



TESIS DOCTORAL

2020

MODELOS DE CLASIFICACIÓN CON MEDIDAS DE CONFIANZA BASADOS EN PREDICTORES CONFORMALES APLICADOS A IMÁGENES DE FUSIÓN NUCLEAR

Msc. D. Alvaro Antonio Olmedo Rodriguez

PROGRAMA DE DOCTORADO EN
INGENIERÍA DE SISTEMAS Y DE CONTROL

Dr. D. JESÚS ANTONIO VEGA SÁNCHEZ

Dr. D. SEBASTIÁN DORMIDO CANTO

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA



Departamento de Informática y Automática
Escuela Técnica Superior de Ingeniería Informática

**CENTRO DE INVESTIGACIONES ENERGÉTICAS,
MEDIOAMBIENTALES Y TECNOLÓGICAS**



Unidad de Análisis Avanzado de Datos del TJ-II
Laboratorio Nacional de Fusión

**MODELOS DE CLASIFICACIÓN CON MEDIDAS DE
CONFIANZA BASADOS EN PREDICTORES CONFORMALES
APLICADOS A IMÁGENES DE FUSIÓN NUCLEAR**

**TESIS DOCTORAL
2020**

Autor: Msc. D. Alvaro Antonio Olmedo Rodriguez

**Directores: Dr. D. Jesús Antonio Vega Sánchez
Dr. D. Sebastián Dormido Canto**

RESUMEN

En esta investigación se estudiaron modelos de clasificación con medidas de confianza basados en predictores conformales y aplicados a imágenes relacionadas con mediciones de densidad y temperatura de los plasmas de fusión nuclear, analizándose cuatro multclasificadores desarrollados y relacionados con los métodos referidos a: la máquina de vectores soporte (SVM) que permitió la construcción de un multclasificador “clásico” donde sólo se obtienen predicciones; la predicción conformal (PC), la predicción conformal inductiva (IPC) y la predicción Venn (PV), que permitieron construir “predictores de confianza” donde además de la predicción se obtienen medidas de fiabilidad.

El objetivo del estudio se centró en “cómo obtener un multclasificador que con el menor tiempo de procesamiento en la fase de entrenamiento, proporcione el mejor rendimiento en términos de tasa de aciertos y fiabilidad”; lográndose mediante la realización de experimentos incrementales y experimentos por lote con las imágenes obtenidas. Con los resultados proporcionados por los experimentos, se realizaron contrastes individuales para cada multclasificador, al confrontar los tiempos de procesamiento frente al rendimiento en términos de tasa de aciertos y fiabilidad. Además de contrastar todos los resultados individuales obtenidos.

Al final del estudio se comprueba que la predicción conformal inductiva permite obtener un multclasificador con un tiempo de procesamiento en la fase de entrenamiento mínimo, proporcionando un buen rendimiento en términos de tasa de aciertos y fiabilidad. Además de comprobar, que la metodología aplicada en esta investigación para la realización de los experimentos incrementales y experimentos por lote, permiten establecer un procedimiento para obtener multclasificadores con un tiempo de procesamiento mínimo en la fase de entrenamiento, garantizando un buen rendimiento en términos de tasa de aciertos y fiabilidad.

Palabras claves: predicción conformal, medidas de confianza, modelos orientados a datos.

ABSTRACT

In this Thesis different classification models with confidence measures based on conformal predictors have been studied. The different models have been applied to classify images from nuclear fusion reactors, these images are related to the electron density and temperature of plasmas. Four multi-classifiers have been analyzed and developed: Support Vector Machines (SVMs) that allowed the construction of a classic multi-classifier where only predictions are obtained. Conformal Prediction (CP), Inductive Conformal Prediction (ICP) and Venn Prediction (VP), that allowed the construction of predictors where also reliability measures were obtained.

The primarily objective of the Thesis was to obtain a multi-classifier that in the least amount of training time can obtain the best performance in terms of hit rate and reliability. This has been achieved by means of the execution of incremental and batch experiments with the obtained images. The results have been analyzed for every multi-classifier being the metrics processing time, hit rate performance and reliability and the results have been compared between the different multi-classifiers.

After conducting the experiments, it can be concluded that ICP allows to obtain a multi-classifier with a minimal processing time in the training stage, giving a good trade-off in terms of hit-rate performance and reliability. Moreover, the methodology followed in this Thesis to carry out the incremental and batch experiments allows to establish a procedure to obtain multi-classifiers with a minimal processing time in the training stage, warranting a good performance in terms of hit rate and reliability.

Key words: conformal prediction, confidence measure, data oriented models.

DEDICATORIA

A mis hijos,

Saúl Antonio, mi príncipe y heredero de mi trono. Te amo hijo.

Albany Sofía, mi florecita encantada, niña de mis ojos. Siempre con una hermosa sonrisa para mí. Te amo hija.

Oriana, mi primer gran amor. Mi vida se llenó de colores con tu llegada. A tu lado aprendí a ser niño. Te amo hija.

A la fuerza de mi fe que me permite vivir y luchar por todas las metas que me propongo.

AGRADECIMIENTOS

A Dios, por darme salud y vida...

Al Dr. Jesús Antonio Vega Sánchez y al Dr. Sebastián Dormido Canto. Les agradezco los años de orientación que han dedicado para sacar adelante la investigación que me permite escribir esta Tesis. Sus consejos y comentarios siempre fueron oportunos.

A la UNED y al CIEMAT.

A mi hermano Freddy Hernández, a su esposa Ana Colmenares y a mis sobrinos Arturo y Anny. Gracias por su hospitalidad.

A mis amigos del nuevo continente, Angel Ramos, David Jaén, Yenny Ocanto, Yamiley Tello, José Aponte, Wilmer Pérez y Pedro López. A pesar de las adversidades siempre están para recordarme, que hay que seguir adelante.

A mis amigos del viejo continente, Marcos Martínez, Ulises Losada, Luis Esteban. Sin dudas, agradezco la gran amistad que me han brindado.

A Josefina Rodríguez, hoy en día miro al pasado y te veo con gran admiración. Bendición.

A Olga Hidalgo. Madre, mis logros, mis metas te las debo a ti. Estoy eternamente agradecido, y no existe riqueza en este mundo para pagarte el amor que me brindas. Te amo.

A las madres de: Saúl Antonio, Albany Sofía y Oriana. Gracias por estar para mis hijos.

ÍNDICE GENERAL

	Pág.
RESUMEN.....	v
ABSTRACT.....	vii
DEDICATORIA.....	ix
AGRADECIMIENTOS.....	xi
ÍNDICE GENERAL.....	xiii
ÍNDICE DE FIGURAS.....	xix
ÍNDICE DE TABLAS.....	xxv
GLOSARIO DE SIGLAS.....	xxix

CAPÍTULO I. INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS

1.1	INTRODUCCIÓN Y MOTIVACIÓN.....	31
1.2	PLANTEAMIENTO DEL PROBLEMA.....	35
1.3	OBJETIVOS DE LA INVESTIGACIÓN.....	50
1.4	METODOLOGÍA SEGUIDA.....	51
1.5	ORGANIZACIÓN DEL TRABAJO.....	53

CAPÍTULO II. LA FUSIÓN

2.1	INTRODUCCION.....	55
2.2	LA FUSIÓN TERMONUCLEAR.....	56
2.3	EL CONFINAMIENTO MAGNETICO DE LOS PLASMAS DE FUSION TERMONUCLEAR.....	58
2.3.1	Las disrupciones.....	59
2.3.2	El colapso radiactivo por calentamiento.....	60

2.3.3	Otros eventos relevantes estudiados en el confinamiento magnético...	61
2.4	EL DISPOSITIVO DE FUSIÓN TERMONUCLEAR TJ-II.....	61
2.4.1	Componentes.....	61
2.4.2	El diagnóstico de esparcimiento Thomson.....	63

CAPÍTULO III. MARCO TEORICO APLICADO

3.1	INTRODUCCIÓN.....	65
3.2	LOS MODELOS.....	65
3.2.1	Fases de construcción de un modelo.....	66
3.2.2	Tipos de modelos.....	68
3.2.3	Definición general de un modelo de clasificación.....	69
3.2.4	El modelo. Los objetos y las etiquetas.....	73
3.2.5	El modelo de clasificación binario y el multclasificador.....	74
3.2.6	Los datos en el entrenamiento de un modelo.....	74
3.3	El APRENDIZAJE.....	76
3.3.1	El aprendizaje automático, la teoría del aprendizaje estadístico y el conocimiento a priori.....	77
3.4	PARADIGMAS DE APRENDIZAJE.....	79
3.4.1	Aprendizaje supervisado.....	80
3.4.2	Aprendizaje no supervisado.....	81
3.4.3	Aprendizaje por refuerzo.....	81
3.5	ALGORITMOS DE APRENDIZAJE.....	82
3.6	MÁQUINAS DE VECTORES SOPORTE.....	82
3.6.1	Máquinas de vectores soporte para la clasificación binaria.....	84
3.6.1.1	Datos linealmente separables.....	85
3.6.1.2	Datos no linealmente separables.....	90
3.6.2	Máquinas de Vectores Soporte para multclasificación.....	100
3.6.2.1	Descripción de la aproximación uno versus el resto.....	101

3.6.2.2	Multclasificador basado en la aproximación uno versus el resto.....	105
3.6.2.3	Tipos de respuestas de un multclasificador SVM.....	108
3.6.2.4	Distancia al hiperplano $h(x)$ como respuesta decisiva de la SVM.....	109
3.7	LA PREDICCIÓN CONFORMAL.....	111
3.7.1	Definición general de la predicción conformal en clasificación.....	114
3.7.2	Funcionamiento de la predicción conformal en clasificación.....	117
3.7.2.1	Medidas de no conformidad a partir de la distancia al hiperplano.....	118
3.7.2.2	Prueba de hipótesis a partir de escenarios basados en la distancia al hiperplano.....	120
3.7.2.3	Asignación de medidas de no conformidad a partir de la distancia al hiperplano.....	123
3.7.2.4	La predicción y las medidas de fiabilidad: confianza y credibilidad.....	125
3.7.3	Multclasificación con predicción conformal.....	127
3.7.3.1	Multclasificación con predicción conformal. Escenarios de prueba de hipótesis adicionales y aproximación modificada uno versus el resto.....	131
3.8	LA PREDICCIÓN CONFORMAL INDUCTIVA.....	135
3.8.1	Predicción conformal inductiva en clasificación.....	135
3.9	LA PREDICCIÓN VENN.....	141
3.9.1	Definición general de la predicción Venn.....	142
3.9.2	Funcionamiento de la predicción Venn.....	144
3.9.3	La predicción y el intervalo de probabilidades.....	145
3.9.4	Taxonomías.....	145
3.9.4.1	Taxonomía: k vecinos más cercanos.....	146
3.9.4.2	Taxonomía: el centroide más cercano.....	150

3.10	LA TRANSFORMADA WAVELET Y EL PRE-PROCESAMIENTO DE IMÁGENES.....	155
3.11	MATLAB Y LA LIBRERÍA LIBSVM.....	157

CAPÍTULO IV. EXPERIMENTOS EN FUSIÓN TERMONUCLEAR

4.1	INTRODUCCIÓN.....	159
4.2	LOS DATOS.....	164
4.2.1	Pre-procesamiento de imágenes mediante la transformada wavelet....	165
4.3	LOS PROGRAMAS Y EL ACCESO A DATOS.....	166
4.4	MULTICLASIFICADOR SVM. DETERMINACIÓN DE LAS OPCIONES PARA LOS MODELOS SVM.....	169
4.5	MULTICLASIFICADOR INCREMENTAL CON SVM.....	193
4.6	MULTICLASIFICADOR POR LOTE CON SVM.....	200
4.7	MULTICLASIFICADOR CON PREDICCIÓN CONFORMAL Y PREDICCIÓN CONFORMAL INDUCTIVA.....	207
4.7.1	SVM como método subyacente y la aproximación modificada uno versus el resto.....	207
4.7.2	Criterio para la selección del parámetro λ	217
4.8	MULTICLASIFICADOR INCREMENTAL Y POR LOTE CON PC.....	219
4.9	MULTICLASIFICADOR INCREMENTAL Y POR LOTE CON IPC.....	230
4.10	MULTICLASIFICADOR CON PREDICCIÓN VENN.....	241
4.10.1	Multclasificador incremental y por lote con PV-KNN.....	241
4.10.2	Multclasificador incremental y por lote con PV-CNN.....	249

CAPÍTULO V. ANÁLISIS DE LOS RESULTADOS

5.1	INTRODUCCIÓN.....	251
-----	-------------------	-----

5.2	CONTRASTE DE LOS RESULTADOS OBTENIDOS EN LOS EXPERIMENTOS CON IMÁGENES PROVENIENTES DEL DIAGNÓSTICO DE ESPARCIMIENTO THOMSON.....	252
5.3	IMPORTANCIA DEL PROCESAMIENTO ALEATORIO.....	258
5.4	IMPORTANCIA DE REPETIR UN EXPERIMENTO MÁS DE UNA VEZ...	261
5.5	IMPORTANCIA DE AJUSTAR MANUALMENTE EL PORCENTAJE FIJADO POR LAS PRUEBAS INCREMENTALES.....	266
5.6	EL EFECTO DEL SORTEO ALEATORIO SOBRE LAS IMÁGENES.....	268
	CONCLUSIONES.....	271
	LINEAS FUTURAS DE TRABAJO.....	275
	BIBLIOGRAFÍA.....	277

ÍNDICE DE FIGURAS

		Pág.
Figura 1.1	Etapas para la construcción de modelos.....	52
Figura 2.1	Representación visual de un reactor de fusión tipo TOKAMAK.....	59
Figura 2.2	Representación visual de un reactor de fusión tipo STELLARATOR..	59
Figura 3.1	Sistema y sus variables de entrada-salida.....	69
Figura 3.2	Separación de objetos a partir de sus atributos mediante un clasificador. Dos atributos diferenciadores: color y forma geométrica.....	71
Figura 3.3	Separación de objetos a partir de sus atributos mediante un clasificador. Sólo un atributo diferenciador: color.....	72
Figura 3.4	Diagrama esquemático del aprendizaje supervisado.....	80
Figura 3.5	Diagrama esquemático del aprendizaje no supervisado.....	81
Figura 3.6	Diagrama esquemático del aprendizaje por refuerzo.....	81
Figura 3.7	Conjunto de datos separado por una SVM.....	83
Figura 3.8	Conjuntos de datos. (a) Linealmente separables. (b) No linealmente separables.....	85
Figura 3.9	Distancia de los objetos x_i al hiperplano $h(x)$	86
Figura 3.10	Ejemplo de aplicación de la SVM para datos linealmente separables...	90
Figura 3.11	Mapeo de objetos de la función de transformación Φ desde el espacio de entrada al espacio de características.....	91
Figura 3.12	Datos disponibles para el Ejemplo 2. (b) Mapa de predicción posible producido por el hiperplano $h(x)$ inferido con los datos disponibles del Ejemplo 2. Región en color rojo: predice la Clase +1. Región en color azul: predice la Clase -1.....	95
Figura 3.13	(a) Datos disponibles para el Ejemplo 3. (b) Mapa de predicción posible producido por el hiperplano $h(x)$ inferido con los datos disponibles del Ejemplo 3. Región en color rojo: predice la Clase +1. Región en color azul: predice la Clase -1.....	99
Figura 3.14	(a) Puntos que conforman un conjunto de datos multiclase. Clase 1: círculos, Clase 2: cuadrados, Clase 3: rombo y Clase 4: triángulos. (b) Inclusión de un objeto asterisco en las coordenadas (-3,2.5) en el conjunto de datos multiclase.....	102
Figura 3.15	Mapas de predicción posibles producidos por los hiperplanos $h(x)$ inferidos con los datos binarios formados a partir de los datos multiclase de la Figura 3.14 (a). Clase +1: signo de suma “+” en color azul. Clase -1: punto “.” en color rojo.....	103
Figura 3.16	Respuesta obtenida para un objeto asterisco (desconocido) ubicado en la coordenada (-3,2.5) usando los mapas de predicción presentados en la Figura 3.15.....	107
Figura 3.17	Respuesta obtenida para un nuevo objeto asterisco (desconocido) ubicado en la coordenada (-3,-2) usando los mapas de predicción presentados en la Figura 3.15.....	110

Figura 3.18	Capas de la predicción conformal.....	117
Figura 3.19	Creación de escenarios para prueba de hipótesis mediante la distancia al hiperplano. (a) Datos disponibles. <i>Clase +1</i> : círculos azules, <i>Clase -1</i> : cuadrados negros y objeto a predecir: asterisco rojo. (b) Hiperplano bajo la hipótesis que el objeto a predecir es de la <i>Clase +1</i> . (c) Hiperplano bajo la hipótesis que el objeto a predecir es de la <i>Clase -1</i>	121
Figura 3.20	Datos disponibles para la creación de escenarios en un problema multiclase. <i>Clase 1</i> : círculos, <i>Clase 2</i> : cuadrados, <i>Clase 3</i> : rombos, asterisco: punto a identificar.....	128
Figura 3.21	Mapas de predicción posibles producido por los hiperplano $h(x)$ inferidos con los datos observados en la Figura 3.20. <i>Clase +1</i> : signo de suma “+” en color azul. <i>Clase -1</i> : punto “.” en color rojo.....	129
Figura 3.22	Modificaciones en la capa 1 de la predicción conformal.....	137
Figura 3.23	Capas de la predicción Venn.....	144
Figura 3.24	Centroides más cercanos. Se asume el asterisco como: (a) un círculo, (b) un cuadrado, (c) un rombo.....	152
Figura 3.25	Comparación de tiempos (en minutos) de procesamiento de las taxonomías KNN y CNN a partir de un conjunto de entrenamiento de 6 muestras hasta alcanzar 456.....	154
Figura 4.1	Agrupación de imágenes para el entrenamiento y pruebas de los modelos SVM, PC, IPC, PV.....	163
Figura 4.2	Imágenes típicas asociadas a las cinco clases. (a) BKG, (b) ECR, (c) NBI, (d) STRAY, (e) COFF.....	165
Figura 4.3	Descomposición de una imagen típica BKG en diferentes niveles de detalle: (a) coeficiente de aproximación, (b) horizontal, (c) vertical y (d) diagonal. Para un nivel de compresión con la Wavelet Haar Nivel 4.....	166
Figura 4.4	Flujo de ejecución de los programas implementados y su acceso a datos.....	168
Figura 4.5	Diagrama de bloques general de la implementación del Programa 1 que explora los valores de las opciones c y g para los modelos SVM acoplados al multclasificador SVM para cuatro clases.....	171
Figura 4.6	Porcentaje de acierto (curva en color azul) versus el tiempo de procesamiento (curva en color rojo) en contraste con los valores n_1 , n_2 , n_3 , n_4 usados para calcular los valores de c de los modelos SVM que son acoplados al multclasificador SVM para cuatro clase.....	175
Figura 4.7	Diagrama de bloques general de la implementación del Programa 2 que explora los conjuntos de muestras tomados por lote en contraste con los diferentes detalles AHVD a partir del multclasificador SVM para cuatro clases.....	177

Figura 4.8	Contraste obtenido entre el porcentaje de aciertos versus el porcentaje por lote al procesar las imágenes en sus diferentes detalles AHVD con el multclasificador SVM para cuatro clases, entrenado a partir de las opciones c y g establecidas con la implementación del Programa 1.....	179
Figura 4.9	Resultado obtenido al ejecutar 100 veces la implementación del Programa 2 para el multclasificador de cuatro clases. Porcentaje de aciertos versus el porcentaje por lote de imágenes para los detalles A y V.....	180
Figura 4.10	Diagrama de bloques general de la implementación del Programa 1 que explora los valores de las opciones c y g para los modelos SVM acoplados al multclasificador SVM para cinco clases.....	181
Figura 4.11	Porcentaje de acierto (curva en color azul) versus el tiempo de procesamiento (curva en color rojo) en contraste con los valores n_1 , n_2 , n_3 , n_4 , y n_5 usados para calcular los valores de c de los modelos SVM que son acoplados al multclasificador SVM para cinco clases.....	182
Figura 4.12	Diagrama de bloques general de la implementación del Programa 2 que explora los conjuntos de muestras tomadas por lote en contraste con los diferentes detalles AHVD a partir del multclasificador SVM para cinco clases.....	183
Figura 4.13	Contraste obtenido entre el porcentaje de aciertos versus el porcentaje por lote al procesar las imágenes en sus diferentes detalles AHVD con el multclasificador SVM para cinco clases, entrenado a partir de las opciones c y g establecidas con la implementación del Programa 1.....	185
Figura 4.14	Resultado obtenido al ejecutar 100 veces el Programa 2 para el multclasificador de cinco clases.....	185
Figura 4.15	Verificación del porcentaje de aciertos para diferentes detalles AHVD medidos a partir de conjuntos de imágenes balanceados y desbalanceados al usar un multclasificador para cuatro clases.....	189
Figura 4.16	Resultado obtenido al ejecutar 20 veces el Programa 2 para el multclasificador SVM para cuatro clases con datos balanceados y desbalanceados. Porcentaje de aciertos versus el porcentaje por lote de imágenes para el detalles V.....	190
Figura 4.17	Verificación del porcentaje de aciertos para diferentes detalles AHVD medidos a partir de los conjuntos de imágenes balanceados y desbalanceados al usar un multclasificador para cinco clases.....	192
Figura 4.18	Resultado obtenido al ejecutar 20 veces el Programa 2 para el multclasificador SVM para cinco clases con datos balanceados y desbalanceados.....	192
Figura 4.19	Diagrama de bloques general de la implementación del programa que permite realizar el entrenamiento y la prueba incremental del multclasificador SVM.....	195

Figura 4.20	Porcentaje de aciertos y tiempo de procesamiento obtenidos para el multclasificador SVM por incremento con los conjuntos iniciales de imágenes según la Tabla 4.9.....	197
Figura 4.21	Contraste del porcentaje de aciertos visto entre los cuatro ensayos desarrollados en la Prueba 5 para el detalle V.....	198
Figura 4.22	Diagrama de bloques general de la implementación del programa que permite realizar entrenamiento y prueba por lote del Multclasificador SVM.....	200
Figura 4.23	Porcentaje de aciertos y tiempo de procesamiento obtenidos para el multclasificador SVM por lote, con los conjuntos de imágenes según la Tabla 4.10.....	203
Figura 4.24	Contraste del porcentaje de aciertos y tiempo de procesamiento visto entre los cuatro ensayos desarrollados en la Prueba 7 para el detalle V. Nota: el eje de las abscisas muestra el progreso por lote de las imágenes tomadas.....	204
Figura 4.25	Contraste entre los resultados obtenidos en la Prueba 5 y 7 para conjuntos desbalanceados con detalle V. Porcentaje de acierto en curva amarilla obtenida con un conjunto de imágenes para entrenamiento por lote: (a) Un 35%. (b) Un 45%.....	206
Figura 4.26	Diagrama de bloques general de la implementación del programa que explora los valores λ y las opciones SVM para un multclasificador PC o IPC.....	209
Figura 4.27	Diagrama de bloques general de la implementación del Programa que clasifica imágenes por incremento a partir del multclasificador PC.....	220
Figura 4.28	Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador PC incremental para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	222
Figura 4.29	Resultados obtenidos al ejecutar diez veces el programa asociado al multclasificador PC incremental para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	223
Figura 4.30	Diagrama de bloques general la implementación del programa que clasifica imágenes por lote a partir del multclasificador PC.....	225
Figura 4.31	Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador PC por lote para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	227
Figura 4.32	Resultados obtenidos al ejecutar diez veces el programa asociado al multclasificador PC por lote para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	228
Figura 4.33	Acceso a las imágenes mediante el algoritmo que implementa la predicción conformal inductiva.....	232

Figura 4.34	Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador IPC incremental para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	234
Figura 4.35	Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador IPC incremental para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	235
Figura 4.36	Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador IPC por lote para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	239
Figura 4.37	Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador IPC por lote para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	240
Figura 4.38	Diagrama de bloques general de la implementación del programa que clasifica imágenes por incremento a partir del multclasificador PV con taxonomía KNN y CNN.....	243
Figura 4.39	Resultados obtenidos al ejecutar la implementación del programa asociado al multclasificador PV-KNN incremental para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V.....	244
Figura 4.40	Resultados obtenidos al ejecutar cinco veces la implementación del programa asociado al multclasificador PV-KNN incremental para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V.....	245
Figura 4.41	Diagrama de bloques general de la implementación del programa que clasifica imágenes por lote a partir del multclasificador PV con taxonomía KNN y CNN.....	246
Figura 4.42	Resultados obtenidos al ejecutar 5 veces el programa asociado al multclasificador PV-KNN por lote para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V.....	248
Figura 4.43	Resultados obtenidos al ejecutar 5 veces el Programa asociado al multclasificador PV-KNN por lote para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V.....	249
Figura 5.1	Graficas del porcentaje de acierto obtenido para un Multclasificador SVM sin incluir e incluyendo el ordenador de imágenes IID tanto para el procesamiento de muestras: (a), (b) por incremento, (c), (d) por lote.....	260

Figura 5.2	Resultados obtenidos al ejecutar 5 veces el Programa asociado al multclasificador IPC incremental para los conjuntos de imágenes con cuatro clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$	262
Figura 5.3	Resultados obtenidos del multclasificador IPC por lote para diferentes conjuntos (porcentajes) de imágenes con cuatro clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$. Porcentajes: (a) 7%, (b) 10%, (c) 14%.....	264

ÍNDICE DE TABLAS

		Pág.
Tabla 1.1	Resumen de las etapas para construcción de modelos.....	53
Tabla 2.1	Sistema de bobinas del TJ-II.....	62
Tabla 3.1	Cálculos a partir de las expresiones (3.22), (3.23) y (3.24).	90
Tabla 3.2	Valores típicos encontrados para los datos del Ejemplo 2.	96
Tabla 3.3	Solución encontrada a partir de los α_i para los datos del Ejemplo 2.....	96
Tabla 3.4	Valores típicos encontrados para los datos del Ejemplo 3.....	99
Tabla 3.5	Solución encontrada a partir de los α_i para los datos del Ejemplo 3.....	99
Tabla 3.6	Resumen del conjunto de datos multiclase de la Figura 3.14 (a).....	105
Tabla 3.7	Tipos de respuesta obtenida por el multclasificador definido a partir de los datos multiclase de la Figura 3.14 (a) cuando procesa el objeto asterisco ubicado en las coordenadas (-3,2.5).	108
Tabla 3.8	Tipos de respuesta obtenida por el multclasificador definido a partir de los datos multiclase de la Figura 3.14 (a) cuando procesa el nuevo objeto asterisco ubicado en las coordenadas (-3,-2).	111
Tabla 3.9	Información implicada al generar un modelo SVM.	122
Tabla 3.10	Forma incorrecta de formular hipótesis según los escenarios de prueba de hipótesis para los datos multiclase de la Figura 3.20.....	128
Tabla 3.11	Medidas de no conformidad obtenidas a partir de la distancia al hiperplano encontradas para los escenarios de la Figura 3.21 (a), (d), (g), (e), (b) y (c) con sus correspondientes $p_valores$	130
Tabla 3.12	Forma correcta de formular hipótesis según los escenarios de prueba de hipótesis para los datos multiclase de la Figura 3.20.....	132
Tabla 3.13	Medidas de no conformidad obtenidas a partir de la distancia al hiperplano según el enfoque dado por la expresión (3.54), con sus correspondientes $p_valores$	134
Tabla 3.14	Datos de la Figura 3.20 usados para contrastar la predicción conformal con la predicción conformal inductiva. PTS*: datos del conjunto de entrenamiento. CP*: dato a predecir. PTS: datos del conjunto de entrenamiento. CS: datos del conjunto de calibración. CP: dato a predecir.....	139
Tabla 3.15	Resumen de los $p_valores$ obtenidos al clasificar el asterisco propuesto en la Figura 3.20 bajo el modelo de la predicción conformal y la predicción conformal inductiva.....	139
Tabla 3.16	Resumen de las medidas de fiabilidad que identifican al asterisco propuesto en la Figura 3.20 como un rombo bajo el modelo de la predicción conformal y la predicción conformal inductiva.....	140
Tabla 3.17	Costo computacional para lograr clasificar el asterisco propuesto en la Figura 3.20 como un rombo bajo el modelo de la predicción conformal y la predicción conformal inductiva.....	140

Tabla 3.18	Centroides vistos en la Figura 3.24, asumiendo la coordenada (-0.5,-1.0) del objeto asterisco como: columna 4: círculo, columna 5: cuadrado, columnas 6: rombo.	152
Tabla 3.19	Cuadro comparativo entre las taxonomías KNN y CNN para la predicción hecha sobre el asterisco rojo de la Figura 3.20.	153
Tabla 3.20	Algunos valores puntuales de las taxonomías KNN y CNN al incrementar el conjunto de entrenamiento en la predicción de una muestra.	154
Tabla 4.1	Cantidad de imágenes por clase a utilizar según el tipo de multclasificador a evaluar.	164
Tabla 4.2	Imágenes tomadas en el DST y su descripción.....	165
Tabla 4.3	Informe proporcionado por la implementación del Programa 1 de la Figura 4.5.	173
Tabla 4.4	Informe proporcionado por la implementación del Programa 2 de la Figura 4.7.	178
Tabla 4.5	Informe proporcionado por la implementación del Programa 2 de la Figura 4.12.	184
Tabla 4.6	Comparación de las opciones encontradas a partir del Programa 1 para los modelos SVM, base de los multclasificadores SVM para cuatro y cinco clases a partir de un conjunto de imágenes desbalanceado.	187
Tabla 4.7	Comparación de las opciones encontradas con imágenes balanceadas y desbalanceadas para los modelos SVM, base de los multclasificadores SVM para cuatro clases.	188
Tabla 4.8	Comparación de las opciones encontradas con imágenes balanceadas y desbalanceadas para los modelos SVM, base de los multclasificadores SVM para cinco clases.	191
Tabla 4.9	Conjuntos de imágenes iniciales para los entrenamientos y pruebas incrementales con el fin de establecer el conjunto mínimo de imágenes para el entrenamiento por lote de un multclasificador SVM.....	196
Tabla 4.10	Cantidades de imágenes dispuestas para los entrenamientos y pruebas por lote con el multclasificador SVM y su convergencia hacia altas tasas de aciertos.....	202
Tabla 4.11	Porcentaje de aciertos y tiempo de procesamiento conseguido mediante la implementación del programa de la Figura 4.19 para un conjunto de imágenes tomando el 20% de muestra para el entrenamiento con las nuevas y antiguas opciones SVM al variar λ	211
Tabla 4.12	Informes arrojado por la implementación del programa de la Figura 4.19 para el multclasificador PC con las nuevas opciones SVM.....	213
Tabla 4.13	Informes arrojado por la implementación del programa de la Figura 4.19 para el multclasificador IPC con las nuevas opciones SVM.....	214
Tabla 4.14	Informes arrojado por la implementación del programa de la Figura 4.19 para el multclasificador PC con las antiguas opciones SVM.....	215

Tabla 4.15	Informes arrojado por la implementación del programa de la Figura 4.19 para el multclasificador IPC con las antiguas opciones SVM.....	216
Tabla 4.16	Selección del parámetro λ y opciones SVM para construir los multclasificadores SVM, PC e IPC para un 20% de muestras para entrenamiento.....	218
Tabla 4.17	Cantidades de imágenes dispuestas para los entrenamientos y pruebas incrementales con el fin de establecer el conjunto inicial de imágenes de entrenamiento con el multclasificador PC.....	221
Tabla 4.18	Cantidades de imágenes dispuestas para los entrenamientos y pruebas por lote con el multclasificador PC y su convergencia hacia altas tasas de aciertos.....	226
Tabla 4.19	Cantidades de imágenes dispuestas para los entrenamientos y pruebas incrementales con el fin de establecer el conjunto inicial de imágenes de entrenamiento con el multclasificador IPC.	233
Tabla 4.20	Cantidades de imágenes dispuestas para los entrenamientos y pruebas por lote con el multclasificador IPC y su convergencia hacia altas tasas de aciertos en la Prueba 15.	237
Tabla 4.21	Cantidades de imágenes dispuestas para los entrenamientos y pruebas incrementales con el fin de establecer el conjunto inicial de imágenes de entrenamiento con el multclasificador PV-KNN.	242
Tabla 4.22	Cantidades de imágenes dispuestas para los entrenamientos y pruebas por lote con el multclasificador PV-KNN y su convergencia hacia altas tasas de aciertos en la Prueba 17.	247
Tabla 5.1	Resultados obtenidos por los experimentos con los multclasificadores SVM, PC, IPC y PV-KNN en el Capítulo IV-Parte 1.....	254
Tabla 5.2	Resultados obtenidos por los experimentos con los multclasificadores SVM, PC, IPC y PV-KNN en el Capítulo IV-Parte 2.	256
Tabla 5.3	Conjuntos de imágenes para entrenar un multclasificador IPC por lote marcado por el cruce del transitorio en las gráficas del porcentaje de acierto de una prueba incremental para cinco sorteos diferentes de imágenes.....	263

GLOSARIO DE SIGLAS

	Significado
AHVD	Detalles wavelet. Coeficiente de aproximación (A), detalle horizontal (H), detalle vertical (V), detalle Diagonal (D)
c	Parámetro del modelo SVM
CB	Clase balanceada
CCD	Cámara con tecnología de carga acoplada
CD	Clase desbalanceada
CIEMAT	Centro de Investigaciones Energéticas Medio Ambientales y Tecnológicas
CNN	Centroide más cercano
CP	Conjunto de prueba o predicción
CS	Conjunto de calibración
d	Valor para ajustar el parámetro g del modelo SVM
DTS	Diagnóstico de esparcimiento Thomson
g	Parámetro del modelo SVM
IPC	Predicción conformal inductiva. También se puede referir al multclasificador IPC
ITER	International Thermonuclear Experimental Reactor. Dispositivo experimental tipo Tokamak
JET	Joint European Torus. Dispositivo experimental de fusión tipo Tokamak
KNN	k Vecinos más cercanos
MC	Multclasificador
n	Valor para ajustar el parámetro c del modelo SVM
NA	No aplica
p_valor	Valor estadístico que define probabilidad
PC	Predicción conformal. También se puede referir al multclasificador PC
PTS	Conjunto propio de entrenamiento
PV	Predicción Venn. También se puede referir al multclasificador PV
PV-CNN	Multclasificador basado en la predicción Venn con taxonomía CNN
PV-KNN	Multclasificador basado en la predicción Venn con taxonomía KNN
RNA	Redes neuronales artificiales
SVM	Máquinas de vectores soporte. También se puede referir al multclasificador SVM

TJ-II Torus JEN II. Dispositivo experimental de fusión tipo Stellarator
 λ Parámetro usado por la aproximación modificada uno versus el resto

CAPÍTULO I

1 INTRODUCCIÓN, MOTIVACIÓN Y OBJETIVOS

1.1 INTRODUCCIÓN Y MOTIVACIÓN

La física de plasmas termonucleares así como otros procesos estudiados por el hombre, está regida por interacciones altamente no lineales y a día de hoy no existen teorías a partir de primeros principios que expliquen su evolución temporal y la aparición de inestabilidades [Murari, 2015].

Debido a esto, los investigadores estudian estos procesos a nivel experimental, estableciendo diagnósticos con el fin de medir variables para crear modelos matemáticos o cualquier otra representación [Makili, 2014], [Murari, 2015], [Martínez, 2017] que permita hacer posteriormente inferencias o predicciones sobre los mismos, procurando así la toma de decisiones [Moraleda y Villalba, 2016], [González, 2012], [Azarang, 1996], [Russell y Norving, 2003], [Chapra y Canale, 2003].

Como ejemplo particular, se menciona el uso de diagnósticos que son acoplados en los dispositivos de fusión termonuclear para medir diferentes parámetros del plasma como la temperatura, densidad, corriente, entre otras características que son recogidas como señales unidimensionales (ondas) y/o bidimensionales (imágenes) [Laboratorio Nacional de Fusión, 2019], [Farias et al, 2004], [Martínez, 2017].

Hay varios aspectos importantes que pueden resaltarse de estos diagnósticos y que hoy en día aún son tema de investigación como se observa en la gran cantidad de artículos científicos, conferencias y tesis existentes. En este sentido se mencionan como ejemplo: [Farias et al, 2004], [Castro, 2010], [Makili, 2014], [Murari, 2015], [Martínez, 2017], [Farias et al, 2018]. Algunos de los temas relevantes tratados en los trabajos de investigación son:

- La generación y adquisición de grandes volúmenes de datos o información provenientes de los sensores (diagnósticos) conectados para registrar las diferentes fluctuaciones que existen por las reacciones del plasma.
- Los sistemas que permiten el almacenamiento, el procesamiento y la seguridad de esta información.
- Los métodos e implementaciones para la extracción e interpretación de información.

El presente trabajo de investigación se puede asociar al último aspecto comentado. En [Makili, 2014] se asegura que *“la extracción e interpretación de la información es un gran desafío”*, y comenta algunos de los problemas tratados en el campo de la fusión tales como: la búsqueda de semejanzas o la clasificación de formas de onda, la clasificación de imágenes, la detección de transiciones en el régimen de confinamiento del plasma y la predicción de interrupciones. En todos estos casos mencionados y en muchos otros que se han revisado, el común denominador apunta al desarrollo de modelos y teorías basados en datos.

Es importante resaltar que no sólo es en el campo de la fusión termonuclear donde se ha requerido y se requiere seguir investigando sobre la extracción e interpretación de la información para la generación de modelos y teorías basados en datos, puesto que en el resto de las áreas de la ciencia e ingeniería, sea cual sea su naturaleza (técnica o social), esta labor se ha convertido en un campo multidisciplinario que utiliza métodos, procesos, algoritmos y sistemas científicos para extraer conocimientos [Armbrust et al, 2010], [Dhar, 2013], [Russell y Norving, 2003], [Chapra y Canale, 2003], [Vovk, Gammerman, y Shafer, 2005], [Vapnik, 1998].

Muchos de los métodos utilizados para la extracción e interpretación de la información, pueden emplearse de dos maneras diferentes por los investigadores cuando tienen el objetivo de construir modelos para responder a preguntas planteadas sobre la naturaleza o proceso que generan los datos disponibles, dependiendo si la información ya está almacenada en alguna bases de datos, o si los datos se generan en un proceso de tiempo real [Cherkassky y Mulier, 2007], [Russell y Norving, 2003],

[Makili, 2014], [Martínez, 2017]. Ambas maneras son comúnmente usadas para conseguir nuevas conclusiones.

Lo anteriormente descrito puede entenderse como “la filosofía” del aprendizaje automático. Es decir, una forma de representar el conocimiento a partir de los datos, dándole la capacidad a una máquina de responder automáticamente ante una entrada determinada. Por esto en la actualidad, el aprendizaje automático se puede considerar la herramienta más utilizada en la búsqueda sistemática de soluciones para el tratamiento de la información almacenada en grandes bases de datos o en procesos que requieren tratamiento en línea. Esto incluye comúnmente los problemas que presenta, el campo experimental de la fusión termonuclear tal y como se observa en [Farias et al, 2004], [Vega et al, 2009], [Castro, 2010], [Dormido-Canto et al, 2013], [Makili, 2014] [Martínez, 2017], [Farias et al, 2018].

No obstante, el desarrollo de un sistema de aprendizaje automático requiere de la construcción de modelos, y según [Cherkassky y Mulier, 2007] se debe cumplir una serie de pasos o etapas que según los intereses de esta investigación se pueden resumir en:

a) Definición del problema: se realiza un análisis global del problema a resolver, permitiendo que se formulen interrogantes o hipótesis que deben ser respondidas o comprobadas a partir del estudio de los modelos o del sistema a desarrollar.

b) Preparación de los datos: se pre-procesan los datos ya que estos pueden presentarse de origen en un formato diferente para el algoritmo utilizado en las siguientes etapas. Además, éstos pueden estar incompletos, presentar redundancia, ser irrelevantes, exhibirse en una dimensionalidad inadecuada y/o contener ruido, entre otras consideraciones que deben tomarse en cuenta antes de realizar el modelo [Cherkassky y Mulier, 2007], [Russell y Norving, 2003], [Proakis y Manolakis, 1998]. En ese sentido, se debe escoger la metodología más adecuada para el pre-procesamiento de los datos.

c) Extracción y selección de características: se extraen y seleccionan las características informativas que forman parte de los componentes que deben

incorporarse mediante el proceso de entrenamiento en la etapa de generación del modelo. En [Cherkassky y Mulier, 2007] se describen consideraciones necesarias para planificar el desarrollo de un sistema de aprendizaje automático, indicando entre otras cosas, la estrecha relación entre esta etapa y la anterior.

d) Generación de modelos: se estudia, diseña y construye (codifica) el *programa*¹ para que realice la inferencia del modelo a partir de los datos de entrenamiento y seguidamente proporcione una predicción ante nuevas muestras. Este programa debe integrar los tres aspectos importantes que se mencionan a continuación:

- *d1)* Los componentes que deben aprenderse, entendiéndose como los atributos o características informativas que serán extraídos de los datos disponibles.
- *d2)* Un paradigma de realimentación (algoritmo de aprendizaje) que permita el procesamiento eficiente de dichos datos, para la extracción e incorporación de patrones o características, que será la base del conocimiento obtenido y usado para inferir sobre algún tipo de respuesta esperada al examinar nueva información. Los paradigmas generalmente utilizados son: el aprendizaje supervisado, el no supervisado y el aprendizaje por refuerzo [Russell y Norving, 2003], [Freeman y Skapura, 1993], [Kosko, 1992], [Olmedo, 2009].
- *d3)* Como último aspecto, pero no menos importante a considerar, está el algoritmo que representará y dará algún tipo de respuesta según los componentes aprendidos, siendo esto en esencia el secreto del aprendizaje, ya que los componentes permiten que el determinado algoritmo mediante el paradigma definido en (*d2*) tenga de la capacidad para dar una respuesta ante nuevos datos que deben ser procesados y sobre los cuales se tomarán decisiones. Cabe acotar que, entre los algoritmos de representación más usados en la actualidad se destacan las redes neuronales artificiales (RNA) [Kosko, 1992], [Freeman y Skapura, 1993], [Olmedo, 2009]], las máquinas de

¹ En esta Tesis se entiende por programa, el conjunto de algoritmos diseñados y codificados para ejecutar un proceso específico. La codificación de estos algoritmos se ha realizado en el lenguaje de programación de Matlab.

vectores soporte (SVM) [Boser et al, 1992], [Cortes y Vapnik, 1995], [Vapnik, 2000], los algoritmos de agrupamiento como los k vecino más cercano (KNN) [Russell y Norving, 2003], [Cherkassky y Mulier, 2007], y más actualmente el aprendizaje profundo (Deep Learning) [Farias et al, 2018].

En concreto, las etapas (b), (c) y (d) sirven para inferir un modelo, regla o dependencia que será utilizado para hacer una “predicción efectiva” sobre los nuevos datos que llegan y el tipo de respuesta o salida que debe obtenerse, todo esto en función de los datos disponibles. Por lo general, este tipo de respuesta define: (1) los modelos de clasificación, si la respuesta se restringe a un conjunto finito de valores; y (2) los modelos de regresión, si la respuesta está dada por un conjunto infinito de valores. El presente trabajo de investigación se centra en los modelos de clasificación.

e) Comprobación y validación de modelos: construido el modelo a partir de las etapas (b), (c) y (d), es necesario medir la idoneidad de la respuesta o “predicción efectiva” que da el sistema al procesar nuevos datos. Esta etapa es la que ha tenido un mayor peso en el desarrollo de esta Tesis Doctoral.

1.2 PLANTEAMIENTO DEL PROBLEMA

Como se ha podido ver en la descripción general realizada, se entiende que entre las etapas (b), (c) y (d) referidas previamente, se ha desarrollado hipotéticamente un modelo de aprendizaje automático que idealmente no debería cometer errores al predecir nuevas entradas. Sin embargo, en la realidad los sistemas de aprendizajes basados en datos tienden a cometerlos. Incluso en el mejor escenario, el sistema desarrollado por cualquier vía pudiera arrojar resultados muy favorables al ser preciso, pero la interrogante obligada sería, si en general: ¿se puede confiar en esa *precisión*²?

En el sentido de responder esta inquietud, el interés de esta tesis se centró en estudiar la inclusión de información adicional que puede obtenerse junto a una

² El término precisión en esta tesis se refiere a la razón entre las muestras predichas correctamente por un modelo y la totalidad de las muestras procesadas.

predicción hecha por un modelo orientado a datos para indicar cuanto de creíble y de confiable puede ser el pronóstico conseguido por este.

Para ser más precisos en el tema que motiva la escritura de esta tesis, se comenta que en muchas aplicaciones críticas desarrolladas a partir de datos para resolver problemas asociados con predicciones, como es el caso de las *disrupciones*³ ocurridas en los dispositivos de fusión termonuclear, que ponen en riesgo la integridad de sus componentes que lo conforman [Vega et al 2013], una simple predicción hecha por cualquier modelo inferido a partir de datos para que avise la aparición de esta inestabilidad, no es muy útil si no se incluyen en el modelo, medidas adicionales que permitan evaluar la confianza y credibilidad.

Como referencia de este fenómeno físico (las disrupciones) así como muchos otros propios de la fusión, véase [Murari, 2015] para leer comentarios bien fundamentados del porqué no se cuenta con información a priori de la naturaleza del proceso o desconocimiento de primeros principios que originan los datos disponibles, o del porqué no pueden describirse los parámetros para generar un modelo fiable de dicho proceso que permita corroborar cualquier tipo de hipótesis hecha. También se puede consultar al respecto en [Farias et al, 2004], [Vega et al, 2009], [Castro, 2010], [Makili, 2014], [Pereira, 2015], [Martínez, 2017], [Farias et al, 2018], entre otros.

A partir del estudio y del análisis del trabajo [Murari, 2015] y la lectura de las referencias mencionadas en el mismo párrafo anterior, se puede abrir una discusión teórica para justificar el hecho de incluir medidas de fiabilidad en modelos desarrollados a partir de datos y con esto, asociar a cada predicción los niveles de confianza y credibilidad necesarios.

Pero siendo simplistas, la inclusión de medidas de fiabilidad se justifica cuando es imposible acceder a las propiedades de un sistema, porque éstas son una vía informativa con la que se pueden formular hipótesis que validan cualquier modelo de análisis de datos planteado (se recuerda nuevamente que este es el caso de la fusión termonuclear). Estas propiedades explicadas en mucha bibliografía tal como

³ Las disrupciones son inestabilidades y se dan cuando la alta energía concentrada del plasma es liberada súbita e inesperadamente.

[Russell y Norving, 2003], [Ogata, 2010], entre otros; se refieren a si *el sistema es: observable, determinista y estático*⁴ (por indicar algunas).

Una razón de fuerza, vinculante a todo lo expuesto, y que justifica definitivamente la inclusión de medidas de fiabilidad en el desarrollo de modelos orientados a datos, es el inconveniente de incertidumbre latente asociado tanto al momento de predecir (una vez que se ha conseguido el modelo) una muestra en particular procesada, como a la selección del conjunto de datos tomado como fuente para generar el sistema predictivo [Russell y Norving, 2003].

Para profundizar en este inconveniente de la incertidumbre latente (riesgo inherente) asociado al uso de modelos orientados a datos con su respetiva inclusión de medidas de fiabilidad, se mencionan los casos y posibles *cuestionamientos*⁵ relacionados a este tipo de soluciones:

- Casos donde se tiene poca información (conjunto de muestras insuficientes), no siendo viable generar un modelo predictivo convencional (por inducción: véase [Vovk, Gammerman, y Shafer, 2005], [Cherkassky y Mulier, 2007]). La construcción por inducción de la regla de aprendizaje, propicia el riesgo inherente por lo insuficiente que pueden ser los datos disponibles del conjunto de muestra seleccionado, imposibilitando una clara asociación del objeto sobre el cual se tiene la necesidad de predecir la etiqueta (clase) desconocida. Los cuestionamientos a plantear en este tipo de casos son:
 - 1) ¿Se puede inferir un modelo con poca información (conjunto mínimo de muestras) que vincule de manera efectiva la predicción de la etiqueta del objeto desconocido con el espacio de etiquetas que soportan los datos usados?.
 - 2) ¿La inclusión de medidas de fiabilidad garantizaría tal inferencia?.

⁴ Un sistema es: observable, si la información medida (salida del sistema o simplemente salida) permiten el acceso completo a los estados del sistema. Es determinista, si la información medida actualmente (estado) junto con la entrada del sistema permite una descripción del próximo estado. Y es estático, cuando la información actual (estado) depende sólo de la entrada actual. Se llama “estado” a la información o conocimiento que se tiene de las variables de un sistema en un instante determinado.

