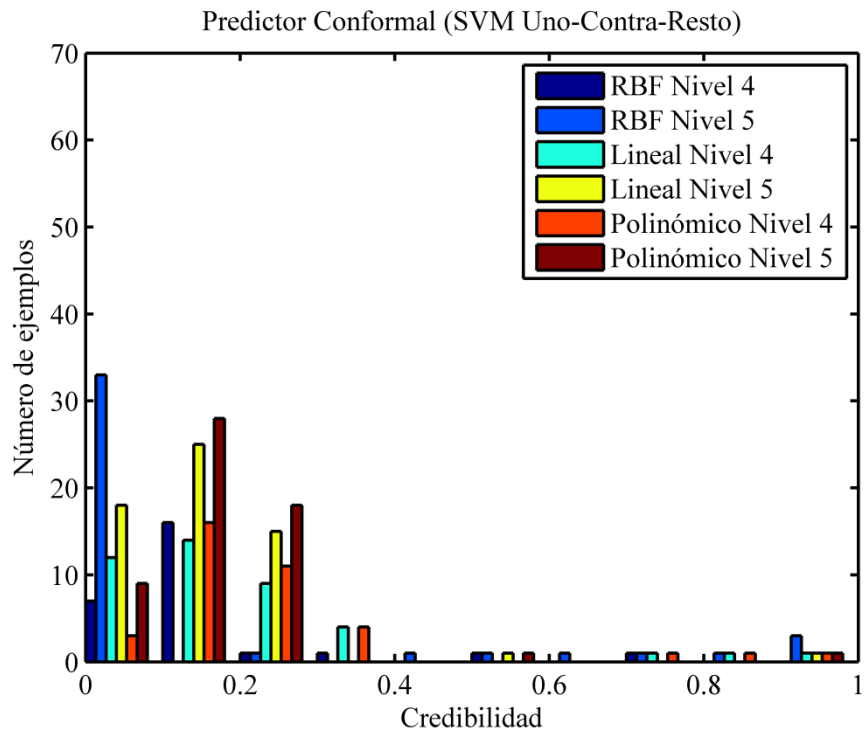


Figura 42: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (*SVM* Uno-Contra-Resto)

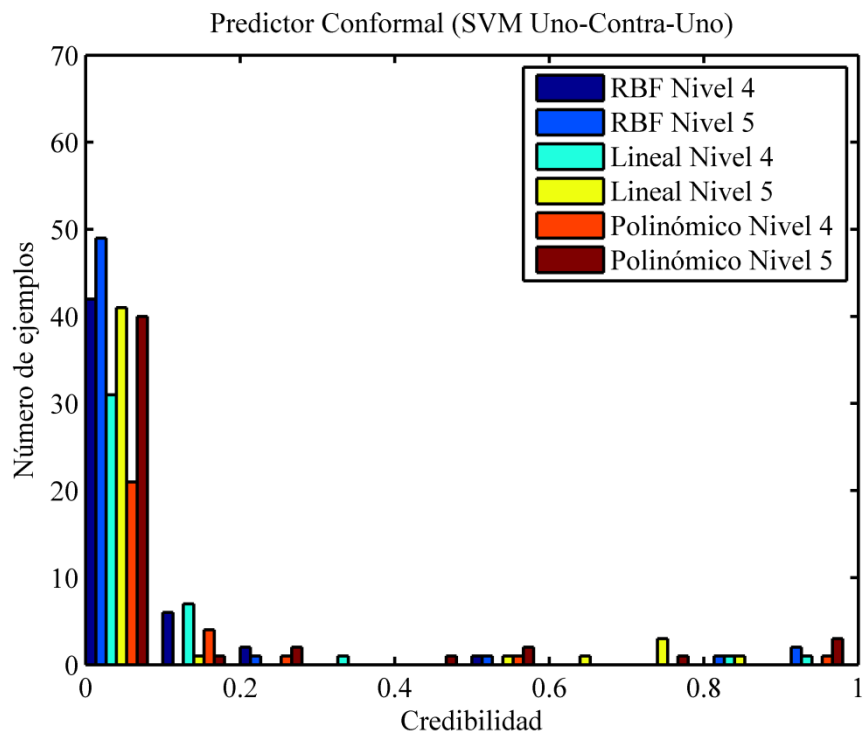


Figura 43: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (*SVM* Uno-Contra-Uno)

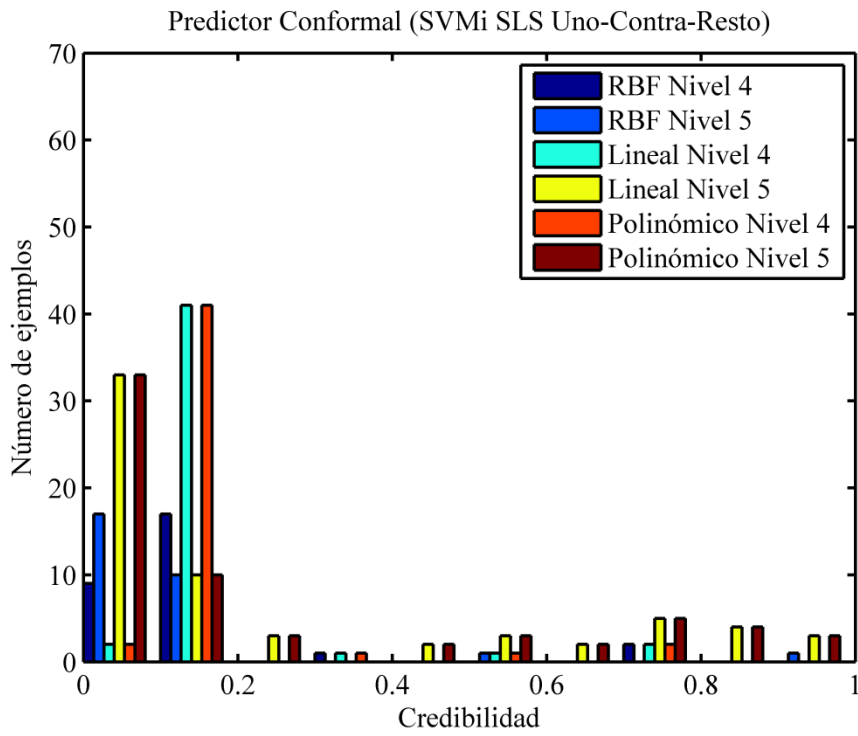


Figura 44: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVMi_SLS Uno-Contra-Resto)

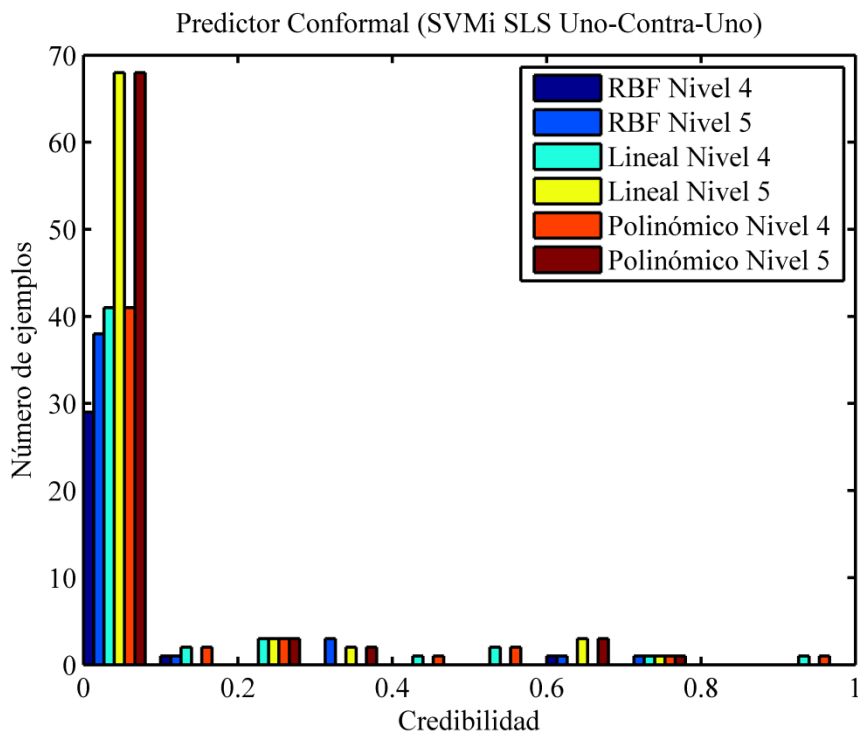


Figura 45: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVMi_SLS Uno-Contra-Uno)

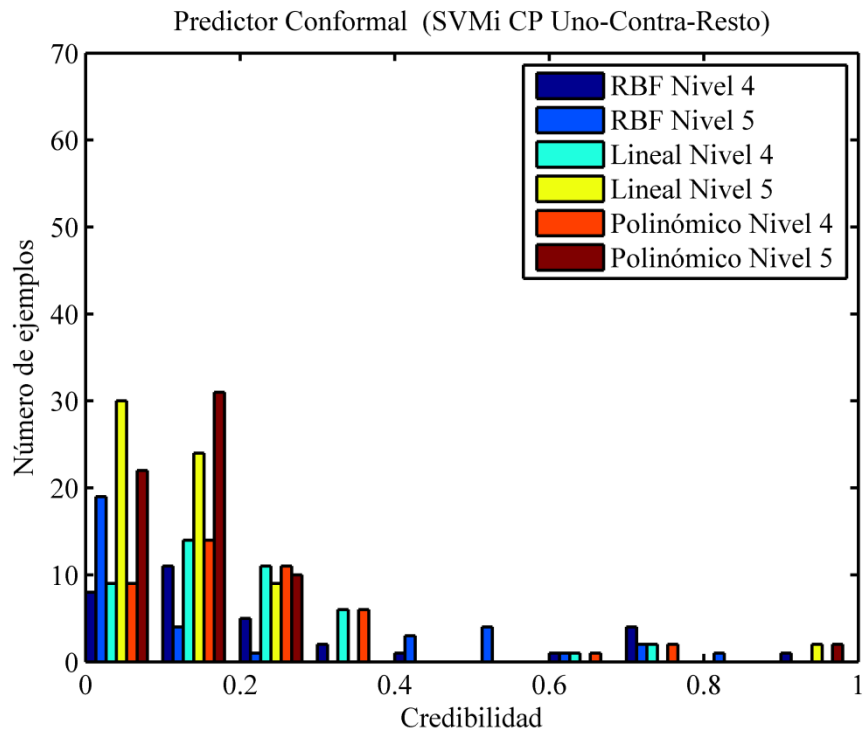


Figura 46: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVMi_CP Uno-Contra-Resto)

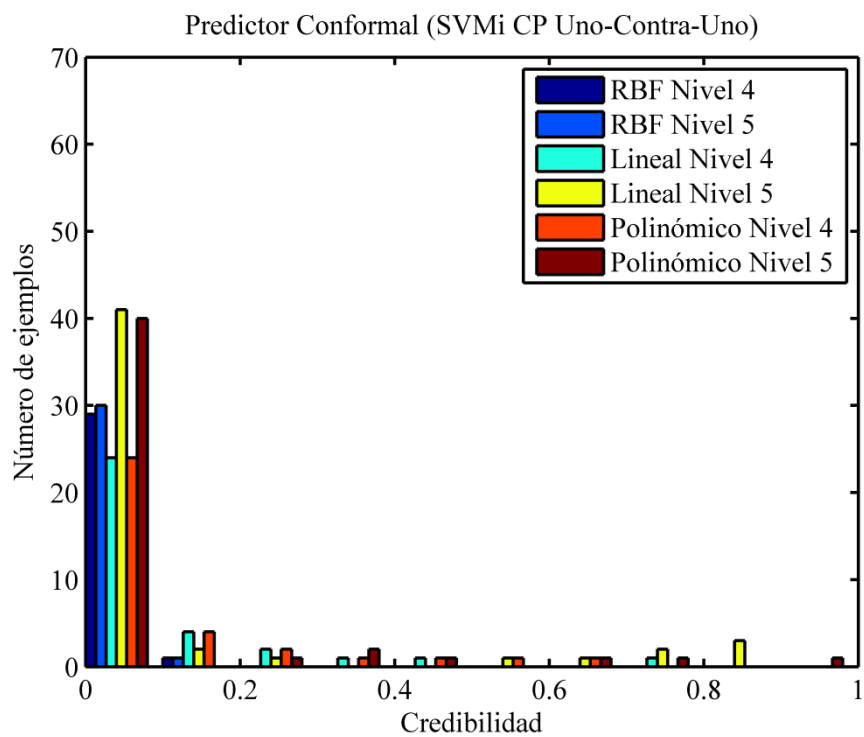


Figura 47: Valores de credibilidad para las imágenes predichas de forma incorrecta; CP (SVMi_CP Uno-Contra-Uno)

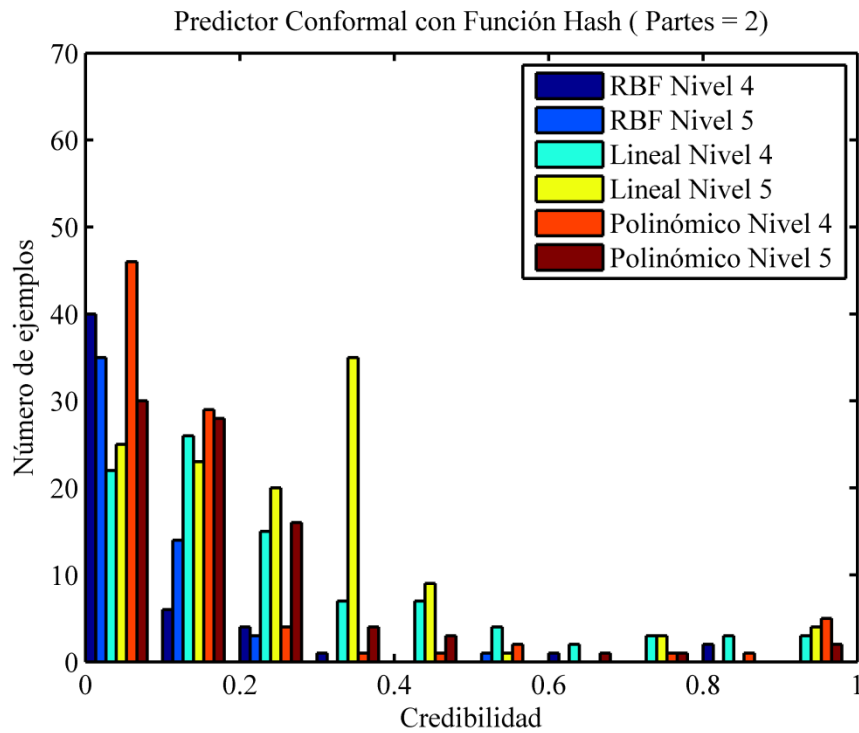


Figura 48: Valores de credibilidad para las imágenes predichas de forma incorrecta; Hashed CP (partes = 2)

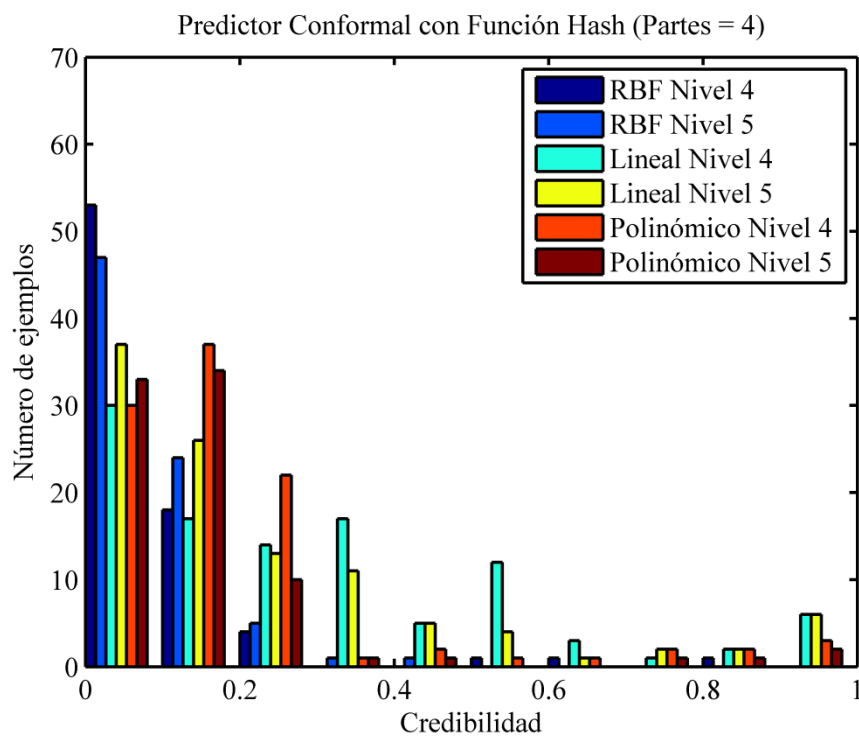


Figura 49: Valores de credibilidad para las imágenes predichas de forma incorrecta; Hashed CP (partes = 4)

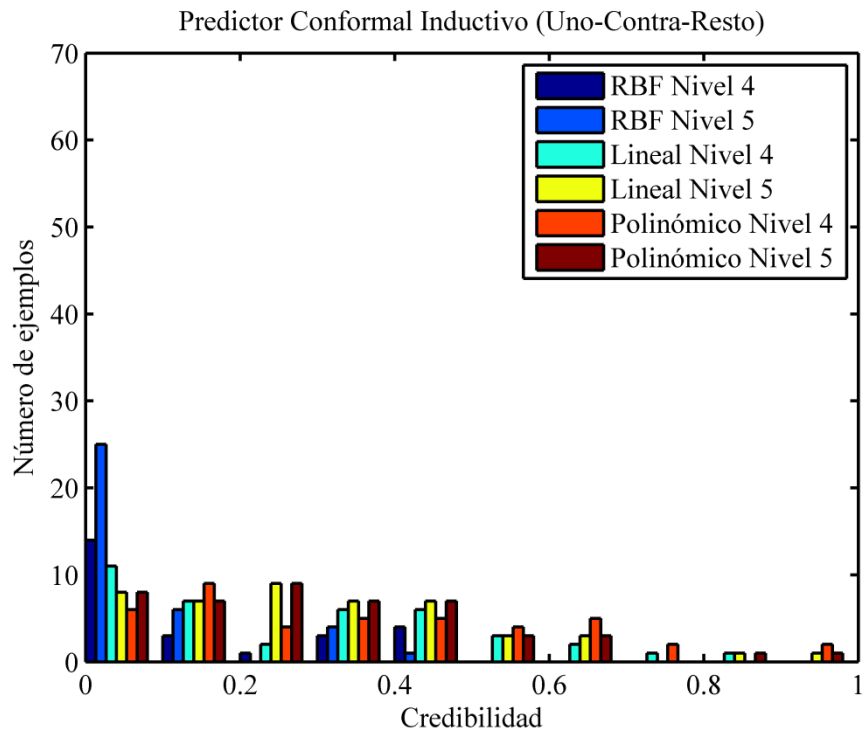


Figura 50: Valores de credibilidad para las imágenes predichas de forma incorrecta; ICP (Uno-Contra-Resto)

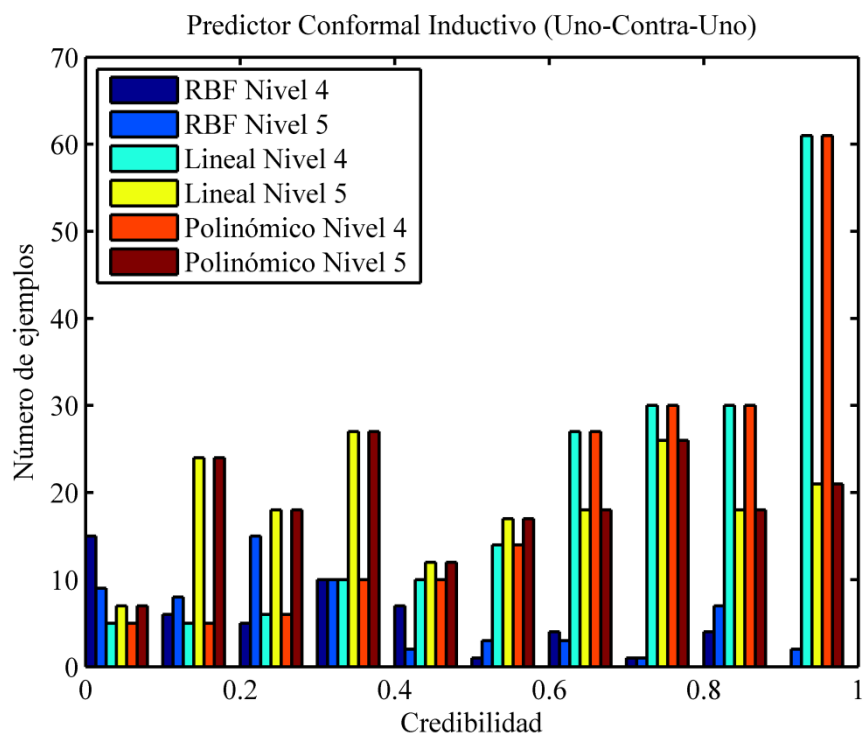


Figura 51: Valores de credibilidad para las imágenes predichas de forma incorrecta; ICP (Uno-Contra-Uno)