⁵ Cuestionar significa ‘poner en duda’ o ‘controvertir un punto dudoso’, tal como señala [REAL ACADEMIA ESPAÑOLA].

- Los casos donde los datos presentes para el procesamiento se obtienen a partir de medidas imperfectas debido a la *naturaleza del entorno*⁶ o contexto. En consecuencia, cualquier problema derivado de esto suele estar asociado al ruido presente al momento de medir los datos, o a imperfecciones de los sistemas de medidas, o intrínsecamente al proceso. A veces estos problemas son intratables [Pereira, 2015], pero otras veces simplemente no se puede percibir y es tarea del sistema de aprendizaje automático poder aprender (de los datos tratados) incluyendo este hecho, afianzando así, los niveles de incertidumbre inherente al momento de entrenar el sistema predictivo y su correspondiente procesamiento de nueva información [Makili, 2014]. Indudablemente, los modelos desarrollados bajo estas condiciones, no garantizan la asociación efectiva del objeto a predecir con su respectiva etiqueta (clase) desconocida, contenida en el espacio de etiquetas soportado por la información ruidosa (datos). Lo descrito en estos casos suma a la lista, los siguientes cuestionamientos:

3) ¿Esta imperfección en las medidas captadas (muestras ruidosas) para la construcción de un conjunto de datos, es particularmente perjudicial en la selección de un subconjunto de datos que permita inferir alguna regla de aprendizaje (modelo) válida?.

4) ¿La inclusión de medidas de fiabilidad minimizaría este riesgo al descartar muestras pocas informativas o simplemente extrañas para la inferencia del modelo?.

- Al contrario del primer caso planteado, se tienen aquellos procesos donde la información recogida es muy grande (manejo de datos masivos, véase [Pereira, 2015]) y muchas veces con ruido. Aquí el tiempo de procesamiento se puede extender, bien sea en la tarea de entrenamiento en el momento de desarrollar diferentes clasificadores e incluso para la evaluación del mismo. Además del tiempo necesario para el procesamiento del resto de los datos que deben ser tratados. Según esto se pueden añadir más cuestionamientos:

⁶ Se puede encontrar información sobre la naturaleza del entorno en [Rusell y Norving, 2003].

5) ¿Se puede escoger del conjunto de datos masivos, un subconjunto realmente representativo que permita inferir una regla que no sólo consiga asociar la etiqueta desconocida de un objeto particular con el espacio de etiquetas que soporta dicho subconjunto, sino que también sirva para asociar al resto de los datos medidos masivamente?.

En realidad este cuestionamiento tal vez es difícil de responder, sobre todo por la cantidad de datos a procesar para escoger un subconjunto mínimo representativo, que permita la generalización sobre el resto. En este sentido se plantea la siguiente cuestión alternativa:

6) ¿La introducción de medidas de fiabilidad para el procesamiento de muestra por muestra del conjunto completo, permitiría ver la evolución del nivel de fiabilidad general (confianza y credibilidad), pudiéndose con esto determinar el momento de escoger un subconjunto que mejore las condiciones globales de inferencia?.

Sobre esto, particularmente se ha observado en los trabajos revisados donde es usada la predicción conformal, la clara tendencia del uso de sus propiedades (condicionalidad, regiones de predicción anidadas y validez) para alcanzar un conjunto válido que garantizan predicciones con niveles de confianza aceptables.

Según esto y en respuesta inmediata a este último cuestionamiento y como consecuencia de la utilización de las propiedades de la predicción conformal, se podría plantear un mecanismo automático que observe la evolución de estos niveles de fiabilidad para decidir el momento en el cual se ha alcanzado el subconjunto de datos mencionado anteriormente. Estos subconjuntos, para el caso donde hay grandes volúmenes de datos e incluso en los procesos con poca información, deben garantizar un tiempo de entrenamiento mínimo.

Esta premisa de garantizar el tiempo de entrenamiento mínimo no es un caso de incertidumbre, pero en muchos procesos estudiados como las investigaciones en fusión termonuclear, actualmente existe la necesidad de reducir aún más los tiempos de entrenamiento en los modelos orientados a

datos que son usados, tanto en los procesos en tiempo real como en aquellos que no lo son.

Para entender la naturaleza de esta necesidad, se tiene que comprender que un modelo orientado a datos extrae información de los datos. Por lo tanto, consume tiempo al procesar una cantidad de muestras. Entonces, si se le suministra más datos más tiempo tardará en entrenarse, aumentando así el tiempo de entrenamiento. También puede ocurrir en los modelos orientados a datos que, al pasársele datos con información redundante, entonces tardará más tiempo en procesar tal información redundante que no aporta nada, pero que del mismo modo aumenta el tiempo de entrenamiento. Otro hecho que ocurre comúnmente, es cuando los procesos arrojan cantidades de datos pero susceptible a cambios en el tiempo. En este caso, es necesario actualizar los modelos orientados a datos ya creados. Esta tarea, pudiera consumir mucho tiempo, afectando negativamente el tiempo de entrenamiento de los modelos. Como puede verse, la premisa de garantizar el tiempo de entrenamiento mínimo no es un caso de incertidumbre, aunque si una necesidad presente en los modelos orientados a datos en la actualidad, y añade un cuestionamiento más:

7) ¿La introducción de medidas de fiabilidad para el procesamiento de un conjunto completo de datos, y el planteamiento del mecanismo automático propuesto con anterioridad para que observe la evolución de las medidas de fiabilidad; permitiría alcanzar un subconjunto de datos que minimice el tiempo de entrenamiento y actualización de los modelos?.

Ante este cuestionamiento, esta investigación también se interesa en estudiar: el tiempo consumido por los algoritmos para aprender (tiempo de entrenamiento), el punto donde ya los algoritmos no pueden incorporar más información al modelo (redundancia de aprendizaje de un modelo) y el tiempo en la cual los modelos se actualizan cuando existe nueva información (actualización de un modelo).

Para el caso del tiempo de entrenamiento invertido por un modelo para que aprenda, sería muy fácil imponer una regla de convergencia que mida cierto estado alcanzado, como el error cometido o tasa de aciertos al clasificar las muestras de un conjunto de test.

Sin embargo, la redundancia de aprendizaje y el tiempo de actualización de un modelo, aunque pudiera imponerse la misma regla de convergencia que garantizará mínimos errores o altas tasas de aciertos, sería poco práctico imponer tiempos para evaluar parte de un espacio de muestras y así optimizar el tiempo de entrenamiento. Por ejemplo, no se podría saber que muestras descartar porque no se conoce tal redundancia en información que será procesada por un modelo de aprendizaje, por lo tanto se tendrían que analizar todas las muestras y esto consumiría mucho tiempo.

Alternativamente, sí se pudiera medir cuantas muestras se han procesado antes de alcanzar cierto punto de convergencia hacia altas tasas de aciertos. A esto se le llama velocidad de aprendizaje. Por lo tanto, para estos dos últimos casos (redundancia de aprendizaje y tiempo de actualización), se puede usar la *velocidad de aprendizaje*⁷, que relaciona el número de muestras necesarias para que un determinado método de aprendizaje alcance altas tasas de acierto. En este sentido, también es de interés para esta investigación, estudiar la velocidad con la cual un modelo aprende.

Finalmente se puede incluir en estos tres casos comentados con antelación, el referido al:

- *Desbalanceo en el tipo de muestra*⁸ (clase) que está presente en muchos conjuntos de datos. De existir este desbalanceo al inferir cualquier modelo bajo los tres casos descritos, las cuestiones que surgen son las siguientes:

⁷ La velocidad de aprendizaje está relacionada con el número de muestras que se necesitan para entrenar un determinado modelo de clasificación y así obtener altas tasas de acierto. Se debe diferenciar del tiempo de entrenamiento, que puede verse como el tiempo que invierte un modelo de clasificación para extraer información de los datos y así converger a un estado donde hay un nivel mínimo de error o máximo de aciertos.

⁸ Se puede ver un estudio realizado sobre el desbalanceo del tipo de muestra con claras definiciones de este problema en [Dormido-Canto, Vega, et al, 2013].

7) ¿El desbalanceo de datos influye en la predicción hecha por el sistema inferido?.

8) ¿La inclusión de medidas de fiabilidad permite detectar esto?.

Sintetizando lo discutido hasta el momento, se tiene por una parte, todas las razones expuestas líneas atrás, incluyendo las encontradas en los cuatro casos analizados que propician un problema latente de incertidumbre al usar modelos orientados a datos, generando cuestionamientos que en gran parte han sido resueltos por la introducción de medidas de fiabilidad que evalúan un conjunto de datos y permiten seleccionar un número de muestras, garantizando la fiabilidad tanto del sistema predictivo inferido como de las predicciones hechas sobre el resto de la información que de manera excluyente y aleatoria siempre queda fuera del conjunto de entrenamiento.

Por la otra, la necesidad discutida de reducir el tiempo de entrenamiento en los modelos orientados a datos, con la finalidad de optimizar la generación y actualización de modelos. Para esto, las medidas de fiabilidad pudieran de igual forma aportar una solución favorable.

En general, las medidas de fiabilidad implantadas sobre los modelos orientados a datos, operan en base al condicionamiento de las muestras necesarias y usadas para la predicción de una muestra en particular (el objeto a predecir). El objeto a predecir, se convierte por decirlo de alguna manera, en el condicionante. Alrededor de esta se anida información referente al conjunto de datos usado para inferir el modelo que arroja definitivamente su predicción (la muestra particular), y al computar (estadísticamente hablando) toda esta información anidada, se solventa la incertidumbre fundada sobre dos aspectos globalmente importantes: (1) cuánto de diferente es la muestra particular a predecir del resto del conjunto de datos de entrenamiento usado para inferir el modelo, y (2) cuánto de representativo es este conjunto de entrenamiento del objeto a predecir.

Ampliando más el tema de la condicionalidad para una mejor comprensión, se añade que la misma es una propiedad presente en los modelos orientados a datos, como los divulgados bajo el enfoque de la predicción conformal. Para el caso de la

clasificación de datos (algo parecido funciona con los modelos de regresión), no sólo permiten indicar la clase (etiqueta) a la cual pertenece un nuevo objeto predicho, sino que aporta información adicional que vincula la muestra procesada con los datos que se usan para inferir el modelo.

Este aporte de información adicional o niveles de fiabilidad, se van incorporando en cada muestra analizada en línea: “del conjunto de entrenamiento se va tomando muestra por muestra hasta llegar al objeto que va ser predicho”. En teoría se anidan regiones de predicción hasta alcanzar una validez automáticamente [Vovk, Gammerman, and Shafer, 2005].

En la actualidad se estudian estas metodologías que permiten la construcción de modelos orientados a datos pero que implementan medidas de fiabilidad, sobre todo si se consideran los cuestionamientos que se han planteado, puesto que su aplicación minimiza el riesgo inherente en el momento de escoger a ciegas (selección aleatoria) las muestras necesarias para inferir un modelo, aumentando la certeza de que dicho conjunto es representativo tanto del resto de los datos que no son escogidos como de aquellos nuevos datos que pudieran procesarse.

Como referencias de este tipo de medidas implementadas, las cuales son motivación de esta tesis, Makili sintetiza en [Makili, 2014] un ampliado análisis hecho en este sentido, introduciendo la predicción conformal como el mecanismo metodológico que permite determinar la calidad en las predicciones realizadas. El mismo autor argumenta la utilización de este tipo de medidas para escoger un conjunto de datos mínimo de partida, que permite entrenar un sistema predictivo donde se garantiza que dicho conjunto asegura regiones de predicción confiables automáticamente.

Por su parte, en [Papadopoulos, 2008] se pueden encontrar comentarios sobre los valores de confianza y credibilidad arrojadas por la predicción conformal, necesarios y utilizados tanto para obtener una predicción como para indicar que tan adecuado son los datos escogidos para este fin.

Por lo que concierne a las observaciones (los datos escogidos) en el momento de generar una predicción, se puede encontrar en [Saunders, Gammerman y Vovk,

1999] una clara explicación de la aportación que da la medida de credibilidad obtenida en la predicción conformal para validar un buen conjunto de datos.

Adicionalmente se puede ver en [Vovk, Gammerman y Shafer, 2005], las propiedades de la predicción conformal, entre las cuales se destaca la condicionalidad que ya se ha introducido en este texto. Al respecto, los autores dejan ver aplicaciones donde se resuelven problemas de predicción (clasificación y regresión) frente a otras metodologías como los algoritmos bayesianos que necesitan conocimiento a priori.

En concreto, el concepto de la condicionalidad se refiere principalmente, a que sólo se necesita un conjunto de muestras escogidas aleatoriamente y una muestra a ser procesada (se debe hacer una predicción), y la predicción conformal construye alrededor de tal conjunto, una regla predictiva capaz de indicar a que clase pertenece dicha muestra analizada, generando adicionalmente medidas que indican cuan precisa ha sido la clasificación efectuada [Vovk, Gammerman y Shafer, 2005].

Tal vez la implicación más importante del concepto de condicionalidad, es que en el proceso de inferencia del modelo, se han generado regiones de predicción que están íntimamente ligadas a cada observación (muestras usadas) conjuntamente con la muestra a ser pronosticada. Esto, intrínsecamente permite que no exista una conclusión incierta sobre la predicción obtenida, sino que más bien dicha predicción se basa en las características de cada objeto visto.

Algo importante que también debe anotarse de la predicción conformal, es que ésta no necesita del conocimiento a priori [Rusell y Norving, 2003], [Papadopoulos, 2008]. Las únicas imposiciones para la construcción de la regla de inferencia son las siguientes: la suposición de que las muestras escogidas son generadas a partir de una distribución de probabilidades desconocida, cumpliendo esto con la hipótesis de que los datos son independientes e idénticamente distribuidos (IID) [Vovk, Gammerman y Shafer, 2005], y de un método tradicional de predicción (SVM, RNA, KNN, entre otros) sobre el cual se puedan medir las características del objeto a inferir del resto de las observaciones. En base al método tradicional de predicción escogido, se define una medida de conformidad que al ser computada, condicionará la inferencia obtenida.

Hay quienes puedan pensar en utilizar los modelos que miden la fiabilidad de las predicciones tan parecido como la predicción conformal (por ejemplo, los modelos bayesianos), pero que requieren entre otros supuestos, de información a priori o contexto de donde se obtienen los datos. Sin embargo y como argumento, véase [Vega et al, 2009], [Makili, 2014], [Pereira, 2015], [Murari, 2015], para comprender que tal contexto no siempre está disponible en muchos problemas como es el caso del estudio de la fusión termonuclear, motivo por la cual se ha ampliado su campo de estudio, siendo incluso tema objeto de investigación en este trabajo.

Ahora bien, la condicionalidad, propiedad que le da a la predicción conformal su ventaja potencial en aplicaciones donde se intenta resolver problemas predictivos, también forma parte de una relevante limitación que debe ser tratada sobre todo cuando se requiere procesar datos en tiempo real.

Esto ocurre porque introduce tiempos de procesamiento excesivos por el costo computacional consumido [Papadopoulos, 2008], debido a que hay que tratar cada muestra que se desea pronosticar junto con el correspondiente conjunto de entrenamiento (cada vez) para producir regiones de predicción anidadas hasta alcanzar la validez.

Esta limitación desde el aspecto computacional advertido, surge por el enfoque del aprendizaje transductivo definido en esta metodología [Vovk, Gammernan y Shafer, 2005]. Para inferir un modelo, se deben observar en línea (*aprendizaje “on line”*⁹) la secuencia de objetos correspondientes del conjunto de muestras usado para el entrenamiento, incluyendo la muestra a pronosticar.

Es evidente que este problema no lo sufren otros modelos predictivos tradicionales, porque usan un conjunto de datos para inferir una regla de aprendizaje y al obtenerla dejan a un lado (no usan) los datos empleados para tal fin

⁹ En el aprendizaje “on line”, se analiza en secuencia (una por una) las muestras de un conjunto de datos. Al final, se toma la muestra (nueva) a ser predicha y se analiza también. Tanto del conjunto de datos visto en secuencia y el nuevo objeto se extrae información. Opcionalmente, se define el aprendizaje “off line” donde se analiza el conjunto de datos completo con el objeto a predecir. La diferencia sustancial en ambos enfoques es la realimentación de la etiqueta verdadera predicha.

[Papadopoulos, 2008], trasladando el problema de predicción exclusivamente al modelo desarrollado.

Sin embargo, los *modelos predictivos tradicionales*¹⁰ que usan un enfoque inductivo-deductivo, tienen la desventaja de no aportar medidas de fiabilidad al momento de hacer una predicción para una nueva muestra, al menos que se le implementen (véase por ejemplo, [Platt, 1999]). También requieren de otros supuestos como el conocimiento a priori o contexto que genera los datos. Al mismo tiempo, necesitan de cierta cantidad de datos para trazar el modelo, lo cual los hace poco prácticos en procesos donde hay poca información. Y en contextos donde se generan mucha información pueden implementarse, pero introducen considerables retardos si tal información cambia, porque los modelos deben ser actualizados. Respecto a este punto, se puede encontrar más información en [Vovk, Gammerman y Shafer, 2005], [Russell y Norving, 2003], entre otras referencias.

Inciendo ahora más a fondo sobre el problema del coste computacional que tiene la predicción conformal, consecuencia de su enfoque de aprendizaje transductivo y por la propiedad de la condicionalidad, también se expone que su algoritmo está integrado por dos capas.

La primera capa, implementa una medida de no conformidad usando un método tradicional de clasificación (como por ejemplo: SVM, RNA, KNN), para generar valores que diferencian (medida de no conformidad) una muestra del resto de las procesadas. Estas medidas se suministran a la siguiente capa. La segunda capa, computa estos valores diferenciadores, generando a su vez una predicción sobre la naturaleza de la muestra procesada, dando así sus respectivas medidas de fiabilidad vital para estimar la calidad de la predicción.

Interiormente, la capa que introduce el inconveniente del coste computacional es la primera. Aquí el algoritmo tradicional llamado método subyacente, debe inferir de manera progresiva (*“on line”*). Aunque también puede ser de forma *“off line”*) y

¹⁰ Se entiende que un modelo de predicción desarrollado bajo el enfoque inductivo-deductivo, al hacer una predicción sobre una muestra particular indicará cierta tendencia o posibilidad, pero no da garantía que la predicción se cumpla para cada caso particular.

en forma transductiva, la regla capaz de medir en un espacio, cuánto de diferente es la muestra particular del resto de los datos usados para el entrenamiento.

En efecto, el proceso descrito es poderoso para hacer estimaciones y aporta información útil que precisa medidas de fiabilidad adicionales. Pero en la práctica computacional es donde se ralentiza el proceso. Además y pese a que se habla de dos capas que integran el mencionado proceso realizado por la predicción conformal, estas no son claramente separables. Sólo se entiende que la primera capa usa un método subyacente y es donde se produce el mayor coste computacional, siendo además este hecho dependiente del tipo de algoritmo subyacente definido, y del tamaño del conjunto de datos a utilizar. El inconveniente será crítico, cuando se requieran utilizar datos suficientes (tal vez un conjunto considerablemente grande) en procesamiento en tiempo real. En el Capítulo IV se analizará este hecho de manera práctica.

Según todo lo expuesto, la predicción conformal es conveniente para generar (inferir) modelos predictivos con medidas de fiabilidad, pero se debe tener en cuenta que tiene sus limitaciones como se ha podido ver en todo lo indicado. Por lo tanto, se deben implementar enfoques diferentes sobre su algoritmo (teóricamente expuesto), para garantizar la optimización en el sentido del coste computacional sin perder sus ventajas. Las preguntas obligadas que surgen alrededor de todo esto, son:

1. ¿Es posible implementar mecanismos en las capas de la predicción conformal que permita minimizar el tiempo de procesamiento sin perder la clara ventaja que esta metodología aporta sobre las predicciones hechas y sobre la forma de determinar la calidad de la predicción?.
2. ¿Estos mecanismos a implementar serán capaces de observar la evolución de los medidas de fiabilidad para medir: el coste computacional para generar un modelo, el punto donde ya no puede incorporar más información al modelo (redundancia de aprendizaje de un modelo), y el tiempo en la cual se actualiza un modelo cuando existe nueva información (actualización de un modelo)?.

La primera pregunta ha tenido respuesta en [Papadopoulos, 2008], [Vovk, Gammerman y Shafer, 2005], por indicar algunos autores que muestran el enfoque de la predicción conformal inductiva. La misma, usa un mecanismo de aprendizaje inductivo-transductivo para crear una regla predictiva capaz de agilizar el manejo de la información, optimizando todo el proceso al dividir en dos subconjuntos los datos disponibles para crear una regla predictiva. El primer subconjunto (conjunto de entrenamiento), se usa para construir una regla de aprendizaje por inducción. El segundo subconjunto, se usa para calibrar esta regla inducida en primera instancia. Luego, la muestra a predecir usa la regla inferida y junto con los valores que arrojan los datos de calibración (medidas de confianza), se consigue una predicción efectiva.

La segunda pregunta es más difícil de responder y por esto forma parte de los objetivos planteados en este estudio, donde se investigará sobre sus consecuencias prácticas.

Para el caso del coste computacional para generar un modelo, se entendió párrafos atrás, que los modelos orientados a datos están vinculados a la cantidad de información a procesar y se dijo que cuanto mayor es el número de muestras, mayor es el tiempo de cálculo. Esto indudablemente perjudica a los modelos tratados bajo la teoría de la predicción conformal, justificando esto cualquier análisis que corrija los inconvenientes, en especial las soluciones aportadas en el campo de la fusión termonuclear.

Tal solución podría obtenerse del análisis basado en un estudio sobre la cantidad de muestras usadas en la fase de entrenamiento y a la implementación de posibles mecanismos automáticos a plantear. Estos mecanismos pudieran observar los niveles de confianza que brinda la predicción conformal, dada por la propiedad de condicionalidad y las regiones de predicción anidadas, para permitir la validación de una cantidad de muestras mínimas.

Para la redundancia de aprendizaje de un modelo, se entiende que pudieran existir muestras que aporten la misma información, cayendo indiscutiblemente en el caso anterior, donde se tendrán más muestras que procesar. Este problema se podría también solucionar, si se vincula con la condición donde se alcanza la validez de un

modelo. Esto es una consecuencia de la propiedad de las regiones de predicción anidadas.

Para el caso de la actualización de un modelo, una nueva muestra evaluada con un modelo válido que arroja un nivel de fiabilidad elevado, sería un indicativo necesario y suficiente para incluir nueva información, teniendo por supuesto que salvar la redundancia de la misma si se tiene presente, las propiedades de: regiones de predicción anidadas y validez.

En resumen, ante todo el planteamiento realizado donde se incluye: (1) la necesidad de desarrollar modelos orientados a datos en el contexto de la fusión termonuclear, (2) la serie de cuestionamientos indicados sobre los modelos orientados a datos usados para resolver problemas de clasificación con inclusión de medidas de fiabilidad y referidos a la incertidumbre inherente y a la garantía de alcanzar tiempos de entrenamiento mínimo para la generación y actualización de modelos, (3) la solución aportada por la predicción conformal ante semejantes cuestionamientos con el inconveniente sobrevenido debido al alto costo computacional que la misma introduce y su consecuente solución dada por el enfoque de la predicción conformal inductiva, y (4) a las preguntas obligadas que surgen alrededor de una posible implementación de mecanismos automáticos capaces de observar la evolución de los medidas de fiabilidad para medir la velocidad de generación y actualización en modelos orientados a datos.

Ante esto, se propuso en esta investigación el estudio de los: “modelos de clasificación con medidas de fiabilidad basados en predictores conformales y aplicados a imágenes de fusión nuclear”, obtenidas a partir del diagnóstico de esparcimiento Thomson, que es un sistema que opera en el dispositivo de fusión termonuclear TJ-II para medir perfiles de densidad y temperatura en los plasmas de fusión [Barth et al, 1999], [Herranz et al, 2003].

Especialmente y a partir de estas imágenes que se corresponden con cinco tipos, correspondientes con la descarga del plasma de fusión nuclear, se plantea el objetivo de: “cómo obtener un multclasificador que con el menor tiempo de

procesamiento en la fase de entrenamiento, proporcione el mejor rendimiento en términos de tasa de aciertos y fiabilidad”.

1.3 OBJETIVO DE LA INVESTIGACIÓN

A partir del estudio de los “modelos de clasificación con medidas de fiabilidad basados en predictores conformales y aplicados a” cinco tipos de “imágenes de fusión nuclear”, se plantea el objetivo de “cómo obtener un multclasificador que con el menor tiempo de procesamiento en la fase de entrenamiento, proporcione el mejor rendimiento en términos de tasa de aciertos y fiabilidad”.

Las imágenes a utilizar se relacionan con medidas de densidad y temperatura en los plasmas de fusión nuclear y se obtienen a partir del diagnóstico de esparcimiento Thomson que opera en el dispositivo de fusión termonuclear TJ-II, ubicado en el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT) [Barth et al, 1999], [Herranz et al, 2003].

Del conjunto de imágenes se tienen: las BCKGND que miden el fondo de la cámara CCD (“background”), las COFF que se obtienen cuando la densidad del plasma calentado por el sistema de calentamiento por resonancia electrónica (ECRH) está por encima de la densidad de corte (“cut off” density), las ECH que se obtienen cuando los plasmas ECRH no sobrepasan la densidad de corte, las NBI que son captadas cuando los plasmas son calentados con haces neutros (“neutral beam injection”) y las STRAY que se obtienen sin plasma para medir la luz parásita.

Para cumplir con el objetivo propuesto, se realizará un análisis individual de los resultados referidos al rendimiento (tasas de aciertos así como la fiabilidad de las predicciones) y a los tiempos de procesamiento (tanto en entrenamiento como en pruebas de clasificación) de los multclasificadores a desarrollar a partir de modelos de predicción que aportan fiabilidad como es el caso de la predicción conformal (PC), la predicción conformal inductiva (IPC) y la predicción Venn (PV), frente a otros que no lo hacen como las máquinas de vectores soporte (SVM). Todo esto al variar sus parámetros de ajuste. Además de realizar un contraste de los resultados individuales

obtenidos para confrontar los tiempos de procesamiento y así verificar la velocidad de aprendizaje frente al rendimiento de los algoritmos bajo estudio.

A fin de cuantificar la fiabilidad en los multclasificadores a estudiar, se utilizarán medidas de confianza, credibilidad y el producto de la confianza por la credibilidad. Las mismas se medirán exclusivamente en los modelos inferidos por la PC y la IPC. Al mismo tiempo se usará la probabilidad e intervalos de probabilidades evaluadas en la PV.

En cuanto al acierto de los multclasificadores, este se cuantificará a partir de las tasas obtenidas en la clasificación de imágenes. También se utilizará el tiempo de cómputo que consume cada predictor al procesar un conjunto de imágenes, el número de imágenes empleado para su entrenamiento y el número de imágenes que deben procesar para alcanzar altas tasas de aciertos.

De igual forma, se tomará como variable el desbalanceo de datos, ya que el banco de imágenes tratado no contiene el mismo número por cada clase procesada. Con esto en mente y en función de ver el efecto que puede provocar éste desbalanceo al rendimiento, todo el estudio se conduce bajo dos grupos: uno con datos totalmente balanceados, y otro con el desbalanceo natural presente en la base de datos a utilizar.

Se indica finalmente, que los modelos generados a partir de la PC y la IPC, harán uso de la SVM como método subyacente bajo el algoritmo de no conformidad, presentado en [Vovk, Gammernan y Shafer, 2005]. Al mismo tiempo, se implementará la PV usando dos taxonomías diferentes: los k vecinos más cercano (KNN) y el centroide más cercano (CNN).

1.4 METODOLOGÍA SEGUIDA

En esta investigación y para todo el proceso experimental que da como resultado la construcción de modelos, se usa como base, las consideraciones hechas en [Cherkassky y Mulier, 2007]. Además, para la construcción de los diferentes modelos propuestos, se usan las cinco etapas que aparecen en la Figura 1.1 y resumidas en la Tabla 1.1, donde se explica a grandes rasgos los resultados obtenidos al cumplir con estas. También se indica, que la implementación de los programas que

permiten la generación de modelos a partir de datos y su posterior uso para la predicción de nuevas muestras, el mismo debe integrar: los componentes a ser aprendidos, el paradigma de realimentación a ser usado y el algoritmo de representación.

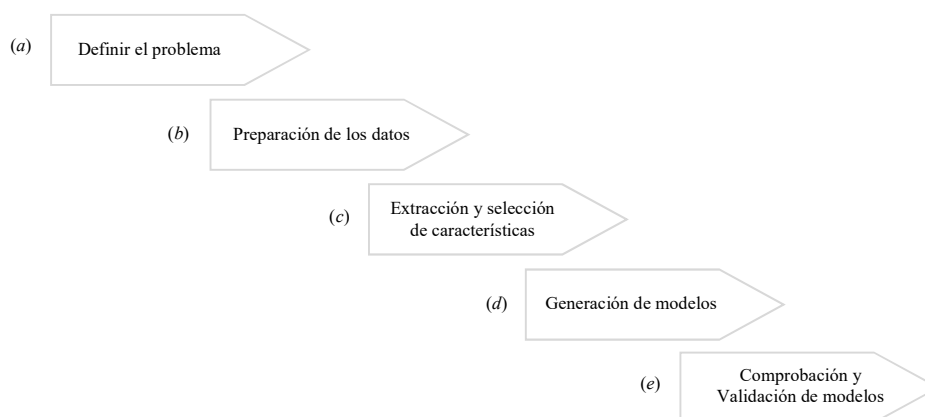


Figura 1.1.- Etapas para la construcción de modelos.

Controlar en teoría cada detalle de estos aspectos mencionados, permite una mejor planificación de las características y condiciones que finalmente serán codificadas. Sobre todo, porque para la generación de los modelos en esta investigación, las etapas de preparación de los datos y extracción y selección de características, son una constante. En ellas, se aplica una única vez la transformada wavelet [Farias et al, 2004], [Hill, 2002], constituyendo el pre-procesamiento de los datos (imágenes tratadas del dispositivo de fusión termonuclear TJ-II), *reduciendo la dimensionalidad, extrayendo y separando características en diferentes detalles*¹¹ (características AHVD).

Por tal motivo, en el momento de diseñar el código que incluye los componentes a ser aprendidos, el paradigma de realimentación a ser usado y el algoritmo de representación, todos estos, deben encajar con los datos a procesar.

¹¹ Un enfoque alternativo para reducir la dimensionalidad y extraer características de los datos usados en forma automática esta explicado en [Farias et al, 2018] donde se usa “aprendizaje profundo” (Deep Learning).

Tabla 1.1.- Resumen de las etapas para construcción de modelos.

Fases	Resultado obtenido
Definir el problema	Descripción general, planteamiento del problema, objetivos y formulación de las hipótesis. Permite una visión global de la planificación (diseño, desarrollo y prueba) de los experimentos (diagnósticos) que básicamente son programas sobre los cuales se ejecutan todos los principios teóricos estudiados. Esto incluye la planificación hecha para el tratamiento de los datos (imágenes) a usar, conduciendo al diseño y desarrollo de un banco que en definitiva es un programa que permite el acceso automático a cada imagen.
Preparación de los datos	El pre-procesamiento de las imágenes realizado con la transformada Wavelet, permitiendo reducir su dimensionalidad y selección de características propias de cada imagen al extraer y separar en detalles (atributos) aproximantes (A), horizontales (H), verticales (V) y diagonales (D). Todo esto se logra mediante el diseño, desarrollo (codificación) e implementación programas específicos que permiten el acceso automático y eficiente de cada imagen. Esta etapa también sirve para definir los datos intermedios manejados por todos los programas implementados que posteriormente usan el pre-procesamiento conseguido. Especialmente se le dan importancia a este tema, porque los códigos desarrollados para diferentes fines en el estudio hecho, generan una gran cantidad de datos (datos auxiliares) al procesar de las imágenes. Estos datos también se recogen dinámicamente y se guardan para su posterior uso.
Extracción y selección de características.	En el pre-procesamiento de cada imagen mediante la transformada wavelet, se extraen características informativas que quedan reducidos en detalles aproximantes (A), horizontales (H), verticales (V) y diagonales (D). Estos detalles son codificados automáticamente y guardados como vectores de características. Luego son seleccionados para la estimación del modelo.
Generación de modelos	La estimación de modelos fue el eje central de esta investigación. Su planificación y desarrollo, tenía que ser independiente del pre-procesamiento de las imágenes, debido a lo complejo de cada mecanismo de inferencia usado en los diferentes enfoques (SVM, PC, IPC, PV), ya que son un tanto diferentes con relación a su principio de funcionamiento. Por lo tanto, para la buena planificación y control en esta fase, por su reiterativa aplicación y por las diferencias en los programas a diseñar, desarrollar e implementar en cada caso, además de los datos a recoger para su posterior procesamiento (que en buena parte debían ser homogéneos por los contrastes a realizar); en la misma se consideran tres aspectos que son: (d1) los componentes, (d2) el paradigma de aprendizaje a usar y (d3) el algoritmo de representación.
Comprobación y validación de modelos	Esta etapa de la planificación se deja exclusivamente para el análisis de los resultados. También son el incentivo para escribir las conclusiones y las recomendaciones de esta Tesis.

1.5 ORGANIZACIÓN DEL TRABAJO

A la vista de los objetivos planteados, el presente trabajo de investigación se ha estructurado en cinco capítulos, las conclusiones y las líneas futuras de trabajo, quedando de la siguiente manera:

En el presente capítulo (Capítulo I) se ha introducido el problema a resolver. Comenzando con una descripción general hasta llegar al planteamiento del problema y seguidamente se menciona el objetivo de la investigación, finalizando con la metodología general seguida.

El Capítulo II se dedica a la fusión como marco de aplicación del trabajo de investigación, realizando inicialmente una breve introducción donde se distinguen la fisión y la fusión. A continuación, se aborda la fusión termonuclear y el confinamiento magnético, mencionando los problemas que la misma mantiene en la

actualidad, y se finaliza con una corta descripción de los componentes que tiene el dispositivo de fusión termonuclear TJ-II.

El Capítulo III se dedica al marco teórico, donde se explica en detalle toda la teoría necesaria para entender los experimentos a ser presentados en el Capítulo IV.

En el Capítulo IV se presentan los experimentos realizados con imágenes por los cuatro modelos principales explicados en el Capítulo III. Sus resultados se muestran en tablas y gráficos pero de forma independiente. Al final de cada experimento se podrá encontrar un análisis de los resultados.

El Capítulo V se dedica al contraste de los resultados obtenidos en el Capítulo IV. Adicionalmente se realizan varias discusiones que surgen del propio hecho experimental, consideradas importantes como un punto de referencia para futuras investigaciones.

Para finalizar, se indica que todos los términos y definiciones usadas en esta tesis que provengan del inglés, serán debidamente traducidos al español. Si en algunos casos esta traducción pudiera dar lugar a confusiones, se mantendrá el término en inglés y se hará notar.

CAPÍTULO II

2 LA FUSION

2.1 INTRODUCCIÓN

El crecimiento de la población humana en los últimos 70 años aumentó exponencialmente pasando de 2000 millones a 8000 millones de personas según las Naciones Unidas [Naciones Unidas, Departamento de Asuntos Económicos y Sociales, División de Población, 2019]. Esto aceleró el consumo en todos los niveles incluyendo el gasto en energía, y ha obligado a los gobiernos a buscar nuevas formas de explotar diferentes fuentes, como la energía que ofrece el átomo.

Al inicio del siglo pasado, ya se veía al átomo como un portador de energía ilimitada. De los muchos científicos e investigadores que trabajaron para descubrir su potencial, Albert Einstein fue quien advirtió su enorme poder al revelar la famosa ecuación $E=mc^2$ en 1905 [Cavero, 1989]. Pese a esto, no fue sino hasta mediados del Siglo XX cuando se construye el primer reactor de fisión nuclear en manos de Enrico Fermi, logrando en 1942 la primera reacción nuclear en cadena [Cavero, 1989]. Sin embargo, accidentes como el de Chernobyl en 1986 y el de Fukushima en el 2011, además de los residuos radiactivos que se producen tanto en la explotación como en el consumo; la consideran extremadamente peligrosa para el medioambiente [Aymerich et al, 2013], [Morones, 2012].

En cambio la fusión nuclear, comparativamente con la fisión es mucho más limpia y segura, pero ciertamente es muy difícil de controlar por los problemas que la misma envuelve aunque esto no ha evitado que se experimente con ella, sobre todo porque no genera reacciones en cadena que colapsan en una súbita explosión, siendo la alternativa futurística para tener energía ilimitada tan parecida como la que nos llega desde el sol pero producida artificialmente en la tierra [Ariola y Pironti, 2008], [Braams y Stott, 2002], [Sheffield, 1994].

Estas dos visiones: la fisión, aunque con sus problemas de inseguridad pero que ya en muchos países actualmente es explotada, y la fusión, segura y vaticinada su

futura producción por los especialistas; ambas ponen al átomo como la fuente ilimitada de energía.

2.2 LA FUSIÓN TERMONUCLEAR

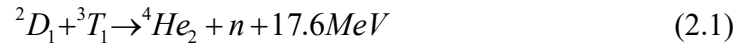
A diferencia de la fisión donde se separa el átomo, la fusión consiste en la unión de núcleos ligeros para formar núcleos más pesados, descargando una gran cantidad de energía en el proceso [Ariola y Pironti, 2008]. Este proceso que se consume en las estrellas a temperaturas muy elevadas, unos 15 millones de grados centígrados en el núcleo, hace que los átomos fusionados pierdan electrones formando de esta manera un gas que es llamado plasma, el cual es contenido por las fuerzas gravitacionales [Ariola y Pironti, 2008], [McCracken y Stott, 2005].

Este mismo proceso, los investigadores por décadas lo han estudiado hasta entender como producir la fusión y obtener plasma de fusión pero lo que no se ha podido lograr, es su producción controlada y mucho menos poder contenerla en algún sitio sin que el mismo sufra un colapso por las altas temperaturas que son superiores a las alcanzadas en las estrellas [Ariola y Pironti, 2008], [McCracken y Stott, 2005].

No obstante, en la actualidad hay una carrera contra el tiempo por conseguir energía a partir de la fusión de átomos de hidrogeno, siendo de los modelos teóricos el más prometedor, los obtenidos por la combinación del deuterio (D) y tritio (T) según [Ariola y Pironti, 2008], [Braams y Stott, 2002] y [Sheffield, 1994]. Esto se debe a su probabilidad de impactar al aumentar su energía cinética, venciendo de esta forma las fuerzas electrostáticas que impiden que dos partículas se acerquen [Ariola y Pironti, 2008].

Para entenderlo con palabras más simples, la condición necesaria para que dos núcleos se unan es aumentar su energía cinética, lo que se traduce en un aumento de la temperatura a cientos de millones de grado y es de esta forma que puede producirse una colisión, creando un núcleo más pesado, inestable y de alta energía. Al procurar un estado más estable (y de baja energía), el núcleo pesado creado por colisión tiene

que liberar energía (*fotones*¹²), neutrones y helio tal y como puede verse en la expresión (2.1) [Ariola y Pironti, 2008], [McCracken y Stott, 2005].



La energía liberada puede ser usada para continuar la fusión en cadena, logrando de esta forma la *ignición*¹³, pero en el tránsito de las partículas liberadas existen problemas de confinamiento debido a que las extremas temperaturas a la que se someten los núcleos hacen que los electrones se liberen de estos *isotopos*¹⁴ formando de esta manera el plasma [Ariola y Pironti, 2008].

Estos plasmas creados por el desprendimiento de los electrones debido a la alta temperatura necesaria para que se produzca la fusión (y de allí el nombre de termonuclear), son confinados en las estrellas o el sol por la atmosfera que ejerce grandes presiones (unas 250 billones de atmosferas en el sol) y facilita a su vez otras colisiones [Ariola y Pironti, 2008], [McCracken y Stott, 2005]. Esta condición de alta presión no puede ser reproducida en la tierra debido a muchas razones técnico-científicas que imposibilitan alcanzar tales atmosferas, constituyendo el confinamiento que esta propicia, uno de los principales problemas que tiene la fusión termonuclear en la actualidad [Pereira, 2015], [Yela, 2007].

Sin embargo, los plasmas tienen propiedades eléctricas por ser nubes de electrones y las mismas pueden ser atrapadas por campos magnéticos, requiriendo así la inversión de mucha energía con dos propósitos principales: (1) la de calentar los núcleos para que alcance altas temperaturas y su consecuente incremento de la energía cinética para que exista una colisión, y (2) la de crear campos magnéticos lo

¹² Los fotones son partículas mínimas (un fotón) de energía luminosa o de otra energía electromagnética que se produce, se transmite y se absorbe [Léxico Powered by Oxford]. Véase también [Soria, 2015].

¹³ La ignición es la condición donde la energía térmica del plasma es completamente sostenida por la energía liberada en la fusión nuclear [Ariola y Pironti, 2008].

¹⁴ Los isotopos son átomos que tienen el mismo número atómico pero diferente masa atómica. Contienen el mismo número de protones pero diferente en el número de neutrones [Léxico Powered by Oxford]. Por ejemplo, en el hidrógeno se contemplan 3 isótopos: el protio, el deuterio y el tritio. Véase también [Soria, 2015].

suficientemente fuertes para atrapar los electrones liberados en la fusión [Ariola y Pironti, 2008], [Braams y Stott, 2002].

2.3 EL CONFINAMIENTO MAGNETICO DE LOS PLASMAS DE FUSION TERMONUCLEAR

Para propiciar la colisión de dos núcleos de hidrogeno (D-T) en un laboratorio, es necesario aumentar la temperaturas a unos 100 millones de grados centígrados, produciendo esto una alta energía cinética en las partículas involucradas, las cuales pueden moverse libremente. Por esto es necesario contenerlas en algún sitio, sin embargo al hablar de temperaturas tan elevadas no existe ningún material que no pueda derretirse [Braams y Stott, 2002].

Como el plasma de fusión es un gas con propiedades eléctricas, puede ser empleado el *confinamiento magnético*¹⁵, que son líneas de campo magnético creadas por un imán. Evidentemente no cualquier imán puede crear una fuerza que atrape los electrones, y particularmente, el plasma de fusión que obtiene una energía cinética tan elevada. Es por esto que se emplean grandes solenoides [Braams y Stott, 2002].

Con esta idea, se diseñaron varios tipos de dispositivos que pueden conducir el plasma, creando las condiciones de confinamiento para dar lugar a las colisiones que provocan la fusión del hidrogeno. Entre estos dispositivos se tiene el TOKAMAK y el STELLARATOR que pueden verse en una representación visual en las Figuras 2.1 y 2.2 respectivamente.

En un TOKAMAK, se inducen corrientes eléctricas a través del propio plasma confinado. Además se añaden campos poloidales colocados tanto en la parte superior como en la inferior del dispositivo y a lo largo de la dirección axial del plasma [Ariola y Pironti, 2008], [Hernández, 2007]. Y en un STELLARATOR los campos magnéticos se colocan en forma toroidal y poloidal, generándose conjuntamente y de forma totalmente externa [Alejaldre et al, 1990], [Pereira, 2015]. Estos dispositivos con sus características muy particulares pueden confinar el plasma, sin embargo en la

¹⁵ Es importante aclarar que el confinamiento magnético no es el único método pensado para atrapar los electrones que forman el plasma creado en la fusión ya que se consideran otros procedimientos como el confinamiento inercial [Ariola y Pironti, 2008], [Makili, 2014].

actualidad presentan sus dificultades técnicas al presentarse eventos que no dejan que la fusión alcance su punto de ignición, entre estos se tiene el de las disrupciones y el colapso radiactivo por calentamiento. Ambos fenómenos entre muchos otros, son ampliamente estudiados en estos dispositivos.

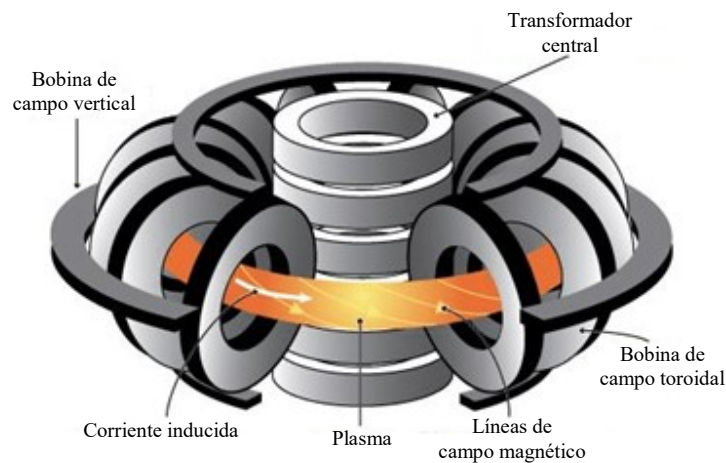


Figura 2.1.- Representación visual de un reactor de fusión tipo TOKAMAK.

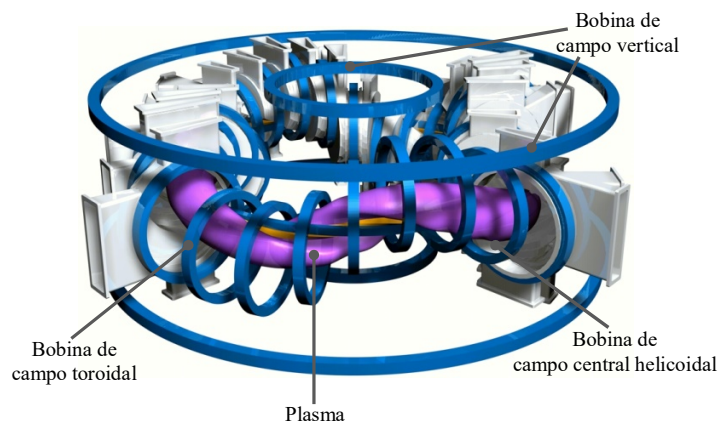


Figura 2.2.- Representación visual de un reactor de fusión tipo STELLARATOR.

2.3.1 Las disrupciones

Las disrupciones es un tipo de inestabilidad propio de un TOKAMAK y tiene lugar cuando el plasma está en un punto de alta energía, unos 100 millones de grados y repentinamente cae la temperatura en un orden de magnitud [Pérez, 2014]. Toda esta energía confinada por los campos magnéticos y movida por la corriente eléctrica

inducida en el propio plasma, termina en las paredes del recipiente que contiene al plasma causando graves problemas en su estructura [Pereira, 2015], [McCracken y Stott, 2005].

Es evidente que la condición de alta energía térmica y el propio confinamiento necesarios para que ocurran colisiones que permitan la fusión y su consecuente ignición, se pierde. Además, el plasma al chocar con las paredes del TOCAMA más que derretirlas, provoca grandes esfuerzos que se transmite al resto de la estructura [Reux, 2010], [Thornton, 2011], [Vega et al 2013].

Actualmente en el dispositivo de fusión JET en Inglaterra se estudia este evento para intentar establecer las causas que lo propician y sobre todo para en un futuro poder controlarlo [Boozer, 2012].

2.3.2 El colapso radiactivo por calentamiento

El colapso radiactivo por calentamiento es un efecto negativo que se produce en el plasma a una temperatura y densidad determinada, impidiendo que los pulsos de microondas emitidos por el sistema de calentamiento ECRH (véase la sección 2.4: sistemas de calentamiento) sigan aumentando la energía térmica. Este problema documentado en Yela [Yela, 2007] es capaz de afectar al tiempo de confinamiento, no permitiendo la ignición del plasma. En el mismo trabajo de Yela, se indica que el evento ocurre bajo ciertas condiciones de temperatura y densidad del plasma donde se alcanza la denominada densidad de corte para la cual las microondas (utilizadas para calentar) dejan de propagarse porque son reflejadas [Yela, 2007]. En [Sánchez et al, 2011] se puede encontrar el resumen de una de las investigaciones que proporcionan una guía para comprender las propiedades del confinamiento de iones rápidos, así como los efectos de las impurezas en los perfiles de radiación, señalándose dos tipos: la campana y la forma del domo, siendo este último el más propenso para que se dé el colapso radiactivo.

2.3.3 Otros eventos relevantes estudiados en el confinamiento magnético

No sólo se estudian las disrupciones y/o el colapso radiactivo por calentamiento, sino que son muchos los temas que hoy por hoy se investigan en el contexto de la fusión por confinamiento magnético, sobre todo porque sus hallazgos o resultados serán implementados en el nuevo dispositivo de fusión *ITER*¹⁶. Por ejemplo: las exploraciones de los requisitos de potencia de entrada para acceder al modo H a baja densidad [Martin y Takizuka, 2008]; el control de partículas, los efectos de configuración y estudios de transporte y fluctuación [Alejaldre et al, 2001]; inestabilidades magneto-hidrodinámicas, control de diferentes medidas de densidades como sus fluctuaciones, transporte y transiciones del plasma en el modo L-H [Sánchez et al, 2011], [González et al, 2010)], entre otros.

2.4 EL DISPOSITIVO DE FUSIÓN TERMONUCLEAR TJ-II

El TJ-II ubicado en el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT) en España, es un dispositivo de fusión termonuclear de tipo heliac flexible o STELLARATOR de tamaño medio donde se estudia el confinamiento, transporte de energía y partículas de los plasmas de fusión [Alejaldre et al, 1999], [Harris et al, 1985].

Según [Hender, 1987], su configuración fue desarrollada entre físicos e ingenieros del CIEMAT en colaboración con el Institut Fur PlasmaPhysik (IPP, Garching, Alemania), quienes pusieron a disposición un conjunto de bobinas que permiten variedades de configuraciones magnéticas y un amplio rango de valores de transformada rotacional [Yoshikawa, 1983], [Wakatani, 1998].

2.4.1 Componentes

Sistema de bobinas: El TJ-II está constituido por un sistema complejo de bobinas que permite el confinamiento del plasma dentro de la cámara de vacío. En la

¹⁶ El Reactor Termonuclear Experimental Internacional o ITER por sus siglas del inglés: International Thermonuclear Experimental Reactor, es un experimento científico de gran magnitud que permitirá producir plasma de fusión con una potencia térmica superior a la usada para su ignición. Puede encontrar información en <https://www.iter.org/>.

Tabla 2.1 se muestra el tipo de bobina, su cantidad y el tipo de campo magnético que genera [Hernández, 2007].

Tabla 2.1.-Sistema de bobinas del TJ-II.

Sistema de bobinas del TJ-II		
Cantidad	Tipo	Campo magnético generado
1	Circular	Poloidal
1	Helicoidal	Helicoidal
32	Toroidal	Toroidal
2	Vertical	Generan campos que mueven el eje del plasma
4	Compensación	Generan campos que cancelan corrientes toroidales
4	Radial	Generan campos radiales que compensan otros no deseados

Cámara de vacío: está constituida por 92 ventanas para observar el plasma y por donde se acoplan diferentes sistemas de diagnósticos. Su interior está recubierto de boro y puede operar con presiones por debajo de 10^{-6} Pa [Hernández, 2007], [Makili, 2014].

Sistemas de calentamiento: En el dispositivo de fusión TJ-II, la etapa de calentamiento se realiza mediante el empleo de dos técnicas.

- En la primera se usa el denominado sistema de calentamiento por resonancia electrónica o ECRH por sus siglas del inglés: electron cyclotron resonance heating, el cual emplea pulsos de microondas de hasta 1 segundo de duración, inyectando al plasma sobre los 300 kW de energía a la frecuencia de 53.2 GHz [Cairns, 1993], [Fernández et al, 2000].
- En la segunda técnica se inyectan gases neutros o haces de átomos neutros denominando al sistema de calentamiento como NBI por sus siglas del inglés: neutral beam injection. Los átomos de alta energía suministrados, al hacer contacto con el plasma (de baja energía) son capaces de transferir alrededor de unos 40 keV^{17} [Fuentes et al, 2005].

Sistema de diagnóstico: El TJ-II cuenta con un conjunto de instrumentos especialmente acoplados para medir diferentes variables del plasma y con esto poder

¹⁷ Según [Nieto, 2001], 1 eV es equivalente a $1.16 \times 10^4 \text{ K}$. El mismo autor señala que el criterio de Lawson (1957) establece que a una temperatura de 10 keV se produce la ignición en la fusión para diferentes condiciones de confinamiento.

conocer su estado. Entre estos sistemas de medición que son conocidos como diagnósticos, se tienen: los magnéticos, el de esparcimiento Thomson, los ópticos y espectroscópicos, de rayos X y bolométricos, de infrarrojos y microondas, de interacción plasma–pared, y de partículas y sondas [Sánchez, 1998].

Otros componentes acoplados al TJ-II: además de los mencionados, existen sistemas adicionales que permiten la adquisición y almacenamiento de datos, sistemas de alimentación eléctrica, entre otros [Hernández, 2007], [Makili, 2014].

2.4.2 El diagnóstico de esparcimiento Thomson

El diagnóstico de esparcimiento Thomson es uno de los sistemas de medida más importantes en los dispositivos de fusión. Se trata de un diagnóstico óptico no perturbativo que permite la determinación simultánea de densidad y temperatura electrónica con resolución espacial. El esparcimiento Thomson consiste en la reemisión de la radiación incidente en el plasma por parte de los electrones libres. La distribución de velocidades de los electrones se traduce en un ensanchamiento espectral de la luz dispersa que está relacionado con la temperatura electrónica (por efecto Doppler). El número total de fotones dispersos es proporcional a la densidad electrónica. Estos efectos son sólo medibles utilizando láseres muy potentes.

En el dispositivo de fusión termonuclear TJ-II, se opera de manera rutinaria un diagnóstico de esparcimiento Thomson [Barth et al, 1999]. En particular, el sistema de esparcimiento Thomson del TJ-II es un sistema espectroscópico bidimensional que utiliza un láser de rubí. Se mide la dispersión de la radiación a lo largo de 450 posiciones definidas por la cuerda del plasma que es atravesada por el láser. La inversión de las medidas determinan perfiles radiales con 90 canales [Herranz et al, 2003].

Como se ha comentado, el diagnóstico de esparcimiento Thomson en el TJ-II es un sistema espectroscópico bidimensional. Una cámara CCD capta la radiación dispersa con longitudes de onda entre 580 nm y 810 nm. A partir de esto, en el TJ-II se originan cinco tipos de imágenes [Makili et al, 2010]. Una de ellas es la que se obtiene para medir el fondo de la cámara CCD (“background”). El conjunto de

imágenes que miden el fondo de la cámara se las etiqueta como BCKGND. Un segundo tipo de imágenes es la que se obtiene cuando la densidad del plasma calentado por ECRH está por encima de la densidad de corte (“cut off” density). A esta clase de imágenes se las etiqueta como COFF. Los plasmas ECRH que no sobrepasan la densidad de corte originan una tercera clase de imágenes que se denominan ECH. Los plasmas calentados con haces neutros (“neutral beam injection”) definen un cuarto tipo de imágenes etiquetadas como NBI. Finalmente, hay imágenes que se obtienen sin plasma para medir la luz parásita. La etiqueta correspondiente se denomina STRAY.

CAPÍTULO III

3 MARCO TEÓRICO APLICADO

3.1 INTRODUCCION

La predicción conformal funciona a manera de capa al estar implementada sobre un método de predicción tradicional y eficientemente ha sido probada en numerosas aplicaciones para desarrollar modelos orientados a datos con el fin de resolver problemas de clasificación y/o regresión (se recuerda que en esta Tesis sólo se analizan modelos de clasificación). Pese a esto, se plantea el hecho donde deja de ser viable computacionalmente, dando paso a la predicción conformal inductiva que permite agilizar el tiempo de procesamiento y hace más eficiente el manejo de datos sobre todo en aplicaciones críticas.

Todas estas ideas ya fueron introducidas pensando en el problema central abordado en esta investigación relativa al desarrollo de “modelos de clasificación con medidas de confianza basados en predictores conformales y aplicados a imágenes de fusión nuclear” y requieren de la comprensión de varios planteamientos, definiciones, teoremas y otras consideraciones teóricas que serán presentadas en este capítulo.

Con relación a esto, se recuerda que todo lo presentado en los siguientes apartados no pretende en ningún momento sustituir la información técnica de la literatura especializada. En consecuencia, si el lector desea profundizar en cualquiera de estos temas que serán puntualizados, puede dirigirse a las referencias sugeridas a lo largo del texto.

3.2 LOS MODELOS

Un experimento, prueba o estudio (se usan indistintamente), es el proceso de extraer datos (o información) de un sistema sobre el cual se ha ejercido una acción externa [Moraleda y Villalba, 2016] con el sentido de estudiar sus posibles variaciones y en el campo de la industria a rasgos generales, es frecuente su uso con la intención de resolver problemas [Chapra y Canale, 2003].

Por ejemplo en fusión nuclear, uno de los tantos intereses está centrado en desarrollar modelos que permitan la predicción de disrupciones [Pereira, 2015]. Para ello se diseñan diagnósticos, que en resumen son experimentos donde se consiguen grandes bancos de datos que deben ser interpretados, no pudiéndose usar directamente la experiencia del investigador por la cantidad de información. Además, cualquier interpretación de los resultados de estos diagnósticos tendría que ser fiable para garantizar en un futuro cualquier sentido de optimización de este, o cualquier otro proceso estudiado [Olmedo, 2016].

Por lo tanto, si se quieren tomar decisiones para propiciar cambios positivos en un proceso como la fusión o cualquier otro sistema, no se puede esperar que los datos reales o experimentales indiquen que hacer y tampoco que indiquen donde está la falla o donde puede presentarse. Entonces sobre estos datos, se debe seguir experimentando para poder analizarlos exhaustivamente y así poder sintetizar soluciones concluyentes [Azarang, 1996], [Chapra y Canale, 2003].

Una de las formas de analizar estos datos es a través de la creación de modelos que capturan las características globales de los mismos y posteriormente pueden explicar o inferir sobre estos [Chapra y Canale, 2003], [Russell y Norving, 2003].

3.2.1 Fases de construcción de un modelo

Ahora bien, un modelo (matemático) describe un objeto que existe fuera del campo de las matemáticas. Por ejemplo, las predicciones del tiempo, los pronósticos económicos, un médico virtual dando un diagnóstico a un paciente, son ejemplos basados en modelos. El éxito o fracaso de esta representación numérica depende de la precisión con la que se construya, la fidelidad con la que se concreten hechos y las situaciones naturales en forma de variables relacionadas entre sí.

Es por esto que en el desarrollo de cualquier modelo, se involucran una serie de fases o etapas que a grandes rasgos se resumen en:

- La construcción, donde se convierte el objeto real a un lenguaje matemático. Hay que notar, que no se llega a construir un modelo sin un análisis previo del

problema a resolver, esto lleva mucho tiempo de planificación para definir lo que se debe realizar. Del mismo modo, la construcción necesita de una etapa de pre-procesamiento de los datos disponibles antes de generar el modelo.

- El análisis o estudio del modelo. En esta etapa se valida el modelo entre otras cosas, existiendo una diversidad de medidas implementables que buscan verificar si el modelo inferido procesa los datos con precisión. Por ejemplo, que la cantidad de muestras predichas correctamente por el modelo y la totalidad de las muestras procesadas sean muy cercanas.
- La interpretación, que será una aplicación de los resultados del análisis sobre el objeto del cual se parte.

Trazar un plan para el desarrollo de modelos como el comentado anteriormente, es una garantía para la obtención de buenos resultados en toda investigación, teniendo en cuenta que la cantidad de datos disponibles cada vez es mayor, convirtiéndose en estructuras difíciles de estudiar o de explicar, no pudiéndose en consecuencia delinear sus características sin la aplicación sistemática de algún tipo de representación.

Claro está, que definir un modelo “matemáticamente hablando” puede ser algo comprometido, así como su desarrollo; y dependiendo del campo de aplicación, esta tarea puede ser simple o compleja. Por lo tanto, en este Capítulo, el interés no es dar una explicación general de las diferentes fases que los investigadores pueden definir para la obtención de un modelo, ni de las etapas para la construcción de un sistema basado en el estudio de dichos modelos. Más bien se está interesado en el desarrollo de modelos orientados a datos basados en un *proceso experimental*¹⁸, cómo el enfoque adoptado por Dowdy y Wearden (1991), quienes son comentado en [Cherkassky y Mulier, 2007].

Este enfoque se enmarca en un plan que abarca las etapas de: definir el problema, preparación de los datos, extracción y selección de características,

¹⁸ El proceso experimental y las consideraciones en las cuales se basan las etapas para el desarrollo de modelos en esta investigación, se pueden ver en [Cherkassky y Mulier, 2007]. También puede ver las etapas definidas para desarrollar esta investigación en la sección 1.4, del Capítulo I de esta Tesis.

generación de modelos, comprobación y validación de modelos. Estas etapas ya fueron introducidas en la sección 1.4, en el Capítulo I de esta Tesis.

3.2.2 Tipos de modelos

Los modelos se pueden tipificar basados en ciertos criterios como por ejemplo, la procedencia de la información y en ese sentido se distinguen en: heurístico, si parten de definiciones de las causas o mecanismos naturales que origina algún fenómeno; o empírico, si el resumen de información procede de la experimentación.

También se pueden clasificar según los resultados obtenidos, conociendo así: los modelos cualitativos, cuando no existe un resultado en concreto sino más bien se reflejan ciertas tendencias. Los cuantitativos, que expresan un resultado preciso como por ejemplo, una fórmula matemática.

Si la aleatoriedad domina la obtención de los datos iniciales, esto da pie a tipificarlos como: modelos estocásticos, cuando en la respuesta devuelve la probabilidad de un resultado con cierto nivel de incertidumbre. Mientras que si los datos y los resultados se conocen en un dominio concreto, se dice que el modelo es determinista, careciendo de niveles de indecisión.

Al ver el tipo de respuesta y comprobar que la misma se restringe a un conjunto finito de valores; permite definir el modelo de clasificación. Por el contrario, si el conjunto de valores que arroja la representación es infinito, se habla de un modelo de regresión.

Finalmente se puede tipificar un modelo por el objetivo perseguido al momento de construirlo, concibiendo por esta vía: el modelo de simulación, cuando se quiera conocer un resultado por adelantado de una situación determinada. Modelo de optimización, usados para evaluar diferentes condiciones con el propósito de lograr la configuración más adecuada en un proceso estudiado. Modelo de control, parecido al anterior, pero los ajustes son para lograr un resultado específico en un proceso.

En general, los modelos son una parte esencial de cualquier sistema en la realidad. Una regla por más simple que parezca, si es capaz de relacionar variables de entrada o excitadoras del sistema con variables de salida o respuestas, creando una regla de dependencia entre las mismas y logrando de esta forma una representación o resumen del contexto y de su información (datos), es un modelo.

3.2.3 Definición general de un modelo de clasificación

Para comenzar, se puede definir a un modelo como una entidad que relaciona un conjunto de variables de entrada, características o atributos de un sistema real con algún tipo de conjunto de variables de salida, tal como muestra la Figura 3.1.

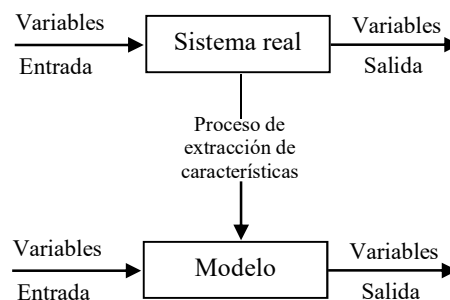


Figura 3.1.- Sistema y sus variables de entrada-salida.

El modelo o entidad emula el funcionamiento del sistema real al extraer características en un contexto determinado, consiguiendo que las salidas del mismo, se repongan al recibir un espacio de entradas similar o igual.

A partir de la Figura 3.1 se acotan varias consideraciones de interés. Estas son:

- El proceso de extracción se puede hacer bien sea por estudio del propio sistema o por análisis de los datos que este arroja al ser excitado.
- Para la extracción de características, se usan diferentes metodologías y dependerá en buena parte del campo de aplicación, del tipo de datos y del tipo de modelo a desarrollar entre otras consideraciones.
- El sistema y el modelo reciben las mismas excitaciones.

- Idealmente, ambos (el sistema y el modelo) deben responder de la misma forma. Cuando esto ocurre, se dice que el modelo es capaz de generalizar.
- Adicionalmente, con el modelo se puede predecir ciertas salidas para algún tipo de entrada que (probablemente) será colocada en el sistema real. Esto comúnmente, es un aspecto dentro del análisis de un sistema real que permite tomar decisiones sobre el mismo.

En el campo del análisis de datos, los modelos suelen usarse como parte esencial de un sistema automático que permite el manejo de los mismos de forma eficiente, especialmente cuando la cantidad (de datos) es grande. Estos sistemas automáticos, al mismo tiempo se conciben mediante la implementación de técnicas de aprendizaje automático, que hoy por hoy conforma un campo de estudio multidisciplinario que utiliza métodos, procesos, algoritmos, entre otros elementos; para extraer conocimientos con el propósito de desarrollar sistemas inteligentes que puedan dar respuestas con las cuales se tomen decisiones.

Definición del modelo de clasificación: es una de dependencia capaz de relacionar un espacio de objetos que gozan de rasgos, atributos o características (se hablará indistintamente) diferenciables con un correspondiente espacio de etiquetas.

Idealmente el modelo de clasificación o clasificador (también se le podrá llamar sistema de clasificación) como se le llamará en adelante, separa la colección de objetos al ver (analizar) estos rasgos, agrupándolas en clases comunes. Cada objeto dentro de un grupo tendrá la misma etiqueta. Este proceso de separación ideal se puede ver en la Figura 3.2.

Nótese, que el espacio de objetos está representado por todas las figuras geométricas (triángulos, cuadrados y círculos). Cada objeto contiene rasgos diferenciables (por ejemplo: color y forma geométrica). El clasificador toma un objeto y define una primera clase (lo separa), por ejemplo un triángulo. A continuación toma un segundo objeto y compara los rasgos de éste con el primero, teniendo en cuenta los atributos comunes (forma geométrica y color). Sí al comparar estas características son iguales, lo agrupa con el primero. Sí se diferencian entonces lo separa de la primera clase ya definida, creando así una segunda clase.

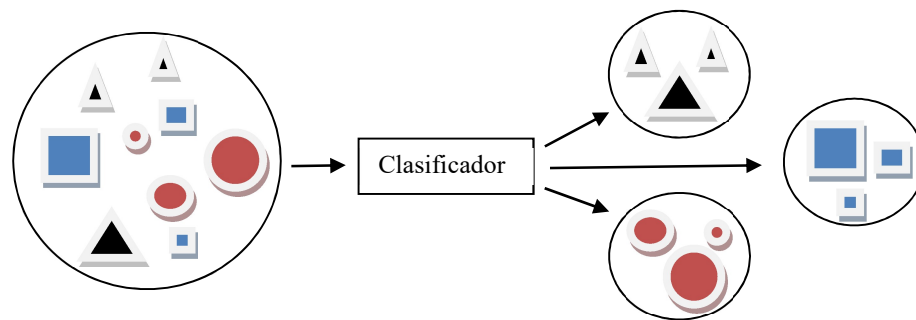


Figura 3.2.- Separación de objetos a partir de sus atributos mediante un clasificador. Dos atributos diferenciadores: color y forma geométrica.

Este proceso continuará hasta que no haya más objetos que separar, conformando de esta forma tres clases bien definidas (triángulos negros, cuadrados azules y círculos rojos). Es importante observar que el clasificador pudo separar cada objeto tomando como criterio un sólo rasgo (forma geométrica o color). Incluso, pudo tomar como criterio otra característica como por ejemplo, el tamaño de la figura (grande, mediana, pequeña).

Comparativamente, una persona de manera intuitiva haría esto mismo. Claramente que en el proceso previo antes de separar los diferentes objetos en clases, la persona debe aprender a distinguirlos. Este aprendizaje dependerá de los objetivos que finalmente persigue la persona; en todo caso debe definir lo que quiere hacer.

Lo importante de la comparación hecha entre el proceso que realiza el clasificador y la persona (un humano), es poder distinguir que la base para poder separar los objetos parte de: un plan que debe ser definido y del proceso de aprendizaje que finalmente da la capacidad de separar los objetos.

Con relación al plan a seguir, el mismo se puede orientar dependiendo del campo de aplicación, en todo caso se plantearía una metodología para conseguir los resultados deseados según todo lo visto al inicio de esta sección.

Con relación al proceso de aprendizaje o entrenamiento como suele llamársele, el clasificador será capaz de conseguir lo que hipotéticamente puede realizar una persona, dotarse de la capacidad para separar los objetos en clases. Claro está, que se debe someterse a dicho proceso, tal cual como un humano se ponga a

estudiar las formas geométricas para distinguir sus rasgos. El clasificador, al someterse al proceso de entrenamiento extrae rasgos que posteriormente usará en las clasificaciones efectuadas (predicciones). Comparativamente con el humano, éste crea conexiones sinápticas [Olmedo, 2009], que luego podrá utilizar en la clasificación (separación de las figuras geométricas).

En el caso del humano, estas conexiones sinápticas pueden perdurar en su estructura biológica cerebral o red neural biológica [Olmedo, 2009]. En un clasificador, las estructuras son numéricas y dependerán de los atributos escogidas según el propósito perseguido [Olmedo, 2010].

Por ejemplo, en el clasificador de la Figura 3.2 que recibió los triángulos negros, cuadrados azules y círculos rojos, y los separa en tres clases bien definidas; se pudiera pensar que el mismo fue entrenado hipotéticamente por dos de sus atributos ya definidos (color y forma geométrica). Incluso por uno sólo, porque claramente las figuras pueden separarse bien sea por su forma geométrica, por su color o incluso por su tamaño. Lo único cierto es que esta tarea se logra al entrenar el clasificador con al menos uno de estos rasgos diferenciadores.

En la Figura 3.3 se muestra otro ejemplo, donde los objetos son claramente diferenciables pero únicamente por su color. Hipotéticamente, ahora se ha entrenado al clasificador con este atributo, consiguiendo separar en tres nuevas clases todas las figuras geométricas.

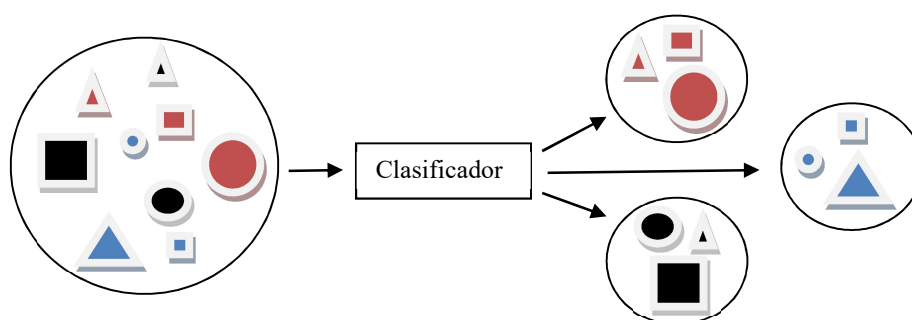


Figura 3.3.- Separación de objetos a partir de sus atributos mediante un clasificador. Sólo un atributo diferenciador: color.

Además de todo lo explicado, en estos ejemplos introductorios se pueden ver otros elementos básicos que conforman a un modelo de clasificación, como son: los datos de entrada u objetos y la información de salida o etiquetas predichas.

3.2.4 El modelo. Los objetos y las etiquetas

La entrada del sistema de clasificación (el modelo) es un vector de atributos o características (variables), que para el caso particular de este trabajo es numérico. Sin embargo y de forma general estos (los atributos) pueden ser también nominales al tomar valores dentro de un conjunto finito. Este conjunto de atributos o características conforman un objeto x_i .

Matemáticamente, se puede definir al objeto x_i como un vector de atributos a_k donde $k=1, 2, 3, \dots$, será el número de atributos o características. En el campo del análisis de datos e incluso en el campo del estudio de sistemas, este número k de atributos suele ser la dimensión del objeto. Cada dimensión puede tomar determinados valores dependiendo del proceso estudiado. Si los datos son tomados de un proceso mediante una medición, por lo general estos atributos corresponden con variables del proceso (temperatura, presión, entre otras) denominando de esta forma al objeto x_i como estado.

En el caso especial de esta investigación, la medición del objeto corresponde con imágenes relacionadas al contexto donde se origina el plasma en un dispositivo de fusión termonuclear. Estas imágenes se comportan como un objeto matemático y poseen un número de atributos a_k relacionados con la ubicación espacial de un pixel. Este espacio de atributos simplemente se denotará por A , tomándose exclusivamente como referencia para los fines de las definiciones que están por desarrollarse.

En consecuencia, para lo que concierne a este trabajo es suficiente con indicar que la entrada de un sistema de clasificación está conformada por la colección de objetos x_i con $i=1,2,3, \dots, n$ y forma un espacio medible de objetos X , es decir: $x_i \in X$. Para soportar el concepto de dimensiones se dirá que el espacio medible X puede encontrarse en R^d , es decir $X \in R^d$.

Cada objeto x_i se refiere a una salida o etiqueta y_i relacionada perfectamente mediante una función f . Es decir, para un objeto x_i dado se puede tener una etiqueta y_i mediante la expresión:

$$y_i = f(x_i) \quad (3.1)$$

Donde f es una regla de dependencia o estrictamente el modelo capaz de relacionar ambas instancias. Este modelo f no siempre se conoce a priori y en el caso de muchos procesos sólo se tiene acceso a excitarlo y con ello medir su respuesta. Tanto la excitación como la respuesta que da el sistema conforman lo que se llama una observación. Es decir, una observación completa está conformada por su entrada y su salida. El conjunto de todas las observaciones se llamará datos disponibles y pueden ser empleados en un proceso de entrenamiento, que por lo general son divididos en un conjunto de entrenamiento propiamente y un conjunto de prueba.

Con referencia a y_i esta pertenece a un espacio de etiquetas finito. En el caso de los problemas de clasificación se escribirá como: $y_i = \{c_1, c_2, \dots, c_j\}$. La idea general de este espacio de etiquetas se formalizará con C .

3.2.5 El modelo de clasificación binario y el multclasificador

Se dice que f es un modelo de clasificación binario cuando sólo es capaz de separar en dos clases (o etiquetas) posibles a un conjunto de objeto x_i dados, es decir: $y_i \in C$, tal que, $C = \{c_1, c_2\}$, donde c_1 y c_2 son las dos clases posibles.

Cuando f es capaz de definir más de dos espacios para los diferentes objetos x_i que separa, se dice que es un multclasificador. En este caso: $y_i \in C$, tal que, $C = \{c_1, c_2, \dots, c_j\}$.

3.2.6 Los datos en el entrenamiento de un modelo

En general, se puede referir a los datos como el conjunto de observaciones z_i , previamente definido que forman un espacio de objetos x_i , medidos y su correspondiente etiqueta y_i . En consecuencia una observación siempre será el par

(x_i, y_i) . Es decir $z_i = (x_i, y_i)$. Estas observaciones z_i permiten inferir un modelo f a partir de un problema de clasificación con lo cual se estima un nuevo objeto x_{i+1} dentro del espacio de etiquetas C . Es decir, el objeto x_{i+1} quedará perfectamente clasificado si el valor \hat{y} predicho pertenece al espacio de etiquetas C . Donde \hat{y} es valor esperado y_{i+1} correspondiente con el objeto x_{i+1} .

Esto implica que el modelo de clasificación, estimador o predictor f es una aplicación $f: A \rightarrow C$, indicando que partir de un conjunto de objetos x_i (que contienen a A) se ha inferido el modelo de clasificación f capaz de estimar la etiqueta \hat{y} perteneciente a C , y definiendo en general al predictor como:

$$\hat{y} = \hat{f}(x_i) \quad (3.2)$$

Obsérvese que la idea del modelo inferido se ha formalizado con \hat{f} que en realidad es una aproximación del modelo desconocido f . En general se quiere indicar que \hat{f} puede estimar una etiqueta \hat{y} contenida en C porque fue sometida a un proceso de entrenamiento mediante un conjunto de datos disponibles Z_i .

Evidentemente, si se utilizan todas las observaciones disponibles para el entrenamiento de un modelo, por consecuencia, no se dispondrían de datos para probar si la estimación del modelo es válida. Este caso ocurre comúnmente en procesos en línea, donde no da tiempo de validar el modelo teniendo necesariamente que imponer medidas de fiabilidad.

En la mayoría de los casos ajenos de los procesos en línea, el conjunto de datos disponibles z_i se divide en dos partes:

- **Conjunto de datos de entrenamiento:** del conjunto z_i con $i=1, \dots, n$, se toman aleatoriamente el sub-conjunto $z_k = \{z_1, \dots, z_m\}$ con $k=1, \dots, m$, con $m < n$, quedando el sub-conjunto z_{k+1} .
- **Conjunto de datos de prueba:** estará formado por el sub-conjunto de datos restante $z_{k+1} = \{z_{n-m+1}, \dots, z_n\}$ con $k=n-m+1, \dots, n$.

También es posible conocer las probabilidades a posteriori para las clases del espacio de etiquetas C a partir del modelo \hat{f} dado un espacio X de objetos

determinado. Es decir, si generaliza a x_i como simplemente x , se puede obtener $f(x)=(P(c_1|x), \dots, P(C_j|x))$. Sin embargo, ya se ha discutido en el Capítulo I los inconvenientes de estimar el modelo f .

3.3 EL APRENDIZAJE

El aprendizaje en general, es la capacidad de extraer (mapear) características de un entorno, para luego y basadas en estas, poder dar una respuesta al propio entorno. Esta respuesta puede ser favorable o no y dependerá de la cantidad de información procesada. Estas características se obtienen como estímulos que entran a través de los sentidos y son memorizados en el caso de un humano, para luego responder de manera satisfactoria cuando son percibidas nuevamente, bien sea de forma parcial o total.

En el caso del cerebro humano, se dice que se han memorizado. Sin embargo, algunas investigaciones han planteado que realmente se crean conexiones sinápticas [Olmedo y Chang, 2010], formando una especie de mapa mental gracias a una experiencia significativa dentro de un entorno concreto. El proceso tras esta experiencia significativa se denomina aprendizaje, y básicamente se da por el procesamiento de la información (del entorno) que va dejando rasgos (conexiones sinápticas) que posteriormente ayudan a reconstruir dicha información, pudiendo darse una respuesta efectiva ante el mismo entorno e incluso extrapolarse a otros contextos.

Este comportamiento, desde los inicios de la inteligencia artificial se ha buscado de emular para el desarrollo de máquinas capaces de aprender un entorno concreto, un proceso e incluso un sistema, y poder operar en el mismo. Específicamente, de un entorno (un proceso o un sistema) bajo estudio, se obtienen datos o información que son el homólogo de los estímulos que recibe un humano y al someter a una máquina (un modelo) con estos datos mediante un mecanismo definido de “aprendizaje”, se forma una regla o modelo que recoge o extrae las características generales embebidas en los datos.

En muchos casos esta regla se puede crear bajo programación, pero para ello se debe estudiar de forma sistemática la información del dominio concreto e incluso estudiar el proceso o sistema desde dentro (principios fundamentales), para poder lograr que la máquina opere y tome decisiones de forma autónoma. Está claro que en muchos casos se sigue esta metodología, pudiendo ser muy efectiva. Otras veces por lo complejo del entorno donde se generan los datos e incluso por la cantidad de información y por el ruido presente; esta regla no puede obtenerse bajo un estudio tradicional y tampoco es válido recurrir a primeros principios porque simplemente no existen, como es el caso del estudio de la fusión [Murari, 2015]. Estas razones dan pie para que entre en juego el término de aprendizaje “Automático”.

3.3.1 El aprendizaje automático, la teoría del aprendizaje estadístico y el conocimiento a priori

La breve descripción con la que finalizó el párrafo anterior deja ver que el aprendizaje se puede programar, pero para ello se requiere un amplio estudio. Sin embargo, uno de los principales objetivos que ha perseguido siempre la inteligencia artificial [Russell y Norving, 2003] es buscar que las máquinas aprendan por si solas. Para ello, este campo de investigación aplicada e interdisciplinaria, ha estudiado, mejorado e incluso, refinado diferentes metodologías y enfoques para lograr tal fin, sólo necesitando datos (observaciones) e información del entorno (hipótesis) donde se generan dichos datos. Claro está que en ocasiones se tienen los datos pero no la información del contexto (conocimiento priori).

Uno de los principales enfoques que está basado en datos e hipótesis, es la teoría del aprendizaje estadístico [Valiant, 1984], [Vapnik, 2000], [Russell y Norving, 2003], [Cherkassk y Mulier, 2007]. A diferencia del enfoque estadístico tradicional para el desarrollo de modelos que busca entre otras cosas la identificación de parámetros y prueba de hipótesis para identificar un sistema desconocido, esta visión (la teoría del aprendizaje estadístico) intenta aproximar el sistema desconocido a partir de un conjunto de datos tomado directamente del contexto o proceso, y con esto genera un modelo. Lógicamente, el modelo depende de los datos usados para

inferirlo, siendo deseable en todo caso, que las muestras escogidas representen la universalidad de todos los datos posibles. De ser así, el modelo generado, regla de predicción o hipótesis como se define en la literatura especializada, podrá producir predicciones con altas probabilidades de acierto.

Para que esto ocurra, además del requisito de los datos medidos directamente del contexto, la teoría del aprendizaje estadístico requiere que tales datos cumplan con ciertas suposiciones que también deben ser probadas. Por ejemplo: si son del tipo independiente e idénticamente distribuidos (IID), si provienen de una misma distribución, si son intercambiables, entre otros.

Además de esto, también es requerido el denominado conocimiento a priori y deseable al aplicar cierto tipo de metodología para inferir el modelo. Este conocimiento a priori son hipótesis de cómo se generan los datos en un entorno específico (cómo funciona el dominio).

Para entender el significado del conocimiento a priori, imagínese a un observador mirando los jóvenes que salen de un colegio. Si sólo ve que salen chicas, pudiera inferir que es un colegio de monjas. Indudablemente, el conocimiento a priori hubiese informado al observador de tal realidad. Pero sin este conocimiento, el observador sólo puede hacer conjeturas (hipótesis). Una buena hipótesis que pudiera plantearse es la tesis de que el colegio es de monjas. Sin embargo, que tal si el día que decidió observar, todos los chicos de la escuela estaban enfermos o fueron castigados en la dirección del plantel. Algo extremo, sería pensar que en la ciudad no hay jóvenes estudiantes.

Volviendo al tema del conocimiento a priori, su existencia o no define dos vertientes en este campo específico de la generación de modelos orientados a datos. Estas son: los métodos que requieren de conocimiento a priori y los métodos que prescinden de hipótesis alguna ya que operan sobre el conjunto de datos medidos directamente del sistema que se quiere aproximar.

Como es de esperar, el presente trabajo de investigación se centra en el desarrollo de modelos orientado a datos donde no existe este conocimiento a priori.

3.4 PARADIGMAS DE APRENDIZAJE

En las secciones anteriores, ya se ha introducido el aprendizaje automático y se ha definido como: la capacidad que tiene una máquina de aprender de un entorno. En el caso de un humano, esta capacidad se ve plasmada en las conexiones sinápticas que desarrolla a nivel de su red neural biológica [Olmedo, 2009]. Además y como aseguran algunos investigadores, tales conexiones se crean por un mecanismo biológico-químico-eléctrico de realimentación [Olmedo, 2009], entre otras teorías.

Para el caso de una máquina, el aprendizaje se construye (y se plasma) a partir de un algoritmo de representación básico como: las redes neuronales artificiales [Kosko, 1992], [Freeman y Skapura, 1993], [Olmedo, 2009]), las máquinas de vectores soporte [Boser et al, 1992], [Cortes y Vapnik, 1995], [Vapnik, 2000], los algoritmos de agrupamiento como el k vecino más cercano [Russell y Norving, 2003], [Cherkassky y Mulier, 2007], o incluso con algoritmos avanzados como es el caso del aprendizaje profundo [Farias et al, 2018].

También pueden emplearse como se verá en esta Tesis, algoritmos mixtos que trabajan en capas como la predicción conformal, la predicción conformal inductiva o la predicción Venn [Vovk, Gammerman y Shafer, 2005]. En adelante se denominarán modelos con medidas de confianza. De todos estos algoritmos mencionados, los utilizados en esta investigación serán definidos y analizados con un mayor detalle en las siguientes secciones.

De forma análoga a un humano que tiene mecanismos biológico-químico-eléctrico de realimentación, el algoritmo de representación usado por una máquina siempre estará dotado de algún mecanismo de realimentación de la información o paradigma de aprendizaje, que les permite ir adquiriendo progresivamente información o rasgos que modulan el tipo de respuesta que debe dar ante algún tipo de entrada. Es decir, el aprendizaje de una máquina depende del entrenamiento que recibe. Este tipo de entrenamiento o paradigmas de aprendizaje son el aprendizaje supervisado, el no supervisado y el aprendizaje por refuerzo [Kosko, 1992], [Freeman y Skapura, 1993], [Russell y Norving, 2003], [Acosta y Zuluaga, 2000], [Olmedo, 2009].

3.4.1 Aprendizaje supervisado

Este tipo de aprendizaje ha sido de los paradigmas más usados. Básicamente está constituido por un algoritmo entrenador que orienta el aprendizaje, informando de los errores, estimulando los logros y además decide el momento de concluir el entrenamiento, la intensidad y frecuencia con la que se presentan los ejemplos (observaciones o datos). El mecanismo por decirlo de alguna manera, permite guiar la respuesta que da el sistema hacia un patrón determinado que ha sido presentado al modelo. Es decir, busca minimizar la diferencia entre la salida del modelo y el patrón presentado (observación). El aprendizaje supervisado, suele usarse en aquellos casos donde se conoce ya la respuesta. También es usado para aproximar funciones donde se conoce sólo un conjunto de datos (interpolación), pero no su modelo matemático. En la Figura 3.4, puede verse un diagrama esquemático de un sistema de entrenamiento supervisado.

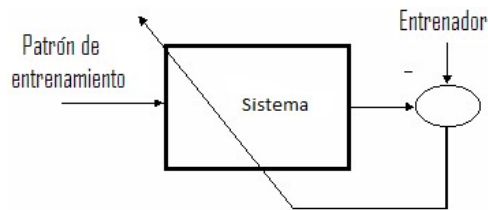


Figura 3.4.- Diagrama esquemático del aprendizaje supervisado.

El algoritmo general del aprendizaje supervisado se basa en los siguientes pasos:

- Presentación de entradas (observaciones) al modelo.
- Producción de una salida (respuesta) por parte del modelo.
- Comparación entre la salida del modelo con la respuesta deseada, estableciendo una medida de error que permite actualizar los parámetros del modelo. Estos parámetros se modifican hasta que la salida del modelo converja hacia la respuesta esperada.

3.4.2 Aprendizaje no supervisado

Este tipo de aprendizaje trabaja solamente con información local. El conjunto de datos de entrenamiento consiste sólo en los patrones de entrada. El mecanismo en este caso busca minimizar el error entre las respuestas dadas y la esperada tratando de descubrir relaciones, similitudes o interconexiones entre los patrones que se tienen para el entrenamiento. Se utiliza en casos donde no se tiene una clasificación establecida o cuando se quiere descubrir relaciones entre los patrones (auto asociación). Es decir, los modelos bajo este enfoque aprenden por adaptación basada en la extracción de patrones de entrenamientos anteriores (véase la Figura 3.5).

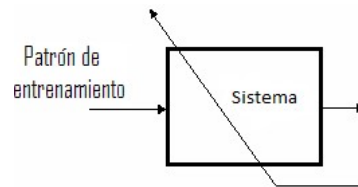


Figura 3.5.- Diagrama esquemático del aprendizaje no supervisado.

3.4.3 Aprendizaje por refuerzo

El mecanismo de aprendizaje por refuerzo funciona parecido al aprendizaje supervisado, salvo que el algoritmo entrenador sólo realimenta en casos particulares. Es decir, el entrenador no realiza modificaciones continuamente por cada respuesta que se obtiene, sino que ajusta los parámetros del modelo cuando ocurre algo específico, funcionando como una especie de penalización. En la Figura 3.6 se observa esta penalización como una especie de conmutador que maneja el algoritmo entrenador para dejar pasar la respuesta puntual.

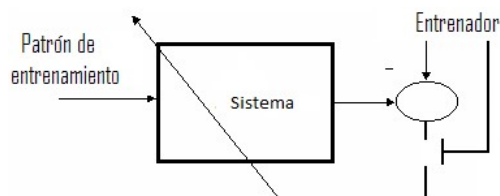


Figura 3.6.- Diagrama esquemático del aprendizaje por refuerzo.

3.5 ALGORITMOS DE APRENDIZAJE

El aprendizaje automático se aplica como solución de problemas que requieren de la generación de modelos orientados a datos y para ello se necesitan: los datos a procesar y algún algoritmo de aprendizaje o de representación como ya se ha comentado en el Capítulo I.

Con relación al algoritmo de representación, hay que indicar que son muchos los que pueden ser utilizados. El uso particular de alguno de estos, dependerá del tipo de problema a resolver, ya que cada uno tiene sus características particulares que el investigador debe conocer para poder concretar con éxito cualquier aplicación a desarrollar, siendo en este sentido hasta obligatorio conocer sus ventajas y desventajas.

Por consiguiente, en las siguientes secciones se explicarán a grandes rasgos los algoritmos de representación básicos o tradicionales que fueron utilizados en esta investigación como son: las máquinas de vectores soporte (SVM), el k vecino más cercano (KNN) y el centroide más cercano (CNN). Además de los algoritmos mixtos que trabajan en capas como la predicción conformal (PC), la predicción conformal inductiva (IPC) y la predicción Venn (PV).

3.6 MÁQUINAS DE VECTORES SOPORTE

Las máquinas de vectores soporte (SVM) es un método de aprendizaje estadístico utilizado para generar modelos a partir de un conjunto de datos disponible. Fue introducido por Vapnik y sus colaboradores [Boser et al, 1992], [Cortes y Vapnik, 1995], [Vapnik, 2000], y hoy por hoy es utilizado para resolver problemas tanto de clasificación como de regresión, contando ya con un gran número de aplicaciones resueltas en los campos de investigación como: la visión artificial, el reconocimiento de patrones, el procesamiento del lenguaje natural, el análisis de series temporales, entre otras soluciones encontradas en las referencias revisadas.

Su algoritmo de representación se basa en el principio de minimización del riesgo estructural, por lo cual se obtienen modelos que sólo dependen de una pequeña proporción de los datos usados en el entrenamiento que son llamados vectores

soporte, tal y como se ve en la Figura 3.7 donde idealmente se tienen dos clases bien definidas [Vapnik, 2000]. La clase formada por cuatro cuadrados azules y la formada por cuatro círculos verdes. El conjunto de datos en total tiene ocho objetos. Al entrenar la SVM, se consigue separar el conjunto de datos disponible mediante un hiperplano equidistante a los tres vectores soportes que se dibujan (véase de nuevo la Figura 3.7). Por lo tanto, un vector soporte no es más que un objeto que está más próximo a un hiperplano de separación.

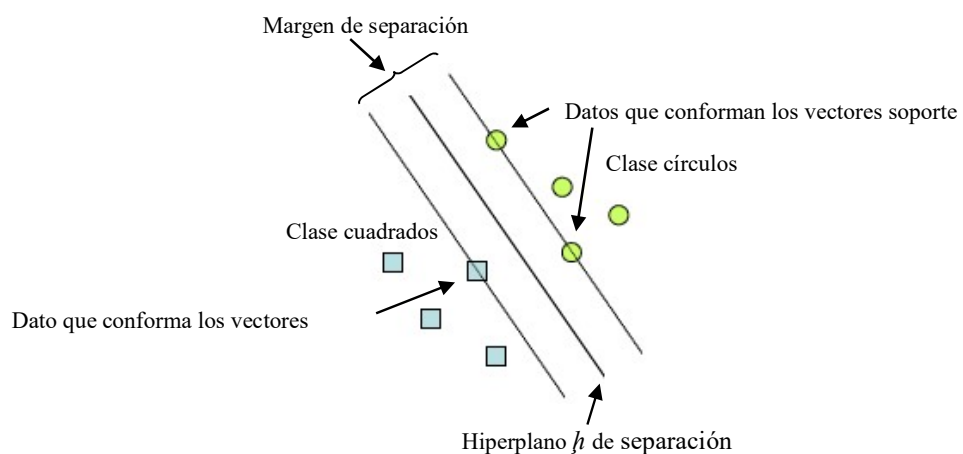


Figura 3.7.- Conjunto de datos separado por una SVM.

En realidad, el proceso de entrenamiento consigue este conjunto mínimo de vectores soporte (objetos), aportado por cada clase del conjunto total de datos disponibles para el entrenamiento. Esto permite formar un margen geométrico como solución al problema de separar ambas clases desde el punto de vista algorítmico. Matemáticamente, el margen geométrico dado en la Figura 3.7 como indica [Carmona, 2014], es “un problema de optimización cuadrático con restricciones lineales” el cual se resuelve con la aplicación de “técnicas estándar de programación cuadrática”. Y “la propiedad de convexidad exigida para su resolución garantiza una solución única”, que se consigue al encontrar el hiperplano que maximiza la distancia de separación entre las clases involucradas.

Todo esto desde un punto de vista simplista, se traduce en encontrar una única línea que pase equidistante a cada objeto considerado vector soporte. También permite definir a las SVM como un método determinístico, siendo la ventaja crucial que tiene frente a los algoritmos de representación como las Redes Neuronales Artificiales que puede aportar más de una solución a este tipo de problemas, muchas de las cuales no necesariamente son soluciones óptimas.

En cuanto al análisis de datos, se ha inferido un modelo o regla de clasificación f a partir del conjunto de datos disponible, tal y como se escribe a continuación:

$$f(\text{datos disponible}) = \text{signo}(h(\text{datos disponible})) \quad (3.3)$$

Donde h se conoce como hiperplano o frontera de decisión, pudiéndose escribir en los siguientes términos:

$$h(x) = w^T x + b \quad (3.4)$$

$$\text{Con } w, x \in \mathbb{R}^d, \text{ y } b \in \mathbb{R}.$$

Al evaluar la regla de decisión f con cada objeto del conjunto de datos disponibles o incluso nuevos datos, esta debe indicar a que clase pertenece.

3.6.1 Máquinas de vectores soporte para la clasificación binaria

Tomando como base el problema más simple en la clasificación de datos referido a un conjunto formado por dos clases, será posible su separación mediante el entrenamiento de una SVM, para lo cual dicho conjunto de datos podrá ser linealmente separable o linealmente no separable [Carmona, 2014] [Chang et al, 2010] tal y como muestra la Figura 3.8.

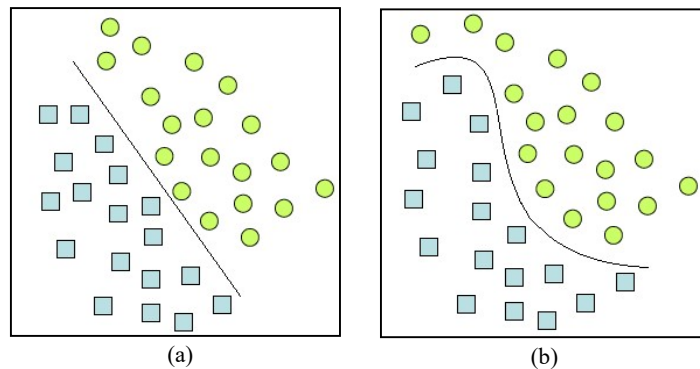


Figura 3.8.- Conjuntos de datos. (a) Linealmente separables. (b) No linealmente separables.

3.6.1.1 Datos linealmente separables

Dado un conjunto de datos disponibles z_i con $i=1, \dots, n$, con $y_i=\{-1, +1\}$ (las clases c_1 y c_2), para $i=1, \dots, n$, como el que se muestra en la Figura 3.8 (a). La SVM debe encontrar un conjunto de vectores soporte único y mínimo para establecer el margen de decisión o hiperplano $h(x)$, tal que:

$$h(x) = w^T x + b = 0 \quad (3.5)$$

Al fijar cualquier referencia con respecto al espacio donde se dibuje el hiperplano, se debe cumplir para todos los objetos x_i , las siguientes condiciones:

$$w^T x_i + b > 0, \text{ para } y_i=+1, \text{ con } i=1, \dots, n \quad (3.6)$$

$$w^T x_i + b < 0, \text{ para } y_i=-1, \text{ con } i=1, \dots, n \quad (3.7)$$

Las expresiones (3.6) y (3.7) pueden reducirse para encontrar:

$$y_i(w^T x_i + b) > 0, \text{ con } i = 1, \dots, n \quad (3.8)$$

Además se debe cumplir para los objetos x_i considerados vectores soporte:

$$w^T x_i + b = +1, \text{ para } y_i = +1, \text{ con } i = 1, \dots, n \quad (3.9)$$

$$w^T x_i + b = -1, \text{ para } y_i = -1, \text{ con } i = 1, \dots, n \quad (3.10)$$

Si los vectores soportes expresados con las condiciones dadas por las expresiones (3.9) y (3.10) son los objetos más próximos al hiperplano, entonces se tiene una restricción completa convirtiendo la expresión (3.8) en:

$$y_i(w^T x_i + b) \geq 1, \text{ con } i = 1, \dots, n \quad (3.11)$$

En la Figura 3.9 se pueden apreciar todas las restricciones impuestas por la expresión (3.11) al problema de hallar un máximo margen de separación, donde sólo los vectores soportes (objetos x_i) están en el borde de dicho margen tanto para las clases +1 y -1. Esto se traduce geoméricamente hablando, que desde el hiperplano $h(x)=0$ a cualquier objeto considerado vector soporte, habrá siempre la mínima distancia posible $\pm d$ (Clase +1 $\rightarrow +d$ y Clase -1 $\rightarrow -d$). En consecuencia para cualquier otro objeto x_i , la distancia d será siempre mayor ($d > \pm 1$).