Métodos	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
1	12	14	9	4	0	0	0	1	1	1
2	31	7	0	1	0	0	0	0	1	1
3	2	41	0	1	0	1	0	2	0	0
4	41	2	3	0	1	2	0	1	0	1
5	9	14	11	6	0	0	1	2	0	0
6	24	4	2	1	1	0	0	1	0	0
7	22	26	15	7	7	4	2	3	3	3
8	30	17	14	17	5	12	3	1	2	6
9	11	7	2	6	6	3	2	1	1	0
10	5	5	6	10	10	14	27	30	30	61

Tabla 30: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; *kernel* lineal, nivel 4 de descomposición

Métodos	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
1	18	25	15	0	0	1	0	0	0	1
2	41	1	0	0	0	1	1	3	1	0
3	33	10	3	0	2	3	2	5	4	3
4	68	0	3	2	0	0	3	1	0	0
5	30	24	9	0	0	0	0	0	0	2
6	41	2	1	0	0	1	1	2	3	0
7	25	23	20	35	9	1	0	3	0	4
8	37	26	13	11	5	4	1	2	2	6
9	8	7	9	7	7	3	3	0	1	1
10	7	24	18	27	12	17	18	26	18	21

Tabla 31: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; *kernel* lineal, nivel 5 de descomposición

Métodos	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
1	3	16	11	4	0	0	0	1	1	1
2	21	4	1	0	0	1	0	0	0	1
3	2	41	0	1	0	1	0	2	0	0
4	41	2	3	0	1	2	0	1	0	1
5	9	14	11	6	0	0	1	2	0	0
6	24	4	2	1	1	1	1	0	0	0
7	46	29	4	1	1	2	0	1	1	5
8	30	37	22	1	2	1	1	2	2	3
9	6	9	4	5	5	4	5	2	0	2
10	5	5	6	10	10	14	27	30	30	61

Tabla 32: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; *kernel* polinómico, nivel 4 de descomposición

Métodos	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
1	9	28	18	0	0	1	0	0	0	1
2	40	1	2	0	1	2	0	1	0	3
3	33	10	3	0	2	3	2	5	4	3
4	68	0	3	2	0	0	3	1	0	0
5	22	31	10	0	0	0	0	0	0	2
6	40	0	1	2	1	0	1	1	0	1
7	30	28	16	4	3	0	1	1	0	2
8	33	34	10	1	1	0	0	1	1	2
9	8	7	9	7	7	3	3	0	1	1
10	7	24	18	27	12	17	18	26	18	21

Tabla 33: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; *kernel* polinómico, nivel 5 de descomposición

Métodos	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
1	7	16	1	1	0	1	0	1	0	0
2	42	6	2	0	0	1	0	0	0	0
3	9	17	0	1	0	0	0	2	0	0
4	29	1	0	0	0	0	1	0	0	0
5	8	11	5	2	1	0	1	4	0	1
6	29	1	0	0	0	0	0	0	0	0
7	40	6	4	1	0	0	1	0	2	0
8	53	18	4	0	0	1	1	0	1	0
9	14	3	1	3	4	0	0	0	0	0
10	15	6	5	10	7	1	4	1	4	0

Tabla 34: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; *kernel* RBF, nivel 4 de descomposición

Métodos	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
1	33	0	1	0	1	1	1	1	1	3
2	49	0	1	0	0	1	0	0	1	2
3	17	10	0	0	0	1	0	0	0	1
4	38	1	0	3	0	0	1	1	0	0
5	19	4	1	0	3	4	1	2	1	0
6	30	1	0	0	0	0	0	0	0	0
7	35	14	3	0	0	1	0	0	0	0
8	47	24	5	1	1	0	0	0	0	0
9	25	6	0	4	1	0	0	0	0	0
10	9	8	15	10	2	3	3	1	7	2

Tabla 35: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; *kernel* RBF, nivel 5 de descomposición

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	7	16	1	1	0	1	0	1	0	0
Rbf Nivel 5	33	0	1	0	1	1	1	1	1	3
Lineal Nivel 4	12	14	9	4	0	0	0	1	1	1
Lineal Nivel 5	18	25	15	0	0	1	0	0	0	1
Polinómico Nivel 4	3	16	11	4	0	0	0	1	1	1
Polinómico Nivel 5	9	28	18	0	0	1	0	0	0	1

Tabla 36: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (*SVM* Uno-Contra-Resto)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	42	6	2	0	0	1	0	0	0	0
Rbf Nivel 5	49	0	1	0	0	1	0	0	1	2
Lineal Nivel 4	31	7	0	1	0	0	0	0	1	1
Lineal Nivel 5	41	1	0	0	0	1	1	3	1	0
Polinómico Nivel 4	21	4	1	0	0	1	0	0	0	1
Polinómico Nivel 5	40	1	2	0	1	2	0	1	0	3

Tabla 37: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (*SVM* Uno-Contra-Uno)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	9	17	0	1	0	0	0	2	0	0
Rbf Nivel 5	17	10	0	0	0	1	0	0	0	1
Lineal Nivel 4	2	41	0	1	0	1	0	2	0	0
Lineal Nivel 5	33	10	3	0	2	3	2	5	4	3
Polinómico Nivel 4	2	41	0	1	0	1	0	2	0	0
Polinómico Nivel 5	33	10	3	0	2	3	2	5	4	3

Tabla 38: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (SVMi_SLS Uno-Contra-Resto)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	29	1	0	0	0	0	1	0	0	0
Rbf Nivel 5	38	1	0	3	0	0	1	1	0	0
Lineal Nivel 4	41	2	3	0	1	2	0	1	0	1
Lineal Nivel 5	68	0	3	2	0	0	3	1	0	0
Polinómico Nivel 4	41	2	3	0	1	2	0	1	0	1
Polinómico Nivel 5	68	0	3	2	0	0	3	1	0	0

Tabla 39: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (SVMi_SLS Uno-Contra-Uno)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	8	11	5	2	1	0	1	4	0	1
Rbf Nivel 5	19	4	1	0	3	4	1	2	1	0
Lineal Nivel 4	9	14	11	6	0	0	1	2	0	0
Lineal Nivel 5	30	24	9	0	0	0	0	0	0	2
Polinómico Nivel 4	9	14	11	6	0	0	1	2	0	0
Polinómico Nivel 5	22	31	10	0	0	0	0	0	0	2

Tabla 40: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (SVMi_CP Uno-Contra-Resto)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	29	1	0	0	0	0	0	0	0	0
Rbf Nivel 5	30	1	0	0	0	0	0	0	0	0
Lineal Nivel 4	24	4	2	1	1	0	0	1	0	0
Lineal Nivel 5	41	2	1	0	0	1	1	2	3	0
Polinómico Nivel 4	24	4	2	1	1	1	1	0	0	0
Polinómico Nivel 5	40	0	1	2	1	0	1	1	0	1

Tabla 41: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; CP (SVMi_CP Uno-Contra-Uno)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	40	6	4	1	0	0	1	0	2	0
Rbf Nivel 5	35	14	3	0	0	1	0	0	0	0
Lineal Nivel 4	22	26	15	7	7	4	2	3	3	3
Lineal Nivel 5	25	23	20	35	9	1	0	3	0	4
Polinómico Nivel 4	46	29	4	1	1	2	0	1	1	5
Polinómico Nivel 5	30	28	16	4	3	0	1	1	0	2

Tabla 42: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; Hashed CP (partes = 2)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	53	18	4	0	0	1	1	0	1	0
Rbf Nivel 5	47	24	5	1	1	0	0	0	0	0
Lineal Nivel 4	30	17	14	17	5	12	3	1	2	6
Lineal Nivel 4	37	26	13	11	5	4	1	2	2	6
Polinómico Nivel 4	30	37	22	1	2	1	1	2	2	3
Polinómico Nivel 5	33	34	10	1	1	0	0	1	1	2

Tabla 43: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; Hashed CP (partes = 4)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	14	3	1	3	4	0	0	0	0	0
Rbf Nivel 5	25	6	0	4	1	0	0	0	0	0
Lineal Nivel 4	11	7	2	6	6	3	2	1	1	0
Lineal Nivel 5	8	7	9	7	7	3	3	0	1	1
Polinómico Nivel 4	6	9	4	5	5	4	5	2	0	2
Polinómico Nivel 5	8	7	9	7	7	3	3	0	1	1

Tabla 44: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; ICP (Uno-Contra-Resto)

Kernel/Nivel	Intervalos de Credibilidad									
	0-0,1	0,1-0,2	0,2-0,3	0,3-0,4	0,4-0,5	0,5-0,6	0,6-0,7	0,7-0,8	0,8-0,9	0,9-1
Rbf Nivel 4	15	6	5	10	7	1	4	1	4	0
Rbf Nivel 5	9	8	15	10	2	3	3	1	7	2
Lineal Nivel 4	5	5	6	10	10	14	27	30	30	61
Lineal Nivel 5	7	24	18	27	12	17	18	26	18	21
Polinómico Nivel 4	5	5	6	10	10	14	27	30	30	61
Polinómico Nivel 5	7	24	18	27	12	17	18	26	18	21

Tabla 45: Distribución de frecuencias de la credibilidad de las predicciones incorrectas; ICP (Uno-Contra-Uno)

Muestras	Métodos																			
	1		2		3		4		5		6		7		8		9		10	
	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Est	Med	Err Estd(10 ⁻³)	Med	Err Estd(10 ⁻³)
100	0,2936	0,0012	0,3934	0,0025	0,3361	0,0017	0,6225	0,0028	0,4490	0,0042	0,7088	0,0042	0,1437	0,0010	0,1435	0,0010	0,0012	0,0441	0,0010	0,0170
200	0,5332	0,0054	0,5002	0,0012	0,3659	0,0036	0,6397	0,0042	0,6463	0,0087	0,8603	0,0047	0,2411	0,0025	0,2501	0,0016	0,0013	0,0369	0,0013	0,0341
300	0,8585	0,0085	0,6432	0,0037	0,4112	0,0050	0,6352	0,0022	0,8622	0,0169	0,9667	0,0104	0,3722	0,0031	0,4008	0,0032	0,0014	0,0553	0,0014	0,0337
400	1,3874	0,0147	0,8128	0,0060	0,4596	0,0073	0,6559	0,0055	1,0887	0,0179	1,0935	0,0186	0,5660	0,0044	0,5999	0,0084	0,0019	0,0306	0,0019	0,0389
500	1,8908	0,0290	0,9631	0,0108	0,5087	0,0131	0,6601	0,0052	1,3211	0,0216	1,2391	0,0229	0,7474	0,0091	0,8069	0,0125	0,0022	0,0202	0,0020	0,0575
600	2,4172	0,0380	1,2151	0,0136	0,5501	0,0158	0,6738	0,0037	1,5164	0,0095	1,3384	0,0179	0,9570	0,0127	1,0337	0,0122	0,0025	0,0486	0,0023	0,0667
700	3,0130	0,0417	1,4614	0,0194	0,5976	0,0126	0,7014	0,0050	1,8335	0,0187	1,5166	0,0220	1,1925	0,0119	1,3064	0,0126	0,0030	0,0469	0,0027	0,0902
800	3,7295	0,0385	1,6990	0,0173	0,6201	0,0105	0,6508	0,0059	2,1784	0,0273	1,7307	0,0376	1,4427	0,0155	1,6000	0,0113	0,0036	0,1170	0,0035	0,2479
900	4,2992	0,0466	1,8947	0,0170	0,6836	0,0116	0,7148	0,0054	2,5039	0,0190	1,8558	0,0261	1,7033	0,0146	1,9033	0,0178				
1000	5,2138	0,0247	2,1778	0,0143	0,7650	0,0095	0,7184	0,0085	2,8532	0,0168	2,0163	0,0310	1,9724	0,0074	2,1832	0,0162				