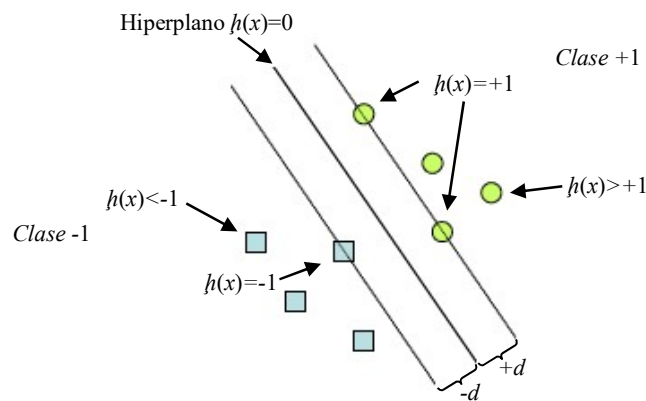


Figura 3.9.- Distancia de los objetos x_i al hiperplano $h(x)$.

Esta distancia desde el hiperplano a cualquier punto se puede hallar mediante la expresión:

$$dist(h, x) = \frac{|h(x)|}{\|w\|} \quad (3.12)$$

Si se quiere optimizar la distancia entre el hiperplano y un objeto sobre el borde del margen, entonces se estaría maximizando el margen que será dos veces la distancia d . Es evidente que estos objetos corresponden con los vectores soportes tal y como se apreció en la misma Figura 3.9, quedando resuelta la expresión (3.12) como:

$$|h(x_{\text{vectores_soportes}})| = 1 \quad (3.13)$$

Al sustituir (3.13) en (3.12) se dispone la función a maximizar y al considerar la restricción impuesta por la expresión (3.11) se obtiene el sistema:

$$\begin{cases} \text{Maximizar:} & \frac{1}{\|w\|} \\ \text{Con la restricción:} & y_i(w^T x_i + b) \geq 1, \text{ con } i = 1, \dots, n \end{cases} \quad (3.14)$$

Al maximizar $\frac{1}{\|w\|}$, se está minimizando $\frac{\|w\|^2}{2}$, con lo que el sistema (3.14) se

reescribe como:

$$\begin{cases} \text{Maximizar:} & \frac{\|w\|^2}{2} \\ \text{Con la restricción:} & y_i(w^T x_i + b) \geq 1, \text{ con } i = 1, \dots, n \end{cases} \quad (3.15)$$

La solución de este sistema se encuentra al hallar w y b . Para ello se usa la *formulación dual*¹⁹ sugerida en los problemas de optimización según [Carmona, 2014]. De esta manera se llega a la ecuación de Lagrange y las condiciones de Karush-Kuhn-Tucker para facilitar que la expresión (3.15) se reescriba como:

¹⁹ La formulación dual reduce el costo computacional al emplear técnicas estándar de programación cuadrática [Carmona S, 2014]. Esto no ocurre en la formulación primal.

$$L(w, b, \alpha) = \frac{1}{2} \|w\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i (w^T x_i + b)) \quad \text{para } i=1, \dots, n \quad (3.16)$$

$$\frac{\partial L(w, b, \alpha)}{\partial w} = 0 \rightarrow w = \sum_{i=1}^n y_i \alpha_i x_i \quad \text{para } i=1, \dots, n \quad (3.17)$$

$$\frac{\partial L(w, b, \alpha)}{\partial b} = 0 \rightarrow \sum_{i=1}^n y_i \alpha_i = 0 \quad \text{para } i=1, \dots, n \quad (3.18)$$

$$\alpha_i (1 - y_i (w^T x_i + b)) = 0 \quad \text{para } i=1, \dots, n \quad (3.19)$$

Las ecuaciones (3.17), (3.18) y (3.19) están expresadas en términos de los multiplicadores de Lagrange α_i . Esto garantiza [Carmona, 2014], que todos los objetos x_i que no sean vectores soporte darán como resultado un $\alpha_i=0$, y permite reescribir el problema dual dado por Lagrange mediante la expresión (3.16) como:

$$L(w, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j x_i^T x_j \quad (3.20)$$

Expresando el funcional en (3.15) con su respectiva restricción al considerar el problema dual resuelto en (3.20) y la combinación de las restricciones conseguidas en (3.18) y (3.19), se obtiene:

$$\left\{ \begin{array}{l} \text{Maximizar:} \\ \text{Con las restricciones:} \end{array} \right. \quad \begin{array}{l} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n y_i y_j \alpha_i \alpha_j x_i^T x_j \\ \sum_{i=1}^n y_i \alpha_i = 0 \quad \text{para } i=1, \dots, n \\ \alpha_i \geq 0 \quad \text{para } i=1, \dots, n \end{array} \quad (3.21)$$

Al resolver (3.21) empleando técnicas estándar de *programación cuadrática*²⁰ [Carmona, 2014] se puede encontrar los multiplicadores de Lagrange (α_i) y con ellos calcular w , b y el hiperplano $h(x)$, tal y como se muestra en las siguientes tres expresiones:

²⁰ Puede leer sobre programación cuadrática en [Merino, 2015].

$$w = \sum_{i=1}^n y_i \alpha_i x_i \quad (3.22)$$

$$h(x) = \sum_{i=1}^n y_i \alpha_i x_i^T x + b \quad \text{para } i=1, \dots, n \quad (3.23)$$

$$b = -\frac{1}{2} (\max_{y_i=-1} \{w^T x_j\} + \min_{y_i=+1} \{w^T x_j\}) \quad (3.24)$$

Se puede hallar b por otra vía, a partir de la expresión (3.19) donde queda claro que su cálculo (el valor de b) sólo se realiza para objetos considerados vectores soportes [Carmona, 2014]²¹. Esto deja como alternativa que su cómputo se pueda realizar mediante la expresión:

$$b = y_{vs} - \langle w, x_{vs} \rangle \quad (3.25)$$

El sub índice “vs” sugiere que sólo se usan objetos considerados vectores soportes con su respectiva etiqueta. De esta manera se consigue el valor directo para b .

A partir de este punto se van a ir mostrando ejemplos prácticos, para poder poner de manifiesto los conceptos teóricos que se van introduciendo con el fin de que queden más claros.

Ejemplo1. Conjunto de datos linealmente separables

Como ejemplo de aplicación, se muestra la Figura 3.10 donde se disponen de 12 objetos. Seis círculos rojos y seis cuadrados azules que definen las *clases* +1 y -1 respectivamente.

Al resolver el problema de optimización para estas dos clases dadas, usando programación cuadrática, se consiguen los α_i multiplicadores de Lagrange. Cada

²¹ Puede hallar en la referencia [Carmona, 2014] mayores detalles de la discusión abordada en esta sesión sobre el uso de las SVM en problemas de clasificación.

multiplicador de Lagrange válido ($\alpha_i \neq 0$) está asociado a un objeto considerado vector soporte (x_{vs} y su respectiva y_{vs}).

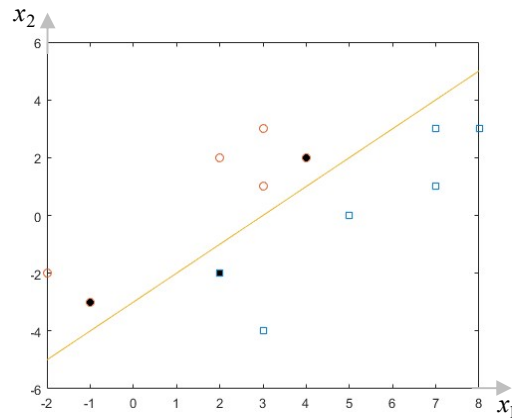


Figura 3.10.- Ejemplo de aplicación de la SVM para datos linealmente separables.

Si se observa nuevamente la Figura 3.10 se notará que se han marcado los vectores soporte con fondo oscuro. Ahora con toda esta información, se procede mediante las expresiones (3.22), (3.23) y (3.24) al cálculo de w , b , además de determinar el hiperplano $h(x)$. Todo esto se muestra en la Tabla 3.1.

Tabla 3.1.- Cálculos a partir de las expresiones (3.22), (3.23) y (3.24).

α			w	b	$h(x)$
$\alpha_{(-1,-3)}$	$\alpha_{(4,2)}$	$\alpha_{(2,-2)}$	$\sum_{i=1}^n y_i \alpha_i x_i$	$y_{vs} - \langle w, x_{vs} \rangle$	$\sum_{i=1}^n y_i \alpha_i x_i^T x + b$
0.5999	0.3999	-0.9998	(-0.9999, 0.9997)	2.9996	$-0.9999x_1 + 0.9997x_2 + 2.9999$

3.6.1.2 Datos no linealmente separables

En este caso se considera un conjunto de datos disponibles z_i con $i=1, \dots, n$, con $y_i = \{-1, +1\}$ (las clases c_1 y c_2), para $i=1, \dots, n$, como el mostrado en la Figura 3.8 (b). Es notorio que una línea recta (un plano recto) no puede dividir este conjunto de objetos hipotéticamente dispuestos.

No obstante, se pudiera dibujar como solución, una línea curva que permita separar las dos clases observadas. Siendo así, la SVM puede encontrar esta “línea

curva” transformando el espacio R^d de los objetos x_i de entrada, a un espacio ℓ de mayor dimensionalidad o espacio de características mediante una función de transformación Φ .

La función de transformación Φ permite que cada punto del espacio de entrada original sea mapeado en el espacio de características, donde sí será posible separar por un hiperplano a dichos puntos, tal y como se muestra en la Figura 3.11.

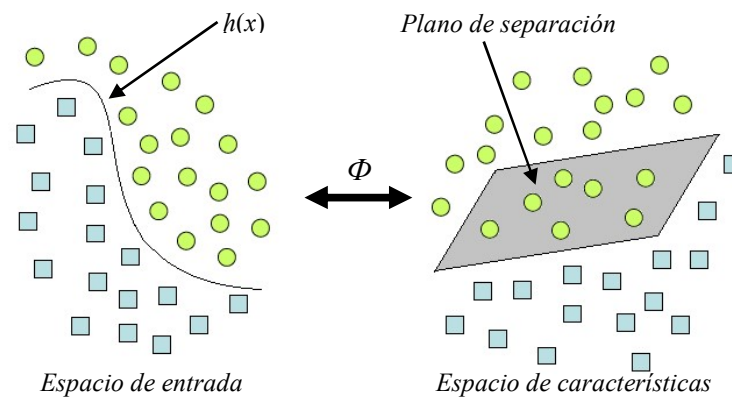


Figura 3.11.- Mapeo de objetos de la función de transformación Φ desde el espacio de entrada al espacio de características.

Debido a esta solución en el espacio de alta dimensionalidad, al mismo tiempo se ha encontrado para el espacio de entrada, un hiperplano no lineal $h(x)$ (la “línea curva”) que sí logra separar los objetos (véase de nuevo la Figura 3.11). Esto matemáticamente puede escribirse como sigue:

$$x = \{x_1, \dots, x_d\} \rightarrow \Phi(x) = \{\Phi_1(x), \dots, \Phi_\ell(x)\} \quad (3.26)$$

Donde d corresponde con la dimensión del espacio de entrada y ℓ con la dimensión del espacio de transformación. Por lo tanto, a cada punto x del espacio de entrada le corresponde en el espacio de características F , un $\Phi(x)$, es decir:

$$x \in R^d \rightarrow \Phi(x) \in R^\ell, \text{ con } d < \ell \quad (3.27)$$

En realidad Φ es una familia de funciones, donde cada miembro por decirlo de alguna manera, representa una característica (homólogo de dimensión) en el espacio de alta dimensionalidad. Es importante entender que las características se forman a partir del espacio de entrada, estando además en función de ellas.

Ahora bien, la asociación efectiva del espacio de entrada con una familia de funciones que proyectan el espacio de características, y que además permiten hallar un hiperplano de separación lineal en un ambiente de alta dimensionalidad asociándolo a su vez con un hiperplano no lineal en el espacio de entrada, se logra con el empleo de una función kernel [Vert et al, 2004], [Carmona, 2014].

Según esto y para delinear lo que hace esta función kernel, detalle la expresión (3.27) para que observe como el espacio de entrada d se ha elevado a ℓ . Es decir, que si se tiene un problema no separable linealmente en R^2 , se puede aumentar el espacio de entrada a R^3 o a cualquiera otra dimensión (características) y verificar allí las condiciones de separación lineal que finalmente asociarán a algún hiperplano $h(x)$ no lineal dentro del espacio de entrada [Vert et al, 2004].

Este “truco kernel” como es denominado por algunos expertos de la materia, y que *eleva efectivamente la dimensionalidad*²² del espacio de entrada o propiamente la función kernel, puede ser escrita como:

$$k(x_i, x_j) = \langle \phi(x) \phi(x') \rangle = \sum_{i=1}^{\ell} \phi_i(x)^T \phi_j(x') \quad (3.28)$$

Si se observa la expresión (3.28), se nota que la propiedad fundamental y clave del “truco kernel” o función kernel, es que en ella se define el producto escalar en el espacio de características. Si ahora se compara esta expresión con el producto escalar dado en la expresión (3.23), correspondiente al espacio de entrada, se concluye lo siguiente:

²² Es importante ver que el espacio de entrada queda intacto. El “truco kernel” sólo vincula, como lo hace cualquier otra transformada estudiada en cualquier carrera de ingeniería, a ambos espacios. Al decir “eleva efectivamente la dimensionalidad”, se está asumiendo transparentemente todo el proceso matemático.

$$k(x, x') = \langle x^T x \rangle = \langle \phi(x)^T \phi(x') \rangle \quad (3.29)$$

Por lo tanto, se puede usar (3.29) para modificar la formulación dual vista en el sistema (3.21) con la condición de que α_i debe estar en el rango $0 \leq \alpha_i \leq C$. Aquí C es un parámetro que controla las muestras mal clasificadas en un problema de separación no lineal, a costa de la maximización de la distancia al hiperplano [Carmona, 2014].

Al resolver el problema de formulación dual empleando técnicas estándar de programación cuadrática, se consiguen los multiplicadores de Lagrange (α_i), quedando determinado de esta forma el hiperplano $h(x)$ mediante la siguiente expresión:

$$h(x) = \sum_{i=1}^n y_i \alpha_i k(x_i, x_j) + b \quad (3.30)$$

Hay que tener un especial cuidado cuando se calcule b , ya que sólo se deben escoger aquellos objetos considerados vectores soporte pero que puedan ser separables. Esto corresponde con los multiplicadores de Lagrange que estén en el rango: $(0 < \alpha_i < C)$ [Carmona, 2014]²³.

Dos notas importantes que deben considerarse son: las *propiedades*²⁴ adicionales de las funciones kernel y los tipos de funciones kernel más usadas para realizar el mapeo del espacio de entrada. Estas funciones son:

- Kernel lineal:

$$k(x, x') = \langle x, x' \rangle \quad (3.31)$$

- Kernel polinomial:

$$k(x, x') = [\gamma \langle x, x' \rangle + \tau]^p \quad (3.32)$$

²³ En [Carmona, 2014] puede encontrar todo el desarrollo matemático que sustenta la introducción del parámetro C en los problemas de separación no lineal.

²⁴ Puede leer sobre las propiedades de los kernel y otras consideraciones relacionadas en [Carmona, 2014] y [Vert et al, 2004].

- Kernel gaussiano:

$$k(x, x') = \exp(-\gamma \|x - x'\|^2) \quad \text{Con } \gamma > 0 \quad (3.33)$$

Como aplicación y a manera de comprender mejor el “truco kernel”, se presentan los siguientes ejemplos donde se ha usado un kernel polinomial.

Ejemplo 2. Datos no linealmente separables. Uso del kernel polinomial

En este primer ejemplo se utiliza un kernel polinomial con los parámetros $\gamma=1$, $\tau=1$, y $p=2$, para clasificar los datos no linealmente separables dibujados en la Figura 3.12 (a). Obsérvese que el espacio de entrada de cada objeto está en R^2 y al aplicar el kernel de la expresión (3.32), se eleva el espacio de características a R^6 .

A continuación se muestra parte del desarrollo matemático realizado para este fin, comenzando por relacionar (3.29) con (3.32), y al sustituir los valores de γ , τ , y p sugeridos, se consigue:

$$k(x, x') = [\langle x, x' \rangle + 1]^2 = \langle \phi(x), \phi(x') \rangle \quad (3.34)$$

Sabiendo que:

$$x = [x_1, x_2] \quad (3.35)$$

Lo que implica al sustituir (3.35) en (3.34):

$$[\langle (x_1, x_2), (x'_1, x'_2) \rangle + 1]^2 = \langle \phi(x), \phi(x') \rangle \quad (3.36)$$

Al desarrollar el lado izquierdo en (3.36) se obtiene:

$$x_1^2(x'_1)^2 + x_2^2(x'_2)^2 + 2x_1x_2x'_1x'_2 + 2x_1x'_1 + 2x_2x'_2 + 1 = \langle \phi(x), \phi(x') \rangle \quad (3.37)$$

Nótese que el producto escalar $\langle \phi(x), \phi(x') \rangle$ proviene de multiplicar dos vectores en el espacio de características, es decir:

$$\langle \phi(x), \phi(x') \rangle = \left\langle \left(1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2 \right), \left(1, \sqrt{2}x'_1, \sqrt{2}x'_2, \sqrt{2}x'_1x'_2, x_1'^2, x_2'^2 \right) \right\rangle \quad (3.38)$$

Quedando claro que la familia de funciones $\Phi(x)$ está contenida dentro del vector:

$$\phi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2] \quad (3.39)$$

También nótese que Φ en (3.39) está en función del espacio de entrada, por lo tanto, al resolver el problema de optimización usando programación cuadrática se consigue como solución los multiplicadores de Lagrange (α_i) válidos ($\alpha_i \neq 0$) que están asociados a los objetos considerados vectores soporte (x_{vs} y su respectiva y_{vs}).

En la Figura 3.12 (a) se han marcado con fondo relleno cada objeto del conjunto de datos observado durante el proceso de entrenamiento, indicando esto que al final todos calificaron como vectores soporte.

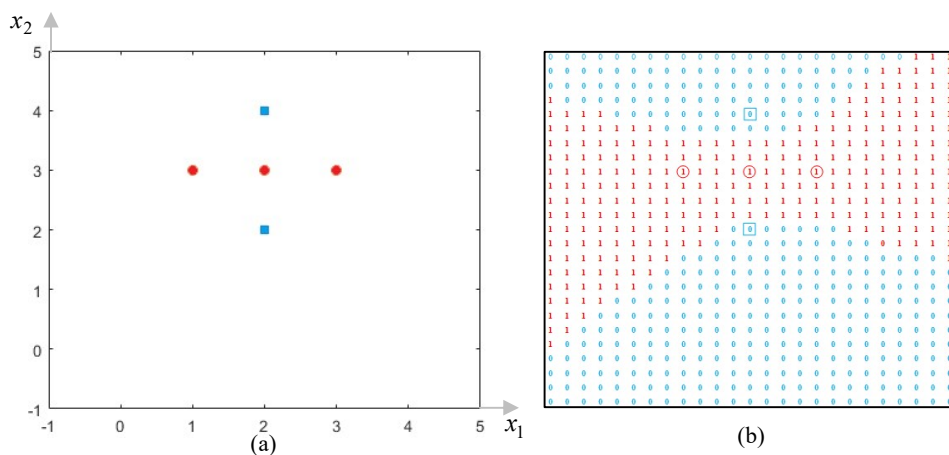


Figura 3.12.- (a) Datos disponibles para el Ejemplo 2. (b) Mapa de predicción posible producido por el hiperplano $h(x)$ inferido con los datos disponibles del Ejemplo 2. Región en color rojo: predice la Clase +1. Región en color azul: predice la Clase -1.

En la Tabla 3.2 se ve un resumen de los valores correspondientes al espacio de entrada (coordenadas) de cada objeto, con su etiqueta y los respectivos α_i asociados y hallados como solución del Ejemplo 2. Por su parte, en la Tabla 3.3, se muestran los valores encontrados para w , b y el modelo del hiperplano $h(x)$, determinados a partir de los multiplicadores de Lagrange α_i .

Tabla 3.2.- Valores típicos encontrados para los datos del Ejemplo 2.

$X_{vs}=[x_1 \ x_2]$		y_{vs}	α_i
1	3	+1	4.51
2	3	+1	10
3	3	+1	2.80
2	2	-1	-10
2	4	-1	-7.31

Tabla 3.3.- Solución encontrada a partir de los α_i para los datos del Ejemplo 2.

w	$\sum_{i=1}^n y_i \alpha_i \phi_i(x)$	(0 , -2.4182 , 3.8112 , 0.3670 , 0.4656 , -1.1374)
b	$y_{vs} - \langle w, x_{vs} \rangle$	-3.5358
$h(x)$	$\sum_{i=1}^n y_i \alpha_i \langle \phi_i(x), \phi_i(x') \rangle + b$	$-3.4199x_1 + 5.3898x_2 + 0.5190x_1x_2 + 0.4656x_1^2 - 1.1374x_2^2 - 3.5330$

A partir del hiperplano $h(x)$ visto en la Tabla 3.3 se ha proyectado en la Figura 3.12 (b) un mapa de la posible clasificación realizada por este modelo, ilustrando de esta manera el comportamiento computacional que tendría tal modelo ante un objeto desconocido que le sea pasado como parámetro.

Como ejemplo, supóngase que los objetos usados en el proceso de entrenamiento son todos desconocidos. Si ahora se usa el modelo del hiperplano $h(x)$ inferido y se les pasa por parámetro tales observaciones (supuestas desconocidas), debería verse, y de hecho se ve, en la Figura 3.12 (b) la predicción de tales muestras.

Para destacar la importancia aún más del mapa, véase las predicciones proyectadas y contemple que hay tres regiones bien delimitadas. Una en color rojo, propiciadas por el uso del kernel polinomial, que ha predicho tres de las muestras

asociándolas con la *Clase +1*. Estas (muestras) se han marcado con un círculo para verificar tal ilustración gráfica (véase la Figura 3.12 (b)). Además se observan otras dos regiones en color azul, que también son debidas al uso del kernel mencionado. Sobre ellas se dibujan con un cuadrado, las otras dos muestras (una en cada una) que han sido etiquetadas como *Clase -1* (véase de nuevo la Figura 3.12 (b)).

Para el caso de los cinco objetos procesados con el modelo encontrado, donde todo el proceso computacional se hizo en MATLAB con ayuda de la librería LIBSVM, eran de esperar los resultados ilustrados porque de antemano se sabía la clase de tales muestras (véase la Tabla 3.2). Pero, si se pasa cualquier otro objeto desconocido, se debe esperar en el mapa de predicción expuesto su clase correspondiente. Esta ilustración, computacionalmente el modelo lo indicaría emitiendo una etiqueta de clase “+1” o “-1”.

Ejemplo 3. Datos no linealmente separables. Uso del kernel polinomial

En este segundo ejemplo para datos no linealmente separables se ha usado nuevamente el kernel polinomial visto en la expresión (3.32) pero variando el parámetros τ (también se puede variar p).

Se ha hecho esto para demostrar cómo se modifica el espacio de características a partir del espacio de entrada de los objetos que se muestran en la Figura 3.13 (a). Obsérvese que aún el espacio de entrada sigue estando en R^2 , pero a diferencia con el Ejemplo 2, ahora al aplicar la expresión (3.22) modificada, se consigue aumentar el espacio de características a R^3 .

A continuación se plantea el mismo desarrollo matemático realizado anteriormente, comenzando por relacionar (3.29) con (3.32), y al sustituir los valores de $\gamma=1$, $\tau=0$, y $p=2$, se consigue:

$$k(x, x') = [\langle x, x' \rangle]^2 = \langle \phi(x), \phi(x') \rangle \quad (3.40)$$

Donde x sigue siendo:

$$x=[x_1, x_2] \quad (3.41)$$

Relacionando (3.40) y (3.41) se obtiene:

$$\left[\langle (x_1, x_2), (x'_1, x'_2) \rangle \right]^2 = \langle \phi(x), \phi(x') \rangle \quad (3.42)$$

Al desarrollar el lado izquierdo en (3.42) se llega a:

$$x_1^2(x'_1)^2 + 2x_1x'_1x_2x'_2 + x_2^2(x'_2)^2 = \langle \phi(x), \phi(x') \rangle \quad (3.43)$$

Como el producto escalar $\langle \phi(x), \phi(x') \rangle$ proviene de multiplicar dos vectores en el espacio de características, entonces al igual que en el Ejemplo 2, se concluye que:

$$\langle \phi(x), \phi(x') \rangle = \left\langle \left(x_1^2, \sqrt{2}x_1x_2, x_2^2 \right), \left(x'^2_1, \sqrt{2}x'_1x'_2, x'^2_2 \right) \right\rangle \quad (3.44)$$

Donde la familia de funciones $\Phi(x)$ corresponde ahora con:

$$\phi(x) = [x_1^2, \sqrt{2}x_1x_2, x_2^2] \quad (3.45)$$

Como era de esperar Φ está en función del espacio de entrada. Por lo tanto, al resolver el problema de optimización usando programación cuadrática para los datos disponibles del Ejemplo 3, se consigue como solución los multiplicadores de Lagrange.

Gracias a los a_i diferentes de cero ($a_i \neq 0$) se dibujan en la Figura 3.13 (a) con fondo relleno, los objetos vectores soporte, distinguiéndose del resto de los datos disponibles para el entrenamiento donde ahora y a diferencia del Ejemplo 2, no todos fueron escogidos.

En las Tablas 3.4 y 3.5 se incluye un resumen de los valores de x_{vs} , y_{vs} y su respectivo α_i asociado. También se muestran valores para w , b y el modelo del

hiperplano $h(x)$ respectivo. Al mismo tiempo, puede verse en la Figura 3.13 (b) el mapa de la posible clasificación realizada por el hiperplano $h(x)$ inferido al usar el conjunto de datos del Ejemplo 3.

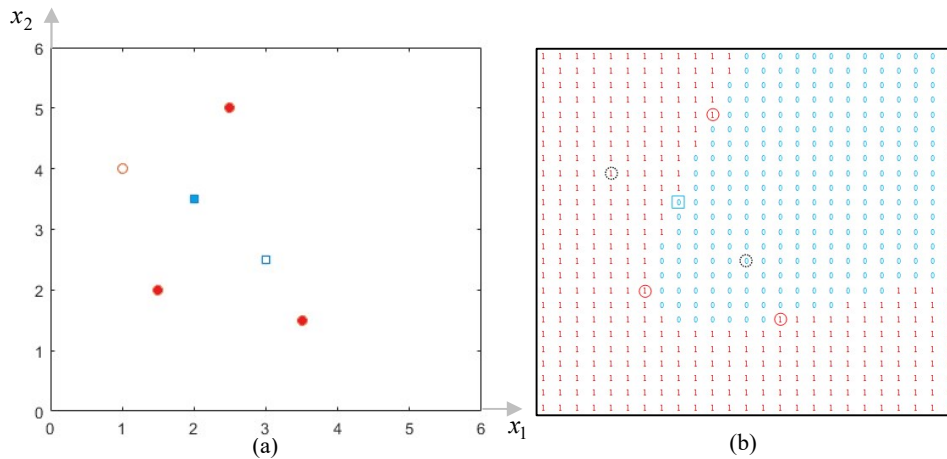


Figura 3.13.- (a) Datos disponibles para el Ejemplo 3. (b) Mapa de predicción posible producido por el hiperplano $h(x)$ inferido con los datos disponibles del Ejemplo 3. Región en color rojo: predice la Clase +1. Región en color azul: predice la Clase -1.

Tabla 3.4.- Valores típicos encontrados para los datos del Ejemplo 3.

$X_{vs}=[x_1 \ x_2]$		y_{vs}	α
1.5	2	+1	18.49
2.5	5	+1	12.84
3.5	1.5	+1	0.75
2.0	3.5	-1	-32.10

Tabla 3.5.- Solución encontrada a partir de los α_i para los datos del Ejemplo 3.

w	$\sum_{i=1}^n y_i \alpha_i \phi_i(x)$	(2.7335 -6.5661 3.6802)
b	$y_{vs} - \langle w, x_{vs} \rangle$	7.9870
$h(x)$	$\sum_{i=1}^n y_i \alpha_i \langle \phi_i(x), \phi_i(x') \rangle + b$	$2.7335x_1^2 - 9.2860x_1x_2 + 3.6802x_2^2 + 7.9870$

Antes de continuar con la SVM y definir la forma como se usa ésta cuando existen más de dos clases, se indicará la importancia de las funciones kernel [Vert et al, 2004] en los problemas donde no pueden separarse linealmente los objetos. Para entenderlo, véase a [Carmona, 2014] y observe que para dos clases bien separables linealmente hablando, siempre será posible encontrar un hiperplano de separación de margen duro. Pero cuando no son claramente separables, se encuentran hiperplanos de separación de margen suave.

Ambos casos (lineal y no linealmente separables) pueden quedar resueltos si se aplican convenientemente las funciones kernel, resolviéndose el problema tal y como se vio para el caso de clases no separables linealmente.

Como comentario final, se resalta que en esta sección se han mencionado tres de las funciones kernel que son aplicadas al resolver algún tipo de problema, sin distinguir cuál pudiera dar mejor resultado. Aunque, en los experimentos realizados en esta investigación a partir de datos sintéticos y otras fuentes de datos como las imágenes obtenidas del dispositivo de fusión termonuclear TJ-II del CIEMAT, la función kernel gaussiana fue la que mejor tasas de aciertos arrojó ante los problemas de clasificación estudiados.

3.6.2 Máquinas de Vectores Soporte para multclasificación

La Máquina de Vectores Soporte (SVM) fue desarrollada originalmente para resolver problemas de clasificación del tipo binario. Indudablemente, esto limita su utilidad real, dado que cotidianamente se requieren soluciones para problemas multiclase. Ante este tipo de inconvenientes, el secreto es razonar siempre como si se estuviese resolviendo un problema binario, tal y como se vio en el apartado anterior, pero teniendo en cuenta que ahora hay más de dos clases.

Para aclarar aún más, ilustrativamente al disponer de un conjunto de datos con k clases, la idea consistiría en crear modelos donde se puedan tratar dos clases. Con esto en mente, se pueden aplicar dos tipos de aproximaciones con el fin de analizar mediante la SVM los datos disponibles:

- **Uno versus uno:** dadas k clases en un conjunto de datos disponible, se buscará siempre de comparar una clase con cada una del resto de las clases presentes. La clase k que se tome, se le asigna la etiqueta “+1” (*Clase +1*) y a la otra “-1” (*Clase -1*). Con cada par de clases formada (conjunto binario), se debe entrenar un modelo SVM. Al final se habrán inferido $k(k - 1)/2$ clasificadores binarios.
- **Uno versus el resto:** dadas k clases presentes en un conjunto de datos disponible, se toma una asignándole la etiqueta “+1” (*Clase +1*). El resto de las clases ($k-1$ restantes) serán agrupadas, asignándole la etiqueta “-1” (*Clase -1*). Esto forma un conjunto binario, que es usado para entrenar una SVM e inferir el modelo $k=1$. Este proceso se realiza con cada una las clases ($k=2, 3, 4, \dots$), siempre viendo al resto (de las clases presentes) como otra clase. Al final de todo el proceso se infieren k clasificadores binarios.

En resumen, ambos procedimientos descritos brevemente, son usadas para desarrollar modelos generales de multclasificación. No obstante, el enfoque utilizado en esta Tesis es la segunda aproximación (uno versus el resto), el cual se analiza en la siguiente sección.

3.6.2.1 Descripción de la aproximación uno versus el resto

En la sección 3.6.1 se mostró como la SVM clasifica datos lineal y no linealmente separables, comprobando que se puede clasificar cualquier tipo información siempre y cuando sea divisible en dos clases, es decir: la *Clase a* y la *Clase b*.

Sin embargo, se puede tener más de dos clases para identificar en un conjunto de datos, con lo que no se podría aplicar una única línea (hiperplano) para clasificarlas. Por esta razón, se definen los multclasificadores que serán estudiados en esta Tesis a través de la aproximación uno versus el resto, con la ayuda de varios ejemplos para ilustrar todo de manera práctica.

Ejemplo 4. Aplicación de la aproximación uno versus el resto para datos multiclase

En este ejemplo, se presentan ocho puntos dibujados en la Figura 3.14 (a), definiendo de esta manera cuatro clases (*Clase 1*: círculos, *Clase 2*: cuadrados, *Clase 3*: rombos y *Clase 4*: triángulos). Las coordenadas (x_1, x_2) de cada grupo geométrico deben ser organizadas siguiendo la aproximación uno versus el resto para entrenar un biclasificador SVM por cada clase presente, tal y como se describe a continuación.

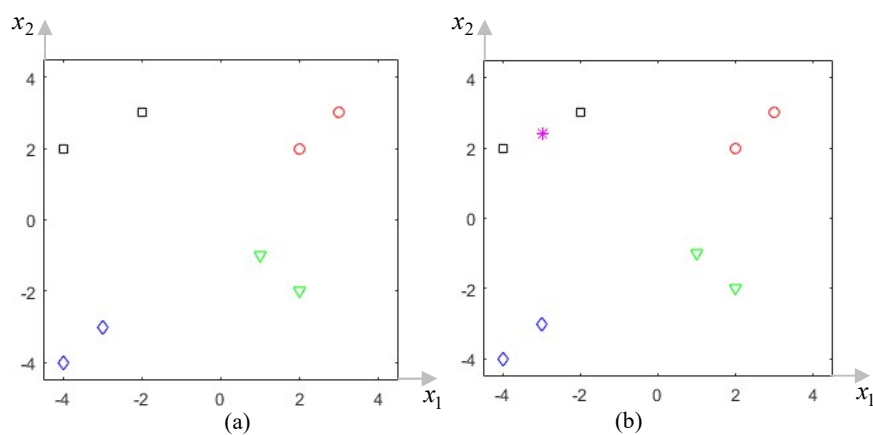


Figura 3.14.- (a) Puntos que conforman un conjunto de datos multiclase. *Clase 1*: círculos, *Clase 2*: cuadrados, *Clase 3*: rombo y *Clase 4*: triángulos. (b) Inclusión de un objeto asterisco en las coordenadas $(-3, 2.5)$ en el conjunto de datos multiclase.

El procedimiento comienza por elegir una de las clases presentes en el conjunto de datos disponibles. Si se elige la *Clase 1* (círculos rojos) asignándole la etiqueta “+1”, entonces al resto de las clases (cuadrados, rombos y triángulos) se les asigna la etiqueta “-1”. Esto conformará el primer conjunto binario de datos a observar (datos observados).

Con el fin de ver el procedimiento en operación y la ayuda del comando *svmtrain* de la librería *LIBSVM*²⁵, se entrena una SVM con función kernel del tipo RBF para inferir el hiperplano de separación $h(x)$ o simplemente el *Modelo_1* para referirse inequívocamente a dicha regla de aprendizaje generada.

²⁵ Se explicaran los detalles de esta librería de funciones para el Manejo de la SVM en la sección 3.11.

Para ver lo que se ha obtenido, fijese en la Figura 3.15 (a) donde se observa un mapa de la predicción posible para los objetos procesados con el *Modelo_1* que pertenezcan a la *Clase +1* y *Clase -1*. La región de la *Clase +1* se distingue con un signo de suma “+” en color azul. Y la región para la *Clase -1* se ha marcado con un punto “.” rojo. En la Figura 3.15 (a) también se muestran todas las observaciones definidas inicialmente con la aproximación uno versus el resto.

Para comprender mejor el mapa creado e incluso el resto que serán presentados a lo largo de este capítulo, se explica la respuesta computacional que puede ser obtenida por un modelo entrenado al procesar un nuevo objeto (una muestra), y como se refleja ilustrativamente en dicho mapa. Para el caso del *Modelo_1* que fue entrenado para responder con una etiqueta “+1” cuando se le pase algún objeto perteneciente a la clase círculo, el mapa debe señalar la predicción para tal etiqueta “+1” en color azul.

Por ejemplo, si se supone que los dos círculos originales del conjunto de datos disponible para crear el *Modelo_1* son desconocidos, y se les pasa al *Modelo_1* (ya creado), ambas muestras deben ser identificadas como *Clase +1*. En este caso el mapa creado para el mencionado *Modelo_1* refleja el resultado dibujado un signo de suma “+” azul encima de cada observación (véase la Figura 3.15 (a)). Para el resto de las figuras geométricas (cuadrados, rombos y triángulos), de ser pasadas como objetos desconocidos al *Modelo_1*, deberían ser predichas como *Clase -1*. En este caso se muestra la respuesta en el mapa correlacionado, dibujándose un punto “.” rojo encima de las mencionadas muestras (puede observarse en la Figura 3.15 (a)).

Después de completar el primer entrenamiento y continuando con la aproximación uno versus el resto, se selecciona otra clase (*Clase 2*: cuadrados). En este caso se le asigna la etiqueta “+1” a la *Clase 2* y al resto de las clases (círculos, rombos y triángulos) la correspondiente etiqueta “-1”, conformándose así el segundo conjunto binario de datos. Al usar nuevamente el comando *svmtrain* de la librería *LIBSVM* con una función kernel del tipo RBF, se infiere otro hiperplano de separación $h(x)$ o *Modelo_2* para referirse a esta otra regla de aprendizaje generada.

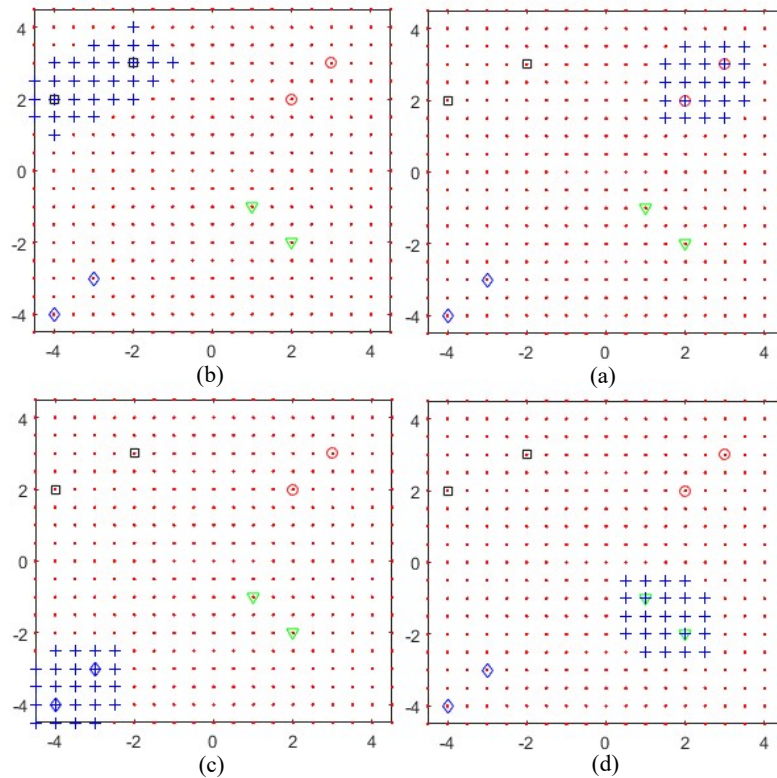


Figura 3.15.- Mapas de predicción posibles producidos por los hiperplanos $h(x)$ inferidos con los datos binarios formados a partir de los datos multiclase de la Figura 3.14 (a). Clase +1: signo de suma “+” en color azul. Clase -1: punto “.” en color rojo.

Al emplear el mismo análisis visual realizado con anterioridad pero ahora sobre la Figura 3.15 (b) donde se ha colocado el mapa de la predicción posible permitido por el nuevo *Modelo_2*, se puede observar la predicción para los objetos cuadrados marcados en esta ocasión con un signo de suma “+” en azul, indicando que pertenecen a la Clase +1. Las otras clases disponibles en este conjunto de datos binario, coinciden como la Clase -1 ya que se les ha marcado con un punto “.” en color rojo, incluyendo los círculos que fueron distinguidos como Clase +1 en el primer conjunto de datos binario formado con anterioridad para este mismo ejemplo.

En este Ejemplo 4, lo que se pretende explicar es que el *Modelo_2* fue entrenado para responder con una etiqueta “+1” cuando se le pase algún objeto

perteneciente a la clase cuadrado. Es evidente que para el resto de los objetos (cuadrados, rombos y triángulos), el *Modelo_2* predecirá una etiqueta “-1”.

Todo el procedimiento explicado para las *Clases* 1 y 2 debe ser repetido sobre las *Clases* 3 y 4 restantes. A modo de resumen se muestra en la Tabla 3.6 toda la información detallada referente al conjunto de datos dispuestos en la Figura 3.14 (a), donde se observan las coordenadas de los 8 puntos en las filas 3 y 4. Se dispone también en las filas 5, 6, 7 y 8, de los cuatro conjuntos binarios conformados al aplicar la aproximación uno versus el resto.

Obsérvese como el procedimiento uno versus el resto opera al organizar las mismas coordenadas (x_1, x_2) , para formar cada conjunto binario, sólo con cambiar las etiquetas de clase y así obtener los modelos SVM que son vistos al final de las filas mencionadas (filas 5, 6, 7 y 8).

Tabla 3.6.- Resumen del conjunto de datos multiclase de la Figura 3.14 (a).

Puntos	1	2	3	4	5	6	7	8	
Objetos	Clase 1 Círculos		Clase 2 Cuadrados		Clase 3 Rombos		Clase 4 Triángulos		
Coordenada x_1	2	3	-2	-4	-3	-4	1	2	
Coordenada x_2	2	3	3	2	-3	-4	-1	-2	Modelo SVM Generado
Etiquetas Conjunto Binario Clase 1	+1	+1	-1	-1	-1	-1	-1	-1	Modelo_1
Etiquetas Conjunto Binario Clase 2	-1	-1	+1	+1	-1	-1	-1	-1	Modelo_2
Etiquetas Conjunto Binario Clase 3	-1	-1	-1	-1	+1	+1	-1	-1	Modelo_3
Etiquetas Conjunto Binario Clase 4	-1	-1	-1	-1	-1	-1	+1	+1	Modelo_4

3.6.2.2 Multiclasificador basado en la aproximación uno versus el resto

En la sección 3.6.2.1 se han desarrollado cuatro biclasificadores SVM (*Modelo_1*, *Modelo_2*, *Modelo_3*, *Modelo_4*) basados en la aproximación uno versus el resto. Cada uno de estos modelos debe responder con la etiqueta “+1” o “-1” cuando se les pase por parámetro, un objeto nuevo del cual no se tiene información sobre qué tipo de figura geométrica es.

Supóngase ahora, que se tiene un objeto desconocido obtenido en el mismo proceso que arrojó los datos de entrenamiento. Tal muestra desconocida, se facilita a cada modelo creado para que la procese y emita un veredicto (de su respuesta).

Obtenida la respuesta por parte de cada modelo (*Modelo_1*, *Modelo_2*, *Modelo_3*, *Modelo_4*), se debe tener presente qué: si algún modelo en particular responde con la etiqueta “+1”, estará indicando que la muestra nueva desconocida se corresponde con aquellos objetos que están en el conjunto de datos binarios observado en el momento de realizar el entrenamiento, pero que fueron definidos como *Clase +1* mediante la aproximación uno versus el resto. Si por el contrario, un modelo en particular responde con la etiqueta “-1”, estará indicando la posibilidad de que la muestra nueva desconocida se corresponda con cualquiera de los objetos que están dentro del conjunto de datos binarios observados en el momento de inferir el modelo correspondiente, pero que fueron definidos como *Clase -1* mediante la aproximación uno versus el resto.

Como no se tiene un único modelo binario, ya que fueron entrenados cuatro, se puede decir que surge el concepto de multclasificador. Que no es más, que utilizar estos cuatro biclasificadores entrenados para que procesen el objeto nuevo desconocido. Cada modelo biclase, debe responder por separado con una etiqueta “+1” o “-1”. Lógicamente, la muestra nueva desconocida pertenecerá a la *Clase +1* del modelo que responda con la etiqueta “+1”. Véase este razonamiento con un nuevo ejemplo.

Ejemplo 5. Construcción de un multclasificador usando el procedimiento uno versus el resto

En este ejemplo, se pretende predecir la muestra dibujada como un asterisco “*” sobre el conjunto de datos dispuestos para el Ejemplo 4, tal y como se observa en la Figura 3.14 (b). Se ha dibujado de esta manera para ilustrar la falta de conocimiento sobre el tipo de figura geométrica que puede ser. Aunque por su ubicación en el plano cartesiano, el nuevo objeto, sin duda pertenece a la *Clase 2* (cuadrados).

Siguiendo con la consideración de que el nuevo objeto asterisco es desconocido y usando el concepto del multclasificador para que ayude en la tarea de identificar la clase de figura geométrica de la muestra, se procede entonces a pasar la observación (“*”) como parámetro a cada uno de los cuatro modelos SVM biclase inferidos (*Modelo_1*, *Modelo_2*, *Modelo_3*, *Modelo_4*). Las respuestas obtenidas por cada regla inferida se ilustran en los mapas que componen la Figura 3.16.

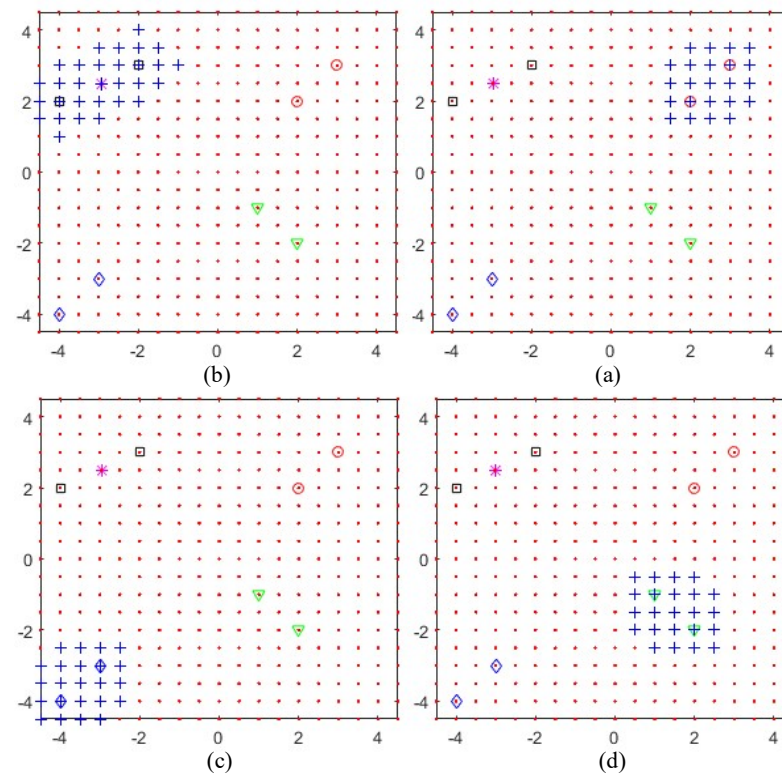


Figura 3.16.- Respuesta obtenida para un objeto asterisco (desconocido) ubicado en la coordenada (-3,2.5) usando los mapas de predicción presentados en la Figura 3.15.

Obsérvese como el *Modelo_2* responde indicando que el objeto es considerado una figura geométrica del tipo cuadrado, ya que se ha dibujado sobre el asterisco “*” de la Figura 3.16 (b) un signo “+” en color azul. Por el contrario, el resto de los modelos (*Modelo_1*, *Modelo_3* y *Modelo_4*) desconocen la naturaleza geométrica del nuevo objeto indicando que muy posiblemente sea una *clase* -1. Esto es evidente porque las Figuras 3.16 (a), (c) y (d), muestran sobre el asterisco un punto “.” rojo dibujado.

3.6.2.3 Tipos de respuestas de un multclasificador SVM

La Figura 3.16 muestra una respuesta gráfica construida para ilustrar el funcionamiento de un multclasificador SVM. Tal respuesta se basó en la aproximación uno versus el resto, pero en realidad cada modelo SVM construido responde (matemáticamente) con un predicción sobre el objeto analizado, arrojando simplemente una etiqueta “+1” si pertenece a tal regla de decisión o con etiqueta “-1” si no pertenece.

Usando de nuevo el objeto asterisco incluido y visto en la Figura 3.14 (b), se puede corroborar lo comentado anteriormente al ver la fila 1 de la Tabla 3.7, donde se muestra la predicción o respuesta computacional obtenida con el comando *svmpredict* de la librería *LIBSVM* para cada biclasificador generado. Nótese que el *Modelo_2* ha respondido con la etiqueta “+1” como era de esperar, y el resto de los modelos han respondido con una etiqueta “-1”. Esto confirma el desconocimiento hipotético hecho sobre el objeto asterisco propuesto. Se indica de antemano, tal y como se verá en la próxima sección, que este tipo de respuesta no siempre logra identificar a un objeto.

Tabla 3.7.- Tipos de respuesta obtenida por el multclasificador definido a partir de los datos multiclase de la Figura 3.14 (a) cuando procesa el objeto asterisco ubicado en las coordenadas (-3,2.5).

Fila	Tipo de Respuesta	Modelo 1	Modelo 2	Modelo 3	Modelo 4
1	Predicción	-1	+1	-1	-1
2	Distancia al Hiperplano $h(x)$	-0.7912	0.3738	-0.7914	-0.7913

Se puede obtener otra respuesta que también se utiliza para decidir a qué clase pertenece cierto objeto nuevo desconocido a identificar. De hecho, esta respuesta es la más certera porque internamente el código básico de cualquier SVM, se construye a partir del planteamiento de un problema para separar geométricamente dos clases, siendo resuelto mediante el empleo de técnicas estándar de programación cuadrática, lo cual arroja un hiperplano $h(x)$ como solución.

En este sentido, es posible realizar la predicción de un objeto nuevo desconocido, sólo con analizar de alguna manera, la distancia desde tal objeto nuevo desconocido al hiperplano de separación $h(x)$ inferido. Para el objeto asterisco

incluido en los datos multiclase de la Figura 3.14 (b), esta respuesta se muestra en la fila 2 de la Tabla 3.7. Razonablemente, si se toma el valor absoluto de las distancias reflejadas, la respuesta esperada se obtiene por el mínimo valor, indicando que el objeto desconocido pertenece a la clase cuadrados.

3.6.2.4 Distancia al hiperplano $h(x)$ como respuesta decisiva de la SVM

Como se ha visto, una SVM puede responder de dos formas posibles. Una, la predicción del objeto, indicando su etiqueta y en consecuencia la clase a la que pertenece. Y la otra, la distancia al hiperplano $h(x)$, indicando que tan cerca está de una clase en particular y como consecuencia la identifica allí.

La primera forma aunque es válida no siempre es capaz de dar una predicción segura e incluso muchas veces ni siquiera hay una decisión, porque depende de donde se encuentre un determinado objeto desconocido. No obstante, la segunda forma pudiera resolver estos inconvenientes, existiendo más de una vía de cómo implementar su usabilidad.

En este apartado se demuestra porqué esta segunda forma de respuesta (la distancia al hiperplano $h(x)$) es poderosa y se verá más adelante cuando se defina la predicción conformal, que se utiliza para generar modelos de predicción con medidas de confianza.

Ejemplo 6. Construcción de un multclasificador usando la aproximación uno versus el resto en combinación con la distancia al hiperplano

Para iniciar la demostración planteada líneas atrás se propone un nuevo objeto asterisco, ubicándolo convenientemente en las coordenadas (-3,-2), tal y como se puede observar en la Figura 3.17. Nótese que el nuevo objeto asterisco no queda dentro de ninguna de las regiones definidas como *Clase +1* (región con signo más “+” en color azul), delimitadas por cada uno de los cuatro modelos SVM binarios construidos a partir de los datos multiclase en el Ejemplo 4 (*Modelo_1*, *Modelo_2*, *Modelo_3*, *Modelo_4*).

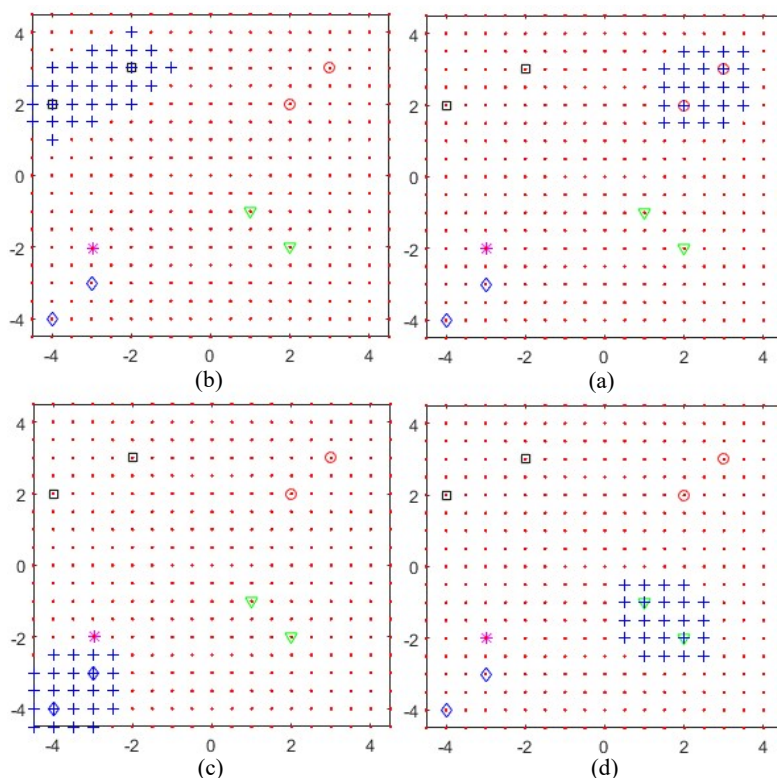


Figura 3.17.- Respuesta obtenida para un nuevo objeto asterisco (desconocido) ubicado en la coordenada $(-3,-2)$ usando los mapas de predicción presentados en la Figura 3.15.

Con relación al hecho de que la muestra asterisco desconocida no queda dentro de las regiones delimitadas en color azul, se debe entender como que ninguno de los cuatro modelos mencionados identifica como suya la muestra desconocida, y sólo se prestan para indicar la posibilidad de que el asterisco pertenece a cualquiera de las otras clases. Es decir, no hay una decisión firme sobre qué tipo de figura geométrica es. Y se dice esto, porque al ver la etiqueta que arroja cada modelo, todas coinciden con “-1”.

Ilustrativamente, este hecho se confirma al ver que cada asterisco dibujado en las Figuras 3.17 (a), (b), (c) y (d), se ha marcado con un punto rojo, señalando que el objeto es de la *Clase* -1. La respuesta computacional enviada por cada modelo SVM biclase, también se puede observar en la fila 1 de la Tabla 3.8.

Tabla 3.8.- Tipos de respuesta obtenida por el multclasificador definido a partir de los datos multiclase de la Figura 3.14 (a) cuando procesa el nuevo objeto asterisco ubicado en las coordenadas (-3,-2).

Fila	Tipo de Respuesta	Modelo 1	Modelo 2	Modelo 3	Modelo 4
1	Predicción	-1	-1	-1	-1
2	Distancia al Hiperplano $h(x)$	-0.6752	-0.6338	-0.0154	-0.6752

La pregunta ahora sería: ¿a qué clase se asigna el asterisco? o ¿qué figura geométrica es?. Razonablemente, la respuesta se puede obtener si se toma la mínima distancia desde el objeto asterisco desconocido al hiperplano de separación creado por cada modelo SVM biclase. Como se aprecia en la fila 2 de la Tabla 3.8, el valor absoluto mínimo lo posee el *Modelo_3* (mínima distancia al hiperplano=0.0154), concluyendo que el objeto se identifica con un rombo. De hecho lo es, porque visualmente y/o geoméricamente hablando, el asterisco en el ejemplo en cuestión está más cerca de los rombos, confirmando esto la tesis hipotéticamente planteada.

Finalizando con este estudio sobre la SVM donde se ha podido comprobar la posibilidad de crear multclasificadores basados en el tipo de respuesta que pueden dar un modelo con estas características y en particular, los modelos basados en la respuesta aportada por la distancia al hiperplano de separación, se verá más adelante en esta Tesis, que será el método elegido con una sencilla modificación para realizar clasificaciones en datos multiclase, generando además medidas adicionales de fiabilidad.

3.7 LA PREDICCIÓN CONFORMAL

La predicción conformal como se le denomina y define en [Vovk, Gammernan y Shafer, 2005] es un método de aprendizaje automático utilizado para resolver problemas tanto de clasificación como de regresión. La predicción conformal a diferencia de otros métodos de aprendizaje automático se caracteriza por incluir en la predicción, medidas de fiabilidad. Esta característica hace que sean predictores de confianza.

Para el caso de problemas de clasificación, como los que se estudian en esta investigación, la predicción conformal puede producir predictores $I^{0.05}$ muy precisos.

El predictor construido, anida regiones de predicción que contiene la etiqueta desconocida con una probabilidad de 1 menos el nivel de significación ε ($1-\varepsilon$). La notación Γ^ε se utiliza para definir al predictor construido bajo un conjunto de etiquetas con un nivel de significación ε que es prefijado y puede tomar valores entre 0 y 1, es decir $\varepsilon \in \{0,1\}$ [Shafer y Vovk, 2008].

Cuando se indica que la predicción conformal produce regiones de predicción, además de que son anidadas; se hace referencia al análisis hecho sobre la experiencia pasada a través de las muestras (objetos, ejemplos u observaciones) que llegan previamente antes de dar una predicción sobre un nuevo objeto.

Dicho de otra forma, el predictor se basa en un conjunto de datos disponible y a partir del análisis de estos, se puede identificar en el caso de la clasificación de datos con presencia de por lo menos dos clases, la clase de una nueva muestra. El proceso usa la información recabada de cada muestra analizada e incorporada al predictor y con ella define medidas de fiabilidad que informan sobre los datos usados y la propia muestra que está siendo analizada. Esta es una característica de la predicción conformal que la convierte en una herramienta implementable en un aproximaciones on line, pero también suele usarse de manera off line [Vovk, Gammerman y Shafer, 2005].

La única imposición que requiere la predicción conformal sobre los datos disponibles y la muestra a clasificar, es la suposición de que estos son independientes e idénticamente distribuidas (IID) [Vovk, Gammerman y Shafer, 2005] [Shafer y Vovk, 2008]. Además de necesitar, de un algoritmo de aprendizaje tradicional o algoritmo subyacente para realizar predicciones \hat{y} .

En cuanto al algoritmo subyacente necesario, hoy por hoy suelen usarse: las máquinas de vectores soporte [Shafer y Vovk, 2008], los algoritmos de agrupamiento [Shafer y Vovk, 2008] y las redes neuronales artificiales [Papadopoulos, 2008]. En esta Tesis, el predictor base utilizado es la máquina de vectores soporte (SVM).

La predicción conformal requiere de un algoritmo subyacente para construir una medida de no conformidad α_i que mide lo inusual que puede ser un objeto con relación a los ejemplos anteriores. Al ser calculadas todas estas medidas de no

conformidad, se convierten en regiones de predicción válidas [Vovk, Gammerman y Shafer, 2005].

Lo más importante del cálculo de las medidas de no conformidad, sin menospreciar la predicción puntual que puede hacer directamente su valor para identificar la clase de un objeto, es que implícitamente se obtienen medidas de confianza (*Conf*) y credibilidad (*Cred*) que apoyan la predicción realizada [Saunders, Gammerman y Vovk 1999] [Vovk, Gammerman y Shafer, 2005] [Shafer y Vovk, 2008].

Estos valores de confianza y credibilidad están restringidos entre cero y uno ($Conf, Cred \in \{0,1\}$), y directamente relacionados con el conjunto de datos disponible para entrenar al predictor y el objeto a predecir. Por un lado, el valor de la confianza, da información sobre el conjunto de entrenamiento. Al aproximarse a uno revela la probabilidad de que ninguna de las etiquetas de los objetos de entrenamiento excluidas sea la correcta. Por otro lado, el valor de credibilidad nos da información sobre el objeto a predecir. Al aproximarse a cero, indica que el objeto a predecir no es conforme con el conjunto de datos disponible [Saunders, Gammerman y Vovk, 1999] [Shafer y Vovk, 2008].

Otra característica importante que se comenta antes de pasar a definir formalmente y explicar la teoría de cómo funciona la predicción conformal, es su naturaleza de aprendizaje transductivo [Cherkassky y Mulier, 2007]. El predictor conformal (regla de aprendizaje o modelo: como regiones de predicción anidadas) que arroja una predicción puntual, se infiere a partir de las observaciones vistas previamente y el propio objeto a predecir.

Lo que se está explicando, sin ánimo de insistir en el tema pero con el sentido de resaltar esta importante característica, es que la predicción conformal no infiere un modelo por inducción [Cherkassky y Mulier, 2007] que luego es usado para analizar muestras. Esta es capaz de generar un modelo que impone la propia muestra que está siendo analizada. De ahí su importancia sobre todo en problemas con poca información.

Todas estas características comentadas a lo largo de esta introducción, que posee la predicción conformal, hacen que este método sea válido y eficiente al incorporar importantes propiedades como son: el conjunto de predicciones anidadas, validez, eficiencia, condicionalidad y flexibilidad [Vovk, Gammernan y Shafer, 2005], [Shafer y Vovk, 2008]. Estas propiedades pueden ser ampliadas en las referencias citadas previamente.

3.7.1 Definición general de la predicción conformal en clasificación

Siendo coherentes con las definiciones dadas con anterioridad a lo largo de este trabajo de investigación y siguiendo a [Papadopoulos, 2008], se proporcionará una definición muy general de la predicción conformal en problemas de clasificación. Aunque esta definición, luego será adaptada desde el punto de vista práctico para resolver el problema de multclasificación estudiado en esta Tesis.

Dado un conjunto de datos disponible $(x_1, y_1), \dots, (x_n, y_n)$ donde x_n representa en sí los objetos del conjunto perteneciente a un espacio medible X con $X \in \mathbb{R}^d$, e y_n su correspondiente etiqueta perteneciente a un espacio medible C con $C \in \{c_1, \dots, c_j\}$. Nótese que se tiene una secuencia de objetos (x_n, y_n) que puede escribirse en forma compacta como el conjunto de observaciones o muestras z_n del tipo IID. Dado también, un objeto nuevo x_{n+1} medido en X con el propósito de predecir (clasificar) a y_{n+1} , que está contenida en C , se define la predicción conformal como el algoritmo que asigna a cada objeto de la secuencia de observaciones (z_n, x_{n+1}) , una medida de no conformidad α_i con $i=1, \dots, n+1$, computables y que se resume como una predicción sobre y_{n+1} , asociando medidas adicionales de fiabilidad. Es decir, para cada:

$$(x_1, y_1), \dots, (x_n, y_n), (x_{n+1}, y_c) \quad (3.46)$$

Se asigna

$$\alpha_i = \alpha_1, \dots, \alpha_n, \alpha_{n+1} = A((x_1, y_1), \dots, (x_n, y_n), (x_{n+1}, y_c)) \quad (3.47)$$

De las expresiones (3.46) y (3.47) se desprenden tres aspectos importantes. El primero es y_c que en (3.46) representa la etiqueta que debe ser predicha en el espacio C , se le denomina: etiqueta de clase. El segundo aspecto se refiere a la función A , que asigna las medidas de no conformidad como se observa en (3.47), que se construye en base a un algoritmo de aprendizaje tradicional (más adelante se verá cómo se define en este trabajo dicha medida a partir de una SVM). El tercer aspecto es muy importante por la implicación que tiene y_c al calcularse las medidas de no conformidad, puesto que su asignación se hace bajo la asunción de tal etiqueta.

Ampliando este último aspecto, se comenta que al ser asignadas las medidas de no conformidad, implícitamente se está diferenciando un objeto con etiqueta y_c de la secuencia dada, con respecto al resto. Por lo tanto, al ser calculadas todas estas medidas de no conformidad bajo el orden (la asunción de hipótesis) de y_c como ya se ha indicado, se construye un estadístico, probabilidad o medida de incertidumbre, definido como p_valor . Este estadístico informativo del objeto x_{n+1} da el carácter predictivo de la predicción conformal y expresa al mismo tiempo, medidas de fiabilidad como se analizará más adelante.

En este sentido, el p_valor se puede ver como una evaluación generalizada de los α_i con respecto al objeto (x_{n+1}, y_c) . Pudiendo calcularse su valor mediante la siguiente expresión:

$$p_valor = \frac{\#\{i = 1, \dots, n+1 : \alpha_i \geq \alpha_{n+1}\}}{n+1} \quad (3.48)$$

Ahora bien, si se reescribe la secuencia de observaciones en (3.46) como z_1, z_2, \dots, z_n , haciendo coincidir el número de observaciones $n+1$, simplemente diciendo que hay n ; se transforma (3.48) en la forma como puede verse en la mayoría de los escritos técnicos, es decir:

$$p_valor = \frac{\#\{i = 1, \dots, n : \alpha_i \geq \alpha_n\}}{n} \quad (3.49)$$

Formalizando ahora la idea conceptual de la evaluación hecha sobre cada α_i a partir de un p_valor , y entendiendo que las mismas son asignadas a cada objeto de la secuencia dada, además de cotejar la expresión (3.49) que refleja su rango posible desde $\frac{1}{n}$ hasta n , se concluye, apoyados en [Vovk, Gammerman y Shafer, 2005], que si su valor tiende a uno ($p_valor \rightarrow 1$) más típico es el objeto correlacionado con α_i dentro de la secuencia observada.

Esta conclusión final es interesante, ya que la evaluación se hace en base al objeto a predecir suponiendo la hipótesis y_c , lo que implica que se está tipificando la muestra analizada.

Se pueden formular otras definiciones para la predicción conformal cuando es aplicada a problemas de clasificación [Vovk, Gammerman y Shafer, 2005]. Más aún, para definir un predictor conformal I^ϵ , se debe prefijar el parámetro ϵ como lo indica su definición vista en la introducción, porque entre otras cosas, su valor tiene que ver con el tamaño del subconjunto de datos utilizado o no excluido durante el entrenamiento del predictor.

Se aporta además, que por la propiedad de regiones de predicción anidadas y validez vistas en [Vovk, Gammerman y Shafer, 2005], al definir la predicción conformal como un predictor de confianza, se obliga a este tipo de predictores a garantizar un subconjunto eficiente y suficiente para alcanzar una predicción válida, por lo tanto ϵ controla el número de observaciones que se incluyen con anterioridad a una predicción.

Esto evidentemente tiene una exigencia computacional mayor porque el predictor está obligado, tal y como ya se ha señalado, a incluir etiquetas en el conjunto predictor que contengan la etiqueta verdadera, dando así la garantía de confianza ante una predicción.

Un valioso análisis en este sentido se hace en [Gammerman y Vovk, 2007] al tratarse el problema de la predicción con confianza. A rasgos generales se indica en la referencia citada, que en la medida que se aumente el nivel de confianza $1-\epsilon$, el

subconjunto de etiquetas predichas debe ser lo suficientemente grande como para contener la etiqueta verdadera.

Finalmente se indica que todos los conceptos teóricos asociados a la predicción conformal y resumidos en las expresiones dadas en (3.46), (3.47) y (3.49), se pondrán de manifiesto con la introducción de algunos ejemplos prácticos. Comenzando inicialmente con un problema de clasificación simple y luego se profundizará en la multclasificación basada en la aproximación uno versus el resto, usando como medida de conformidad la distancia al margen de separación que proporciona la SVM.

3.7.2 Funcionamiento de la predicción conformal en clasificación

Vista la definición previamente realizada y específica para un problema de clasificación, la predicción conformal puede verse como un algoritmo implementado en dos capas (véase la Figura 3.18). Una primera capa que genera un conjunto de medidas de no conformidad, de una secuencia de observaciones y un objeto a predecir condicionadas a este último. Y una segunda capa que computa estas medidas de no conformidad, resumiéndolas en un valor informativo definido como p_valor que lleva implícito la predicción del objeto y medidas de fiabilidad adicionales.

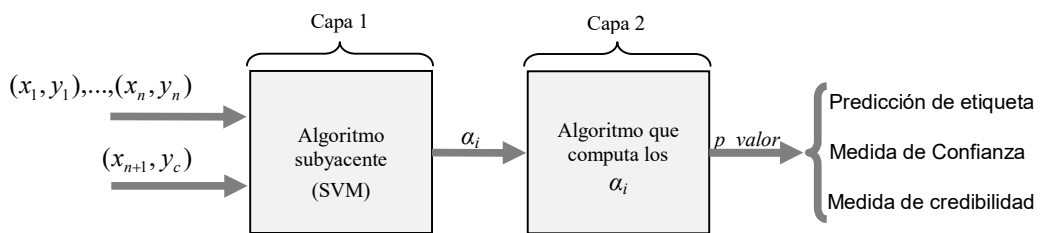


Figura 3.18.- Capas de la predicción conformal

En las siguientes secciones se analizará la predicción conformal sin centrarse en el concepto de capas mencionado, pero es relevante la ilustración hecha para entender que no es un algoritmo compacto (único), sobre todo por las modificaciones futuras que también serán explicadas para lograr mejoras a nivel computacional.

3.7.2.1 Medidas de no conformidad a partir de la distancia al hiperplano

Como ya se ha introducido, estas *medidas de no conformidad*²⁶ se toman a partir de un algoritmo de aprendizaje tradicional utilizando sus propiedades predictivas, y basadas en estas se construye una regla que indica lo diferente que es un objeto del resto en una secuencia de observaciones. Para el caso de las SVM, como *propiedad*²⁷ predictiva se usan los vectores soportes conseguidos tras un entrenamiento para resolver el problema de biclasificación planteado por un conjunto de datos dispuesto, incluyendo el objeto a ser predicho (véase la sección 3.6). Al conjunto de datos dispuesto y el objeto a predecir, se les denominará en adelante: “*conjunto observado*” o simplemente “*observaciones*”.

Con los vectores soportes comentados anteriormente, se obtiene un hiperplano que separa las clases dispuestas en el conjunto observado como solución del problema planteado. Al mismo tiempo, el hiperplano sirve para referenciar las observaciones a partir de la distancia desde cada muestra en particular al propio hiperplano encontrado. Como consecuencia inmediata de esto, se le ha proporcionado a la distancia al hiperplano, capacidades predictivas. Además, esto sugiere que la distancia al hiperplano se puede considerar como otra propiedad predictiva de la SVM. Como argumento ante lo dicho, se recuerda que en la sección 3.6.2, en las Figuras 3.16 y 3.17 se propusieron los puntos (-3,2.5) y (-3,-2) que fueron perfectamente identificados por su ubicación visual. Aunque el segundo objeto (-3,-2) no pudo ser identificado mediante la predicción que daba la SVM (bajo la discusión realizada en el Ejemplo 6) y se tuvo que recurrir a la distancia al hiperplano como medida decisiva para eliminar la incertidumbre hipotéticamente creada, confirmado de esta manera la propiedad predictiva que posee la distancia al hiperplano.

²⁶ En las referencias bibliográficas citadas en todo la sección en cuestión, se puede ver que cualquier algoritmo de aprendizaje tradicional se puede implementar para conseguir medidas de no conformidad. Sin embargo, en esta investigación sólo se utiliza la SVM para generar y asignar tales medidas de conformidad.

²⁷ En [REAL ACADEMIA ESPAÑOLA] se define “propiedad” como: “Atributo o cualidad esencial de alguien o algo. Una búsqueda en google permite ampliar la definición con: “una condición, una característica, un estado o una facultad de algo”.

Ahora bien, no sólo se trata de identificar un objeto particular sobre estos planos, sino también de implementar medidas de fiabilidad para evaluar de alguna manera la incertidumbre latente a la hora de clasificar una muestra, tal y como sugiere la teoría de la predicción conformal a rasgos generales. Sobre todo cuando se tienen casos de clasificación como el visto en el Ejemplo 6 y que fue ilustrado en la Figura 3.17.

No obstante, en este ejemplo mencionado, el punto estudiado no es de los más extremos, porque visualmente el objeto asterisco se inclinaba indudablemente hacia la clase rombo y con él los resultados vistos. Por esto y a manera de introducir el tema de los escenarios de prueba, se propone un tercer punto que sea realmente difícil de clasificar a simple vista.

Ejemplo 7. Introducción a los escenarios de prueba de hipótesis

Con el fin de introducir los escenarios de prueba, se propone colocar punto en las coordenadas (0,0) del plano cartesiano, donde se dibujan los datos disponibles usados para los ejemplos referentes de las Figuras 3.16 y 3.17. Al analizar esta nueva ubicación con la ayuda de la SVM, se pudiera encontrar para tal objeto, la clase correspondiente como solución. Pese a esto, se tendría la incertidumbre nuevamente de sí el modelo inferido por el conjunto de observaciones arroja una predicción confiable.

Esto ocurre porque para la clasificación efectuada en este nuevo planteamiento de datos multiclase formulado a partir de las figuras en cuestión, es imprescindible controlar los parámetros de la SVM, pudiéndose en consecuencia sintonizar valores inadecuados que conduzca a una clasificación ineficiente. Y por la ubicación del objeto (punto en (0,0) planteado), es poco práctico e improbable que a partir de un análisis visual sobre los cuatro escenarios de predicción creados se resuelva tal incertidumbre.

A la postre de esto y desde el punto de vista de la predicción conformal se pudiera generar una predicción confiable, si se transforma cualquiera de los problemas anteriormente planteados, creando escenarios adicionales que permitan

validar alguna hipótesis sobre a qué clase se le atribuye el objeto a clasificar. Véase esto en la siguiente sección.

3.7.2.2 Prueba de hipótesis a partir de escenarios basados en la distancia al hiperplano

Para entender este tema de los escenarios y la confirmación de hipótesis de una manera sencilla, se propone inicialmente un ejemplo con datos biclase para posteriormente utilizar datos multiclase y de esta manera llegar a definir de manera práctica a los multclasificadores basados en la predicción conformal.

Ejemplo 8. Introducción a los escenarios de prueba de hipótesis. Un ejemplo con datos biclase

En la Figura 3.19 (a) se plantea un problema con datos biclase inicialmente, conformado por cinco puntos bien diferenciados. Obsérvese las *Clase +1* y *-1*, definidas por los dos cuadrados negros y círculos azules respectivamente. Además nótese el objeto asterisco rojo, el cual no se identifica a qué clase de figura geométrica pertenece.

Al ver ahora la Figura 3.19 (b) se observa una línea dibujada con ayuda de la SVM, la cual separa al objeto asterisco rojo junto con los círculos azules del resto de las figuras geométricas (cuadrados negros). Aquí, se ha supuesto que el asterisco pertenece a la clase círculos azules para poder inferir el modelo que dibuja tal hiperplano. De esta manera, se define la primera hipótesis (un escenario) de que el asterisco corresponde con la *Clase +1* (un círculo azul).

Análogamente, se realiza la hipótesis de que el asterisco pertenece a la clase de figuras geométricas cuadrados negros, y en consecuencia se genera un segundo hiperplano que puede separar estos objetos de los círculos azules, tal y como puede verse en la Figura 3.19 (c). Esto define la otra hipótesis (otro escenario) de que el asterisco corresponde pero ahora con la *Clase -1* (un cuadrado negro).

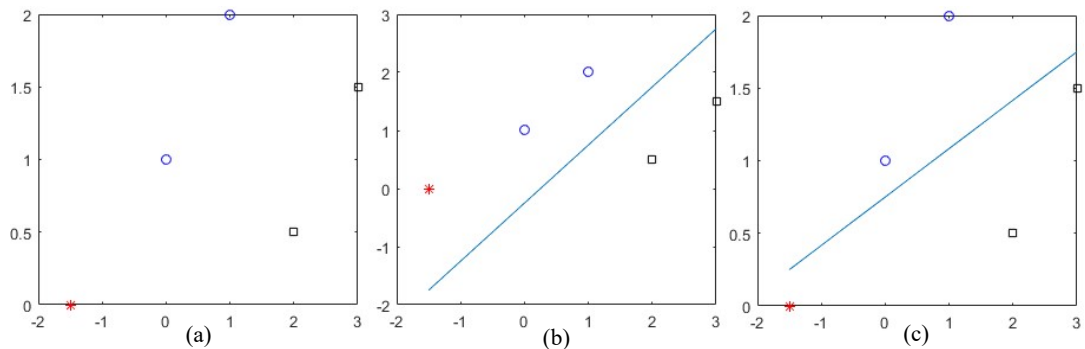


Figura 3.19.- Creación de escenarios para prueba de hipótesis mediante la distancia al hiperplano. (a) Datos disponibles. *Clase +1*: círculos azules, *Clase -1*: cuadrados negros y objeto a predecir: asterisco rojo. (b) Hiperplano bajo la hipótesis que el objeto a predecir es de la *Clase +1*. (c) Hiperplano bajo la hipótesis que el objeto a predecir es de la *Clase -1*.

Como se puede apreciar, se han creado dos escenarios de los cuales, la hipótesis válida será aquella donde se confirme la máxima distancia desde el asterisco rojo a cualquiera de los hiperplanos dibujados en la Figura 3.19 (b) o (c), bajo la premisa de que el asterisco rojo pertenecerá a aquella clase que plantee un mejor hiperplano de separación para el conjunto de datos observado.

Esta premisa se da, porque bajo cada hipótesis planteada se ha condicionado el hiperplano generado, al objeto a predecir. Esto ocurre por definición de la SVM, puesto que la misma debe optimizar las distancias de todos los objetos involucrados, incluyendo las del objeto supuesto como una clase u otra (véase la sección 3.6 relativa a las SVM). Más aún, al plantear el cambio de etiquetas del objeto a predecir (asunción de hipótesis), se han condicionado todas las distancias que se ven desde cada observación al hiperplano, en cualquiera de los dos escenarios mostrados en la Figura 3.19 (b) y (c).

En resumen, la premisa adoptada será: “dados los diferentes escenarios con su correspondiente hiperplano que separe las clases observadas, el objeto bajo hipótesis estará más cerca de la clase que registre la mayor distancia desde dicho objeto hipótesis al hiperplano”. Este criterio adoptado, puede ser corroborado en la Tabla 3.9, donde se muestra la información que implica generar un modelo SVM. En las columnas 1, 2 y 3 se tienen la información referente a los datos dispuestos. En la

última fila de estas columnas, se muestran las coordenadas del objeto asterisco rojo con etiqueta “+1”. Véase en las columnas 4 y 5, la predicción hecha por cada modelo generado donde se asume la hipótesis de que el objeto asterisco rojo pertenece a la Clase +1 y -1 respectivamente.

Tabla 3.9.- Información implicada al generar un modelo SVM.

Lo que Implica Generar el Modelo SVM											
Objetos		Predicción Según la Hipótesis			Distancia al Hiperplano bajo la Hipótesis		Multiplicadores de Lagrange bajo la Hipótesis		Medida de No Conformidad bajo la Hipótesis		
X	Y	Clase +1	Clase -1	Clase +1	Clases -1	Clase +1	Clase -1	Clase +1	Clase -1		
0.0	1.0	+1	+1	+1	1.0000	0.9994	0.1600	8.8847	-1.0000	-0.9994	
1.0	2.0	+1	+1	+1	1.0000	3.6648	0.4800	0	-1.0000	-3.6648	
2.0	0.5	-1	-1	-1	-1.0000	-3.6651	-0.6400	0	-1.0000	-3.6651	
3.0	1.5	-1	-1	-1	-1.0000	-0.9997	0	-3.2577	-1.0000	-0.9997	
-1.5	0.0	+1	+1	-1	1.4000	-0.9997	0	-5.6269	-1.4000	-0.9997	
									<i>p_valor</i>	1.0	0.6

Lo que implica que cambien las distancias relativas al hiperplano de separación ante cada modelo. Esto se muestra en las columnas 6 y 7. Hasta aquí se puede confirmar la hipótesis de que la muestra en cuestión es de la Clase +1, porque se observa que la mayor distancia al hiperplano se ha obtenido al asumir que la hipótesis es un círculo azul.

Quizás el lector no experimentado piense que la premisa de tomar como válida aquella hipótesis que confirme la mayor distancia, sea apresurada, arbitraria o simplemente trivial. Pero nada más alejado de la realidad, esta premisa se fundamenta en la teoría relacionada con el algoritmo de aprendizaje o predictor base, es decir la SVM. Con su aplicación, se ha optimizado el plano de separación entre los objetos usados en el problema de biclasificación de en la Figura 3.19 y que incluye el asterisco rojo como previamente se ha comentado, formulándose dos problemas en lugar de uno, resultando como solución dos escenarios posibles tal y como se ha visto.

Estas soluciones encontradas para ambos problemas mediante el empleo de *técnicas estándar de programación cuadrática*²⁸ como sugiere [Carmona, 2014], se

²⁸ Véase la sección 3.6 donde se discute cómo solucionar un problema de clasificación aplicando la SVM.

logran porque en el proceso se determinan los llamados multiplicadores de Lagrange (también suelen denotarse como α_i), los cuales son mostrados en las columnas 8 y 9 de la Tabla 3.9.

A través de estos valores es que se ha podido, entre otras cosas, trazar el hiperplano de separación para la Figura 3.19 (b) y (c), ya que los mismos apuntan directamente sobre los objetos vectores soportes seleccionados como solución óptima que encuentra el algoritmo de la SVM al problema de separar el conjunto de datos observado en uno u otro escenario.

Para el caso de los objetos vectores soporte como solución al problema dado, estos siempre tendrán asociado un multiplicador de Lagrange o α_i diferente de cero ($\alpha_i \neq 0$) según las restricciones impuestas en la expresión (3.21) de la sección 3.6. Al ser las observaciones más cercanas (los objetos vectores soporte) al hiperplano dibujado según la teoría estudiada sobre la SVM, por lo tanto: “aquellas observaciones que no cumplan con tal condición de ser un vector soporte, son muestras consideradas alejadas del hiperplano”.

Con esto se puede confirmar que el objeto hipótesis, visto en la Figura 3.19 (b), está obviamente distanciado. Por las dudas que puedan quedar por lo revelado, se puede contrastar el valor α_i ($\alpha_i = 0$) mostrado en la última fila de la columna 8 en la Tabla 3.9, para validar de esta manera la premisa escrita.

3.7.2.3 Asignación de medidas de no conformidad a partir de la distancia al hiperplano

Vistos los diferentes escenarios planteados en el apartado anterior, propiciados por la asunción de hipótesis, y entendiendo la forma como se reflejan las diferencias significativas en la distancia de cada objeto del conjunto de datos observado al respectivo hiperplano de separación, además de permitir la comprobación de la distancia más extrema de un objeto hipótesis, toca ahora convertir tales distancias en medidas de no conformidad.

Esto se puede hacer pensando en el concepto del p_valor visto en la expresión (3.48) que no es más que un resumen informativo de la *tipicidad*²⁹ de un objeto, pudiéndose ver desde varios ángulos. Uno, el dado en [Vovk, Gammerman y Shafer, 2005] donde se indica que “*el p_valor es una medida de cuán extraño*” es un “*ejemplo de prueba*” (objeto) “*cuando se le da una determinada clasificación*”, señalando además que su valor “*dice la probabilidad de observar*” las medidas de no conformidad bajo el orden de que el objeto de prueba es la clasificación correcta. Dos, lo dicho en [Saunders, Gammerman y Vovk, 1999], que el p_valor es “*la fracción de los ejemplos en una secuencia tan disconforme como el*” objeto a predecir.

Utilizando esto dos puntos de vista como argumento, se puede entonces adaptar las distancias desde cada objeto al hiperplano de separación vistas en las Figuras 3.19 (b) y (c), de tal manera que reflejen de alguna forma lo extraño que es el asterisco rojo en la Figura 3.19 (b) al ocupar la distancia más larga de los dos escenarios, además de medir por esta vía la fracción del resto de los objetos que son tan igual o más extremos que ellos mismos.

Es así como se formaliza la regla que asigna las medidas de no conformidad a partir de la distancia al hiperplano, pudiéndose escribir como sigue:

$$\alpha_i = \begin{cases} -|d| & \text{si } y_{\text{predicha}} = y_c \\ +|d| & \text{si } y_{\text{predicha}} \neq y_c \end{cases} \quad (3.50)$$

Donde d es la distancia desde un objeto cualquiera hasta el hiperplano, y_{predicha} es la etiqueta pronosticada por la SVM, e y_c es la etiqueta de clase, que para el caso del objeto a pronosticar, se ha supuesto para los dos escenarios. Si se observan las columnas 10 y 11 de la Tabla 3.9, allí se encuentran las medidas de no conformidad obtenidas a partir de la expresión (3.50) para los datos del Ejemplo 8.

²⁹ En este trabajo se da por entendido que la tipicidad está relacionada por algo que es típico, extremo o extraño.

Antes de computar estos valores de no conformidad usando la expresión (3.49) bajo la hipótesis supuesta para cada objeto a pronosticar y así determinar la fracción por columna de las observaciones que están tan cercanas o más como el propio objeto bajo hipótesis, se explicará metafóricamente el sentido práctico de todo lo señalado.

Para ello, véase al escenario propuesto en la Figura 3.19 (b) y nótese que se tienen cuatro muestras que están más cercanas al hiperplano que el propio asterisco rojo. Es decir, estas cuatro observaciones pueden confirmar que el objeto bajo hipótesis (se asume que es un círculo azul) es extraño o extremo. Para el caso de la Figura 3.19 (c), sólo se cuenta con dos observaciones que están tan cercanas o más del hiperplano correspondiente, y aunque tal fracción puede confirmar que el objeto a pronosticar bajo la otra hipótesis (se asume ahora, que es un cuadrado negro) es extraño, no tendrían la *potencia suficiente*³⁰ para ratificarlo.

Al final de la columna 10 de la Tabla 3.9 está representada una parte de la metáfora, al reflejarse la fracción de las observaciones que indican para este escenario lo extraño que es el objeto bajo hipótesis, al sumar todos aquellos valores que sean mayores o iguales a -1.4, dando un *p_valor* igual a 1 (*p_valor*=1). Véase que se ha incluido en tal suma el objeto bajo estudio. Para la otra parte de la metáfora, lo propio se hizo al final de la siguiente columna ya mencionada, obteniendo un *p_valor* de 0.6 (*p_valor*=0.6).

3.7.2.4 La predicción y las medidas de fiabilidad: confianza y credibilidad

En la Tabla 3.9 también se observan dos *p_valor*, de los cuales por todo el análisis hecho y bajo los argumentos señalados con anterioridad, se ha tomado el mayor *p_valor*, concluyendo con esto que el asterisco rojo pertenece a la clase de círculos azules.

Para el caso del ejemplo en cuestión (Ejemplo 8) sólo se tienen dos clases. Pero, si se generaliza el concepto para más de dos clases *c* presentes en un conjunto

³⁰ La frase “potencia suficiente” se atribuye en esta tesis al mayor número de muestras posibles que confirmar lo alejado que esta un objeto bajo hipótesis. Esta confirmación se da por implícita, porque las muestras realmente están tan cercanas o más que el propio objeto estudiado.

de datos observados donde se debe predecir una etiqueta y_c , entonces se tendría como resumen a un conjunto de p_valor . Es decir, existirían tantos p_valor como clases existan. Por lo tanto, la etiqueta y_c correcta, pertenecerá al conjunto de datos observados que obtenga el máximo p_valor , pudiendo escribirse esto como sigue.

$$\hat{y}=y_c|\max\{p_valor_1,\dots,p_valor_c\} \quad (3.51)$$

Además de esto, también se tienen que tomar las medidas de fiabilidad que para el caso de una predicción puntual conseguida, mide la fuerza con la cual se asume la decisión en cuanto a la fracción de observaciones que aportan información ante la predicción. Muchos autores afirman que estos valores estiman la calidad de la predicción hecha como por ejemplo [Saunders, Gammerman y Vovk, 1999].

Según la teoría, como medida de credibilidad se debe tomar el mayor p_valor encontrado [Saunders, Gammerman y Vovk, 1999]. Por lo tanto la credibilidad para el Ejemplo 8 tiene un valor igual a uno ($Cred=1$). Traduciendo esto a un problema multiclase, se tendría que la credibilidad debe coincidir con el máximo p_valor encontrado, pudiéndose escribir directamente como P_1 . Por lo tanto:

$$Cred=P_1 \quad (3.52)$$

Por otro lado, para el valor de la confianza se recurre al segundo máximo p_valor [Saunders, Gammerman y Vovk, 1999] que por comodidad se escribe como P_2 . Siendo entonces el cálculo de la confianza igual a uno menos el segundo p_valor . Para el Ejemplo 8, tal confianza da como resultado 0.4 ($Conf=0.4$) y generalizado a un problema multiclase, después de obtener al segundo máximo p_valor (P_2), se tiene el valor de la confianza:

$$Conf=1-P_2 \quad (3.53)$$

Para finalizar, se recuerda que por definición al inferir el predictor I^ϵ , necesariamente debe ser prefijado el parámetro ϵ o nivel de significación. Comentarios sobre esto se dejaron en la definición hecha en la sección 3.7.1.

3.7.3 Multclasificación con predicción conformal

En la sección 3.6.2, se analizó el problema de multclasificación mediante la SVM pero sin utilizar predicciones con mediadas de fiabilidad. Posteriormente se analizó, como obtener predicciones confiables (sección 3.7.2) en un problema biclase, al introducir lo que se denominó “escenarios de prueba de hipótesis”.

Ahora se analizará el problema multiclase, sin embargo la metodología de los “escenarios de prueba de hipótesis” no puede ser aplicada directamente a un problema multiclase, si no se realizan modificaciones que introducen escenarios adicionales. Para comprender estas modificaciones, a continuación se verá un ejemplo donde fallaría el concepto inicial de los escenarios de prueba de hipótesis ya vistos.

Ejemplo 9. Introducción a los escenarios de prueba de hipótesis. Un ejemplo con datos multiclase fallido

En la Figura 3.20 se muestra un conjunto de datos multiclase conformado por seis puntos, distribuidos en tres grupos de figuras geométricas (*Clase 1*: círculos, *Clase 2*: cuadrado y *Clase 3*: rombos), y un objeto asterisco rojo que debe ser identificado en alguna de las clases de las figuras mencionadas.

En este caso de datos multiclase, se puede usar un predictor conformal para decidir a qué grupo pertenece el objeto a clasificar, pero se deben tener presente dos aspectos:

- El primero, es el hecho de que al estar en presencia de un problema multiclase, se tiene que aplicar alguno de las dos aproximaciones vistas en la sección 3.6.2. En particular se utilizará la aproximación uno versus el resto.
- El segundo, es que ahora no se va a usar simplemente una SVM para hacer la predicción, sino que se desea conseguir predicciones con medidas de

fiabilidad y por lo tanto, hay que asumir de alguna forma que el asterisco rojo pertenece a una clase determinada, y en contra de tal hipótesis plantear también, que no pertenece a las otras dos clases.

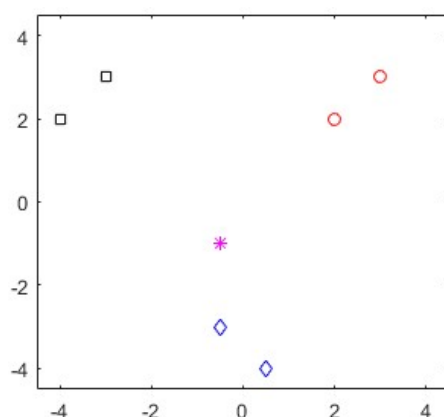


Figura 3.20.- Datos disponibles para la creación de escenarios en un problema multiclase. *Clase 1:* círculos, *Clase 2:* cuadrados, *Clase 3:* rombos, asterisco: punto a identificar.

Según estos dos aspectos, al razonar como se hizo en el Ejemplo 8, se obtendrán los seis escenarios mostrados en la Figura 3.21 (a), (d), (g), (e), (b) y (c)³¹. Si se observa bien la secuencia de estos escenarios, se podrá notar que se han formulado seis hipótesis como se indica en la Tabla 3.10. Las primeras tres hipótesis expresan que el objeto asterisco pertenece a las figuras geométricas círculo, cuadrado y rombo respectivamente. Y las otras tres expresan que el objeto asterisco no pertenece a tales figuras geométricas respectivamente.

Tabla 3.10.- Forma incorrecta de formular hipótesis según los escenarios de prueba de hipótesis para los datos multiclase de la Figura 3.20.

Escenario	Formula que	Escenario	Formula que
(a)	$* \in \circ$	(e)	$* \notin \circ$
(d)	$* \in \square$	(b)	$* \notin \square$
(g)	$* \in \diamond$	(c)	$* \notin \diamond$

³¹ Los escenarios (e), (b) y (c) son iguales a (h), (i), (f) respectivamente.

Además y como consecuencia de estos escenarios mostrados, se tendrán las distancias al hiperplano, y a partir de la expresión (3.50) se asignarán las correspondientes mediadas de no conformidad, logrando finalmente un p_valor como resumen a través de la expresión (3.49) para cada conjunto observado (en cada escenario) según la hipótesis asumida.

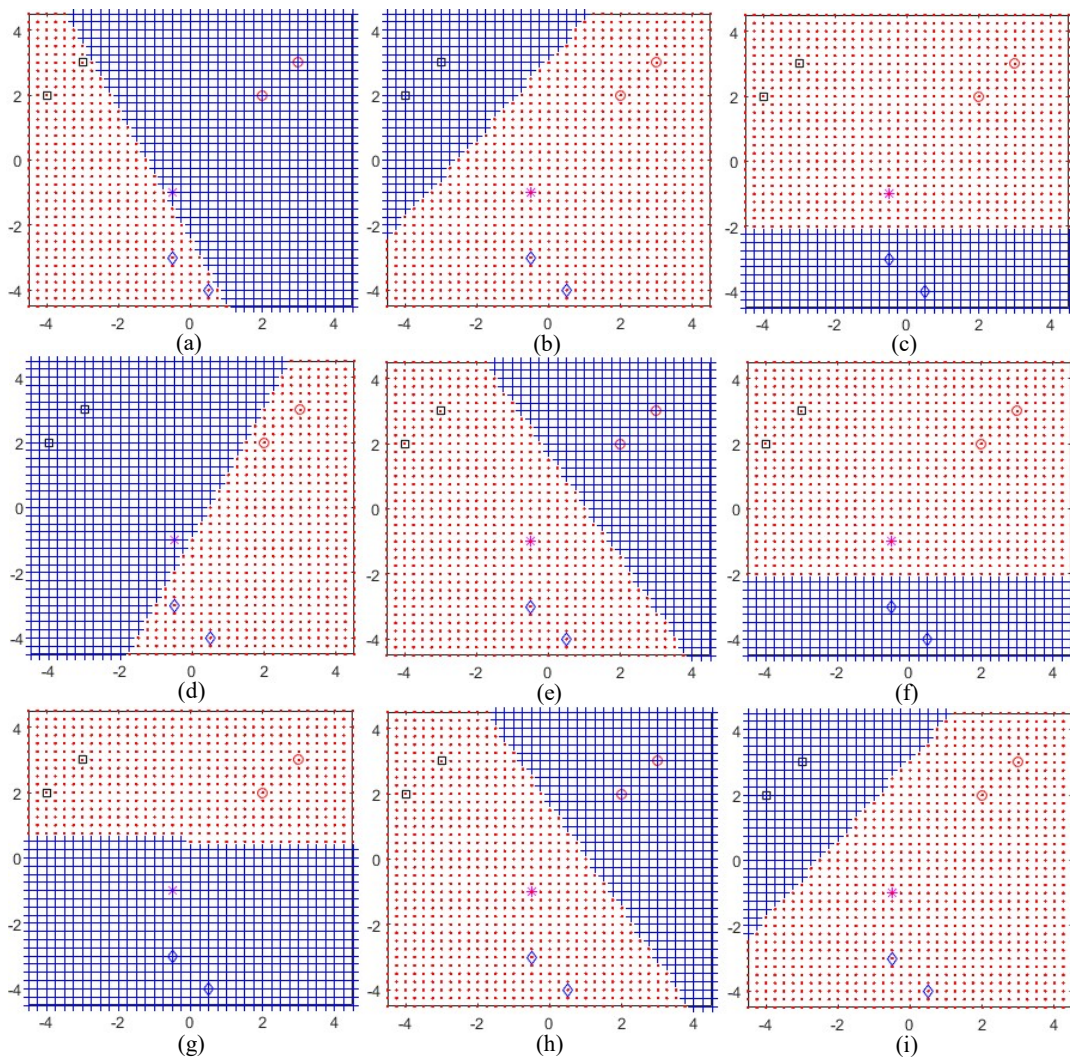


Figura 3.21.- Mapas de predicción posibles producido por los hiperplano $h(x)$ inferidos con los datos observados en la Figura 3.20. Clase +1: signo de suma “+” en color azul. Clase -1: punto “.” en color rojo.