Tabla 46: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* lineal, nivel 4 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador

Muestras	Métodos																			
	1		2		3		4		5		6		7		8		9		10	
	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Est	Med	Err Estd(10 ⁻³)	Med	Err Estd(10 ⁻³)
100	0,2971	0,0017	0,3878	0,0014	0,3425	0,0023	0,6441	0,0052	0,4522	0,0041	0,6946	0,0071	0,1543	0,0079	0,1506	0,0008	0,0010	0,0233	0,0011	0,0178
200	0,5315	0,0095	0,5057	0,0044	0,3720	0,0037	0,6460	0,0062	0,6181	0,0070	0,8375	0,0127	0,2369	0,0022	0,2553	0,0013	0,0013	0,0409	0,0013	0,0256
300	0,8987	0,0248	0,6392	0,0020	0,4205	0,0044	0,6405	0,0029	0,8230	0,0140	0,9344	0,0114	0,3709	0,0073	0,4068	0,0057	0,0015	0,0515	0,0015	0,0364
400	1,3340	0,0151	0,7932	0,0037	0,4562	0,0044	0,6689	0,0032	1,0633	0,0186	1,0913	0,0204	0,5579	0,0072	0,6048	0,0069	0,0019	0,0487	0,0018	0,0282
500	1,8103	0,0343	0,9868	0,0117	0,4988	0,0073	0,6758	0,0038	1,3024	0,0161	1,2587	0,0216	0,7646	0,0111	0,8474	0,0128	0,0024	0,0615	0,0021	0,0313
600	2,3875	0,0253	1,1822	0,0127	0,5675	0,0116	0,6813	0,0044	1,5388	0,0213	1,4147	0,0201	0,9746	0,0101	1,0673	0,0139	0,0027	0,0877	0,0025	0,0431
700	3,0613	0,0336	1,4497	0,0147	0,6059	0,0129	0,6997	0,0048	1,8392	0,0195	1,5547	0,0356	1,1924	0,0161	1,3528	0,0095	0,0031	0,0784	0,0028	0,0876
800	3,6723	0,0322	1,6801	0,0173	0,6096	0,0112	0,6682	0,0060	1,7347	0,0351	1,7133	0,0271	1,4590	0,0257	1,6332	0,0078	0,0037	0,1582	0,0032	0,2106
900	4,1983	0,0353	1,9590	0,0100	0,6624	0,0113	0,6816	0,0048	2,4353	0,0262	1,8166	0,0235	1,7674	0,0327	1,8945	0,0136				
1000	5,1186	0,0376	2,1919	0,0200	0,7151	0,0078	0,6905	0,0096	2,8275	0,0194	1,9574	0,0145	2,0467	0,0263	2,2320	0,0166				

Tabla 47: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* polinómico, nivel 4 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador

Muestras	Métodos																			
	1		2		3		4		5		6		7		8		9		10	
	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Est	Med	Err Estd(10 ⁻³)	Med	Err Estd(10 ⁻³)
100	0,2935	0,0021	0,3927	0,0009	0,3241	0,0013	0,6067	0,0051	0,4432	0,0029	0,6630	0,0027	0,168	0,0045	0,1551	0,0011	0,0011	0,0114	0,0012	0,0123
200	0,5064	0,0029	0,4938	0,0020	0,3626	0,0033	0,6102	0,0074	0,6743	0,0077	0,8294	0,0053	0,2909	0,0032	0,2557	0,0019	0,0014	0,0393	0,0014	0,0233
300	0,8019	0,0081	0,6291	0,0032	0,3838	0,0044	0,6731	0,0198	0,9384	0,0213	0,9963	0,0079	0,4463	0,0051	0,3965	0,003	0,0018	0,0206	0,0017	0,0563
400	1,2808	0,0158	0,8047	0,0059	0,4159	0,0040	0,6419	0,0042	1,2063	0,0221	1,1396	0,0140	0,6576	0,006	0,5931	0,0113	0,002	0,054	0,0021	0,036
500	1,6819	0,0217	0,9869	0,0061	0,4536	0,0091	0,6916	0,0107	1,4837	0,0172	1,3415	0,0161	0,8981	0,0095	0,7701	0,0112	0,0024	0,0688	0,0023	0,0495
600	2,2235	0,0255	1,2000	0,0069	0,5022	0,0063	0,7073	0,0070	1,9777	0,0264	1,5514	0,0223	1,1234	0,0124	0,9983	0,013	0,0028	0,049	0,0025	0,092
700	2,6824	0,0297	1,4158	0,0102	0,5418	0,0064	0,6583	0,0032	2,2394	0,0397	1,7069	0,0232	1,3858	0,0117	1,2228	0,0162	0,0033	0,0495	0,003	0,0546
800	3,3266	0,0500	1,7720	0,0133	0,5575	0,0092	0,0132	0,0042	2,7325	0,0653	2,0210	0,0377	1,6687	0,0145	1,4755	0,0154	0,004	0,1009	0,0035	0,1171
900	3,8188	0,0373	2,0139	0,0132	0,6367	0,0081	0,7153	0,0038	3,1342	0,0216	2,2039	0,0392	1,9374	0,0159	1,735	0,0146				
1000	4,5604	0,0327	2,3007	0,0402	0,6559	0,0058	0,7239	0,0032	3,7102	0,1019	2,5045	0,0298	2,2785	0,0241	2,0492	0,0381				

Tabla 48: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* RBF, nivel 4 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador

Muestras	Métodos																			
	1		2		3		4		5		6		7		8		9		10	
	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Est	Med	Err Estd(10 ⁻³)	Med	Err Estd(10 ⁻³)
100	0,1870	0,0009	0,3328	0,0007	0,2981	0,0004	0,5436	0,0039	0,3867	0,0050	0,6384	0,0037	0,0971	0,0004	0,0994	0,0011	0,0008	0,0111	0,0008	0,0100
200	0,2324	0,0015	0,3568	0,0017	0,3056	0,0017	0,5409	0,0014	0,4669	0,0088	0,7262	0,0070	0,1189	0,0013	0,1211	0,0010	0,0008	0,0173	0,0009	0,0132
300	0,3045	0,0033	0,3801	0,0040	0,3259	0,0033	0,6188	0,0123	0,5649	0,0072	0,8211	0,0051	0,1462	0,0012	0,1588	0,0021	0,0009	0,0439	0,0009	0,0183
400	0,3875	0,0024	0,4125	0,0017	0,3376	0,0043	0,6686	0,0060	0,6895	0,0131	0,8412	0,0137	0,1822	0,0012	0,1950	0,0021	0,0010	0,0505	0,0010	0,0043
500	0,4454	0,0068	0,4421	0,0020	0,3379	0,0052	0,6084	0,0137	0,7915	0,0181	0,9266	0,0092	0,2208	0,0030	0,2465	0,0024	0,0011	0,0593	0,0010	0,0303
600	0,5616	0,0067	0,4821	0,0032	0,3550	0,0078	0,5326	0,0019	0,9311	0,0119	0,9324	0,0150	0,2772	0,0032	0,3032	0,0048	0,0012	0,0518	0,0011	0,0268
700	0,7057	0,0099	0,5197	0,0017	0,3765	0,0060	0,6367	0,0089	1,0381	0,0139	0,9643	0,0168	0,3351	0,0082	0,3645	0,0050	0,0012	0,1015	0,0012	0,0691
800	0,8984	0,0120	0,5523	0,0033	0,3973	0,0079	0,6387	0,0140	1,2661	0,0192	1,0825	0,0219	0,3959	0,0026	0,4369	0,0065	0,0016	0,0541	0,0017	0,1629
900	1,0954	0,0148	0,6118	0,0057	0,4105	0,0058	0,5566	0,0028	1,4501	0,0132	1,1262	0,0135	0,4527	0,0045	0,5084	0,0050				
1000	1,2972	0,0183	0,6475	0,0069	0,4008	0,0046	0,5633	0,0011	1,5293	0,0200	1,2206	0,0128	0,5307	0,0048	0,6024	0,0066				

Tabla 49: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* lineal, nivel 5 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador

Muestras	Métodos																			
	1		2		3		4		5		6		7		8		9		10	
	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Est	Med	Err Estd(10 ⁻³)	Med	Err Estd(10 ⁻³)
100	0,1590	0,0010	0,3332	0,0012	0,2457	0,0010	0,5605	0,0025	0,3513	0,0037	0,6513	0,0069	0,0977	0,0012	0,0944	0,0002	0,0008	0,0051	0,0008	0,0168
200	0,2028	0,0011	0,3620	0,0025	0,2585	0,0012	0,5404	0,0011	0,4327	0,0069	0,7268	0,0113	0,1158	0,0009	0,1153	0,0015	0,0009	0,0296	0,0008	0,0156
300	0,2793	0,0041	0,3436	0,0337	0,2684	0,0025	0,5490	0,0014	0,5179	0,0057	0,8584	0,0245	0,1454	0,0011	0,1456	0,0012	0,0009	0,0176	0,0010	0,0162
400	0,3490	0,0035	0,3998	0,0017	0,2861	0,0047	0,5627	0,0015	0,6007	0,0107	0,8622	0,0117	0,1791	0,0020	0,1794	0,0017	0,0011	0,0194	0,0009	0,0174
500	0,4401	0,0081	0,4421	0,0025	0,2993	0,0039	0,5562	0,0020	0,7103	0,0146	0,8872	0,0160	0,2243	0,0068	0,2232	0,0034	0,0012	0,0135	0,0011	0,0446
600	0,5569	0,0082	0,4747	0,0016	0,3126	0,0038	0,5570	0,0030	0,8229	0,0136	0,9607	0,0160	0,2673	0,0034	0,2719	0,0023	0,0013	0,0428	0,0012	0,0366
700	0,6765	0,0110	0,5171	0,0021	0,3468	0,0108	0,5563	0,0032	0,9597	0,0164	1,0207	0,0176	0,3224	0,0050	0,3323	0,0050	0,0015	0,1187	0,0014	0,0307
800	0,7759	0,0116	0,5627	0,0052	0,3606	0,0099	0,5540	0,0048	1,1132	0,0152	1,0334	0,0181	0,3802	0,0048	0,3977	0,0030	0,0016	0,1217	0,0017	0,1368
900	0,9866	0,0154	0,6068	0,0023	0,3821	0,0045	0,5833	0,0203	1,2629	0,0118	1,0881	0,0160	0,4447	0,0040	0,4629	0,0040				
1000	1,1320	0,0077	0,6443	0,0083	0,3973	0,0110	0,5683	0,0024	1,3958	0,0147	1,1282	0,0151	0,5178	0,0048	0,5386	0,0034				