De todos los $p_valores$ encontrados según lo visto con anterioridad, se debe tomar, tanto la predicción como las medidas de confianza y credibilidad. Sin embargo, se puede demostrar que este análisis puede conducir a una encrucijada, no pudiendo hallar una respuesta satisfactoria. Todo ello debido principalmente a que en un problema multiclase no necesariamente se debe aplicar un kernel lineal como el usado para plantear los referidos mapas de la Figura 3.21.

Para confirmar todo esto, se deja un resumen en la Tabla 3.11 de las medidas de no conformidad conseguidas a partir de las distancias al hiperplano obtenidas para los seis escenarios mostrados, con su respectivo p_valor . Nótese que existen un p_valor alto (0.4285), pero apunta a un escenario que indica que el objeto asterisco no es un círculo. Los otros dos $p_valores$ altos son iguales (0.2857), pero ambos se contradicen ya que apuntan a los escenarios donde se tienen las hipótesis de que le objeto asterisco “es un rombo” y “no es un rombo” respectivamente. Finalmente se tiene cuatro $p_valores$ también iguales (0.1429), dejando este análisis como ya se ha indicado en una encrucijada. Al ver todo esto surge la pregunta: ¿Cómo se toma la predicción del objeto asterisco?.

En conclusión, se deja en evidencia que la metodología de los “escenarios de prueba de hipótesis” aplicados en la sección 3.7.2 al problema biclase, no resuelven el problema multiclase planteado en este ejemplo fallido.

Tabla 3.11.- Medidas de no conformidad obtenidas a partir de la distancia al hiperplano encontradas para los escenarios de la Figura 3.21 (a), (d), (g), (e), (b) y (c) con sus correspondientes $p_valores$.

Objetos			Medida de No Conformidad bajo la Hipótesis de que el Asterisco					
X	Y		Es Círculo	Es Cuadrado	Es Rombo	No es Círculo	No es Cuadrado	No es Rombo
2.0	2.0	Círculo	-16.9916	-0.9997	-1.0004	-0.9995	-1.0005	-4.0000
3.0	3.0	Círculo	-22.9887	-1.9994	-1.6673	-1.7421	-1.0579	-5.0000
-3.0	3.0	Cuadrado	-0.9997	-9.9960	-1.6667	-0.9998	-1.0002	-5.0000
-4.0	2.0	Cuadrado	-6.9968	-0.9956	-0.9998	-1.7424	-1.0577	-4.0000
-0.5	-3.0	Rombo	-2.9987	-0.9997	-2.3338	-1.5710	-1.5710	-1.0000
0.5	-4.0	Rombo	-0.9997	-3.9986	-3.0005	-1.3996	-2.1996	-2.0000
-0.5	-1.0	Asterisco	-0.9994	-0.9995	-1.0002	-0.9998	-0.9998	-1.0000
p_valor			0.1429	0.1429	0.2857	0.4285	0.1429	0.2857

3.7.3.1 Multiclasificación con predicción conformal. Escenarios de prueba de hipótesis adicionales y aproximación modificada uno versus el resto

La dificultad vista en el fallido Ejemplo 9, radica en el hecho de que se han formulado mal las hipótesis. Esto ha ocurrido porque se estaba pensando en un problema biclase, cuando realmente se está en presencia de un problema multiclase. Por lo tanto, la forma correcta de plantear las hipótesis, conduce a tener nueve formulaciones como se puede observar en la Tabla 3.12 y no seis como se vio en la Tabla 3.10.

Para que esto ocurra, es imprescindible contar con escenarios de prueba de hipótesis adicionales. Por esta razón se tienen nueve escenarios en la Figura 3.21, pero en el fallido Ejemplo 9, sólo se usaron seis. Claramente, se puede pensar que seguirá existiendo el mismo problema de no poder identificar el objeto asterisco en alguna de las clases asociadas a las figuras geométricas, porque ahora en vez de tener seis $p_valores$ que analizar, se tendrán nueve e indudablemente, la encrucijada que se tenía inicialmente aumenta su dificultad.

Sin embargo y alternativamente a la dificultad vista en el Ejemplo 9, se plantea para resolver el problema de predicción multiclase, el uso de la expresión vista en [Vovk, Gammerman y Shafer, 2005], la cual se escribe como:

$$\begin{aligned}
 & A((x_1, y_1), \dots, (x_n, y_n), ((x_{n+1}, y_c))) = \tag{3.54} \\
 & \lambda A((x_1, I_{y_c}), \dots, (x_n, I_{y_c}), ((x_{n+1}, y_c))) + \\
 & \frac{1-\lambda}{|Y|-1} \sum_{y_c' \neq y_c} A((x_1, I_{y_c'}), \dots, (x_n, I_{y_c'}), ((x_{n+1}, y_c')))
 \end{aligned}$$

Con $y_c = \{-1, +1\}$, con $I = \begin{cases} 1 & \text{si } x \in y_c \\ 0 & \text{si } x \in y_c' \end{cases}$, y $\lambda = [0, 1]$

Con un análisis minucioso de la expresión (3.54) se puede ver que los nueve escenarios creados bajo las hipótesis formuladas en la Tabla 3.12, son organizados de manera diferente para obtener directamente un p_valor por cada clase (de objeto:

círculos, rombos y cuadrados), en vez de los dos que se conseguían con la lógica aplicada originalmente en el problema biclase (véase la sección 3.7.2).

Al aplicando esta nueva lógica sobre el fallido Ejemplo 9 donde se tienen tres clases, se consigue finalmente reducir a tres $p_valores$ en vez de los seis que ya se habían obtenido. Con estos tres $p_valores$ si se podría obtener la predicción y las medidas de confianza y credibilidad correspondientes.

Tabla 3.12.- Forma correcta de formular hipótesis según los escenarios de prueba de hipótesis para los datos multiclase de la Figura 3.20.

Escenario	Formula que	Escenario	Formula que	Escenario	Formula que
(a)	* \in \circ	(b)	* \square	(c)	* \diamond
(d)	* \in \square	(e)	* \circ	(f)	* \diamond
(g)	* \in \diamond	(h)	* \circ	(i)	* \square

El análisis minucioso de la citada expresión (3.54), revela la suma ponderada que se efectúa con las medidas de no conformidad ya obtenidas y que fueron asignadas a partir de las distancias al hiperplano que muestra cada escenario en la Figura 3.21, pero organizando los datos observados bajo el orden $\{(a), (b), (c)\}$, $\{(d), (e), (f)\}$, $\{(g), (h), (i)\}$ (véase horizontalmente la Figura 3.21). El resumen final logrado al organizar los escenarios en este orden previsto, permite hallar los tres $p_valores$ comentados. Véase la siguiente expresión como referencia:

$$\begin{cases} p_valor_1 = \text{resumen de los escenarios } \{(a), (b), (c)\} \\ p_valor_2 = \text{resumen de los escenarios } \{(d), (e), (f)\} \\ p_valor_3 = \text{resumen de los escenarios } \{(g), (h), (i)\} \end{cases} \quad (3.55)$$

Hay que hacer notar, en el análisis correcto realizado sobre el problema multiclase, que los escenarios se plantean en el orden previsto, pero la asignación de medidas de no conformidad es independiente, y se sigue haciendo en esta investigación bajo la expresión (3.50). Además se indica que en la expresión (3.54), las medidas promediadas de no conformidad obtenidas por las hipótesis alternativas

formuladas e ilustradas en los escenarios {(b), (c)}, (e), (f)}, (h), (i)}, contribuyen de manera moderada a las hipótesis principales según el valor del parámetro λ .

Esto último establece el único inconveniente que aparece debido a este enfoque, teniéndose en consecuencia que controlar el parámetro λ ya que su valor puede ir desde cero hasta uno ($\lambda=(0,1)$), *a parte del control de los parámetros propios que deben ajustarse por el kernel usado en el método subyacente*³².

Ejemplo 10. Análisis correcto de los escenarios de prueba de hipótesis en problemas con datos multiclase

Para poner de manifiesto los conceptos que deben ser correctamente considerados en el problema multiclase, se plantean nuevamente los datos del Ejemplo 9 donde se hizo un análisis fallido a propósito, pero ahora se presenta el estudio y los resultados correctos que logran clasificar el asterisco rojo dibujado en la Figura 3.20.

Al razonar bajo la visión de la expresión (3.54) se apreciará en la Tabla 3.13, las nuevas medidas de no conformidad obtenidas para un valor de λ igual a uno ($\lambda=1$). El valor de λ se ha fijado después de explorar su espacio posible para identificar efectivamente la muestra desconocida, y asumiendo por supuesto las hipótesis planteadas en la Tabla 3.12.

Viéndolo detenidamente, con el valor de λ y las mismas medidas de no conformidad presentadas en la Tabla 3.11, se han obtenido las nuevas medidas de no conformidad de la Tabla 3.13. Este comentario es importante porque no se han determinado nuevos hiperplanos, de hecho se han usado las mismas reglas inferidas conseguidas usando un *kernel lineal*³³ bajo el mismo ajuste. Esto se notó, al filtrar los datos internos manejados por el código y ver que se tienen las mismas distancias al hiperplano tanto para el Ejemplo 9 como para el propio Ejemplo 10 que se está presentando.

³² Recuérdese que en esta investigación se está usando la SVM como método subyacente, y las medidas de no conformidad presentadas en los ejemplos se han obtenido a través de la librería LIBSVM, que fue compilada para trabajar con MATLAB.

³³ El parámetro c en la librería LIBSVM se colocó alto después de haber explorado sus posibles valores.

Tabla 3.13.- Medidas de no conformidad obtenidas a partir de la distancia al hiperplano según el enfoque dado por la expresión (3.54), con sus correspondientes $p_valores$.

Objetos			Medida de No Conformidad		
Escenarios			{(a),(b),(c)}	{(d),(e),(f)}	{(g),(h),(i)}
X		Y	Es Circulo	Es Cuadrado	Es Rombo
2.0	2.0	Círculo	-16.9916	-0.9997	-1.0004
3.0	3.0	Círculo	-22.9887	-1.9994	-1.6673
-3.0	3.0	Cuadrado	-0.9997	-9.9960	-1.6667
-4.0	2.0	Cuadrado	-6.9968	-0.9956	-0.9998
-0.5	-3.0	Rombo	-2.9987	-0.9997	-2.3338
0.5	-4.0	Rombo	-0.9997	-3.9986	-3.0005
-0.5	-1.0	Asterisco	-0.9994	-0.9995	-1.0002
p_valor			0.1429	0.1429	0.2857

Lo único que ha cambiado, es la manera de organizar las distancias al hiperplano planteadas por las hipótesis formuladas en la Tabla 3.12, y promediadas por la expresión (3.54) que propicia la obtención de un p_valor a partir de la hipótesis principal y la contribución de dos hipótesis alternativas generada alrededor de cada clase.

Con esto se concluye el estudio de la predicción conformal, observando que ahora sí se ha obtenido un resultado concreto de predicción para el problema multiclase del Ejemplo 9, el cual indica que el asterisco rojo pertenece a la clase rombo, puesto que se obtiene inequívocamente un valor máximo de los tres $p_valores$ presentados también en la Tabla 3.13. Al mismo tiempo, se obtienen las medidas de credibilidad y confianza, cuyos valores son 0.2857 y 0.8571 según las expresiones (3.52) y (3.53) respectivamente.

El valor de credibilidad proporciona, según la teoría vista y referenciada, que la muestra no es conforme con todas las observaciones vistas. Lógicamente es lo que ocurre porque sólo una muestra confirma junto con el propio objeto asterisco en el escenario (g) de predicción, que realmente tal observación desconocida es extrema (está alejada). Esto no ocurre en los otros escenarios (a) y (d) porque la potencia es débil, ya que la muestra que confirma en tales escenarios que el objeto asterisco es extremo, es la propia muestra bajo hipótesis porque el resto son aún más extremos (Véase la Tabla 3.11).

Por su parte, el valor de la confianza explica que, en los escenarios donde se excluyen las etiquetas, las mismas no son las correctas. Para entender este concepto, véanse los casos donde se tiene poca información (pocos datos) porque los ejemplos excluidos son más extremos que el propio objeto.

3.8 LA PREDICCIÓN CONFORMAL INDUCTIVA

La predicción conformal inductiva es una modificación realizada sobre el algoritmo tradicional de la predicción conformal que permite acelerar, por decirlo de alguna manera, el tiempo de computo en problemas tanto de clasificación como de regresión.

Esta modificación propuesta como muy bien se indica en [Papadopoulos, 2008], se hace en base al costo computacional que debe pagar la predicción conformal por su enfoque transductivo [Cherkassky y Mulier, 2007] en el momento de inferir la regla que arroja una predicción puntual a partir de un conjunto de datos disponibles.

En teoría, la ineficiencia de la predicción conformal definida bajo el enfoque transductivo, se presenta al examinar muestra por muestra de un conjunto de datos disponibles para construir el predictor, “*cada vez*”³⁴ que va a estimar la etiqueta de un objeto nuevo desconocido [Vovk, Gammernan y Shafer, 2005]. A pesar de esto, el procedimiento se justifica y es utilizado en los casos donde no se tiene conocimiento a priori del contexto que origina los datos, cosa que imposibilita la estimación de un modelo por inducción [Cherkassky y Mulier, 2007] teniendo que invocarse el principio de transducción.

3.8.1 Predicción conformal inductiva en clasificación

Si se analiza el problema desde el punto de vista propuesto en la introducción realizada en la sección 3.7.2 donde se presentó la predicción conformal como un algoritmo de dos capas (véase la Figura 3.18), se observa que el problema de

³⁴ Para cada objeto nuevo a estimar su etiqueta la predicción conformal repite el examen sobre los datos observados.

ineficiencia infringido por la inferencia del modelo en forma transductiva se presenta en la primera capa.

En esta capa, se usa la SVM como método subyacente, que para el caso de la clasificación de datos manejada en esta investigación permite generar el modelo del hiperplano de separación, y a partir del mismo, se determinan las denominadas distancias al hiperplano. Cabe recordar que en la sección 3.7, este procedimiento se debe repetir cada vez que se deba estimar la clase de un nuevo objeto debido a la asunción de hipótesis que son manejadas por la predicción conformal. También se debe tener presente, que la obtención del modelo del hiperplano se concreta aplicando técnicas estándar de programación cuadrática para hallar los multiplicadores de Lagrange, y en consecuencia determinar los vectores soporte (véase la sección 3.6).

Computacionalmente, todo este proceso recordado en el párrafo anterior, consume tiempos excesivos, agudizándose cuando la cantidad de datos es considerable. Y es aquí especialmente, donde se dan las modificaciones principales sobre la predicción conformal, para lograr que la inferencia del modelo se realice bajo un enfoque inductivo [Cherkassky y Mulier, 2007].

Es importante notar que desde el punto de vista práctico, no sólo se modifica el proceso de inferencia del modelo que se hace de manera transductiva en la predicción conformal, cambiándolo ahora por un enfoque inductivo y con esto construir la predicción conformal inductiva. Sino que, también se debe modificar en la primera capa, el proceso previo referente a la organización del conjunto de datos dispuesto, el cual debe ser dividido en dos partes: el conjunto de entrenamiento propio y el conjunto de calibración [Papadopoulos, 2008].

Con el conjunto de entrenamiento propio, se genera por inducción la regla de aprendizaje o hiperplano de separación $h(x)$. Luego, la regla de aprendizaje es usada para generar las distancias al hiperplano a partir del conjunto de calibración y el nuevo objeto a predecir; y con la ayuda de la expresión (3.50) se obtienen las medidas de no conformidad. Estas modificaciones comentadas a rasgos generales pueden apreciarse en la Figura 3.22 (véase también la Figura 3.18 para que compare y note los cambios introducidos).

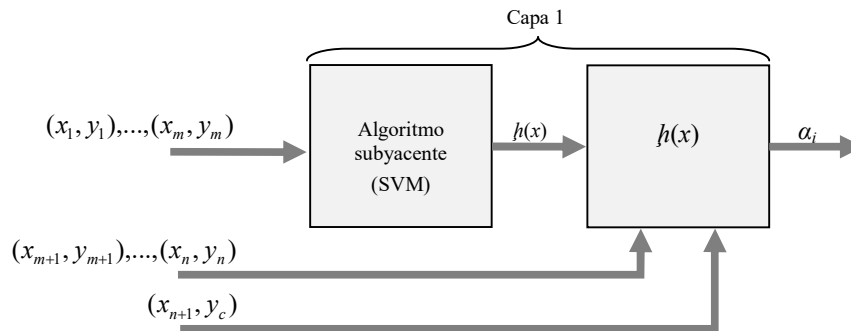


Figura 3.22.- Modificaciones en la capa 1 de la predicción conformal.

Como comentario final se pueden apreciar varios detalles:

- Los datos se dividen bajo el supuesto de que se generan mediante un proceso IID (se asignan las muestras a cada conjunto aleatoriamente).
- Si los datos disponibles a dividir son n : el conjunto de entrenamiento propio contendrá m muestras, por lo tanto el conjunto de calibración debe conformarse con las restantes $n-m$ muestras (véase la Figura 3.22).
- Con el conjunto de entrenamiento propio se genera el modelo de hiperplano $h(x)$ sólo una vez. La reducción del costo computacional radica aquí.
- Con el modelo $h(x)$ se calculan las distancias al hiperplano, tanto para el conjunto de calibración, como para el objeto nuevo a ser predicho.
- El proceso que permite el cálculo de las distancias al hiperplano (lea el punto anterior), se puede replantear de tal manera que se reduzca el costo computacional. Esto es posible, si se crea un algoritmo que sólo permita determinar una única vez tales distancias a partir del conjunto de calibración. Esto no aplica para cada objeto a ser predicho. Nota: este tipo de algoritmo fue diseñado e implementado en esta investigación.
- Calculadas las distancias al hiperplano, se asignan las correspondientes medidas de no conformidad α_i , haciendo uso de la expresión (3.50).
- La segunda capa vista en la predicción conformal no se modifica (Véase la Figura 3.18). La misma se sigue usando para computar las medidas de no conformidad α_i haciendo uso de la expresión (3.49), consiguiendo de esta

forma los $p_valores$, para finalmente obtener la predicción y las medidas de confianza y credibilidad, tal y como se explicó en la sección dedicada a la predicción conformal.

Ejemplo 11. Contraste entre los resultados obtenidos en el Ejemplo 10 mediante la predicción conformal y los nuevos resultados obtenidos con la predicción conformal inductiva

Con el fin de contrastar el funcionamiento de la predicción conformal inductiva frente a la predicción conformal, se tomaron los datos dispuestos en la Figura 3.20. Con el conjunto de datos completo como se observa en la Tabla 3.14, se ajustó el modelo bajo la predicción conformal usando como medida de no conformidad, la aproximación modificada uno versus el resto vista en la expresión (3.54). Estas medidas fueron asignadas con ayuda de la expresión (3.50) a partir de la distancia al margen de separación que determina la SVM al ser usado como algoritmo subyacente con kernel lineal (*parámetros*³⁵: $c=0.010$ y $\lambda=0.3$).

Al final del entrenamiento se logra con ayuda del modelo inferido, clasificar el asterisco como un rombo entre el resto de las figuras geométricas (círculos y cuadrados), gracias a que el p_valor más alto ($p_valor=0.7143$) se detectó en esa clase como se puede comparar entre las fila 1, 2 y 3 de la Tabla 3.15. El mismo p_valor más alto será la medida de credibilidad ($Cred=0.7143$), y el segundo p_valor la confianza ($Conf=0.8571$) como se muestra en la fila 2 y 3 de la Tabla 3.16.

Continuando con el ejemplo para lograr el contraste propuesto, se toma nuevamente el conjunto de datos de la Figura 3.20, procediendo a su división para conformar con tres de las seis observaciones el conjunto de entrenamiento propio (PTS), y con el resto el conjunto de calibración (CS).

Empleando el conjunto de datos marcados con PTS en la Tabla 3.14, se ajustó el modelo bajo la predicción conformal inductiva, consiguiendo el hiperplano $h(x)$

³⁵ En todos los ejemplos vistos en esta tesis, los parámetros kernel y λ fueron optimizados mediante un algoritmo de iteración que explora sus valores posibles hasta lograr buenos resultados referentes a medidas de fiabilidad y/o porcentaje de aciertos (clasificaciones correctas) altas.

mediante la SVM con kernel lineal, manteniendo el valor del parámetro c encontrado con anterioridad ($c=0.010$).

A continuación se tomaron los datos marcados con CS y el objeto a clasificar (CP) mostrados también en la Tabla 3.14, para determinar las medidas de no conformidad empleando la aproximación modificada uno versus el resto vista en la expresión (3.54), conservando el valor del parámetro λ en 0.3 ($\lambda=0.3$) y usando la expresión (3.50) para asignar las medidas de no conformidad a partir de la distancia al hiperplano $h(x)$ inferido.

Tabla 3.14.- Datos de la Figura 3.20 usados para contrastar la predicción conformal con la predicción conformal inductiva. PTS*: datos del conjunto de entrenamiento. CP*: dato a predecir. PTS: datos del conjunto de entrenamiento. CS: datos del conjunto de calibración. CP: dato a predecir.

		Datos usados por el modelo a partir de la		
		Predicción Conformal		Predicción Conformal Inductiva
X	Y			
2.0	2.0	Círculo	PTS*	PTS
3.0	3.0	Círculo	PTS*	CS
-3.0	3.0	Cuadrado	PTS*	PTS
-4.0	2.0	Cuadrado	PTS*	CS
-0.5	-3.0	Rombo	PTS*	CS
0.5	-4.0	Rombo	PTS*	PTS
-0.5	-1.0	Asterisco	CP*	CP

Realizado todo esto, se confirma que la predicción conformal inductiva también identifica al asterisco rojo como un rombo, al arrojar el p_valor más alto para esta clase ($p_valor=0.75$) entre el resto de las figuras geométricas (círculos y cuadrados). Esto se puede corroborar al contrastar las filas 4, 5 y 6 de la Tabla 3.15.

Tabla 3.15.- Resumen de los $p_valores$ obtenidos al clasificar el asterisco propuesto en la Figura 3.20 bajo el modelo de la predicción conformal y la predicción conformal inductiva.

Resumen de los $p_valores$ obtenidos al clasificar el asterisco bajo el modelo de la					
Predicción Conformal			Predicción Conformal Inductiva		
Círculo	Cuadrado	Rombo	Círculo	Cuadrado	Rombo
0.1429	0.1429	0.7143	0.2500	0.2500	0.7500

Nótese además dos detalles interesantes. Primero, se ha elevado la medida de credibilidad a 0.75 ($Cred=0.75$) como puede verse en la fila 5 de la Tabla 3.16. Segundo, y aunque se nota una reducción de la confianza de 0.85 a 0.75 (nueva $Conf=0.75$) como se puede cotejar al comparar las filas 3 y 6 de la Tabla 3.16, aún ésta sigue siendo alta ($Conf \approx 1$).

Tabla 3.16.- Resumen de las medidas de fiabilidad que identifican al asterisco propuesto en la Figura 3.20 como un rombo bajo el modelo de la predicción conformal y la predicción conformal inductiva.

Resumen de las medidas de fiabilidad que identifican al asterisco como un rombo bajo el modelo de la					
Predicción Conformal			Predicción Conformal Inductiva		
Predicción	Credibilidad	Confianza	Predicción	Credibilidad	Confianza
0.7143	0.7143	0.8571	0.7500	0.7500	0.7500

Como se ha visto, la predicción conformal inductiva ha logrado bajo condiciones similares, garantizar tanto la predicción del objeto asterisco como las medidas de fiabilidad; manteniéndose cercana a su predecesora, la predicción conformal, al reflejar valores cercanos y válidos.

Adicionalmente a esto, y como justificación principal para aplicar la predicción conformal inductiva, se compararon los tiempos de procesamiento invertidos por cada método, desde el entrenamiento hasta la clasificación obtenida con sus respectivas medidas de fiabilidad.

Esta comparación se distingue en la Tabla 3.17, notando que la predicción conformal invierte para la clasificación en cuestión, un tiempo de 0.026 segundos, mientras que la predicción conformal inductiva propicia un costo computacional para hacer el mismo trabajo de 0.011 segundos.

Tabla 3.17.- Costo computacional para lograr clasificar el asterisco propuesto en la Figura 3.20 como un rombo bajo el modelo de la predicción conformal y la predicción conformal inductiva.

Costo computacional bajo el modelo de la	
Predicción Conformal	Predicción Conformal Inductiva
0.026 segundos	0.011 segundos

Tal vez, alguien con poca visión pudiera preguntarse si vale la pena todo el esfuerzo hecho para modificar la predicción conformal y así conquistar la predicción conformal inductiva, tan sólo para reducir a un poco más de la mitad, los tiempos insignificantes reflejados en este Ejemplo 11.

Sin embargo y como se verá en el Capítulo IV, donde se explicará el desarrollo de un multclasificador para procesar imágenes provenientes del dispositivo de fusión termonuclear TJ-II ubicado en el CIEMAT de España, la predicción conformal inductiva puede reducir drásticamente el tiempo de cómputo invertido respecto a la predicción conformal, en aplicaciones donde clasifican datos masivos multidimensionales, manteniendo la calidad de las predicciones al reflejar buenas medidas de fiabilidad.

3.9 LA PREDICCIÓN VENN

Los predictores Venn, son predictores multiprobabilísticos bien calibrados y pertenecen a la familia de los predictores conformales. Estos se combinan con clasificadores convencionales para obtener predicciones, y a diferencia de la predicción conformal que proporciona medidas de fiabilidad las cuales no son probabilidades directas; la predicción Venn dispone de estimaciones que evalúan el error cometido al construir distribuciones de probabilidades para cada etiqueta predicha de un objeto desconocido.

Para la etiqueta verdadera del objeto nuevo desconocido, las distribuciones de probabilidades definen un intervalo acotado de probabilidades donde está contenida la misma [Vovk, Gammerman y Shafer, 2005] [Nouretdinov et al, 2012]. Esto se logra, similarmente a la predicción conformal, ya que la predicción Venn hace predicciones sobre el objeto nuevo, adaptándolo al conjunto de datos disponibles, sólo requiriendo que sean independientes e idénticamente distribuidos [Vovk, Gammerman y Shafer, 2005].

Sin embargo, la idea general de los predictores Venn según [Nouretdinov et al, 2012], se centra en que el algoritmo que lo implemente debe dividir el conjunto de

datos dispuesto en *categorías*³⁶. Cada categoría asignada dependerá de la implicación de un objeto con el resto de las observaciones contenidas en el conjunto completo.

Para el caso del objeto nuevo a clasificar se asume una hipótesis por categoría. Luego se clasifica el nuevo objeto en una de las categorías, usando la frecuencia asociada por categoría como distribuciones de probabilidades de la etiqueta para el nuevo objeto. Al final se tendrán tantas distribuciones como categorías.

Dadas las categorías (o divisiones), cada observación del conjunto completo debe ser clasificada incluyendo el objeto nuevo bajo hipótesis en una división mediante algún algoritmo de aprendizaje automático tradicional denominado como taxonomía. La taxonomía seleccionada funciona al estilo de los algoritmos subyacentes usados en la predicción conformal [Vovk, Gammerman y Shafer, 2005].

Teniendo en cuenta esto, cualquier algoritmo de aprendizaje puede ser usado como taxonomía, entre los cuales tenemos: las redes neuronales artificiales, las máquinas de vectores soporte, los algoritmos de agrupamiento, entre otros.

En esta investigación sólo se aplicaron los predictores Venn en combinación con los algoritmos de agrupamiento de los k vecinos más cercanos [Johansson et al, 2018] y el centroide más cerca, para realizar clasificaciones de las imágenes obtenidas del dispositivo de fusión termonuclear ubicado en el CIEMAT de España. El estudio comprende los siguientes objetivos:

- Comparar los resultados obtenidos por la predicción Venn al utilizar las taxonomías dada por los k vecinos más cercanos y los centroide más cerca.
- Contrastar los resultados obtenidos con la predicción Venn y con resto de los resultados que serán presentados en el Capítulo IV referente a la clasificación de imágenes obtenidas del dispositivo de fusión termonuclear ubicado en el CIEMAT de España.

3.9.1 Definición general de la predicción Venn

Siguiendo a [Nouretdinov et al, 2012] y en coherencia con todo lo visto hasta el momento, se definirá de manera general la predicción Venn.

³⁶ Las categorías pueden ser entendidas como las posibles clases o etiquetas.

Dado un conjunto de datos disponible $(x_1, y_1), \dots, (x_n, y_n)$, donde x_n representa en si los objetos del conjunto perteneciente a un espacio medible X con $X \in \mathcal{R}^d$, e y_n su correspondiente etiqueta perteneciente a un espacio medible C con $C = \{c_1, \dots, c_j\}$.

Dado además, un objeto nuevo x_{n+1} medido en X con el propósito de predecir (clasificar) a y_{n+1} , que está contenida en C . Y ahora, asumiendo para este objeto x_{n+1} diferentes hipótesis y_c con $y_c \in C$, se formarán las j -ésimas secuencias de observaciones (o conjuntos observados) según:

$$(x_1, y_1), \dots, (x_n, y_n), (x_{n+1}, y_c) \quad (3.56)$$

Como se aprecia, para cada y_c se formará una secuencia. Formalizando ahora la expresión (3.56) como z_1, z_2, \dots, z_n , haciendo coincidir el número de observaciones $n+1$, simplemente diciendo que hay n observaciones. También se puede referenciar a cualquier observación en particular como $z_i = (x_i, y_i)$ con $i = 1, \dots, n$.

Con esto y disponiendo ahora de una taxonomía A_n que clasifica a cada objeto x_i , en una de las τ_j categorías (o clases) según la relación con el resto de los objetos x , es decir:

$$\tau_j = A_n(\{z_1, \dots, z_n\}, z_i) \quad (3.57)$$

Ahora bien, para cada una de las categorías τ_j existirá una distribución de probabilidades por clase de etiqueta, según:

$$p^{y_c}(y_k) = \frac{\#\{(x_n, y_n) \in \tau : y_c = y_k\}}{|\tau|} \quad (3.58)$$

Donde p^{y_c} será un vector que contiene la distribución de probabilidades para cada etiqueta y_k , contenida para una categoría τ_j asumiendo y_c . Es decir:

$$p^{y_c}(y_k) = [p^{y_c}(y_1), \dots, p^{y_c}(y_k)] \quad (3.59)$$

Evidentemente, se tomará como vector de distribución de probabilidades válido, a aquella categoría donde este contenida la hipótesis y_c asumida. Esto formará una matriz de distribuciones de probabilidades con la forma:

$$p^{y_c}(y_k) = \begin{bmatrix} p^{y_1}(y_1) & \cdots & p^{y_1}(y_k) \\ \vdots & \ddots & \vdots \\ p^{y_c}(y_1) & \cdots & p^{y_c}(y_k) \end{bmatrix} \quad (3.60)$$

3.9.2 Funcionamiento de la predicción Venn

Como se ha visto hasta el momento, la definición de la predicción Venn se ha realizado alrededor de una muestra nueva a clasificar, pero tal vez esto sea algo tedioso para ser visto con facilidad, y es por esto que se explicará en aras de comprender todo pero desde una perspectiva más práctica.

Para ello, al igual que la predicción conformal, la predicción Venn puede verse como un algoritmo implementado sobre dos capas, tal y como se muestra en la Figura 3.23. Con esto en mente, se puede percibir que la taxonomía se implementa en la primera capa, formando las diferentes categorías τ_i al asumir diferentes hipótesis y_c para el objeto a clasificar y donde tales hipótesis asumidas estarán asociadas a la variedad de clases presentes en los datos dispuestos.

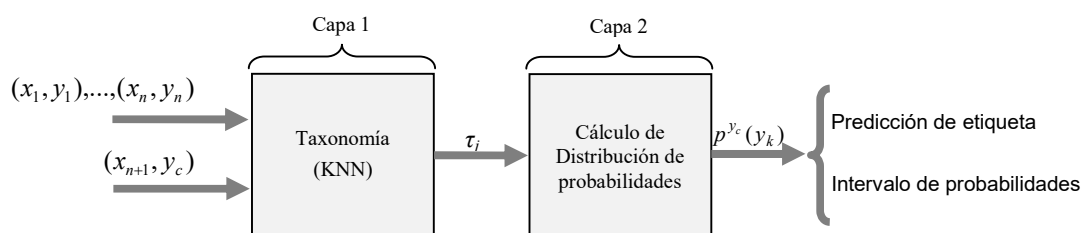


Figura 3.23.- Capas de la predicción Venn.

A continuación, estas categorías pasan a la segunda capa donde se determinan las distribuciones de probabilidades $p^{y_c}(y_k)$. Es importante notar que en la salida de la segunda capa se tiene una matriz bajo la forma de la expresión (3.60). De aquí se

obtienen finalmente, tanto la predicción del objeto nuevo, como los valores que forma el intervalo de probabilidad dado por $\{p^-, p^+\}$.

3.9.3 La predicción y el intervalo de probabilidades

A partir de la matriz $p^{y_c}(y_k)$ vista en la expresión (3.60), se obtiene la predicción \hat{y} de la etiqueta y_n desconocida para un objeto nuevo $z_n=(x_n, y_n)$. Esta predicción estará dada por el valor máximo descubierto al promediar la suma de los elementos en cada columna, según la expresión:

$$\hat{y} = \max[\overline{\sum p^{y_c}(y_1)} \cdots \overline{\sum p^{y_c}(y_k)}] \quad (3.61)$$

Además, el intervalo de probabilidades será asignado por el mínimo y máximo valor presentes al inspeccionar los elementos de la referida columna encontrada por la expresión (3.61). Este intervalo puede escribirse como:

$$\{p^-, p^+\} = \{\min[p^{y_1}(y_k) \cdots p^{y_c}(y_k)], \max[p^{y_1}(y_k) \cdots p^{y_c}(y_k)]\} \quad (3.62)$$

3.9.4 Taxonomías

Al igual que los métodos subyacentes en la predicción conformal, las taxonomías A_n juegan un papel vital en la predicción Venn, ya que son algoritmos tradicionales de aprendizaje automático capaces de agrupar (clasificar) por su clase (categorías), a los diferentes objetos presentes en un conjunto de observaciones dispuestas.

Dentro de estos algoritmos tradicionales de aprendizaje automático, que pueden ser empleados como taxonomías para hacer agrupamiento se tienen, las redes neuronales artificiales, las máquinas de vectores soporte y los algoritmos de agrupamiento, entre otros. Aunque en esta investigación sólo se utilizarán los algoritmos de agrupamiento asociados con los métodos de los k vecinos más cercanos

(KNN) y el centroide más cercano (CNN) agrupar los datos y conseguir clasificaciones.

3.9.4.1 Taxonomía: k vecinos más cercanos

Los k vecinos más cercanos es un *método de aprendizaje no paramétrico*³⁷ [Rusell y Norving, 2003] perteneciente al paradigma de aprendizaje perezoso [Johansson et al, 2018], caracterizándose por la ausencia de una fase de extracción de conocimiento de los datos. Según [Rusell y Norving, 2003], “*la idea clave de los modelos de vecinos más cercanos es que es probable que las propiedades de un punto de entrada particular x sean similares a las de los puntos cercanos a x* ”. Esto, en términos de un modelo de clasificación, los k vecinos más cercanos deben asignarle a un objeto desconocido, la etiqueta de aquellos objetos que estén más próximos.

Claro está que para un objeto en un espacio multidimensional, este pudiera estar rodeado de muchos otros objetos con diferentes etiquetas. En estos casos, se debe apelar por alguna regla que permita determinar la mayor contribución de aquellas clases de etiquetas que mejor identifiquen al objeto desconocido [Rusell y Norving, 2003] [Johansson et al, 2018].

Las reglas en general, pueden entenderse como las diferentes métricas que pueden ser tomadas entre un objeto desconocido y sus k vecinos más cercanos. Entre estas métricas se mencionan las más comúnmente utilizadas: la distancia euclidia, la mahalanobis y la de hamming. En [Rusell y Norving, 2003] se dan ideas de porqué usar una medida u otra, destacándose el uso de la distancia euclidia en aplicaciones de agrupamiento donde las dimensiones o características de un objeto son medidas de manera homogénea. En [Johansson et al, 2018] se discute la utilización de los k vecinos más cercanos como taxonomía de los predictores Venn bajo este tipo de medida euclidia.

³⁷ En el aprendizaje estadístico se denomina a un método como no paramétrico, cuando a través de él se construyen modelos que no están centrados en el ajuste de los parámetros de una familia restringida de modelos de probabilidad [Rusell y Norving, 2003].

Con el fin de poder poner de manifiesto los conceptos de la teoría de la predicción Venn y el uso de las taxonomías, se introducirán algunos ejemplos prácticos.

Ejemplo 12. La predicción Venn con taxonomía de los k vecinos más cercanos

Para ver el funcionamiento de la predicción Venn desde el punto de vista práctico, se tomó el mismo conjunto de datos dispuesto en la Figura 3.20, dejando la muestra (asterisco rojo) colocada en la posición $(-0.5,-1)$, y que fue objeto de estudio en los últimos tres ejemplos vistos.

Para iniciar el trabajo de la predicción Venn, se tiene que establecer el número de categorías (clases), siendo evidente al observar la Figura 3.20 que se tienen tres clases, las cuales son: los dos círculos rojos, los dos cuadrados negros y los dos rombos azules.

Conociendo el número de clases presentes, se deben asumir tres hipótesis. Cada una supondrá que el asterisco rojo pertenece a una de estas clases. Con esto en mente y usando la expresión (3.56) se construyen los tres conjuntos observados (secuencias) que se muestran a continuación:

$$\begin{aligned} \text{Conjunto}_1 &= \{\text{círculo}, \text{círculo}, \text{cuadrado}, \text{cuadrado}, \text{rombo}, \text{rombo}, \text{asterisco}^{\circ}\} \\ \text{Conjunto}_2 &= \{\text{círculo}, \text{círculo}, \text{cuadrado}, \text{cuadrado}, \text{rombo}, \text{rombo}, \text{asterisco}^{\square}\} \\ \text{Conjunto}_3 &= \{\text{círculo}, \text{círculo}, \text{cuadrado}, \text{cuadrado}, \text{rombo}, \text{rombo}, \text{asterisco}^{\diamond}\} \end{aligned} \quad (3.63)$$

Fíjese, que cada conjunto observado se ha construido asumiendo que el asterisco es de una clase a la vez. A continuación, se debe tomar cada una de estas secuencias vistas en (3.63) y se pasa a la taxonomía A_n seleccionada, que en el caso del ejemplo tratado se está usando los k vecinos más cercanos.

Entonces, con cada secuencia se obtienen tres categorías τ_i según la expresión (3.57), donde se mide seguidamente y con la ayuda de la expresión (3.58), la distribución de probabilidades por clase presente (categorías τ_j). Los resultados

conseguidos por ambos procesos obtenidos mediante las expresiones (3.57) y (3.58) se resumen a continuación.

- Categorías y distribuciones de probabilidades obtenidas, asumiendo el asterisco rojo como círculo:

$$\begin{aligned}\tau_{\circ} &= \{\text{círculo}, \text{círculo}\} \\ \tau_{\square} &= \{\text{cuadrado}, \text{cuadrado}\} \\ \tau_{\diamond} &= \{\text{rombo}, \text{rombo}, \text{asterisco}^{\circ}\} \\ P_{\circ} &= 1/3; P_{\square} = 0; P_{\diamond} = 2/3\end{aligned}\tag{3.64}$$

- Categorías y distribuciones de probabilidades obtenidas, asumiendo el asterisco rojo como cuadrado:

$$\begin{aligned}\tau_{\circ} &= \{\text{círculo}, \text{círculo}\} \\ \tau_{\square} &= \{\text{cuadrado}, \text{cuadrado}\} \\ \tau_{\diamond} &= \{\text{rombo}, \text{rombo}, \text{asterisco}^{\square}\} \\ P_{\circ} &= 0; P_{\square} = 1/3; P_{\diamond} = 2/3\end{aligned}\tag{3.65}$$

- Categorías y distribuciones de probabilidades obtenidas asumiendo el asterisco rojo como rombo:

$$\begin{aligned}\tau_{\circ} &= \{\text{círculo}, \text{círculo}\} \\ \tau_{\square} &= \{\text{cuadrado}, \text{cuadrado}\} \\ \tau_{\diamond} &= \{\text{rombo}, \text{rombo}, \text{asterisco}^{\diamond}\} \\ P_{\circ} &= 0; P_{\square} = 0; P_{\diamond} = 1\end{aligned}\tag{3.66}$$

Antes de continuar, es importante notar dos conclusiones en los procesos ejecutados hasta el momento:

- Primera. La construcción de los conjuntos de datos observados como consecuencia de aplicar la expresión (3.56) y resumidos en la expresión (3.63), no necesariamente deben generarse todos al mismo tiempo antes de pasar a las siguientes etapas. El detalle se indica por razones de optimización de memoria, y alternativamente el algoritmo que ejecuta la predicción Venn, puede construir una secuencia a la vez al asumir la hipótesis de turno sobre la muestra desconocida (en el ejemplo tratado es el asterisco rojo) y ser pasada a los procesos que siguen dados por las expresiones (3.57) y (3.58).
- Segunda. Está relacionada con el cálculo de la distribución de probabilidades por cada categoría obtenida, que para el ejemplo tratado se cuantifican nueve grupos. Si se analiza esta conclusión viendo la expresión (3.58) que mide la

posible distribución de probabilidades para cada etiqueta (clase) existente en una categoría construida al aplicar la taxonomía escogida, se debe tener presente que tal cálculo de probabilidades no debe realizarse en todos los grupos construidos por la expresión (3.57). Es decir, sólo se deben calcular las probabilidades en aquellos grupos (o categorías) que contenga la etiqueta bajo hipótesis. Si se observan nuevamente las expresiones (3.64), (3.65) y (3.66), se podrá notar que el asterisco rojo bajo hipótesis, exclusivamente está en tres de las nueve categorías, aplicándose la expresión (3.58) únicamente en estas categorías.

Continuando con el proceso de la predicción Venn, toca ahora recoger en un vector, las distribuciones de probabilidades halladas en las expresiones (3.64), (3.65) y (3.66) según la expresión (3.59), para luego y con la ayuda de la expresión (3.60) tomar la matriz de distribución de probabilidades del ejemplo que se comenta y que se muestra a continuación:

$$p^{y_c}(y_k) = \begin{bmatrix} 1/3 & 0 & 2/3 \\ 0 & 1/3 & 2/3 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.67)$$

Es importante observar las siguientes tres conclusiones adicionales:

- Tercera. La matriz (3.67) se puede obtener directamente, al implementar un algoritmo para la predicción Venn que vaya guardando convenientemente en un vector, los resultados de aplicar la expresión (3.59).
- Cuarta. Cada fila de la matriz (3.67) representa en sí, la distribución de probabilidades de cada clase, al asumir una hipótesis para la muestra asterisco. Por ejemplo, para la primera fila se tiene la distribución de probabilidades de cada clase de figura geométrica (círculos, cuadrados y rombo) asumiendo que la muestra asterisco es un círculo. Entonces, para la segunda y tercera fila se asume que la muestra asterisco es un cuadrado y rombo respectivamente.

- Quinta. En cada columna se tienen como elementos, las distribuciones de probabilidades para una clase determinada bajo diferentes hipótesis. Al sumar y promediar cada elemento de cada columna, se consigue la clase pronosticada como lo sugiere la expresión (3.61). Además, aquí se alojan los valores que conformaran según la expresión (3.62), el intervalo de probabilidades de la clase ganadora.

Según la quinta conclusión y como paso previo a la aplicación de las expresiones (3.61) y (3.62), se han sumado y promediados los elementos de cada columna de la matriz vista en (3.67). Los resultados se han colocado al final de cada columna en la expresión (3.68), para que se aprecie que la columna ganadora está relacionada con la categoría rombo la cual obtuvo un promedio $7/9$ (0.7778) frente a las columnas círculo y cuadrado con sólo $1/9$ (0.1111) cada una.

$$p^{y_c}(y_k) = \begin{array}{c} \left[\begin{array}{ccc} 1/3 & 0 & 2/3 \\ 0 & 1/3 & 2/3 \\ 0 & 0 & 1 \end{array} \right] \\ \hline [1/9 \quad 1/9 \quad 7/9] \end{array} \quad (3.68)$$

Con esto se concluye el estudio realizado a través del Ejemplo 12. Y como era de esperarse, se obtiene definitivamente la predicción de que el asterisco rojo es un rombo, con un intervalo de probabilidades de $\{p^-, p^+\} = \{2/3, 1\}$.

3.9.4.2 Taxonomía: el centroide más cercano

La taxonomía del centroide más cercano, es una variante de la taxonomía de los k vecinos más cercanos pero sigue usando como núcleo a este algoritmo de aprendizaje tradicional.

Esta modificación lo que hace es promediar la suma de las observaciones pertenecientes a cada categoría τ_i , incluyendo a aquella categoría que acepta al objeto a predecir supuesto bajo hipótesis, construyendo de esta manera un centroide por clase (categoría).

Conseguidos los centroides por categoría, se debe proceder al mismo estilo que lo hace la taxonomía de los k vecinos más cercanos, pero a diferencia de buscar la relación de un objeto de la secuencia observada con el resto de los objetos de tal secuencia; ahora busca la relación de cada objeto de la secuencia observada pero usando los centroides.

Esta variante se utiliza (según el argumento encontrado bajo los experimentos realizados en esta investigación) para reducir el costo computacional en el proceso que realiza la predicción Venn, al usar los k vecinos más cercanos como taxonomía pero incluyendo en el núcleo el concepto del centroide.

En los experimentos realizados, se observó una dependencia del número de muestras de entrenamiento para justificar el uso de esta variante. Viendo que, en la medida que sean pocas observaciones, los tiempos de cómputo de ambas taxonomías se acercan; mientras que al aumentar considerablemente los datos, el centroide más cercano incrementa su ventaja al lograr reducir los tiempos del proceso de predicción.

Para ver el funcionamiento de esta taxonomía, se propone el siguiente ejemplo.

Ejemplo 13. La predicción Venn con taxonomía del centroide más cercano

La taxonomía del centroide más cercano, toma el promedio de la suma de los objetos en cada categoría presente dentro del conjunto observado, resumiéndolas en un único punto (centroide), tal y como se observa gráficamente en la Figura 3.24, donde se muestran tres “x” por imagen (Figura 3.24 (a), (b) y (c)). Cada “x” representa el centroide de los objetos círculos, cuadrados y rombos incluida la observación asterisco asumiéndola como una figura geométrica a la vez.

Según esto, fíjese por ejemplo en la Figura 3.24 (a), donde se ha tomado el centroide con los dos círculos rojos y el asterisco bajo la hipótesis de que también lo es un círculo rojo. De la misma forma, se toma el baricentro en las Figuras 3.24 (b) y (c), aunque para ello se asume la hipótesis de que el asterisco es un cuadrado negro y rombo azul respectivamente.

Nótese además, como el centroide calculado con los datos (figuras geométricas) donde se asume al asterisco afín a los mismos, la “x” (que ilustra al centroide) se inclina hacia el centro en cada uno de los tres dibujos, en dirección al asterisco. Esto ocurre porque el objeto desconocido aporta información, reflejándose matemáticamente en el promedio calculado.

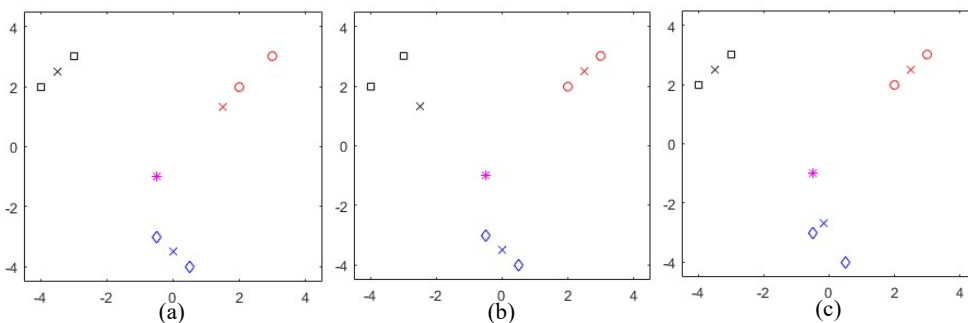


Figura 3.24.- Centroides más cercanos. Se asume el asterisco como: (a) un círculo, (b) un cuadrado, (c) un rombo.

Para una mejor comprensión, se ha colocado por columna en la Tabla 3.18, los resultados de los promedios de las coordenadas de cada centroide para cada dibujo de la Figura 3.24, resaltando en negrita los baricentros donde el asterisco está bajo hipótesis.

Tabla 3.18.- Centroides vistos en la Figura 3.24, asumiendo la coordenada (-0.5,-1.0) del objeto asterisco como: columna 4: círculo, columna 5: cuadrado, columnas 6: rombo.

Centroide asumiendo la coordenada (-0.5,-1.0) del asterisco como					
X		Y	Figura 3.24 (a) Círculo	Figura 3.24 (b) Cuadrado	Figura 3.24 (c) Rombo
2.0	2.0	Círculo	(1.5000,1.3333)	(2.5000,2.5000)	(2.5000,2.5000)
3.0	3.0	Círculo			
-3.0	3.0	Cuadrado	(-3.5000,2.5000)	(-2.5000,1.3333)	(-3.5000,2.5000)
-4.0	2.0	Cuadrado			
-0.5	-3.0	Rombo	(0,-3.5000)	(0,-3.5000)	(-0.1667,-2.6667)
0.5	-4.0	Rombo			

El resultado de clasificar el asterisco rojo mediante la predicción Venn bajo esta otra taxonomía, es similar a los obtenidos con la taxonomía de los k vecinos más cercanos como se revela en el cuadro comparativo de la Tabla 3.19.

Tabla 3.19.- Cuadro comparativo entre las taxonomías KNN y CNN para la predicción hecha sobre el asterisco rojo de la Figura 3.20.

Algunos resultados interesantes a comparar	Taxonomía KNN	Taxonomía CNN
Expresión (3.48) $\begin{bmatrix} p^{y_1}(y_1) & \cdots & p^{y_1}(y_k) \\ \vdots & \ddots & \vdots \\ p^{y_c}(y_1) & \cdots & p^{y_c}(y_k) \end{bmatrix}$	$\begin{bmatrix} 0.3333 & 0 & 0.6667 \\ 0 & 0.3333 & 0.6667 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.3333 & 0 & 0.6667 \\ 0 & 0.3333 & 0.6667 \\ 0 & 0 & 1 \end{bmatrix}$
Expresión (349): sólo promedios $[\overline{\sum p^{y_c}(y_1)} \cdots \overline{\sum p^{y_c}(y_k)}]$	0.1111 0.1111 0.7778	0.1111 0.1111 0.7778
Expresión (349) $\hat{Y} = \max[\overline{\sum p^{y_c}(y_1)} \cdots \overline{\sum p^{y_c}(y_k)}]$	0.7778	0.7778
Clase predicha	3	3
Intervalo de probabilidades	{0.6667,1}	{0.6667,1}
Tiempo de procesamiento (en segundos)	0.0286	0.0270

Como se aprecia, básicamente se tienen los mismos resultados para un mismo objeto supuesto desconocido, coincidiendo ambas taxonomías al ayudar a la predicción Venn a decidir la clase resultante. Aunque si se fija la mirada en la última fila de la Tabla 3.19, se notará una insignificante diferencia entre los tiempos de procesamiento consumido por ambos métodos.

Si se recuerda que, la taxonomía del centroide más cercano es usada con el argumento de qué reduce el costo computacional del proceso efectuado por la predicción Venn, en detrimento de los k vecinos más cercanos al clasificar una muestra, entonces y a fin de demostrar tal argumento, se propuso realizar un experimento donde se confrontan los dos enfoques.

Para ello, a cada taxonomía se le pasó un conjunto inicial de entrenamiento de 6 observaciones para clasificar 450 muestras. Por cada predicción, se va acumulando el tiempo e incrementando en uno el conjunto de entrenamiento hasta alcanzar las 456 muestras, obteniéndose la respuesta gráfica que se observa en la Figura 3.25. Nótese que, con el paso de las primeras muestras al conjunto de entrenamiento, ambas taxonomías mantienen un tiempo de clasificación para la observación vista muy

cercano (tiempo de procesamiento despreciable hasta 75 muestras en el conjunto de entrenamiento), pero en la medida que se va incrementando el conjunto observado, el costo computacional comienza a despegarse.

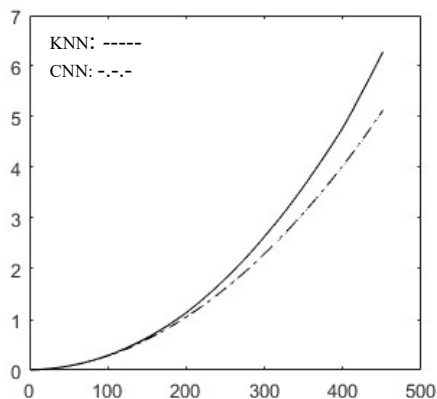


Figura 3.25.- Comparación de tiempos (en minutos) de procesamiento de las taxonomías KNN y CNN a partir de un conjunto de entrenamiento de 6 muestras hasta alcanzar 456.

En la Tabla 3.20 se pueden ver algunas comparaciones puntuales para entender la ventaja de acelerar la predicción Venn usando la taxonomía del centroide más cercano. En la columna 1, se observa el número de muestras que conforma el conjunto de entrenamiento por incremento de sus observaciones. En las columnas 2 y 3, se muestran los tiempos que ha tardado la predicción Venn en analizar una muestra e incorporarla al conjunto de entrenamiento con las diferentes taxonomías, y en la columna 4, la diferencia de estos tiempos.

Tabla 3.20.- Algunos valores puntuales de las taxonomías KNN y CNN al incrementar el conjunto de entrenamiento en la predicción de una muestra.

Número de muestras para entrenamiento	Taxonomía KNN (tiempos en minutos)	Taxonomía CNN (tiempos en minutos)	Diferencia entre KNN y CNN (en minutos)
1-74	Despreciable	Despreciable	Despreciable
75	0.1737	0.1699	0.0038
150	0.6471	0.6154	0.0317
225	1.4627	1.3264	0.1363
300	2.6563	2.3122	0.3441
375	4.2106	3.5729	0.6377
450	6.2798	5.1262	1.1536

De esta manera y como resumen, se observa que para incorporar 450 muestras, la predicción Venn con taxonomía KNN consume 6.2798 minutos. Mientras que su modificación por vía de CNN hace lo mismo pero en 5.1262 minutos, reduciendo el tiempo de cómputo para la predicción vista en 1.1536 minutos.

Claro está, que los tiempos discutidos dependerán del equipo computacional donde se ejecute el proceso de la predicción Venn, pero en todo caso, siempre la taxonomía CNN tomará ventajas computacionalmente hablando.

Para finalizar con este estudio sobre la predicción Venn, se quiere indicar que los resultados recientemente mostrados sólo incluye la ventaja en el sentido exclusivo del costo computacional a la hora de procesar un conjunto de muestras al asumir una u otra taxonomía.

Sin embargo, el estudio no incluye el rendimiento en el sentido de la precisión en cuanto a la cantidad de muestras acertadas versus no acertadas, y tampoco incluye medidas de fiabilidad que pueden conseguirse al aplicar ambas taxonomías, ya que el experimento se controló para sólo contrastar los tiempos de procesamiento.

No obstante, en el Capítulo IV se verá experimentalmente, que la ventaja del costo computacional obtenida bajo la taxonomía CNN, no necesariamente se mantiene en otras líneas de estudio asociadas a la efectividad que puede conseguirse con la predicción Venn bajo la taxonomía KNN, en cuanto a precisión y medidas de fiabilidad.

3.10 LA TRANSFORMADA WAVELET Y EL PRE-PROCESAMIENTO DE IMÁGENES

La transformada wavelets es una herramienta de análisis de multiresolución utilizada en numerosas aplicaciones en ciencia e ingeniería, como por ejemplo: la reducción de ruido, la compresión para la transmisión y almacenamiento de información y la detección de anomalías; tanto en el procesamiento de señales como de imágenes [Misiti et al, 1997], [Melo et al, 2005]. Su aplicación en esta

investigación quedó confinada al tratamiento de imágenes, con el fin de: reducir la cantidad de información y la selección de características.

Con relación a la reducción de información y viéndolo en sentido práctico, esto sugiere la compresión en niveles de información por píxeles en una imagen, haciendo que los procesos implementados en este trabajo sean más accesibles desde el punto de vista computacional [Melo et al, 2005].

Ampliando un poco más, para la reducción de información en imágenes, la transformada wavelet realiza su descomposición en diferentes niveles [Misiti et al, 1997]. Cada nivel retiene y mantiene información considerable y de calidad que representan características de la imagen original [Melo et al, 2005]. Este proceso se logra porque la transformada wavelet rastrea, por decirlo de alguna manera, la correlación existente de características comunes (píxeles) en una vecindad de la imagen, calculando de esta forma la redundancia e irrelevancia de píxeles [Melo et al, 2005] y depositándola como una representación en otro nivel pero a escala diferente. Esto se puede ver como una de-correlación.

Con relación a la selección de características, la transformada wavelet deja una vía para su elección por defecto. Para entender esto, se debe ver el esquema de compresión [Melo et al, 2005] que utiliza esta herramienta para lograr los diferentes niveles de reducción de la información contenida en la imagen.

Como se dijo líneas atrás, la descomposición de una imagen mediante la transformada wavelet en diferentes niveles, retiene información representativa, la cual es a su vez es incorporada en diferentes niveles de detalles, siendo estos: la aproximación o coeficiente de aproximación (A), los detalles horizontales (H), los detalles verticales (V) y los detalles diagonales (D) [Misiti et al, 1997], [Melo et al, 2005]. Cada nivel de detalle mencionado de un nivel de descomposición determinado, representan rasgos de la imagen original, pero con una cantidad de información menor (compresión) retenida [Misiti et al, 1997], [Melo et al, 2005].

Esto último, es la característica más interesante y utilizable en aplicaciones de clasificación e identificación, ya que sólo basta con seleccionar aquel nivel de detalle

conveniente que permita concretar una aplicación con éxito. Un estudio en este sentido y relacionado con la investigación realizada para escribir ésta tesis, se encuentra en [Farias et al, 2004].

En el Capítulo IV se explicará con algo más de detalle, la forma como se utilizó la transformada wavelet para el pre-procesamiento de las imágenes obtenidas en el dispositivo de fusión termonuclear TJ-II del CIEMAT de España.

3.11 MATLAB Y LA LIBRERÍA LIBSVM

MATLAB es el nombre abreviado de “MATrix LABoratory” [García et al, 2005]. Este es un programa para realizar cálculos con números escalares, pero su poder se expresa cuando es usado en el procesamiento de datos que involucran vectores y matrices. Quizás la capacidad más potente de MATLAB radica en la realización de gráficos en dos y tres dimensiones, dando de esta manera una gran versatilidad a la hora de desarrollar y probar cualquier aplicación que involucra el análisis de datos.

Otra capacidad que tiene este paquete de desarrollo técnico-científico, es su lenguaje de programación que no es más potente que otros espacios de programación como el Lenguaje C, pero al unirlo con las funciones propias de MATLAB desarrolladas para aplicaciones particulares y sus propias librerías, hacen de éste una verdadera arma de aplicación, simulación y pruebas de sistemas. Este software también cuenta con toolboxes (herramientas de trabajo) que permiten desarrollar ambientes gráficos amigables y así hacer la tarea más fácil para el usuario final. Su uso en esta investigación fue total, ya que mediante su entorno de programación, se desarrollaron y ejecutaron todos los experimentos implementados.

Por su parte, LIBSVM [Chang y Lin, 2018] es una librería en línea que implementa un grupo de funciones que emulan el algoritmo de la máquina de vectores soporte (SVM) y compilable en diferentes lenguajes de programación. Para los propósitos de esta tesis y por la utilización de MATLAB, esta librería se compiló

en su Versión³⁸ 3.22 sobre MATLAB en su Versión R2015b, permitiendo el desarrollo y prueba de las principales experiencias en esta investigación.

Dentro de las funciones de la librería LIBSVM compiladas en MATLAB, sólo fueron usadas: la función *svmtrain*, que permite el entrenamiento de un modelo SVM bajo diferentes parámetros a partir de un conjunto de datos. La función *svmpredict* que acepta como parámetro el modelo SVM entrenado previamente, para realizar predicciones, bien sea con los mismos datos de entrenamiento o sobre nuevos datos.

Con relación a los parámetros de ajustes de las funciones de LIBSVM, los mismos pueden ser consultados, además de ver ejemplos de aplicación en [Chang y Lin, 2018].

De manera general y particularmente, los parámetros a ser usados para entrenar los diferentes modelos SVM a desarrollar en el Capítulo IV, se escriben en el vector de opciones que puede verse en la expresión (3.51).

$$\text{Opciones_Modelo} = '-s 0 -t 2 -c 1 -g 0.0011 -e 0.001'; \quad (3.69)$$

Como se puede observar, se definen los parámetros “-s, -t, -c, -g, -e” con sus valores por defecto, definidos al lado izquierdo. Cada uno de estos valores, debe ser sintonizado adecuadamente para poder ajustar el modelo SVM a los datos presentados, para que responda tal y como se estudió en la sección 3.6.

³⁸ Nota: también fueron usadas versiones anteriores de LIBSVM pero compiladas sobre MATLAB Versión R2009a.

CAPÍTULO IV

4 EXPERIMENTOS EN FUSIÓN TERMONUCLEAR

4.1 INTRODUCCIÓN

En este capítulo se muestran todos los experimentos realizados con las imágenes provenientes del diagnóstico de esparcimiento Thomson, que opera en el dispositivo de fusión termonuclear TJ-II ubicado en el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), atendiendo al objetivo y la metodología (ver Figura 1.1) planteada en el Capítulo I, y a los fundamentos teóricos vistos de manera práctica a través de ejemplos en el Capítulo III.

En general, los experimentos se centraron en el estudio de cuatro tipos de modelos, los cuales son: la máquina de vectores soporte (SVM), la predicción conformal (PC), la predicción conformal inductiva (IPC) y la predicción Venn (PV). Con cada uno de estos modelos se desarrolló un multclasificador, presentando también en este capítulo, los resultados de su estudio individual a partir de las variables medidas y contrastadas; *referidas a las altas tasas de aciertos* al procesar las imágenes disponibles, así como *las medidas de confianza, de credibilidad*³⁹ y la confianza por la credibilidad, además de *la probabilidad e intervalos de probabilidades*⁴⁰.

Conjuntamente con las variables ya mencionadas y registradas por cada multclasificador analizado de manera individual, donde se buscó el compromiso entre las altas tasas de aciertos y las medidas de fiabilidad mediante el ajuste de diferentes parámetros; también se incluyó para su optimización (minimizar), el tiempo de procesamiento de los datos utilizados, presentándose en contraste con las anteriores variables ya señaladas.

³⁹ Las medidas de confianza y credibilidad son computadas por la predicción conformal y la predicción conformal inductiva.

⁴⁰ La probabilidad e intervalos de probabilidades son computadas por la predicción Venn.

La tasa de aciertos se calcula como la relación entre el número de muestras acertadas correctamente (aciertos) y el número de muestras procesadas por un multclasificador, tal y como se escribe en la expresión 4.1.

$$tasa_de_aciertos(\%) = \frac{aciertos}{muestras_procesadas} \times 100 \quad (4.1)$$

Donde las muestras procesadas son el conjunto de imágenes reservado para evaluar el multclasificador previamente entrenado, conociendo siempre la clase de cada imagen examinada.

Para definir, los *aciertos* que son el número de muestras predichas correctamente, se debe definir el *acierto* que es la predicción particular obtenido por muestra procesada.

Según esto, se tendrá como *acierto*, al resultado de comparar: la clase pronosticada para una muestra determinada con su clase real. Si en la comparación, las clases predicha y real coinciden, se anota como uno sumando un acierto. Por el contrario, si las clases no coinciden, el multclasificador ha cometido un error y se anota el registro con un cero. Por lo tanto, se puede escribir el *acierto* como a *los aciertos* acumulados de la siguiente manera:

$$acierto = \begin{cases} 1 & \text{si } muestra_pronosticada = muestra_real \\ 0 & \text{si } muestra_pronosticada \neq muestra_real \end{cases} \quad (4.2)$$

$$aciertos = \sum_{de_todo} acierto = 1 \quad (4.3)$$

En cuanto a los procedimientos empleados para clasificar una muestra en relación al número de clases existentes en el problema multiclase abordado por el análisis de las imágenes obtenidas del diagnóstico de esparcimiento Thomson, a partir de los modelos de predicción analizados en esta investigados, con el fin de calcular la tasa de aciertos; se indica para el caso del multclasificador SVM, que se usó la aproximación clásica uno versus el resto descrita la sección 3.6.2, específicamente se

empleó como respuesta para su comparación (según la expresión 4.2), la predicción de etiquetas que puede verse en la sección 3.6.2.3.

Para los multclasificadores desarrollados mediante la predicción conformal y la predicción conformal inductiva, se usó como respuesta para su comparación y con esto conseguir las tasas de aciertos, la predicción de etiquetas obtenidas mediante el empleo de la SVM como método subyacente, computándose la distancia al hiperplano (véase la sección 3.6.2.4) como medida de no conformidad bajo la aproximación modificada uno versus el resto descrita en la sección 3.7.3.1.

Y para los multclasificadores desarrollados mediante la predicción Venn (véase la sección 3.9), la predicción de etiquetas se logra aplicando la taxonomía de los k vecino más cercano y el centroide más cercano (véase la sección 3.9.4).

Con relación a los datos usados para la experimentación con los multclasificadores mencionados, se aprovechó un banco de 1.149 imágenes conformado por 5 clases (se definen los datos en la sección 4.2), permitiendo realizar dos tipos de pruebas, las cuales fueron:

- Los experimentos por lote: que consisten en separar aleatoriamente el banco de imágenes disponible en dos conjuntos. Un conjunto de entrenamiento y otro de prueba (cada uno se conforma con un porcentaje de la totalidad de las imágenes). El primer conjunto se emplea para entrenar el multclasificador. El segundo conjunto se usa para realizar pruebas de clasificación, verificando altas tasas de aciertos con el porcentaje de imágenes escogido para entrenamiento.
- Los experimentos incrementales: que consisten en tomar aleatoriamente un número mínimo de muestras para entrenamiento y el resto se reserva para prueba. Con el conjunto mínimo se entrena el multclasificador y seguidamente se elige aleatoriamente una muestra del conjunto de prueba para su clasificación. La muestra elegida y clasificada, inmediatamente es incorporada al conjunto de entrenamiento respetando su etiqueta original, haciendo que el conjunto de entrenamiento aumente en una muestra. Luego se toma el conjunto de entrenamiento nuevamente (con la muestra incorporada)

y se procede a elegir aleatoriamente otra muestra del conjunto de prueba para su clasificación, repitiendo todo el proceso descrito hasta examinar la totalidad de las muestras disponibles.

A continuación, es importante distinguir los experimentos incrementales de los realizados por lote. Como se ha podido ver, el procedimiento incremental va aumentando el conjunto de entrenamiento con cada muestra examinada, al mismo tiempo la tasa de aciertos gradualmente va aumentando hasta mostrar valores elevados, que indican un número mínimo de imágenes que debe ser usado en un entrenamiento por lote. Por su parte, el procedimiento por lote, verifica que tal número de imágenes encontrado por incremento es un conjunto válido para lograr predicciones precisas, al favorecer altas tasas de aciertos.

Se aclara que, el número de imágenes encontrado en el experimento incremental y el número de imágenes verificado en el experimento por lote, se pueden expresar en porcentaje. Al mismo tiempo se debe recordar que, en cada prueba realizada se procesa un conjunto de imágenes, consiguiendo un alto porcentaje de aciertos siempre que tal medida computada (porcentaje de aciertos) en cualquier multclasificador ensayado, alcance o supere el 90%. Siempre que ocurra esto, se pudiera estar mencionando también, que el porcentaje de aciertos converge.

En los experimentos realizados, los datos fueron agrupados de forma tal, que sus clases existentes quedaran balanceadas o desbalanceadas dentro de los conjuntos definidos tanto para entrenamiento como para pruebas de clasificación. Según esto, los experimentos por lote e incrementales mencionados anteriormente, se subdividen cada uno en experimentos desbalanceados y experimentos balanceados. Este procedimiento se justifica debido al desbalanceado natural que presentan las imágenes obtenidas, siendo entonces el balanceo de imágenes, objeto de estudio y por tal motivo en esta investigación, se mide su efecto sobre la predicción obtenida en cada modelo y sus respectivas medidas de fiabilidad (*donde aplique*⁴¹). Por lo tanto, se consideró en ese sentido los dos tipos de agrupaciones definidas como:

⁴¹ Se recuerda que los multclasificadores SVM no arrojan medidas de fiabilidad.

- Balanceo de imágenes: las clases consideradas contienen el mismo número de imágenes.
- Desbalanceo de imágenes: las clases consideradas no contienen el mismo número de imágenes.

En la Figura 4.1 se muestra un esquema general de las ocho posibles agrupaciones que pueden ser seleccionadas para evaluar (entrenar y probar) los diferentes multclasificadores: SVM, PC, IPC y PV. Además, en la Tabla 4.1 se muestra la cantidad de imágenes por clases usadas según el *tipo de multclasificador*⁴² a evaluar.

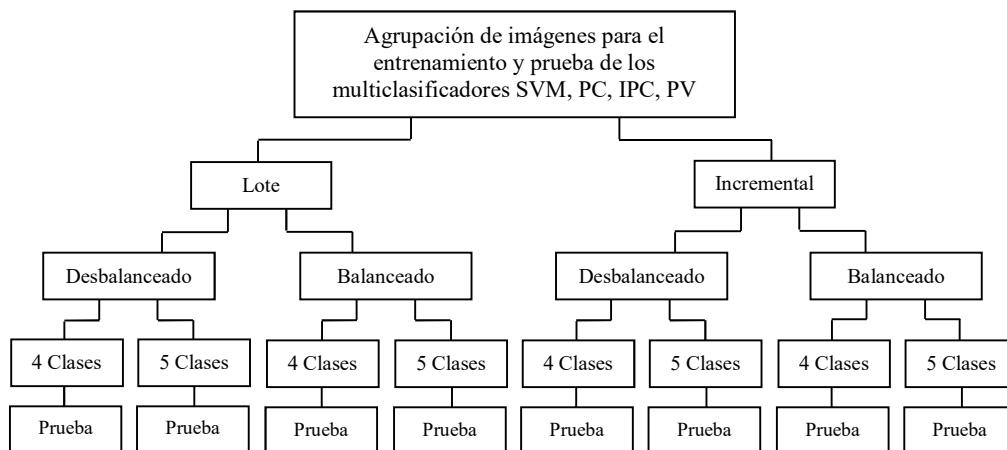


Figura 4.1.- Agrupación de imágenes para el entrenamiento y pruebas de los modelos SVM, PC, IPC, PV.

Se acota finalmente, que este capítulo no se dedica a explicar en profundidad cómo funcionan los multclasificadores a ser presentados, ni los modelos usados como base para su desarrollo; ya que en el Capítulo III, se dio una amplia discusión en este sentido y se incluyeron ejemplos para analizar toda la teoría tras las pruebas que se presentan en adelante. Aunque ante cada prueba presentada, se darán detalles

⁴² En este trabajo se define al tipo de multclasificador según el modelo (SVM, PC, IPC, PV) usado, el número de clases (BKG, ECR, NBI, STR, CUT) existente en el conjunto total de imágenes que procesa y al balanceo o desbalanceo de imágenes.

suficientes para entender las mismas, se mostraran sus resultados y su respectivo análisis individual.

Finalmente, se invita a revisar el Capítulo V donde se mostrará el contraste de los resultados entre los diferentes multclasificadores que serán presentados a lo largo de este capítulo, así como algunas discusiones que surgen por las pruebas (evaluaciones) realizadas.

Tabla 4.1.- Cantidad de imágenes por clase a utilizar según el tipo de multclasificador a evaluar.

		Cantidad de imágenes usadas en un multclasificador para			
		Cuatro clases		Cinco clases	
		Desbalanceado	Balanceado	Desbalanceado	Balanceado
Clase de imágenes → → → → →	BKG	107	107	107	42
	ECR	461	107	461	42
	NBI	334	107	334	42
	STR	205	107	205	42
	CUT	x	x	42	42
Total		1107	428	1149	210

4.2 LOS DATOS

Los datos usados en todos los experimentos realizados fueron tomados del diagnóstico de esparcimiento Thomson, acoplado al dispositivo de fusión termonuclear TJ-II ubicado en el CIEMAT de España [Farias et al, 2004]. Este diagnóstico es uno de los sistemas de medida más importantes en los dispositivos de fusión, ya que permite determinar simultáneamente la densidad y temperatura de los electrones libres en el plasma (véase la sección 2.4.2 para mayor información).

La reemisión de la radiación incidente en el plasma se relaciona con la luz espectral emitida, la cual es captada por una cámara CCD, originándose cinco tipos de imágenes. Cada tipo de imagen constituye una clase (véase la Tabla 4.1) y se corresponde con una medida que puede caracterizar al plasma [Makili et al, 2010]. En la Figura 4.2 se muestra una imagen típica por cada clase y al mismo tiempo en la

Tabla 4.2 su descripción. Para esta investigación se utilizó un banco de 1.149 muestras con una cantidad de imágenes por clase distribuidas como se muestra en la Tabla 4.1.

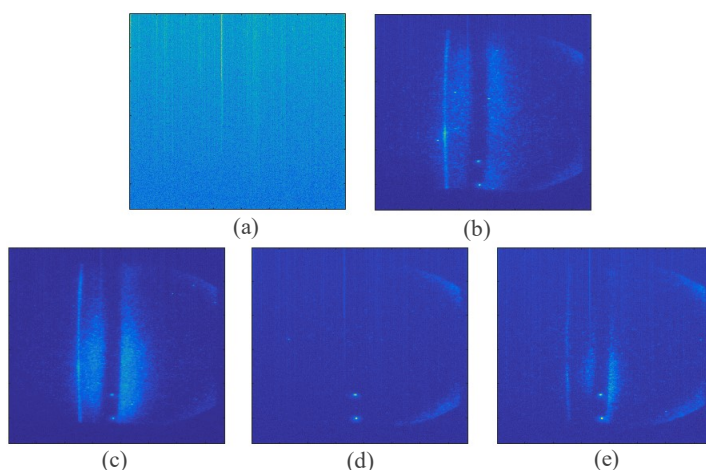


Figura 4.2.- Imágenes típicas asociadas a las cinco clases. (a) BKG, (b) ECR, (c) NBI, (d) STRAY, (e) COFF.

Tabla 4.2.- Imágenes tomadas en el DST y su descripción.

Tipo de Imagen	Descripción
BKG	Fondo de la cámara ICCD
ECR	Plasma con calentamiento ECRH
NBI	Plasma con calentamiento NBI
STRAY	Luz parásita sin plasma
COFF	Plasma en corte ECRH

4.2.1 Pre-procesamiento de imágenes mediante la transformada wavelet

Con el propósito de hacer que los procesos ejecutados por cada multclasificador sea más eficiente desde el punto de vista del costo computacional, se utiliza la transformada wavelet (véase la sección 3.10) para el pre-procesamiento, logrando dos fines concretos que son:

- La reducir de la cantidad de información por imagen a utilizar.
- La selección de características.

En relación a la reducción de información, la transformada wavelet permite que cada imagen con una resolución original de 221.760 píxeles (576 x 385) sea descompuesta en 4 diferentes niveles de detalles como puede apreciarse en la Figura 4.3. Cada nivel de detalle queda finalmente con 900 píxeles (36 x 25). Esto para un nivel 4 de compresión (véase la sección 3.10).

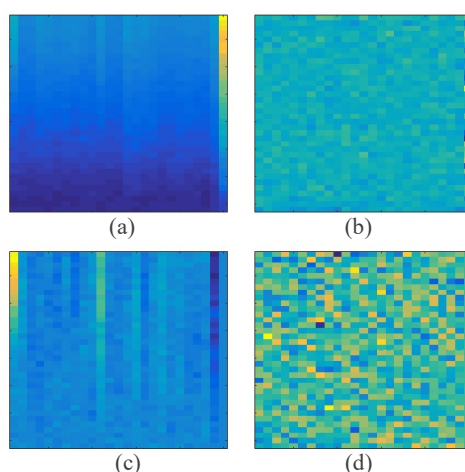


Figura 4.3.- Descomposición de una imagen típica BKG en diferentes niveles de detalle: (a) coeficiente de aproximación, (b) horizontal, (c) vertical y (d) diagonal. Para un nivel de compresión con la Wavelet Haar Nivel 4.

En relación a la selección de características, cada nivel de detalle representa en sí a la imagen original. Por lo tanto, sólo basta con seleccionar cualquiera de los cuatro detalles (en adelante se mencionaran estos detalles como: detalles AHVD) para realizar los diferentes experimentos, pero es evidente que se tendría que probar cuales de estos propicia la mayor tasa de aciertos en los diferentes multclasificadores propuestos; y como se verá más adelante, el detalle vertical (véase la Figura 4.3 (c)), es la que resulta seleccionada.

4.3 LOS PROGRAMAS Y EL ACCESO A DATOS

Toda la parte experimental se realizó en el entorno de programación de MATLAB donde se compilo la librería LIBSVM (véase la sección 3.11). Aquí se diseña e implementan todos los programas (códigos o rutinas) para desarrollar los

diferentes modelos, base de los cuatro multclasificadores (SVM, PC, IPC y PV) estudiados. Además, se implementan todos los códigos de apoyo que dan el acceso, pre-procesamiento, almacenamiento y/o selección de los datos; así como aquellas rutinas que permiten el entrenamiento, prueba y diagnóstico de los diferentes multclasificadores desarrollados.

En líneas generales, cualquier prueba realizada con estas implementaciones mencionadas, sigue un flujo de ejecución y acceso a datos que puede verse a gran escala en la Figura 4.4, donde se observa que las imágenes están alojadas en un directorio raíz (archivos en formato.txt-B1) dentro del disco duro.

Desde aquí se inicia el flujo de ejecución y acceso a datos con la lectura de las imágenes e inmediatamente se construye un array numérico por cada una, pudiendo ser guardado en un banco intermedio (estructura de datos-B2).

Pasar del banco de archivos en formato.txt a la estructura de datos-B2 puede llevar varios minutos y dependerá de la velocidad del procesador donde se ejecuten estos programas. Por esta razón, se crea el banco intermedio (estructura de datos-B2), que puede alojarse en el disco duro como un archivo.dat y al ser requerido; su carga hacia la RAM sólo tardará unos segundos. Esto sucede porque la versión en array de la totalidad de las imágenes alojadas en tal estructura, ocupa 188 Megabytes en comparación con la carpeta original de imágenes en formato.txt que ocupa 5.93 Gigabytes.

Siguiendo con el flujo de ejecución y acceso a datos, a continuación un programa toma las imágenes desde la estructura de datos-B2 alojada en la RAM (recuérdese que una versión está alojada en el disco duro en formato.dat para su carga inmediata) y las pre-procesa mediante la transformada Wavelet Haar, descomponiéndola hasta un nivel 4 de compresión, obteniéndose los cuatro niveles de detalles AHVD. Estos archivos de datos que ocupan en su totalidad unos 14 Megabytes, son los utilizados para el resto de los procesos que se muestran en la Figura 4.4. Una versión de estos archivos de imágenes puede ser alojada en el disco duro (estructura de datos-B3) para tener el acceso eficiente a ellos (aproximadamente dos segundos) para iniciar el trabajo con las imágenes.

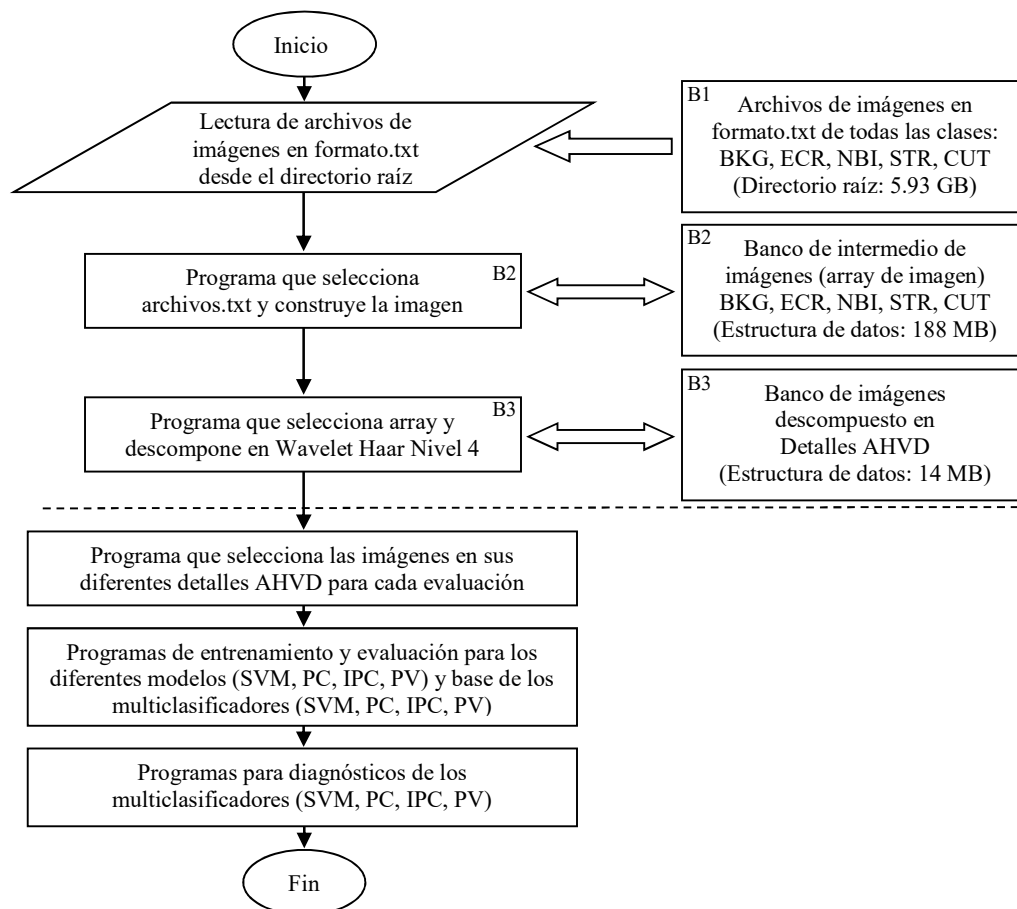


Figura 4.4.- Flujo de ejecución de los programas implementados y su acceso a datos.

A partir de este momento, comienza la obtención de los modelos y de los multclasificadores, ejecutándose el programa que selecciona las imágenes en sus diferentes detalles AHVD para pasarlas a las rutinas de entrenamiento y evaluación, y así obtener información valiosa que finalmente será diagnosticada con otros códigos de apoyo.

Para culminar con esta sección, se hará un comentario especial sobre el proceso ilustrado a partir de la Figura 4.4, con el sentido de validarlo, sin menospreciar cualquier sugerencia para su optimización. Por ejemplo, al no tener únicamente 5 Gigabytes de datos de imágenes, que pueden cargarse para su procesamiento en la RAM de una computadora actual y tener que procesar mucha más información, no se podrá cargar toda en la RAM.

Por lo tanto, el procedimiento de lectura, pre-procesamiento y almacenamiento intermedio sería válido, pudiendo recurrir a la lectura por lotes o secciones de la información que se alojará en un momento determinado sobre la RAM para procesarlas y obtener las estructuras de datos intermedia mencionadas e inmediatamente eliminar sus versiones anteriores.

Seguidamente se debe incorporar el siguiente lote hasta completar la cantidad de información que se tenga que procesar. Toda esta labor comentada, se le deja al lector para su imaginación y comprensión, si llega a tener este tipo de inconvenientes.

4.4 MULTICLASIFICADOR SVM. DETERMINACIÓN DE LAS OPCIONES PARA LOS MODELOS SVM

Los primeros experimentos con la SVM fueron realizados como referencia para el resto de los ensayos que debían hacerse con la PC, IPC y PV. Este punto de partida se toma porque con la SVM ya se han desarrollado e implementado sistemas automáticos de clasificación en el dispositivo de fusión del TJ-II en el CIEMAT [Vega et al, 2005]. En ese sentido y sabiendo que un modelo SVM es funcional para la clasificación de las imágenes disponibles en esta investigación, se procedió a realizar las pruebas con los siguientes condicionantes:

1. Establecer criterios para la selección de un conjunto mínimo de imágenes para entrenar un modelo SVM. Este conjunto mínimo de imágenes debe seleccionarse por cada clase presente en el banco (de imágenes) disponible. Al final se debe construir un multclasificador SVM que permita alcanzar altas tasas de aciertos en las imágenes que quedan en el conjunto de prueba. Este criterio, será adoptado en el resto de los experimentos con la PC, IPC y PV, al hacer pruebas por lote.
2. Seleccionar el detalle AHVD más apropiado para obtener altas tasas de aciertos. Una vez establecido el detalle AHVD, debe mantenerse para el resto de los experimentos con la PC, IPC y PV, con el fin de poder realizar los análisis de la manera más homogénea y coherente posible.

3. Seleccionar las opciones o parámetros de los modelos SVM con kernel RBF que permitan alcanzar altas tasas de aciertos con la reducción consecuente del tiempo de procesamiento. Una vez establecidos, también deben mantenerse para el resto de los experimentos con la PC, IPC y PV.
4. Verificar la influencia del desbalanceo natural de muestras por clase presente en el banco de imágenes disponibles. Esto también se verificará en los experimentos con la PC, IPC y PV.

Teniendo presentes estos condicionantes, se muestran los experimentos realizados con la SVM usando: por una parte, porcentajes o conjuntos de imágenes para ejecutar pruebas por lote; y por otra, conjunto mínimo de cinco imágenes para ejecutar pruebas incrementales.

Prueba 1.- Clasificación por lote de cuatro clases-desbalanceado

En esta primera prueba se experimenta con 1107 imágenes, tomando la totalidad de muestras por clase según la Tabla 4.1. Nótese que el conjunto está desbalanceado con relación al número de imágenes por clase. Antes de pasar a explicar los detalles y resultados de esta primera prueba, se resalta la importancia de la misma.

Inicialmente, en esta primera prueba con las imágenes para clasificarlas, no se conocía a priori cual es el número de muestras mínimas por clase para entrenar los modelos SVM que deben ser incorporados al multclasificador (recuérdese que este experimento se realizó con cuatro clases, por lo tanto hay que entrenar un modelo SVM por clase. Véase la sección 3.6).

Tampoco se sabe cuál de los detalles AHVD daría un mejor resultado en las tasas de aciertos. Y mucho menos se conocían a priori, las opciones de los diferentes modelos SVM que usan un kernel RBF (véase la expresión 3.69).

Por lo tanto, bajo todo este desconocimiento mencionado, se tenía que pensar en una solución que resolviera estas interrogantes. En este sentido se implementaron dos programas para resolver la prueba descrita.

Programa 1. Determinación de las opciones para cuatro modelos SVM a partir de un conjunto de imágenes desbalanceado

El primer programa realiza una búsqueda automática de las opciones para los cuatro modelos SMV (*Opciones_Modelo*. Véase la expresión (3.69)) en un rango fijado manualmente (ajuste grueso) y luego se realiza un ajuste fino para obtener los mejores valores. En la Figura 4.5 se observa el diagrama de flujo de la implementación del Programa 1, notándose que tiene embebido el código para entrenar los cuatro modelos SVM necesarios para que funcione el multclasificador SVM.

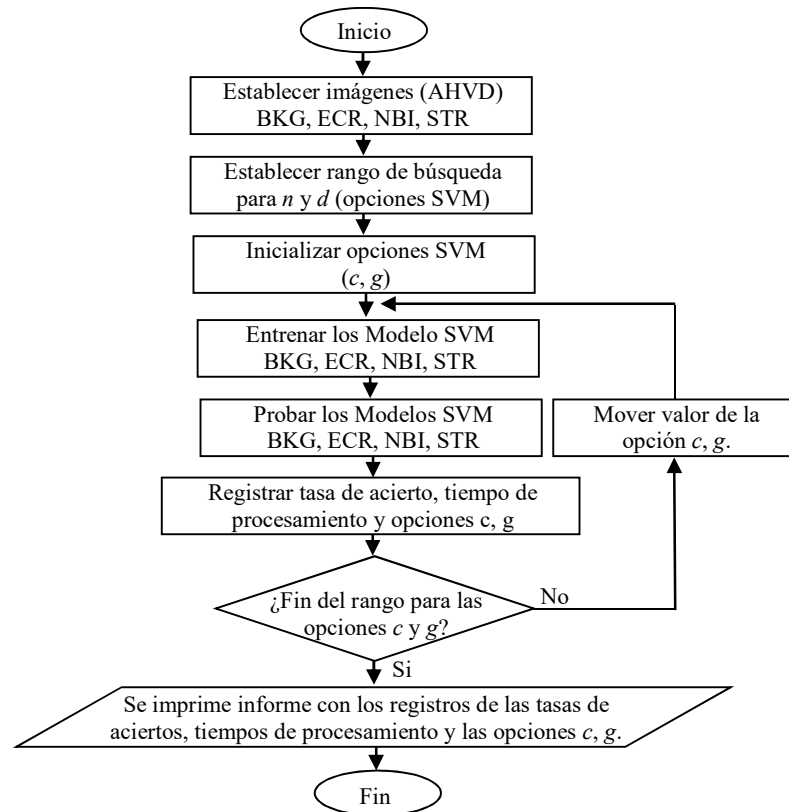


Figura 4.5.- Diagrama de bloques general de la implementación del Programa 1 que explora los valores de las opciones c y g para los modelos SVM acoplados al multclasificador SVM para cuatro clases.

Además, tiene incorporado el código para probar el multclasificador SVM para cuatro clases; y una rutina que registra la tasa de aciertos, el tiempo de

procesamiento y las opciones exploradas en el rango establecido para cada modelo SVM. Al final de la ejecución de este primer programa se imprime un informe donde se pueden comprobar todos los parámetros SVM, con el fin de seleccionar aquellos que den la mejor tasa de acierto en la clasificación con un mínimo de tiempo de proceso.

En la Tabla 4.3 se muestra un extracto del informe mencionado. En la columna 1 (PIE) puede observar el porcentaje de imágenes por lote empleado para entrenar los modelos SVM, de la totalidad de las muestras propuestas para este ensayo (véase la Tabla 4.1). Las columnas 2 y 3 (PID y TDP) muestran el porcentaje de aciertos y el tiempo de procesamiento en segundos, obtenidos sobre el resto de las imágenes usadas para la prueba de clasificación con el multclasificador SVM. Desde la columna 4 a la 11, se muestra la variación de los valores n_1, n_2, n_3, n_4 y d_1, d_2, d_3, d_4 , necesarios para calcular las opciones c y g de los modelos SVM, según las expresiones (4.4) y (4.5).

$$c = 2^n \quad (4.4)$$

$$g = \frac{1}{d} \quad (4.5)$$

En cuanto a las opciones c y g , se debe entender que el Programa 1, explora rangos para los diferentes n y d , y luego se emplean las expresiones (4.4) y (4.5) para calcular los valores c_1, c_2, c_3, c_4 y g_1, g_2, g_3, g_4 . Posteriormente, estos valores son pasados como parámetros a la función que entrena los diferentes modelos SVM que son parte del multclasificador SVM para cuatro clases. Esta labor se realiza según las relaciones que se muestran en la expresión (4.6).

$$\begin{aligned} (c_1, g_1) &\rightarrow \text{Modelo_SVM_1_BKG} \\ (c_2, g_2) &\rightarrow \text{Modelo_SVM_2_ECR} \\ (c_3, g_3) &\rightarrow \text{Modelo_SVM_3_NBI} \\ (c_4, g_4) &\rightarrow \text{Modelo_SVM_4_STR} \end{aligned} \quad (4.6)$$

Continuando con el informe, sólo se observa el uso del detalle V de las imágenes disponibles, aunque fueron usados todos los detalles AHVD. También muestra que se mantuvo constante el porcentaje de imágenes por lote (PIE). En cuanto a la opción *g* para cada modelo SVM, se varían los valores *d* con un ajuste grueso hasta lograr una disminución considerable del tiempo de procesamiento, viéndose este efecto entre el rango de 35.000.000 a 50.000.000.

Tabla 4.3.- Informe proporcionado por la implementación del Programa 1 de la Figura 4.5.

PIE	PIP	TDP	n ₁	n ₂	n ₃	n ₄	d ₁	d ₂	d ₃	d ₄
Detalle V										
50	56	9	0	0	0	0	50000000	50000000	50000000	50000000
50	69	8	0	0	0	4	50000000	50000000	50000000	50000000
50	71	7	0	0	0	8	50000000	50000000	50000000	50000000
50	60	9	0	0	5	0	50000000	50000000	50000000	50000000
50	73	7	0	0	5	4	50000000	50000000	50000000	50000000
50	72	6	0	0	5	8	50000000	50000000	50000000	50000000
50	58	8	0	0	10	0	50000000	50000000	50000000	50000000
50	72	7	0	0	10	4	50000000	50000000	50000000	50000000
50	72	7	0	0	10	8	50000000	50000000	50000000	50000000
50	68	5	0	10	0	0	50000000	50000000	50000000	50000000
50	83	3	0	10	0	4	50000000	50000000	50000000	50000000
50	85	2	0	10	0	8	50000000	50000000	50000000	50000000
50	69	4	0	10	5	0	50000000	50000000	50000000	50000000
50	85	2	0	10	5	4	50000000	50000000	50000000	50000000
50	86	2	0	10	5	8	50000000	50000000	50000000	50000000
50	71	4	0	10	10	0	50000000	50000000	50000000	50000000
50	86	3	0	10	10	4	50000000	50000000	50000000	50000000
50	85	2	0	10	10	8	50000000	50000000	50000000	50000000
50	65	8	8	0	0	0	50000000	50000000	50000000	50000000
50	84	7	8	0	0	4	50000000	50000000	50000000	50000000
50	81	7	8	0	0	8	50000000	50000000	50000000	50000000
50	68	8	8	0	5	0	50000000	50000000	50000000	50000000
50	83	6	8	0	5	4	50000000	50000000	50000000	50000000
50	83	6	8	0	5	8	50000000	50000000	50000000	50000000
50	67	8	8	0	10	0	50000000	50000000	50000000	50000000
50	79	6	8	0	10	4	50000000	50000000	50000000	50000000
50	82	6	8	0	10	8	50000000	50000000	50000000	50000000
50	78	4	8	10	0	0	50000000	50000000	50000000	50000000
50	93	3	8	10	0	4	50000000	50000000	50000000	50000000
50	94	2	8	10	0	8	50000000	50000000	50000000	50000000
50	79	3	8	10	5	0	50000000	50000000	50000000	50000000
50	95	2	8	10	5	4	50000000	50000000	50000000	50000000
50	95	1	8	10	5	8	50000000	50000000	50000000	50000000
50	80	3	8	10	10	0	50000000	50000000	50000000	50000000
50	97	2	8	10	10	4	50000000	50000000	50000000	50000000
50	98	1	8	10	10	8	50000000	50000000	50000000	50000000

PIE: Porcentaje Imágenes para entrenamiento
PIP: Porcentaje de imágenes predichas
TDP: Tiempo de procesamiento (en segundos)
n₁, n₂, n₃, n₄: son los valores usados para el cálculo de las opciones c en los modelo SVM.
d₁, d₂, d₃, d₄: son los valores usados para el cálculo de las opciones g en los modelo SVM.
Los modelos SVM se toman para las clases: BKG, ECR, NBI, STR.

Cabe resaltar, que el informe muestra sólo el valor para 50.000.000 debido a que el tiempo de exploración para establecer todas las opciones c y g de los cuatro modelos SVM puede alcanzar incluso días, produciéndose al final un informe demasiado largo para mostrar en este trabajo. Este último comentario deja en evidencia que, aunque se automatice esta parte del proceso siempre se tendrán que establecer rangos manuales para explorar las opciones de la SVM, procediendo inicialmente con un ajuste grueso y luego seguir con un ajuste fino para ubicar más rápido cualquier parámetro.