Tabla 50: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* polinómico, nivel 5 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador

Muestras	Métodos																			
	1		2		3		4		5		6		7		8		9		10	
	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Estd	Med	Err Est	Med	Err Estd(10 ⁻³)	Med	Err Estd(10 ⁻³)
100	0,1805	0,0006	0,3250	0,0014	0,3434	0,0045	0,6788	0,0155	0,3769	0,0062	0,6413	0,0031	0,1001	0,0015	0,1001	0,0009	0,0008	0,1111	0,0009	0,1902
200	0,2227	0,0010	0,3502	0,0007	0,3018	0,0009	0,5896	0,0012	0,4728	0,0073	0,7218	0,0077	0,1195	0,0010	0,1222	0,0006	0,0009	0,0577	0,0010	0,2270
300	0,2892	0,0021	0,3739	0,0025	0,3542	0,0101	0,6486	0,0199	0,5740	0,0102	0,7854	0,0066	0,1467	0,0007	0,1504	0,0009	0,0010	0,0367	0,0010	0,1945
400	0,3472	0,0033	0,2806	0,0011	0,2953	0,0012	0,5615	0,0006	0,6926	0,0126	0,8437	0,0178	0,1809	0,0012	0,1848	0,0011	0,0011	0,1978	0,0010	0,1392
500	0,4196	0,0036	0,4297	0,0016	0,3009	0,0016	0,5643	0,0023	0,9007	0,0233	0,9402	0,0132	0,2192	0,0027	0,2249	0,0013	0,0011	0,1899	0,0011	0,3484
600	0,5163	0,0030	0,4604	0,0013	0,3054	0,0029	0,5667	0,0014	1,1869	0,0187	1,0439	0,0234	0,2671	0,0070	0,2708	0,0023	0,0013	0,3874	0,0012	0,4568
700	0,5829	0,0087	0,5010	0,0020	0,3081	0,0008	0,5905	0,0103	1,5014	0,0361	1,2776	0,0222	0,3178	0,0035	0,3296	0,0053	0,0015	0,4498	0,0013	0,2715
800	0,7144	0,0071	0,5602	0,0042	0,3122	0,0008	0,5780	0,0079	2,0229	0,0513	1,2417	0,0277	0,3680	0,0050	0,3789	0,0022	0,0018	0,5176	0,0016	0,3823
900	0,8561	0,0081	0,6099	0,0030	0,3187	0,0014	0,5600	0,0017	2,1698	0,0622	1,4254	0,0325	0,4380	0,0078	0,4396	0,0041				
1000	1,0127	0,0087	0,6633	0,0107	0,3521	0,0069	0,5905	0,0013	2,5423	0,0715	1,4797	0,0187	0,5039	0,0096	0,5138	0,0038				

Tabla 51: Tiempo de CPU (s) requerido para la clasificación de una imagen; *kernel* RBF, nivel 5 de descomposición. Los valores tabulados constituyen la media obtenida en 10 ejecuciones de cada clasificador

ANEXO D: LIBRERÍA DE *SOFTWARE*

En este anexo se hace una breve descripción de los componentes *software* fundamentales desarrollados e incluidos en la librería. Los referidos componentes fueron escritos en Matlab, teniendo en vista su fácil reutilización, por lo que se ha hecho uso de los recursos incorporados en ese lenguaje para la implementación de la metodología de programación orientada a objetos. Como soporte de la programación se ha empleado la librería Spider (Weston, 2006), la cual permite la explotación e implementación de métodos de aprendizaje automático en Matlab, por lo que su utilización implica la instalación previa de la referida librería.

A continuación se hace la descripción de los objetos desarrollados más importantes.

Onpr: clasificador conformal *online*

$A = \text{onpr}(C, H)$ devuelve una instancia del objeto clasificador conformal *online*, basado en el algoritmo C e inicializado con los hiperparámetros H.

Hiperparámetros y sus valores por defecto:

- `smoothed = 0`: define una versión aleatorizada del clasificador
- `región_predictor = 0`: define si las salidas del clasificador son presentadas en forma de regiones de predicción o no

- $\epsilon = 0,05$: nivel de significancia
- $\text{teacher} = 0$: estrategia de tutoría empleada
 - 0 : ideal
 - 1 : lenta y perezosa
 - 2 : selectiva
- $\text{lag} = 0$: retraso introducido en el caso de tutoría lenta
- $\text{step} = 1$: paso empleado en el caso de un tutoría perezosa
- $\text{criterion} = 0$: criterio empleado en caso de tutoría selectiva
 - 0 : confianza
 - 1 : credibilidad
 - 2 : confianza * credibilidad
 - 3 : credibilidad – (1 – confianza)
- $\text{threshold} = 0,2$: umbral utilizado para la tutoría selectiva. El algoritmo debe ser “*enseñado*” si el valor del criterio es igual o menor que el umbral.

Métodos:

- `train, test`

Ejemplo:

- `O = onpr(cp(svm), {'teacher = 1' 'lag = 10'});`

Offpr: clasificador conformal *offline*

`A = offpr(C, H)` devuelve una instancia del objeto clasificador conformal *offline*, basado en el algoritmo C e inicializado con los hiperparámetros H.

Hiperparámetros y sus valores por defecto:

- `smoothed = 0`: define una versión aleatorizada del clasificador
- `región_predictor = 0`: define si las salidas del clasificador son presentadas en forma de regiones de predicción o no
- `epsilon = 0,05`: nivel de significancia

Métodos:

- `train, test`

Ejemplo:

- `O = offpr(cp(svm), {'smoothed = 1'});`

Cp: predictor conformal

`A = cp(C, H)` devuelve una instancia del objeto inicializado con el algoritmo `C` y los hiperparámetros `H`. Es un objeto virtual, que constituye el núcleo de la predicción conformal, utilizado con los objetos `onpr`, `offpr`, `one_vs_1` y `one_vs_r`. Determina los valores de no conformidad de la predicción conformal para un único vector de test.

Hiperparámetros y sus valores por defecto:

- `full_cycle = 1`: determina si utiliza un ciclo de entrenamiento completo o abreviado

Métodos:

- `train, test`

Ejemplo:

- `O = cp(svm);`

One_vs_1: objeto uno-contra-uno

$A = \text{one_vs_1}(C, H)$ devuelve una instancia del objeto uno-contra-uno, el cual permite entrenar un objeto C en varios subproblemas del tipo clase i contra clase j . Es un objeto virtual, utilizado con los objetos *onpr* y *offpr*, y calcula los valores de no conformidad para cada uno de los subproblemas.

Hiperparámetros y sus valores por defecto:

- `summary_fcn = 'mean'` define la función de sumario utilizada en el cálculo de los valores de no conformidad ('mean' o 'max')

Métodos

- `train`, `test`

Ejemplo

- `O = one_vs_1(cp);`

One_vs_r: objeto uno-contra-el-resto

$A = \text{one_vs_r}(C, H)$ devuelve una instancia del objeto uno-contra-el-resto, el cual permite entrenar un objeto C en varios subproblemas del tipo clase i contra las demás clases. Es un objeto virtual, utilizado con los objetos *onpr* y *offpr*, y calcula los valores de no conformidad para cada uno de los subproblemas.

Hiperparámetros y sus valores por defecto:

- `summary_fcn = 'mean'`: define la función de sumario utilizada en el cálculo de los valores de no conformidad ('mean' o 'max')
- `lambda = 0,8`: factor de peso utilizado para calcular la media ponderada al determinar los valores de no conformidad
- `shortcut_alpha_eval = 0`: utiliza o no una versión abreviada del cálculo de los valores de no conformidad en el caso en que `lambda = 1`

Métodos:

- `train`, `test`

Ejemplo:

- `O = one_vs_r(cp(svm), {'lambda = 0.9'});`

Hashed_cp: clasificador conformal con función *hash*

`A = hashed_cp(C, H)` devuelve un clasificador conformal con base en el algoritmo `C` e inicializado con los hiperparámetros `H`. Permite la determinación de los valores de no conformidad de un predictor conformal multiclase, utilizando el método de la función *hash*, para un único vector de test.

Hiperparámetros y sus valores por defecto:

- `splits = 2`: número de subdivisiones del conjunto de entrenamiento a emplear al utilizar el método.

Métodos:

- `train`, `test`

Ejemplo:

- `O = hashed_cp(svm, {'splits = 4'});`

Onicp: predictor conformal inductivo *online*

`A = onicp(C, H1, H2)` devuelve una instancia del objeto clasificador conformal inductivo, basado en el algoritmo `C` e inicializado con los hiperparámetros `H1` y `H2`.

Hiperparámetros y sus valores por defecto:

- `H1`
 - `smoothed = 0`: define una versión aleatorizada del clasificador

- `training_trials = [100]`: define los puntos de actualización del clasificador
- `región_predictor = 0`: define si las salidas del clasificador son presentadas en forma de regiones de predicción o no
- `epsilon = 0,05`: nivel de significancia
- H2
 - `lambda = 0.8`: parámetro utilizado en la ponderación de la media al calcular los valores de no conformidad en el caso de utilización con el método multiclase uno-contra-el-resto

Métodos:

- `train, test`

Ejemplo:

- `O = onicp(svm, {'smoothed = 1'}, {'lambda = 1'});`
- `O = onicp(one_vs_rest(svm));`
- `O = onicp(svm, {'smoothed = 1'});`

Officp: predictor conformal inductivo *offline*

`A = officp(C, H1, H2)` devuelve una instancia del objeto clasificador conformal inductivo *offline*, basado en el algoritmo `C` e inicializado con los hiperparámetros `H1` y `H2`.

Hiperparámetros y sus valores por defecto:

- H1
 - `smoothed = 0`: define una versión aleatorizada del clasificador
 - `calibDat = []`: datos de calibración

- `divideFcn = 'rand'`: función utilizada para subdividir el conjunto de entrenamiento en subconjuntos de entrenamiento propio y de calibración ('rand', 'strat' o 'index'); utilizada cuando `calibDat = []`.
- `divideParam = 0.4`: parámetro utilizado con `divideFcn`
- `semi_offline = 0`: define el modo de entrenamiento semi-offline
- `región_predicor = 0`: define si las salidas del clasificador son presentadas en forma de regiones de predicción o no
- `epsilon = 0,05`: nivel de significancia
- H2
 - `lambda = 0.8`: parámetro utilizado en la ponderación de la media al calcular los valores de no conformidad en el caso de utilización con el método multiclase uno contra el resto

Métodos:

- `train, test`

Ejemplo:

- `O = officp(svm, {'smoothed = 1'}, {'lambda = 1'});`
- `O = officp(one_vs_one(svm));`
- `O = officp(svm, {'smoothed = 1'});`

Svmi_cp: máquina de vectores soporte incremental

`A = svmi_cp(H)` devuelve una instancia de un objeto *SVM* incremental, inicializado con los hiperparámetros *H*. Implementa una *SVM* incremental “adiabática” de acuerdo al enfoque

introducido por Cawenberghs y Poggio (2001). Esta es una versión del código distribuido por Christopher P. Diehl (Diehl, 2003).

Hiperparámetros y sus valores por defecto:

- `child = kernel`: el *kernel* es guardado en este campo
- `C = 10`: parámetro de regularización; puede ser:
 - Escalar : parámetro de regularización universal
 - Vector bidimensional : parámetro de regularización condicionado por clases
 - Vector n-dimensional : parámetro de regularización por cada ejemplo (n = cantidad de ejemplos)
- `deps = 10-3`: factor de *jitter* para la matriz *kernel*
- `max_reserve_vectors = 3000`: cantidad máxima de vectores de reserva acumulados
- `incremental_mode = 0`: define si realiza el entrenamiento de una nueva máquina *SVM* o incrementa los nuevos ejemplos a una solución ya existente
 - 0 : entrena nueva máquina *SVM*
 - 1 : incrementa una solución ya existente
- `batch_mode = 0`: define si incrementa los nuevos ejemplos uno a uno o en lote al entrenar la máquina *SVM*

Métodos:

- `train, test`

Ejemplo:

- `d = gen(spiral({'m = 200', 'n = 2', 'noise = 0.35'}));`
- `[r, a] = train(svmi_cp({'kernel('rbf', 1), 'batch_mode = 1'}), d);`

Svmi_sls: máquina de vectores soporte incremental

$A = \text{svmi_sls}(H1, H2)$ devuelve una instancia del objeto *SVM* incremental, inicializado con los hiperparámetros $H1$ y $H2$. Implementa una máquina *SVM* entrenada incrementalmente de acorde al enfoque de Syed, Liu y Sung (1999a). Este es una extensión del objeto *SVM* incorporado en el Spider.

Hiperparámetros y valores por defecto:

- $H1$
 - `incremental_mode = 0`: define el modo de entrenamiento
 - `0` : entrena una nueva máquina *SVM*
 - `1` : incrementa los nuevos ejemplos a una solución existente y hace la remoción de los alfas cuyos valores sean iguales a cero
 - `2` : entrena la máquina *SVM* incrementalmente y preserva todos los alfas
 - `cutoff = 10^{-10}` : conserva valores de alfa tales que $\text{abs}(a_i) > \text{cutoff}$
- $H2$
 - Hiperparámetros heredados del objeto *SVM* incorporado en el Spider

Métodos:

- `train, test`

Ejemplo:

- `d = gen(spiral({'m = 200', 'n = 2', 'noise = 0.35'}));`
- `[r, a] = train(svmi_sls({'kernel('rbf', 1), 'batch_mode = 1'}), d);`

Data_cp: contenedor de datos utilizado por los métodos de predicción conformal

$D = \text{data_cp}(\text{'name'}, X, Y)$, $d = \text{data_cp}(X, Y)$, $d = \text{data_cp}(X)$ o $d = \text{data_cp}()$, devuelven instancias del objeto `data_cp`. Constituye una extensión del objeto `data` incorporado en la librería `Spider` (Weston, 2006). Utiliza sus distintos atributos para guardar los datos utilizados en o resultados de la predicción conformal.

Atributos:

- `Pv`: p-valores
- `cf`: confianza
- `cr`: credibilidad
- `mult`: predicciones múltiples
- `err`: predicciones incorrectas
- `emp`: predicciones vacías
- `data`: objeto `data`

Métodos:

- `get`, `get_x`, `get_y`, `get_xy`, `get_pv`, `get_cr`, `get_cf`, `get_mult`, `get_err`, `get_emp`, `set_x`, `set_y`, `region_pred`, `confident_pred`

Samplingsel: selección activa de conjuntos de entrenamiento utilizando el paradigma del muestreo selectivo

$A = \text{samplingsel}(C, H)$ devuelve una instancia del objeto inicializado con el algoritmo `C` y los hiperparámetros `H`. Selecciona las mejores muestras del conjunto de entrenamiento y entrena un modelo empleando las muestras seleccionadas.

Hiperparámetros y valores por defecto:

- `divideFcn = 'rand'`: función utilizada para la selección de los ejemplos para la inicialización del algoritmo ('rand', 'strat', 'index' o [])
- `divideParam = 50`: parámetro utilizado con la función anterior
- `init_sel_dat`: vector de tipo lógico con un elemento por cada punto de datos, conteniendo unos para los puntos seleccionados para el conjunto de inicialización y ceros en caso contrario
- `mode = 'pool'`: define el modo de selección, 'pool', 'stream' o 'mixed'
- `block_marker = []`: vector utilizado para agrupar los datos en diferentes bloques; utilizado en el modo 'mixed'
- `batch = 50`: cantidad máxima de vectores a incluir en el lote de datos a seleccionar en el modo 'pool'
- `representativeness = 0`: activa/desactiva el criterio de representatividad
- `diversity = 0`: activa/desactiva el criterio de diversidad
- `criteria = 1`: criterio de información utilizado para la selección de las muestras; puede ser utilizado como un valor escalar o un vector para la combinación de varios criterios.
 - 1 : confianza
 - 2 : credibilidad
 - 3 : confianza * credibilidad
 - 4 : credibilidad – (1 – confianza) (QBT)
 - 5 : GQBT; mayor valor propio de la matriz $C(i, j) = |p_i - p_j|$, donde p_i y p_j son los p -valores

- $\text{threshold} = 0,2$: umbral utilizado en la selección de las muestras; muestras con un valor de criterio menor o igual que el umbral son seleccionadas. Puede ser un valor escalar o un vector con dos valores $[\text{min_tr } \text{max_tr}]$ para la implementación de un umbral adaptativo
- $\text{sigma_d} = 500$: parámetro de escala utilizado al determinar la semejanza entre vectores al calcular el criterio de diversidad; $\text{sim}(v1, v2) = \exp(-d(v1, v2)/\text{sigma_d})$
- $\text{sigma_r} = 2 \cdot 10^3$: parámetro de escala utilizado al determinar la semejanza entre vectores al calcular el criterio de representatividad
- $\text{lbda_i} = 0,5$: parámetro utilizado en la determinación de los pesos al combinar varios criterios de información
- $\text{lbda_rd} = 0,5$: parámetro utilizado en la determinación de los pesos al combinar los criterios de diversidad y representatividad
- $\text{lbda_ird} = 0,5$: parámetro utilizado en la determinación de los pesos al combinar los criterios de información y diversidad/representatividad
- $\text{mode_comb} = 1$: define la manera de combinar a los criterios
 - 1 : marcador (*scores*)
 - 2 : rango (*rank*)
- $\text{scheme_comb} = 1$: esquema utilizado al determinar los pesos para la combinación de los criterios
 - 1 : $w1 = \text{lbda}, w2 = 1 - \text{lbda}$
 - 2 : esquema de Akaike, $w_j = \exp(-\text{lbda} * 0.5 * \text{delta}_j) / \sum(\exp(-\text{lbda} * 0.5 * \text{delta}_r))$
- $\text{dist} = \text{distance}$: medida utilizada al calcular la distancia entre vectores

- `clust = kmeans`: método utilizado para la determinación de los centroides representativos del *pool* de datos para la implementación del criterio de representatividad
- `num_samples = 500`: cantidad máxima de ejemplos a incluir en el conjunto seleccionado
- `max_rej_samples = 100`: cantidad máxima de rechazos consecutivos de muestras permitida
- `k1 = 2`: cantidad permitida de *max_rej_samples*, o selecciones vacías, alcanzadas antes de abandonar el ciclo de selección; si no se ha alcanzado *k1*, el umbral de selección es actualizado, en lugar de abandonar el ciclo, si el umbral es dado en la forma `[min_tr max_tr]`
- `calibration = 0`: define la utilización o no de la funcionalidad de calibración
- `calib_freq = 10`: frecuencia (en ciclos) del proceso de calibración; utilizado en los modos 'stream' y 'mixed'
- `calib_dat = 1`: define los datos a ser utilizados en la calibración
 - 1 : conjunto *hold-out* separado
 - 2 : lote separado del *pool* de datos
 - 3 : *pool* de datos completo
- `calib_meth = [1 0 0]`: métodos utilizados para efectos de calibración, dado por un vector de unos y ceros (activado/desactivado), [error confianza cambio-clasificación].
- `calib_comb = 1`: forma de combinación de varios métodos de calibración
 - 1 : primero; la selección termina si una de las condiciones es satisfecha
 - 2 : todos; la selección termina si todas las condiciones son satisfechas

- `divideCalibFcn = 'rand'`: función utilizada para la selección del conjunto *hold-out* de calibración, cuando sea aplicable; ('rand', 'strat', 'index' o [])
- `divideCalibParam = 100`: parámetro a ser utilizado con *divideCalibFcn*
- `calib_sel_dat = []`: vector de tipo lógico con un elemento por cada punto de datos, conteniendo unos para los puntos seleccionados para el conjunto de calibración y ceros en caso contrario.
- `stop_condition = 0`: activa/desactiva la condición de parada temprana
- `k2 = 2`: cantidad permitida de iteraciones consecutivas sin mejorar el criterio de parada temprana
- `epsln = 0,001`: umbral utilizado en la condición de parada temprana; el algoritmo se detiene si durante $k2$ ciclos consecutivos, $i - k2, \dots, i$:
 - Error: $c(i) - c(i - 1) \geq -epsln$
 - Confianza: $c(i) - c(i - 1) \leq epsln$
 - Cambio clasificación: $c(i) \leq epsln$
- `hybrid = 0`: activa/desactiva selección híbrida

Métodos:

- `train, test`

Ejemplos:

- `O = one_vs_r(cp(svm));`
- `[r, a] = train(samplingsel(o, {'batch' 'num_samples'}, {20 200}), gen(toy)) ;`

Gridcfsel: selección de modelo en malla con un enfoque de refinamiento de la búsqueda derivado del campo del *Diseño de Experimentos* (Staelin, 2002)

`S = gridcfsel(A, P, V, H)` devuelve una instancia del objeto inicializado con el algoritmo `A`, hiperparámetros `H` y un espacio de búsqueda compuesto por los parámetros `P` delimitados por los valores `V`. Determina los mejores parámetros en el espacio de búsqueda y entrena y guarda el correspondiente modelo.

Hiperparámetros y valores por defecto:

- `loss = 'class_loss'`: función de pérdida a utilizar
- `score = cv('folds = 5')`: método de evaluación de los algoritmos
- `param = []`: parámetros a ser evaluados
- `values = []`: definición de los límites del espacio de búsqueda [min max]
- `design = 'mix'`: diseño experimental
 - '3d' : 3 niveles estándar (-1, 0, 1)
 - '2d' : 2 niveles estándar (-1/2, 1/2)
 - 'mix' : combinación de los dos anteriores
- `epochs = 10`: cantidad máxima de iteraciones
- `scale = 'log2'`: escala utilizada al definir los rangos de valores de los parámetros ('log2' o 'linear')
- `stop_condition = 0`: activa/desactiva condición de parada temprana; si en k iteraciones consecutivas no se alcanza un estado que es mejor que el actual en al menos $epsln$, la búsqueda se detiene
- `k = 3`: cantidad permitida de iteraciones consecutivas sin mejorar al mejor valor ya alcanzado
- `epsln = 0,001`: valor de $epsln$ utilizado al testar la condición de parada

Métodos:

- train, test

Ejemplo:

- `[r, a] = train(gridcfsel(svm, 'C', [1 10], {'score = cv; score.folds = 3'}), gen(toy));`
- `[r, a] = train(gridcfsel(svm, {'C', 'kerparam'}, {[1 10], [8 64]}), gen(toy));`

Hcwrapsel: selección de modelo a través de un *wrapper* (Kohavi, 1995b) utilizando como componente de búsqueda un método de la familia *hill climbing* (Russel, 2003)

`S = hcwrapsel(A, P, V, H)` devuelve una instancia del objeto inicializado con el algoritmo *A*, hiperparámetros *H* y un espacio de búsqueda compuesto por los parámetros *P*, delimitados por los valores *V*. Determina los mejores parámetros en el espacio de búsqueda y entrena y guarda el correspondiente modelo.

Hiperparámetros y valores por defecto:

- `loss = 'class_loss'`: función de pérdida a utilizar
- `score = cv('folds = 5')`: método de evaluación de los algoritmos
- `param = []`: parámetros a ser evaluados
- `values = []`: definición del espacio de búsqueda [min max]
- `scale = 'log2'`: escala utilizada al definir los rangos de valores de los parámetros ('log2' o 'linear')
- `alfa_i = 0.1`: parámetro utilizado en el cálculo del tamaño del rectángulo de vecindad interno, $hi = alfa_i * (max - min)/2$
- `alfa_e = 0.1`: parámetro utilizado en el cálculo del tamaño del rectángulo de vecindad externo, $he = alfa_e * (max - min)/2$

- $\text{alfa_p} = 0.1$: coeficiente utilizado para el cálculo del tamaño de la vecindad hiperrectangular utilizada en el cálculo de la proximidad entre dos puntos, $hp = \text{alfa_p} * (\text{max} - \text{min})/2$
- $\text{neighb} = 3$: número de vecinos a utilizar en cada expansión
- $\text{partitioning} = \text{'geom'}$: método de división del espacio alrededor del punto a expandir ('geom' , 'linear')
- $\text{restarts} = 5$: número de reinicios aleatorios
- $\text{epochs} = 50$: número máximo de evaluaciones de la función objetivo
- $\text{epsln} = 0.001$: parámetro de precisión para medición de la evolución del objetivo
- $\text{goal} = 0$: valor objetivo de la minimización

Métodos:

- train , test

Ejemplo:

- $\text{o} = \text{offpr}(\text{one_vs_r}(\text{cp}(\text{svm})));$
- $[\text{r}, \text{a}] = \text{train}(\text{hcwrapsel}(\text{o}, \{\text{'C' 'kerparam'}\}, \{[5 \ 18] [5 \ 15]\}), \text{gen}(\text{toy})) ;$

Sawrapsel: selección de modelo a través de un *wrapper* (Kohavi, 1995b) utilizando como componente de búsqueda el método de recocido simulado (en inglés *simulated annealing*) (Kirkpatrick, 1983)

$S = \text{sawrapsel}(A, P, V, H)$ devuelve una instancia del objeto inicializado con el algoritmo A , los hiperparámetros H y un espacio de búsqueda compuesto por los parámetros P , delimitados por los valores V . Determina los mejores parámetros en el espacio de búsqueda y entrena y guarda el correspondiente modelo. Utiliza para el recocido simulado el algoritmo *Adaptive Simulated Annealing* (ASA) distribuido por Lester Ingber (Ingber, 2011).

Hiperparámetros y valores por defecto:

- `loss = 'class_loss'`: función de pérdida a utilizar
- `score = cv('folds = 5')`: método de evaluación de los algoritmos
- `param = []`: parámetros a ser evaluados
- `values = []`: definición del espacio de búsqueda [min max]
- `scale = 'log2'`: escala utilizada al definir los rangos de valores de los parámetros ('log2' o 'linear')
- `rand_seed = []`: raíz del generador de números aleatorios
- `asa_out_file = []`: archivo para la impresión de los resultados de la simulación
- `limit_acceptances = 50`: número máximo de estados aceptados antes del término de la simulación
- `limit_generated = 99999`: número máximo de estados generados antes del término de la simulación
- `limit_invalid = 1000`: límite permitido de estados inválidos consecutivos
- `accepted_to_generated_ratio = 10e-4`: menor ratio de estados aceptados por generados
- `cost_precision = 0,001`: parámetro de precisión de la función de coste
- `maximum_cost_repeat = 3`: cantidad permitida de repeticiones de la función de coste
- `number_cost_samples = 5`: cantidad de valores de la función de coste utilizados para el cálculo de la temperatura inicial
- `temperatura_ratio_scale = 10-5`: guía para el valor esperado del coste de la temperatura de convergencia

- `cost_parameter_scale_ratio = 1,0`: ratio entre las escalas de coste y temperatura de recocido
- `temperature_anneal_scale = 100,0`: guía para el alcance de la temperatura coste buscada por *temperature_ratio_scale* dentro de los límites determinados por *limit_acceptances*
- `include_integer_parameters = 0`: si ‘verdadero’, ASA incluye parámetros enteros en los cálculos de derivadas y recocido
- `user_initial_parameters = 0`: si ‘verdadero’, el primero estado es calculado a partir de la primera conjetura hecha por el usuario
- `sequential_parameters = -1`: define el método de generación de nuevas muestras
- `initial_parameter_temperature = 1,0`: temperatura inicial para todos los parámetros
- `acceptance_frequency_modulus = 10`: frecuencia de test para el test periódico y recocido, dependiente del número de estados aceptados
- `generated_frequency_modulus = 10000`: frecuencia de test para el test periódico y recocido, dependiente del número de estados generados
- `reannel_cost = 1`: controla el proceso de recocido
 - 0 : evita el recocido de la temperatura de coste
 - 1 : permite que el recocido por defecto de la temperatura de coste sea parte de del proceso de ajuste
 - > 1 : el recocido inicial de la temperatura de coste es calculado como la desviación sobre un conjunto de `reannel_cost` funciones de coste

- < 1 : el índice de coste es reinicializado a 1 y las temperaturas de coste inicial y actual son calculados como la desviación sobre un conjunto de `reannel_cost` funciones de coste
- `reannel_parameters = 1`: permite que el recocido de los parámetros de temperatura forme parte del proceso de ajuste
- `delta_x = 0,001`: incremento fraccional de los parámetros utilizados para la determinación numérica de las derivadas al calcular las tangentes para recocido, para cada parámetro a ser recocido
- `asa_pipe_file = 0`: si 1, ASA imprime en el archivo `asa_pipe` líneas de datos que pueden ser utilizadas para el examen de la ejecución

Métodos:

- `train, test`

Ejemplo:

- `o = offpr(one_vs_r(cp(svm)))`;
- `[r, a] = train(sawrapssel(o, {'C' 'kerparam'}, {[5 18] [5 15]}), gen(toy))` ;

Tswrapssel: selección de modelo a través de un *wrapper* (Kohavi, 1995b) utilizando como componente de búsqueda el método de búsqueda tabú (Siarry, 1997), (Chelouah, 2000)

`S = sawrapssel(A, P, V, H)` devuelve una instancia del objeto inicializado con el algoritmo `A`, los hiperparámetros `H` y un espacio de búsqueda compuesto por los parámetros `P`, delimitados por los valores `V`. Determina los mejores parámetros en el espacio de búsqueda y entrena y guarda el correspondiente modelo.

Hiperparámetros y valores por defecto:

- `loss = 'class_loss'`: función de pérdida a utilizar

- `score = cv('folds = 5')`: método de evaluación de los algoritmos
- `param = []`: parámetros a ser evaluados
- `values = []`: definición del espacio de búsqueda [min max]
- `scale = 'log2'`: escala utilizada al definir los rangos de valores de los parámetros ('log2' o 'linear')
- `alfa_i = 0.1`: parámetro utilizado en el cálculo del tamaño del rectángulo de vecindad interno, $hi = alfa_i * (max - min)/2$
- `alfa_e = 0.1`: parámetro utilizado en el cálculo del tamaño del rectángulo de vecindad externo, $he = alfa_e * (max - min)/2$
- `alfa_p = 0.1`: coeficiente utilizado para el cálculo del tamaño de la vecindad hiperrectangular utilizada en el cálculo de la proximidad entre dos puntos, $hp = alfa_p * (max - min)/2$
- `neighb = 3`: número de vecinos a utilizar en cada expansión
- `partitioning = 'geom'`: método de división del espacio alrededor del punto a expandir ('geom', 'linear')
- `epochs = 50`: número máximo de evaluaciones de la función objetivo
- `epsln = 0.001`: parámetro de precisión para medición de la evolución del objetivo
- `goal = 0`: valor objetivo de la minimización
- `k = 3`: cantidad permitida de iteraciones consecutivas sin mejorar el mejor valor alcanzado
- `tabu_list_size_min = 4`: tamaño mínimo para la lista tabú
- `tabu_list_size_max = 12`: tamaño máximo para la lista tabú

- `use_surrogate = 0`: define si utilizar o no una función sustituta para la evaluación de los vecinos
- `surrogate_fcn = ho`: función sustituta a utilizar (defecto *hold-out*)

Métodos:

- `train`, `test`

Ejemplo:

- `o = offpr(one_vs_r(cp(svm)));`
- `[r, a] = train(tswrapsel(o, {'C' 'kerparam'}, {[5 18] [5 15]}), gen(toy)) ;`

ANEXO E: TUTORIAL SOBRE UTILIZACIÓN DE LA LIBRERÍA SPIDER EN UN ENTORNO DE 64 BITS

Con la evolución ocurrida en los últimos años al nivel de los recursos de computación se está tornando cada vez más vulgar la utilización de plataformas de 64 bits. Esto trae anexada la necesidad de alguna adaptación en lo que concierne al *software* de aplicación, lo que en algunos casos, y no obstante el esfuerzo emprendido por los proveedores de *software*, puede resultar en contrariedades para los utilizadores.

En el caso de la librería Spider (Weston, 2006), varias de las rutinas de optimización de máquinas *SVM* fueron escritas originariamente en lenguajes como el C o el C++ y son utilizadas como rutinas ejecutables ligadas dinámicamente al Matlab, designadas como *MEX-files* (MathWorks, 2009). Tales son los casos de las rutinas de optimización como LIBSVM (Chang, 2011), SVMlight (Joachims, 1999) o SVM Torch (Collobert, 2001), por ejemplo. Al haber sido compiladas en entornos de 32 bits, la utilización de las referidas rutinas en entornos de 64 bits supone un proceso de adaptación y creación de versiones compatibles, esfuerzo que normalmente se encuentra a cargo del usuario una vez que el soporte técnico a la librería por parte de sus autores está discontinuado.

Con esta guía se pretende reunir en un documento único un conjunto de instrucciones que permitan auxiliar a eventuales utilizadores de la librería Spider que puedan experimentar dificultades en el proceso de migración hacia una plataforma de 64 bits. En esta se trata de reflejar cierta experiencia en el proceso de migración, experiencia adquirida durante la implementación de diferentes métodos de aprendizaje, teniendo máquinas *SVM* como *algoritmo base*, a lo largo del periodo de elaboración de esta Tesis.

En la guía la atención es centrada fundamentalmente en dos aspectos, (i) la instalación de un compilador compatible con Matlab, para la creación de *MEX-files* y (ii) la compilación o adaptación de las rutinas fornecidas con el Spider para utilización en un ambiente de 64 bits. Los referidos aspectos son tratados a continuación.

Instalación de un compilador compatible

Estas instrucciones están redactadas de acuerdo a información disponible en las páginas de soporte de Matlab (<http://www.mathworks.es/support/>). Las mismas fueron testadas con el Sistema Operativo Windows 7 Profesional y la versión R2008a (7.6) de Matlab, por lo que en el caso de utilización de otras versiones quizás sea necesaria la utilización de otras herramientas, compatibles con las versiones utilizadas, y no exactamente las mencionadas en la guía. El proceso de instalación se resume a lo siguiente.

1. Caso tenga ya instalado en el sistema un compilador de 64 bits para la generación de *MEX-files* se puede pasar directamente a la sección de *compilación/adaptación de las rutinas de la librería*.
2. Instalar un compilador compatible con la versión de Matlab existente en el sistema. En este caso, para la versión R2008a de Matlab fue utilizado el *Microsoft Visual Studio 2008 Express Edition*. Este es distribuido libremente, pero su enlace ya no se encuentra en la página *Visual Express* de la Microsoft (<http://www.microsoft.com/>

- visualstudio/eng/products/visual-studio-express-products). De todos modos se puede descargar en <http://download.microsoft.com/download/8/B/5/8B5804AD-4990-40D0-A6AA-CE894CBBB3DC/VS2008ExpressENUX1397868.iso>. Caso utilice una versión más reciente de Matlab, puede consultar la lista de compiladores compatibles en <http://www.mathworks.com/support/compilers>.
3. Por defecto, al instalar el *Visual Studio* este solamente compila aplicaciones de 32 bits. Para la compilación de aplicaciones de 64 bits es necesario instalar algunos componentes adicionales, integrados en el *Microsoft Windows Software Development Kit* (SDK). El SDK se puede obtener para instalación en <http://msdn.microsoft.com/en-us/windowsserver/bb980924.aspx>, siendo recomendable la instalación del *SDK for Windows 7 and .NET Framework 3.5 SP1*.
 4. Al efectuar la instalación del SDK, se debe seleccionar la opción “*X64 Compilers and Tools*”.
 5. Para verificar si todas las componentes fueron instaladas, verifique si el SDK contiene la versión “*amd64*” del compilador C/C++ “*cl.exe*”. Esta es usualmente instalada en el directorio *C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\bin\amd64\cl.exe*.
 6. Con esto ya se puede compilar *MEX-files* normalmente. Para tal inicie el Matlab y a través del comando “*mex -setup*” seleccione la opción que permite que el Matlab localice automáticamente los compiladores instalados.

Compilación/adaptación de las rutinas

A partir de la versión R2006b (7.3) de Matlab se hizo la introducción del soporte para indexación con 64 bits, lo que permite la creación de variables con hasta $2^{48} - 1$ elementos en plataformas de 64 bits. En el proceso de compilación de *MEX-files*, la nueva API de 64 bits es ac-

tivada a través de la utilización de la opción *-largeArrayDims* con el comando *mex*. La simple compilación y ejecución de *MEX-files* en plataformas de 64 bits no garantiza *per se* el pleno acceso a las nuevas funcionalidades, siendo para tal necesaria la actualización del código fuente de los *MEX-files*.

Como alternativa, Matlab conserva una camada de compatibilidad que permite la compilación y ejecución de *MEX-files* en entornos de 64 bits empleando la API de 32 bits, lo que facilita el proceso de transición por no requerir alteraciones en el código fuente de los *MEX-files*. La referida camada es activada al emplear el comando *mex* con la opción *-compatibleArrayDims*.

Las orientaciones constantes en esta guía se refieren a la compilación de las rutinas de optimización de máquinas *SVM* integradas en el Spider para utilización en un entorno de 64 bits. En esta se considera apenas, y sin perder generalidad, el caso de los optimizadores empleados en la rutina de entrenamiento de máquinas *SVM* para clasificación, siendo utilizada para la compilación, en el proceso descrito, la camada de compatibilidad, es decir, la API de 32 bits. Considerando que algunos de los optimizadores y sus respectivas interfaces para Matlab fueron escritos para utilización en entornos de 32 bits, el acceso con los mismos a la plenitud de las funcionalidades implementadas en la API de 64 bits requiere una ligera adaptación de algunas partes del código fuente de los respectivos *MEX-files*. Como base para la referida adaptación, caso se tome esta como objetivo, se puede tomar las instrucciones existentes en (MathWorks, 2013).

A continuación se hace la descripción de los procedimientos para la compilación de las referidas rutinas. Dichos procedimientos se refieren a la compilación de tres rutinas de optimización de máquinas *SVM* distintas, LIBSVM, SVMlight y SVMTorch.

LIBSVM

En este caso existen dos alternativas. La primera consiste en compilar la función *libsvm_classifier_spider*. El código fuente correspondiente se encuentra en el directorio de la librería Spider, en la carpeta *...spider/Optimization/libsvm_classifier_spider/*. Para tal se debe:

1. Editar el archivo *make.m* existente en la carpeta, adicionando la opción *-compatibleArrayDims*. El comando final sería en este caso,


```
mex -compatibleArrayDims libsvm_classifier_spider.cpp svm.cpp -I ..\compiletools  
..\compiletools\mexarg.cpp.
```
2. Ejecutar el archivo *make.m* para la compilación de la función.
3. Copiar el *MEX-file* resultante, *libsvm_classifier_spider.mexw64*, para el directorio donde reside habitualmente los *MEX-files* de optimización relacionados con la librería Spider, en este caso *.../spider/Optimization/*.

La segunda alternativa consiste en compilar la interfaz para Matlab fornecida en el paquete LIBSVM y adaptar la rutina de entrenamiento para clasificación con máquinas *SVM* de la librería Spider para que emplee las funciones fornecidas con la referida interfaz. Dicha interfaz puede ser obtenida en el sitio web de LIBSVM (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>). Para tal se debe:

1. Hacer la descarga y descompresión de la interfaz de LIBSVM para Matlab.
2. Editar el archivo *make.m* fornecido con la librería acrecentando la opción *-largeArrayDims* en todas las instrucciones *mex* existentes; por ejemplo cambiar

```
mex -O -c svm.cpp para
```

```
mex -largeArrayDims -O -c svm.cpp;
```

en el caso de la utilización de versiones recientes de la interfaz, la compilación puede ser llevada a cabo utilizándose a la API de 64 bits, sin necesidad de adaptación del código, una vez que el código fue escrito con las opciones necesarias para que sea independiente de la plataforma.

3. Ejecutar el archivo *make.m* para compilar la librería.
4. Copiar el *MEX-file svmtrain.mexw64* para el directorio donde reside habitualmente los *MEX-files* de optimización relacionados con la librería Spider, en este caso, *.../spider/Optimization/*.
5. En la rutina de entrenamiento de máquinas SVM (*.../spider/pat/@svm/training.m*) substituir la llamada a la función *libsvm_classifier_spider* por una llamada a la función *svmtrain*. Para tal, se puede substituir el código de llamada, existente entre las líneas 225 y 232, por el siguiente:

```
svmOpt = ['-s ' num2str(svm_type) ' -t ' num2str(kernelType)...
         ' -d ' num2str(degree) ' -g ' num2str(gamma) ' -r '...
         num2str(coef0) ' -m ' num2str(cachesize) ' -c '...
         num2str(C) ' -e ' num2str(eps) ' -n ' num2str(nu) ...
         ' -p ' num2str(p) ' -h ' num2str(shrinking)];
model = libsvmtrain(y, x, svmOpt);

alpha = model.sv_coef;
xSV = full(model.SVs);
bias0 = -model.rho;
```

SVMlight

El código fuente correspondiente se encuentra en la carpeta *...spider/Optimization/svm_light.src/*. Para tal se debe:

1. Editar el archivo *svm_common.h*, retirando el comentario existente en la línea 19, `/*#define MICROSOFT*/`.
2. Ejecutar en Matlab el comando

```
mex -compatibleArrayDims svm_mex.c svm_learn_main.c svm_learn.c
      svm_common.c svm_hideo.c;
```

3. Cambiar el nombre del *MEX-file* resultante, *svm_mex.mexw64*, para *svmlight.mexw64*.
4. Copiar el archivo *svmlight.mexw64* para el directorio donde reside habitualmente los *MEX-files* de optimización relacionados con la librería Spider, en este caso *.../spider/Optimization/*.

SVM Torch

El código fuente correspondiente se encuentra en la carpeta *...spider/Optimization/SVMTorch/*. Para la compilación se debe:

1. Ejecutar el comando

```
mex -compatibleArrayDims SVMTorch.cc SVM.cc standardSVM.cc cache.cc ker-
      nel.cc IOTorch.cc general.cc Userkernel.cc;
```

2. Si se genera el error de compilación “*fatal error C1083: Cannot open include file: 'values.h': No such file or directory*”, en la sección de inclusiones del archivo *general.h*, acrecentar la línea ‘*#include <float.h>*’.
3. Si en la compilación ocurre el error “*fatal error C1083: Cannot open include file: 'sys/times.h': No such file or directory*”, en la sección de definiciones del archivo *general.h*, comentar la línea ‘*#define I_WANT_TIME*’ (línea 42).
4. Copiar el archivo resultante, *SVMTorch.mexw64*, para el directorio donde reside habitualmente los *MEX-files* de optimización relacionados con el Spider, en este caso *.../spider/Optimization/*.