Por esta razón se observa en el informe, que sólo varían las opciones c (se van cambiando sólo los valores n). Sin embargo, nótese también, la variación que sufre el porcentaje de acierto y el tiempo de procesamiento por las clasificaciones efectuadas a partir del multclasificador SVM para cuatro clases, reparando que ambas lecturas son inversas, ya que al incrementar una disminuye la otra.

Una ilustración de esta variación inversa puede verse en la Figura 4.6, donde se muestra el porcentaje de acierto (curva en color azul) versus el tiempo de procesamiento (curva en color rojo). Se aclara que los valores del porcentaje de aciertos y tiempo de procesamiento han sido normalizados para poder confrontarlos con los valores n_1 , n_2 , n_3 y n_4 , usados para calcular cada valor c de los modelos SVM, los cuales como ya se ha mencionado son acoplados al multclasificador SVM para cuatro clases.

Como comentario final del resultado obtenido por la implementación del Programa 1 mostrado en la Figura 4.5, observe la última fila del informe visto en la Tabla 4.3 donde los valores de n y d se corresponden con la última descarga mostrada en la Figura 4.6. Tales valores de n y d son los que finalmente establecen las opciones c y g en cada modelo SVM del multclasificador SVM para cuatro clases que se estudiará en la segunda parte de esta Prueba 1. También se mantendrán para el resto de las pruebas que involucran imágenes con 4 clases, incluyendo los experimentos que se verán más adelante con la PC, IPC y PV; pero se adelanta que los mismos deben sufrir un pequeño ajuste para poder lograr finalmente, altas tasas de aciertos en los multclasificadores PC e IPC.

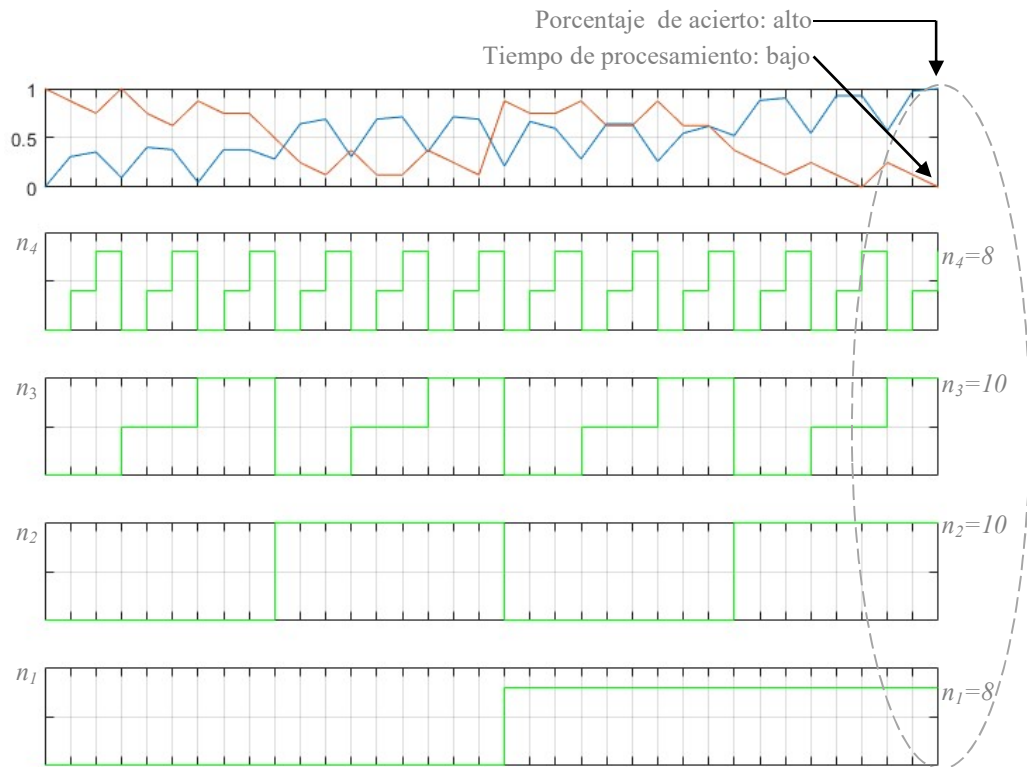


Figura 4.6.- Porcentaje de acierto (curva en color azul) versus el tiempo de procesamiento (curva en color rojo) en contraste con los valores n_1 , n_2 , n_3 , n_4 usados para calcular los valores de c de los modelos SVM que son acoplados al multclasificador SVM para cuatro clases.

Programa 2.- Exploración del conjunto mínimo de muestras por lote y detalle AHVD a partir de un conjunto de imágenes desbalanceado para entrenar un multclasificador SVM-cuatro clases

Anteriormente fue presentado el Programa 1, que se encarga de explorar un rango para encontrar las opciones c y g y así lograr altas tasas de aciertos con su correspondiente minimización del tiempo de procesamiento al examinar un conjunto de imágenes para su correcta clasificación. Después, estas opciones son establecidas en la implementación del Programa 2, para entrenar los modelos SVM acoplados en el multclasificador SVM para cuatro clases que serán analizados en esta segunda parte de la Prueba 1.

Antes de comenzar a explicar lo que se consigue con la implementación del Programa 2, se comenta brevemente, que los Programas 1 y 2 son muy similares.

Incluso todo el proceso pudo resumirse en un único código. Sin embargo, durante los ensayos preliminares con las imágenes obtenidas del dispositivo de fusión termonuclear TJ-II, se decide *separar*⁴³ la prueba en dos partes porque inicialmente se colocaron las opciones SVM por defecto que sugiere [Chang y Lin, 2018] y al ver que no respondían los diferentes modelos SVM en el momento de hacer las pruebas de clasificación, se tuvo que diseñar un programa (Programa 1) que inspeccionara tales opciones hasta que la tasa de aciertos convergiera a un alto valor con su correspondiente tiempo de procesamiento mínimo, esto para un determinado conjunto de imágenes de prueba.

En tal circunstancia, se decide arbitrariamente que, el Programa 1 tomaría el 50% de las imágenes para entrenar los diferentes modelos SVM y el resto para probar el multclasificador SVM. De esta manera se consiguen los valores de las opciones c y g que son enviadas por parámetro al Programa 2.

El Programa 2, que se muestra con el diagrama de bloques de la Figura 4.7, se encarga de explorar un rango de valores para encontrar el porcentaje mínimo de muestras para realizar los entrenamientos del multclasificador SVM por lote, en contraste con los diferentes niveles de detalle AHVD, siempre verificando la condición más beneficiosa en dicho compromiso (porcentaje mínimo versus detalle AHVD) para alcanzar altas tasas de aciertos al realizar las pruebas de clasificación.

Obsérvese también en el Programa 2, que se han embebido los códigos: para el entrenamiento, las pruebas de clasificación, además de incorporarse las rutinas que recogen y realizan un diagnóstico automático de los datos de interés arrojados por el multclasificador SVM para cuatro clases.

Al final de su ejecución, se imprime el informe mostrado en la Tabla 4.4 donde puede observarse: los porcentajes de imágenes establecidos (columna 1: PIE) para realizar el entrenamiento por lote, junto con el porcentaje de imágenes predichas (columnas 2: PIP) a partir del conjunto reservado para prueba de clasificación (véase la Tabla 4.2), con su respectivo tiempo de procesamiento (columnas 3: TDP).

⁴³ Otros problemas que obligaron a separar la Prueba 1 en dos partes fueron: las diferentes clases presentes en el conjunto de datos, el desbalanceo en el número de muestras por clase, los diferentes niveles detalles AHVD y el lote mínimo de imágenes para un entrenamiento efectivo.

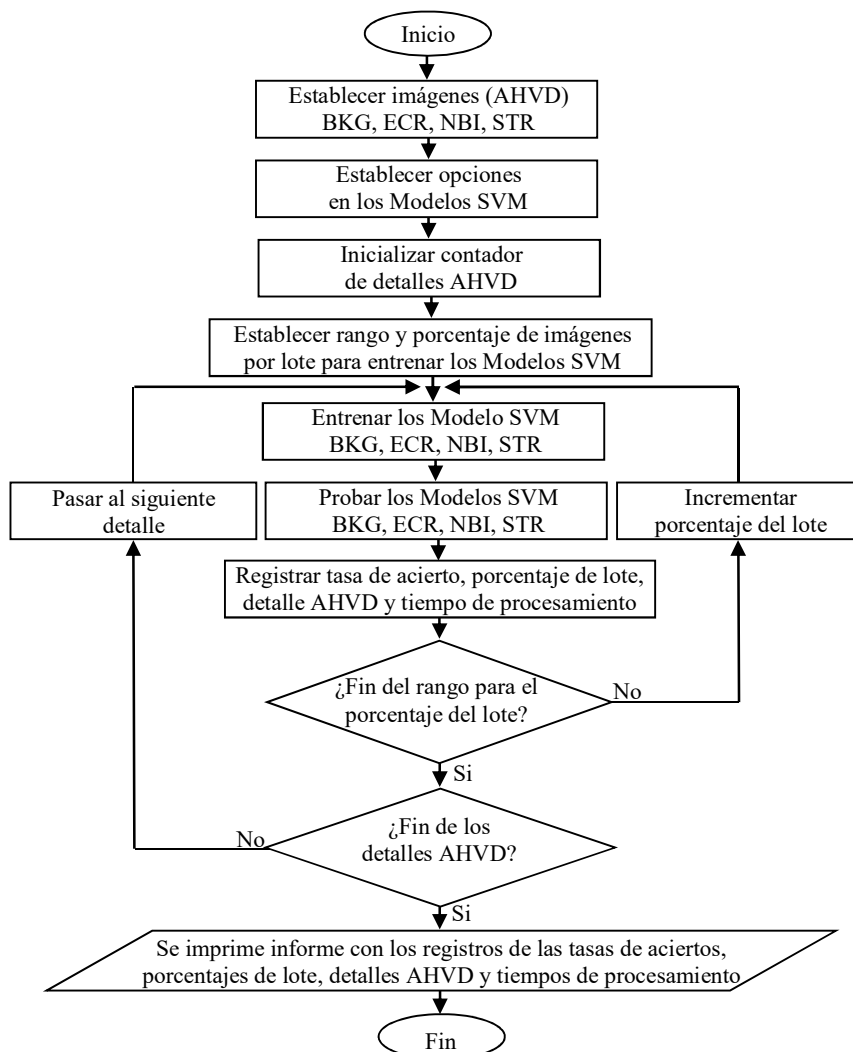


Figura 4.7.- Diagrama de bloques general de la implementación del Programa 2 que explora los conjuntos de muestras tomados por lote en contraste con los diferentes detalles AHVD a partir del multclasificador SVM para cuatro clases.

También se observa desde la columna 4 a la 11, los valores de n_1, n_2, n_3, n_4 y d_1, d_2, d_3, d_4 , encontrados mediante el Programa 1 y pasados por parámetros al Programa 2 para poder calcular las opciones c y g en el instante de entrenar los modelos SVM, en esta segunda parte de la prueba. Nótese además, los altos porcentajes de imágenes predichas al emplear los valores c y g en los entrenamientos y pruebas por lote del multclasificador para cuatro clases.

Tabla 4.4.- Informe proporcionado por la implementación del Programa 2 de la Figura 4.7.

PIE	PIP	TDP	n ₁	n ₂	n ₃	n ₄	d ₁	d ₂	d ₃	d ₄
Detalle A-----										
10	96	0	8	10	10	8	50000000	50000000	50000000	50000000
20	97	0	8	10	10	8	50000000	50000000	50000000	50000000
30	97	0	8	10	10	8	50000000	50000000	50000000	50000000
40	98	1	8	10	10	8	50000000	50000000	50000000	50000000
50	97	1	8	10	10	8	50000000	50000000	50000000	50000000
60	98	1	8	10	10	8	50000000	50000000	50000000	50000000
70	99	1	8	10	10	8	50000000	50000000	50000000	50000000
80	97	1	8	10	10	8	50000000	50000000	50000000	50000000
90	97	0	8	10	10	8	50000000	50000000	50000000	50000000
Detalle H-----										
10	70	2	8	10	10	8	50000000	50000000	50000000	50000000
20	70	5	8	10	10	8	50000000	50000000	50000000	50000000
30	75	6	8	10	10	8	50000000	50000000	50000000	50000000
40	75	7	8	10	10	8	50000000	50000000	50000000	50000000
50	81	7	8	10	10	8	50000000	50000000	50000000	50000000
60	80	7	8	10	10	8	50000000	50000000	50000000	50000000
70	78	6	8	10	10	8	50000000	50000000	50000000	50000000
80	82	6	8	10	10	8	50000000	50000000	50000000	50000000
90	84	4	8	10	10	8	50000000	50000000	50000000	50000000
Detalle V-----										
10	94	0	8	10	10	8	50000000	50000000	50000000	50000000
20	94	1	8	10	10	8	50000000	50000000	50000000	50000000
30	96	1	8	10	10	8	50000000	50000000	50000000	50000000
40	94	1	8	10	10	8	50000000	50000000	50000000	50000000
50	96	1	8	10	10	8	50000000	50000000	50000000	50000000
60	96	1	8	10	10	8	50000000	50000000	50000000	50000000
70	96	1	8	10	10	8	50000000	50000000	50000000	50000000
80	96	1	8	10	10	8	50000000	50000000	50000000	50000000
90	98	1	8	10	10	8	50000000	50000000	50000000	50000000
Detalle D-----										
10	45	2	8	10	10	8	50000000	50000000	50000000	50000000
20	52	5	8	10	10	8	50000000	50000000	50000000	50000000
30	56	8	8	10	10	8	50000000	50000000	50000000	50000000
40	57	10	8	10	10	8	50000000	50000000	50000000	50000000
50	58	10	8	10	10	8	50000000	50000000	50000000	50000000
60	59	10	8	10	10	8	50000000	50000000	50000000	50000000
70	61	10	8	10	10	8	50000000	50000000	50000000	50000000
80	61	8	8	10	10	8	50000000	50000000	50000000	50000000
90	66	7	8	10	10	8	50000000	50000000	50000000	50000000

PIE: Porcentaje Imágenes para entrenamiento
PIP: Porcentaje de imágenes predichas
TDP: Tiempo de procesamiento (en segundos)
n₁, n₂, n₃, n₄, son los valores usados para el cálculo de las opciones c en los modelo SVM.
d₁, d₂, d₃, d₄, son los valores usados para el cálculo de las opciones g en los modelo SVM.
Los modelos SVM se toman para las clases: BKG, ECR, NBI, STR.

Como ejemplo, puede verificarse para el 50% del porcentaje de imágenes usado para entrenamiento (columna PIE), que se han obtenido al examinar los detalles A y V un 97% y 96% de tasa de aciertos respectivamente con un tiempo de procesamiento aproximado de 1 segundo en ambos casos. Por el contrario, y al mismo tiempo puede observarse bajo las mismas circunstancias, que el

multiclasificador discrimina los detalles H y D al reflejar un 81% y 58% de imágenes predichas con un tiempo aproximado de procesamiento de 7 y 10 segundos respectivamente.

Un análisis con mayor detalle del estudio hecho en esta primera prueba, vinculante con los resultados vistos en la Tabla 4.4, se muestra en la Figura 4.8 donde se han contrastado los porcentajes de aciertos versus los porcentajes por lote de imágenes en los diferentes detalles AHVD usados para entrenar *el mismo multiclasificador SVM*⁴⁴. Tal contraste permite comentar que los detalles A y V son los más prometedores, ya que propician un alto porcentaje de aciertos para cualquier porcentaje de lote de imágenes encogido al entrenar el multiclasificador SVM para cuatro clases.

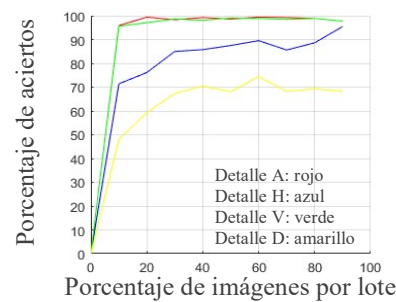


Figura 4.8.- Contraste obtenido entre el porcentaje de aciertos versus el porcentaje por lote al procesar las imágenes en sus diferentes detalles AHVD con el multiclasificador SVM para cuatro clases, entrenado a partir de las opciones *c* y *g* establecidas con la implementación del Programa 1.

Para verificar que este resultado prometedor no depende del sorteo aleatorio hecho en el momento de escoger por lote el conjunto de imágenes para efectuar el entrenamiento del multiclasificador SVM para cuatro clases, se planteó realizar una prueba intensiva que consiste en ejecutar el Programa 2, cien veces.

Los resultados obtenidos se muestran en la Figura 4.9 donde se observa que el porcentaje de acierto bajo las consideraciones expuestas, se mantienen por encima del 95% a partir de un 20% de la totalidad de las imágenes procesadas con el Detalle A;

⁴⁴ Cuando se dice que es el mismo multiclasificador SVM, es porque en todos los casos se han mantenido las opciones *c* y *g* encontradas con el Programa 1

siendo muy parecidos los resultados que se tienen mediante las imágenes procesadas con Detalle V.

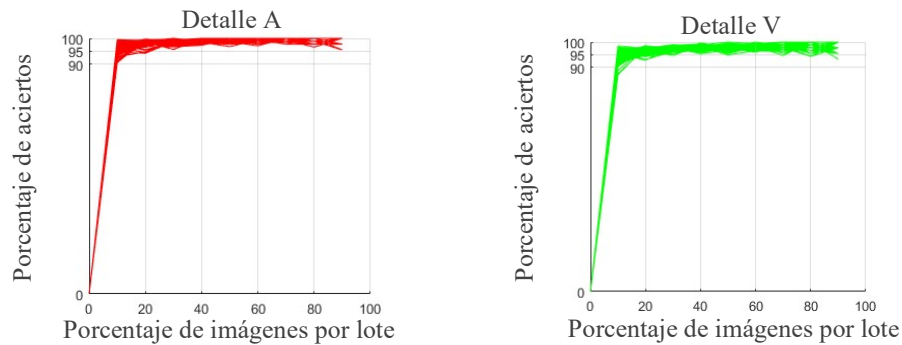


Figura 4.9.- Resultado obtenido al ejecutar 100 veces la implementación del Programa 2 para el multclasificador de cuatro clases. Porcentaje de aciertos versus el porcentaje por lote de imágenes para los detalles A y V.

Prueba 2.- Clasificación por lote de cinco clases-desbalanceado

En la Prueba 1 se consideraron sólo cuatro clases. Ahora, para esta Prueba 2 se experimentó con la totalidad de las imágenes obtenidas en el dispositivo de fusión termonuclear TJ-II, incrementando de esta manera el número de clases a cinco para un total de 1149 muestras. Obsérvese en la Tabla 4.1 el resumen de imágenes por clase, apreciando que este conjunto también está desbalanceado (con relación al número de imágenes por clase).

Recuérdese que, al incluir la clase CUT, se incrementa el número de modelos SVM en el multclasificador SVM, ya que debe predecir datos para cinco clases. Por lo tanto, se tienen que determinar las opciones de todos los modelos SVM, incluyendo el modelo para la clase CUT. Esta implementación se realiza siguiendo todo el procedimiento anteriormente visto en la Prueba 1, verificando sí se modifican las condiciones que permitieron los resultados antes de agregar la nueva clase. En definitiva, se estudia la influencia que tiene aumentar el número de clases.

Programa 1. Determinación de las opciones para cinco modelos SVM a partir de un conjunto de imágenes desbalanceado

La Prueba 2 comienza buscando las opciones para los modelos SVM mediante el Programa 1, incluyendo la nueva clase como se observa en el diagrama de bloques de la Figura 4.10. Luego de la exploración, se consigue que los valores de n y d (para calcular las opciones c y g respectivamente) se mantienen para los modelos SVM asociados a las clases BKG, ECR, NBI y STR. Para el modelo CUT, el mismo Programa 1 establece sus valores para n y d en 7 y 50.000.000 respectivamente.

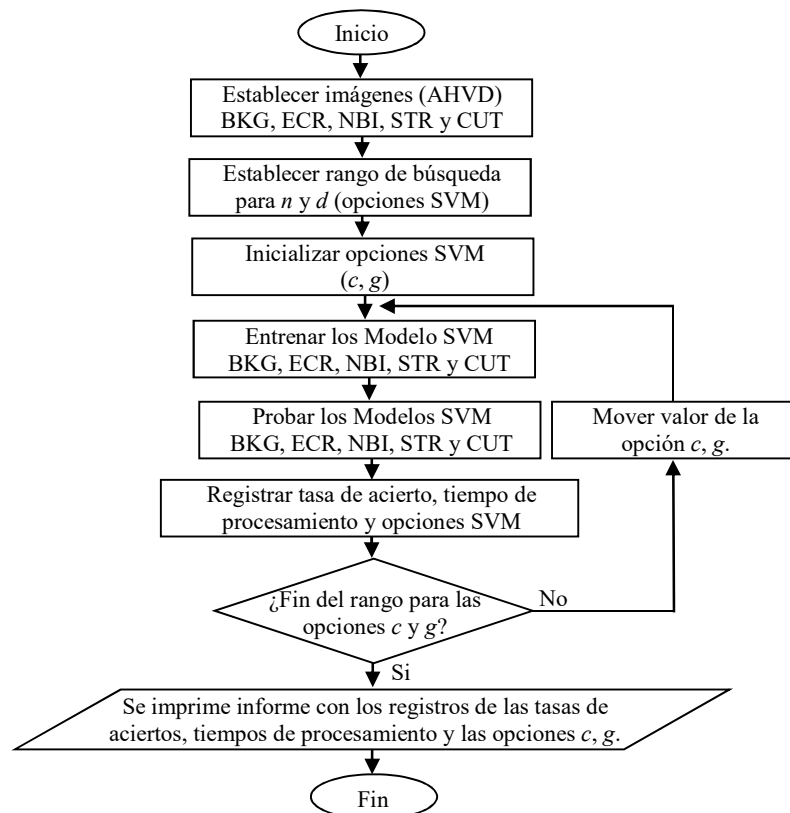


Figura 4.10.- Diagrama de bloques general de la implementación del Programa 1 que explora los valores de las opciones c y g para los modelos SVM acoplados al multclasificador SVM para cinco clases.

En la Figura 4.11 se contrasta el porcentaje de acierto (curva en color azul) versus el tiempo de procesamiento (curva en color rojo) arrojados por el Programa 1. Las mediciones fueron normalizadas para confrontarlas con los valores n_1 , n_2 , n_3 , n_4 y

n_5 . Recuérdese que estos valores son usados para calcular las opciones c de los modelos SVM acoplados al multclasificador SVM para cinco clases.

Con relación a los valores d para todos los modelos SVM en el momento de realizar esta primera parte de la Prueba 2, los mismos fueron cambiándose mediante un ajuste grueso en el rango de 1 a 100.000.000, notándose que entre el rango de 35.000.000 a 50.000.000 se obtenían mínimos tiempos de procesamiento en algunos casos. Por esto se establece su valor en 50.000.000 nuevamente.

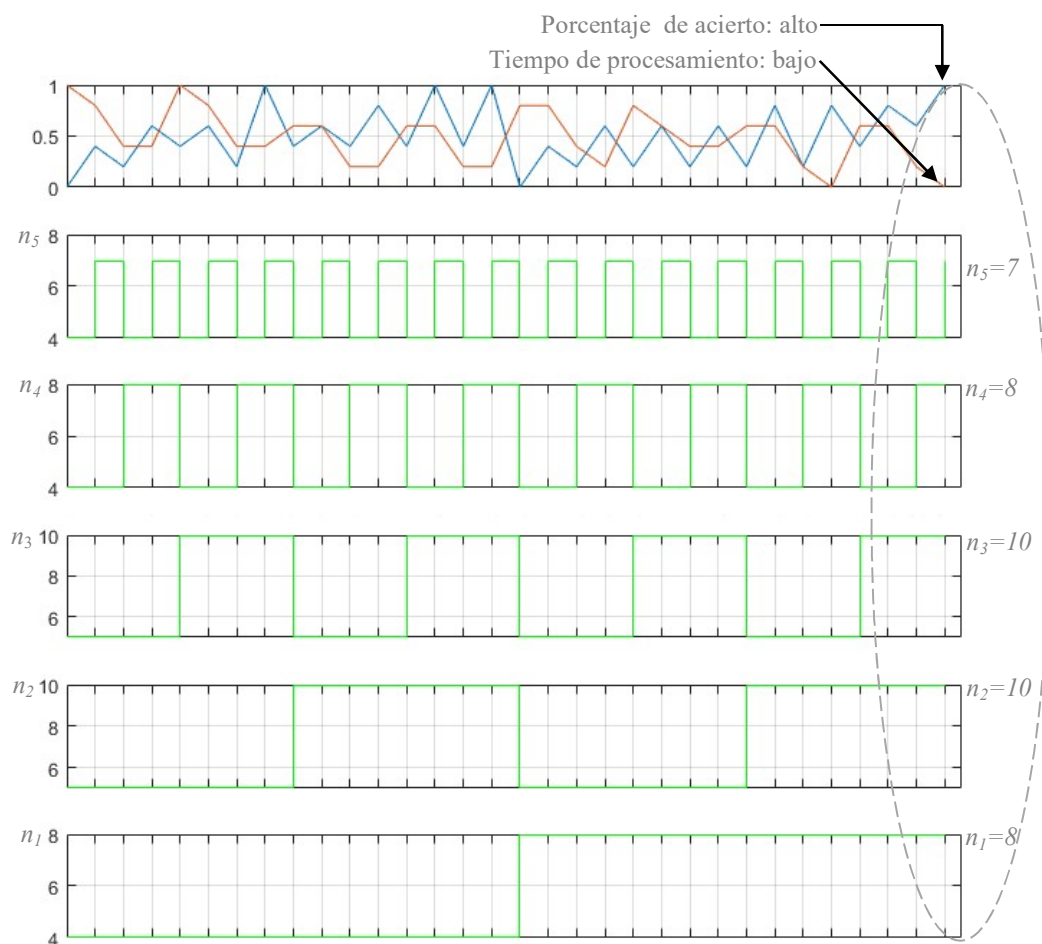


Figura 4.11.- Porcentaje de acierto (curva en color azul) versus el tiempo de procesamiento (curva en color rojo) en contraste con los valores n_1 , n_2 , n_3 , n_4 , y n_5 usados para calcular los valores de c de los modelos SVM que son acoplados al multclasificador SVM para cinco clases.

Programa 2. Exploración del conjunto mínimo de muestras por lote y detalle AHVD a partir de un conjunto de imágenes desbalanceado para entrenar un multclasificador SVM-cinco clases

Encontradas las opciones de los cinco modelos SVM mediante el Programa 1, se pasan al Programa2 (véase la Figura 4.12) para explorar el porcentaje mínimo de muestras por lote necesario para entrenar el multclasificador SVM para cinco clases en contraste con los diferentes niveles de detalle AHVD, verificándose en el informe correspondiente en la Tabla 4.5, que los detalles A y V siguen dando altas tasas de aciertos.

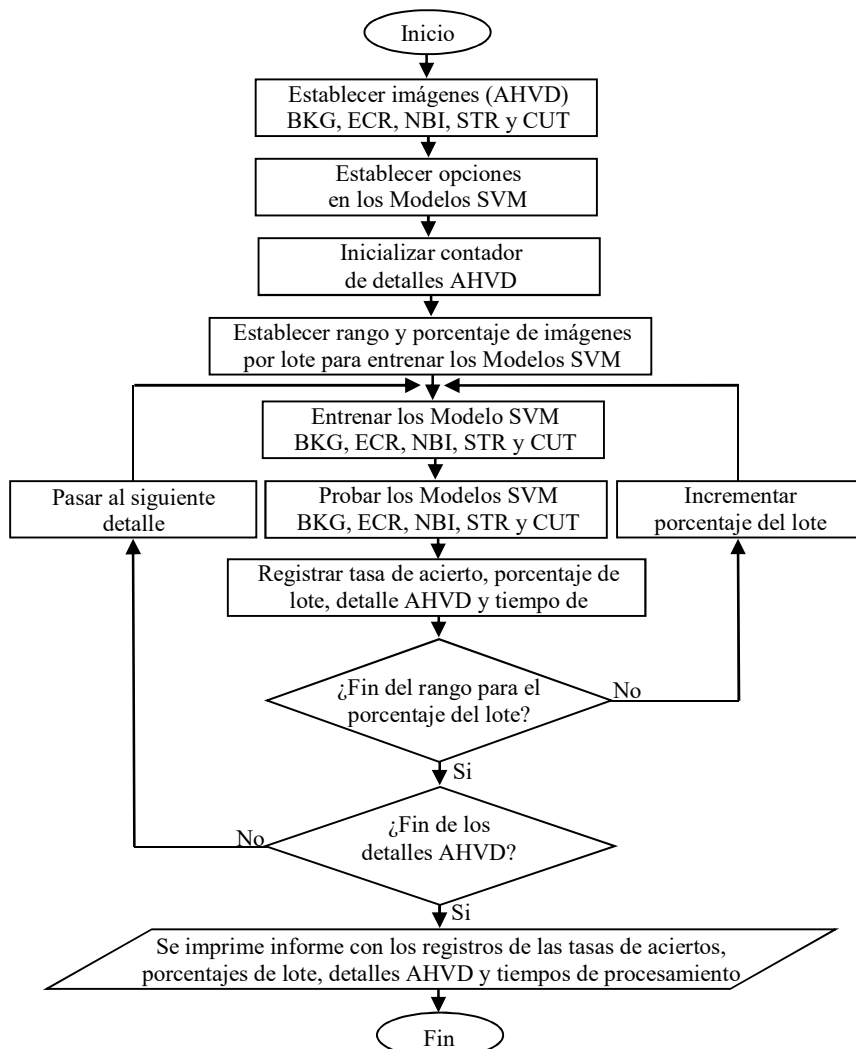


Figura 4.12.- Diagrama de bloques general de la implementación del Programa 2 que explora los conjuntos de muestras tomadas por lote en contraste con los diferentes detalles AHVD a partir del multclasificador SVM para cinco clases.

Tabla 4.5.- Informe proporcionado por la implementación del Programa 2 de la Figura 4.12.

PIE	PIP	TDP	n ₁	n ₂	n ₃	n ₄	n ₅	d ₁	d ₂	d ₃	d ₄	d ₅
Detalle A-----												
10	99	4	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
20	100	4	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
30	99	4	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
40	99	3	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
50	99	3	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
60	99	2	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
70	99	2	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
80	100	2	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
90	100	1	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
Detalle H-----												
10	95	30	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
20	95	27	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
30	95	24	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
40	94	21	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
50	95	18	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
60	95	15	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
70	95	12	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
80	95	9	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
90	94	6	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
Detalle V-----												
10	99	7	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
20	99	7	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
30	99	6	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
40	99	5	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
50	99	5	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
60	99	4	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
70	99	3	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
80	100	2	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
90	99	2	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
Detalle D-----												
10	88	46	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
20	88	41	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
30	88	37	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
40	87	32	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
50	87	28	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
60	88	23	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
70	90	18	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
80	86	15	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
90	90	10	8	10	10	8	7	50000000	50000000	50000000	50000000	50000000
PIE: Porcentaje Imágenes para entrenamiento PIP: Porcentaje de imágenes predichas TDP: Tiempo de procesamiento (en segundos) n ₁ , n ₂ , n ₃ , n ₄ , n ₅ : son los valores usados para el cálculo de las opciones c en los modelo SVM. d ₁ , d ₂ , d ₃ , d ₄ , d ₅ : son los valores usados para el cálculo de las opciones g en los modelo SVM. Los modelos SVM se toman para las clases: BKG, ECR, NBI, STR, CUT.												

Este resultado también se confirma visualmente en la Figura 4.13, donde además se puede observar que los aciertos cuando se incluye una clase más, aumentan casi al 100 %, en detrimento del tiempo de procesamiento. Esto último asociado al tiempo de procesamiento, se puede ver en la columna 3 (TDP) de la Tabla 4.5.

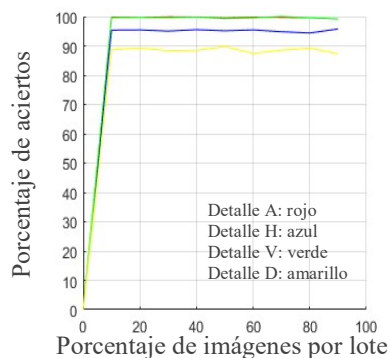


Figura 4.13.- Contraste obtenido entre el porcentaje de aciertos versus el porcentaje por lote al procesar las imágenes en sus diferentes detalles AHVD con el multclasificador SVM para cinco clases, entrenado a partir de las opciones *c* y *g* establecidas con la implementación del Programa 1.

Para probar que el resultado anterior no depende del sorteo aleatorio realizado en el momento de escoger por lote las imágenes para el entrenamiento del multclasificador SVM para cinco clases, se plantea nuevamente la ejecución de cien veces del Programa 2, mostrando estos nuevos resultados en la Figura 4.14.

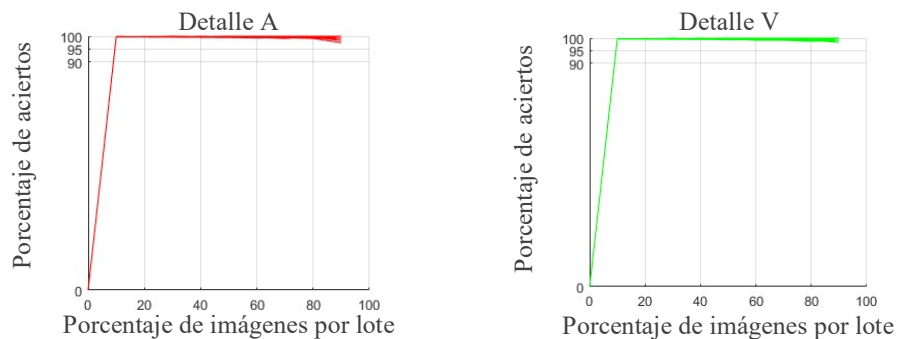


Figura 4.14.- Resultado obtenido al ejecutar 100 veces el Programa 2 para el multclasificador de cinco clases.

Obsérvese que el porcentaje de acierto obtenido con las consideraciones expuestas se mantienen muy por encima del 95%, a partir de un 10% de la totalidad de las imágenes procesadas con Detalle A y V. Esto indica que al agregar una clase más, la clasificación global mejora considerablemente.

Análisis de los resultados obtenidos en las Prueba 1 y 2

Al comparar las Pruebas 1 y 2, se concluye que:

- Las opciones de los modelos SVM asociados a los datos BKG, ECR, NBI, y STR, permanecen constantes. Esto puede ser observado en la Tabla 4.6, donde se comparan los multclasificadores SVM desarrollados para cuatro y cinco clases.
- Se comprueba que al emplear una clase adicional, la clasificación global mejora para cualquiera de los detalles AHVD. Esto se puede observar al comparar las Figuras 4.8 y 4.13, donde se tienen los porcentajes de aciertos obtenidos versus los porcentajes por lote tomados para entrenar los dos multclasificadores SVM bajo diferentes detalles AHVD. Recuérdese que al entrenar un multclasificador, se entrena sus modelos SVM base (modelos para cada clase: BKG, ECR, NBI, STR y CUT⁴⁵)
- Además, al comparar las Figuras 4.8 y 4.13, se nota que los detalles A y V son los mejores para alcanzar altas tasas de aciertos.
- Adicionalmente, se observa que para cinco clases se puede tomar el conjunto mínimo de imágenes para el entrenamiento, a partir de un 10%. Esto se debe a que en este punto ya se alcanza un porcentaje de acierto muy por encima del 95%. Mientras que para cuatro clases, el conjunto mínimo de imágenes para entrenamiento se encuentra alrededor del 20%, alcanzándose un porcentaje de acierto alrededor del 95%.
- Con relación al procedimiento establecido en el Programa 1 para explorar automáticamente las opciones SVM (c y g), el mismo no es suficiente para lograr de forma efectiva el establecimiento de tales opciones. Esto se debe a que en la heurística general de exploración, hay que aplicar ajustes gruesos de forma manual por un usuario experto para poder dar con un espacio de valores donde se observen mejoras tanto de la tasa de aciertos como del tiempo de

⁴⁵ Recuérdese que la clase de imágenes CUT sólo se utiliza en el multclasificador SVM para cinco clases.

procesamiento. Una vez ubicada la zona, se deben aplicar ajustes manuales de manera fina para dar finalmente con los valores de c y g .

- El Programa 2, permite explorar el conjunto de imágenes mínimo necesario para lograr altas tasas de aciertos. Además de permitir la exploración de los detalles AHVD más apropiados para tal fin. Además de ser efectivo en las tareas mencionadas, el Programa 2 sirve para obtener un diagnóstico visual que permite tomar decisiones rápidamente.

Tabla 4.6.- Comparación de las opciones encontradas a partir del Programa 1 para los modelos SVM, base de los multclasificadores SVM para cuatro y cinco clases a partir de un conjunto de imágenes desbalanceado.

Opciones	n_1	n_2	n_3	n_4	n_5	d_1	d_2	d_3	d_4	d_5
Multclasificador SVM 4 clases	8	10	10	8	-	50	50	50	50	-
Multclasificador SVM 5 clases	8	10	10	8	7	50	50	50	50	50

Nota: los valores de d se expresan en millones

Prueba 3.- Clasificación por lote de cuatro clases-balanceado

Este ensayo es similar a la Prueba 1, aplicándose consecuentemente los Programas 1 y 2. El Programa 1, debe explorar las opciones SVM para ubicar los valores que permitan un alto porcentaje de acierto y un tiempo mínimo de procesamiento en la clasificación de un conjunto balanceado de 428 imágenes para las clases: BKG, ECR, NBI y STR. Cada clase contiene 107 muestras (véase de nuevo la Tabla 4.1). El Programa 2, debe explorar el conjunto de imágenes mínimo por lote necesario para entrenar el multclasificador SVM para cuatro clases, también con imágenes balanceadas, intentando obtener altas tasas de aciertos, además de ubicar el detalle AHVD más apropiado para esta tarea.

Programa 1. Determinación de las opciones para cuatro modelos SVM a partir de un conjunto de imágenes balanceado

Al usar el Programa 1 con un 50% (fijado de forma arbitraria) de imágenes para entrenamiento tomadas aleatoriamente y de esta manera explorar las opciones c y g según las especificaciones de la Prueba 3, se observa que la variación en tales

opciones no cambia significativamente con relación a los resultados ya obtenidos en las Prueba 1 y 2, tal y como se muestra en la Tabla 4.7. Se indica que esta medición se realizó usando el otro 50% de las imágenes reservadas para la prueba de clasificación.

Tabla 4.7.- Comparación de las opciones encontradas con imágenes balanceadas y desbalanceadas para los modelos SVM, base de los multclasificadores SVM para cuatro clases.

Opciones	n_1	n_2	n_3	n_4	d_1	d_2	d_3	d_4
Multclasificador SVM-desbalanceado	8	10	10	8	50	50	50	50
Multclasificador SVM-balanceado	10	9	10	8	50	50	50	50

Nota: los valores de d se expresan en millones

Con el fin de comprobar, si esta insignificante variación de los valores de las opciones c y g afecta el porcentaje de aciertos al procesar diferentes lotes de imágenes del conjunto balanceado, se procede a ejecutar el Programa 2 para verificar:

- Si el porcentaje de acierto cambia al procesar el conjunto de imágenes balanceado.
- Si el porcentaje mínimo de imágenes (balanceado) por lote para entrenar el multclasificador SVM para cuatro clases cambia para conjuntos de imágenes balanceados.
- Si los detalles A y V siguen siendo los más apropiados para propiciar altas tasas de aciertos.

Programa 2.- Verificación del porcentaje de acierto a partir de un conjunto de imágenes balanceadas por cambiar las opciones SVM para entrenar un multclasificador SVM-cuatro clases

En la Figura 4.15 se muestra el resultado de procesar el conjunto de imágenes balanceado en la Prueba 3, junto con el resultado ya obtenido en la Prueba 1 para su contraste. De esta comparación se revela que el porcentaje de aciertos sigue siendo

alto para los detalles A y V al procesar conjuntos de imágenes tanto balanceados como desbalanceados.

En cuanto al conjunto mínimo de imágenes por lote a utilizar, se observa que puede tomarse un porcentaje a partir del 10% para un superior 95% de aciertos, tanto en el detalle A como en el V para los conjuntos balanceados, quedando los detalles H y D por debajo del 90%, como era de esperar. En contraparte, se observa que los conjuntos desbalanceados requieren de un porcentaje mayor de imágenes (alrededor del 20%) para reflejar altos porcentajes de aciertos.

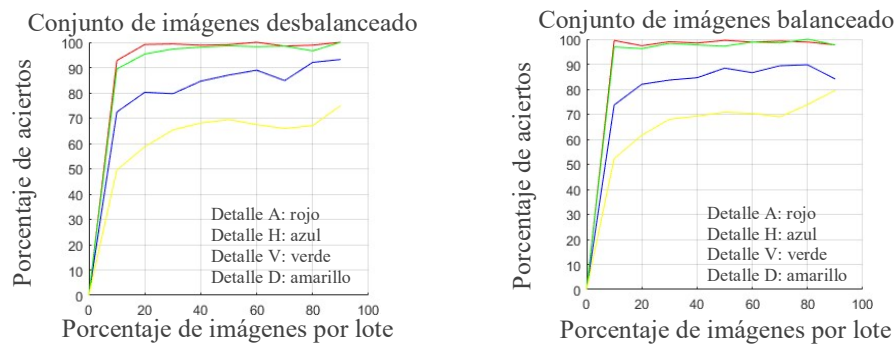


Figura 4.15.- Verificación del porcentaje de aciertos para diferentes detalles AHVD medidos a partir de conjuntos de imágenes balanceados y desbalanceados al usar un multclasificador para cuatro clases.

Para confirmar los porcentajes (lotes) mínimos de imágenes a tomar para entrenar el multclasificador de cuatro clases, tanto en conjuntos balanceados como desbalanceados; se ejecuta 20 veces el Programa 2 con sólo el detalle V. Al confrontar los resultados en la Figura 4.16 se observa que el entrenamiento puede comenzar fijando cualquier conjunto por encima del 20%. Bajo este perfil se podrá obtener un oscilante porcentaje de acierto alrededor del 95%.

Análisis de los resultados obtenidos en la Prueba 3

Al comparar las Pruebas 1 y 3, se concluye que:

- Las opciones *c* y *g* de los modelos SVM asociados a los datos BKG, ECR, NBI y STR, permanecen cercanas. Esto se observó en la Tabla 4.7 donde se comparan los valores para *n* y *d* asociados a los dos multclasificadores SVM

para cuatro clases, tanto para el conjunto de imágenes balanceado como desbalanceado.

- El porcentaje de aciertos arrojado por el multclasificador SVM para cuatro clases al procesar conjuntos de imágenes tanto balanceado como desbalanceado, está alrededor del 95% para los detalles A y V; si se toma para el entrenamiento un 20% de imágenes, en ambos casos. Esto se confirma en la Figura 4.16.

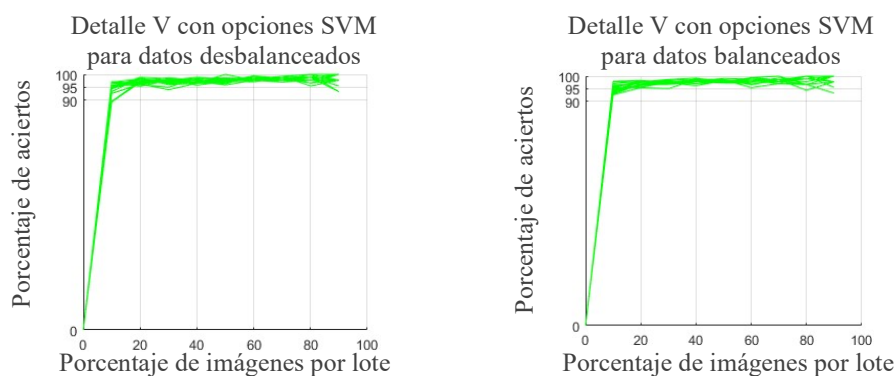


Figura 4.16.- Resultado obtenido al ejecutar 20 veces el Programa 2 para el multclasificador SVM para cuatro clases con datos balanceados y desbalanceados. Porcentaje de aciertos versus el porcentaje por lote de imágenes para el detalles V.

Prueba 4.- Clasificación por lote de cinco clases-balanceado

Este ensayo es parecido a la Prueba 2, y también se aplicarán consecuentemente los Programas 1 y 2. El Programa 1, debe explorar las opciones de los modelos SVM para ubicar los valores que permitan un alto porcentaje de acierto y un tiempo mínimo de procesamiento en la clasificación de un conjunto balanceado de 210 imágenes para las clases: BKG, ECR, NBI, STR y CUT. Cada clase contiene 42 muestras (véase la tabla 4.1). El Programa 2 debe explorar el conjunto de imágenes mínimo por lote necesario para entrenar el multclasificador SVM para cinco clases con imágenes balanceadas, intentando obtener altas tasas de aciertos, además de ubicar el detalle AHVD más apropiado.

Programa 1. Determinación de las opciones para cinco modelos SVM a partir de un conjunto de imágenes balanceado

Al usar el Programa 1 sobre el conjunto de imágenes balanceadas, pero con cinco clases, se consiguen las opciones c y g según las especificaciones de esta Prueba 4. El resultado obtenido se muestra en la Tabla 4.8, junto con el resultado obtenido en la Prueba 2 para su contraste. De esta manera se llega a la conclusión de que las opciones SVM que se habían calculado pueden mantenerse.

Tabla 4.8.- Comparación de las opciones encontradas con imágenes balanceadas y desbalanceadas para los modelos SVM, base de los multclasificadores SVM para cinco clases.

Opciones	n_1	n_2	n_3	n_4	n_5	d_1	d_2	d_3	d_4	d_5
Multclasificador SVM-desbalanceado	8	10	10	8	7	50	50	50	50	50
Multclasificador SVM-balanceado	8	10	10	8	7	50	50	50	50	50

Nota: los valores de d se expresan en millones

Programa 2. Verificación del porcentaje de acierto a partir de un conjunto de imágenes balanceado manteniendo las opciones SVM conseguidas en la Prueba 2 al entrenar un multclasificador SVM-cinco clases

En la Figura 4.17, se muestran los resultados obtenidos en la Prueba 4 al procesar un conjunto de imágenes con detalles AHVD balanceado, notándose que son muy similares a los resultados ya obtenidos en esta prueba 2, pero realizados con datos desbalanceados bajo condiciones similares.

Se puede observar que, el porcentaje de aciertos sigue estando por encima del 98 % para los detalles A y V. Esto permite fijar el conjunto mínimo de imágenes por lote para entrenar el multclasificador SVM para cinco clases a partir del 10% en ambos casos (Prueba 2 y 4).

Para confirmar estos resultados, sólo para el detalle V, se toman conjuntos de imágenes balanceados y desbalanceados y se procede nuevamente a ejecutar 20 veces el Programa 2. En la Figura 4.18 se muestran estos resultados, deduciéndose que pueden tomarse como mínimo el 10% de muestras para lograr un porcentaje de acierto por encima del 98%.

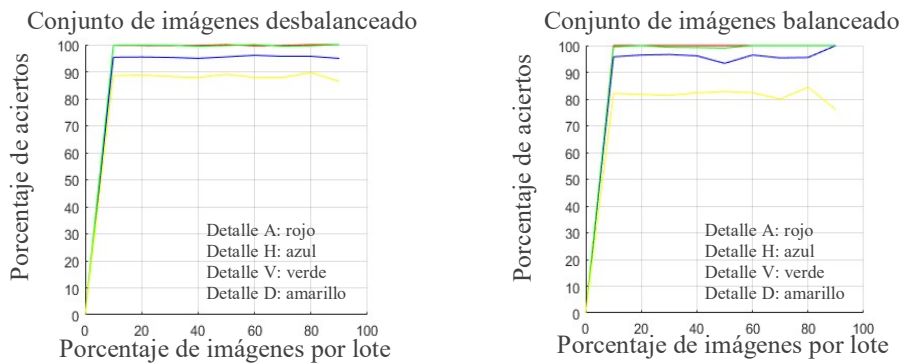


Figura 4.17.- Verificación del porcentaje de aciertos para diferentes detalles AHVD medidos a partir de los conjuntos de imágenes balanceados y desbalanceados al usar un multclasificador para cinco clases.

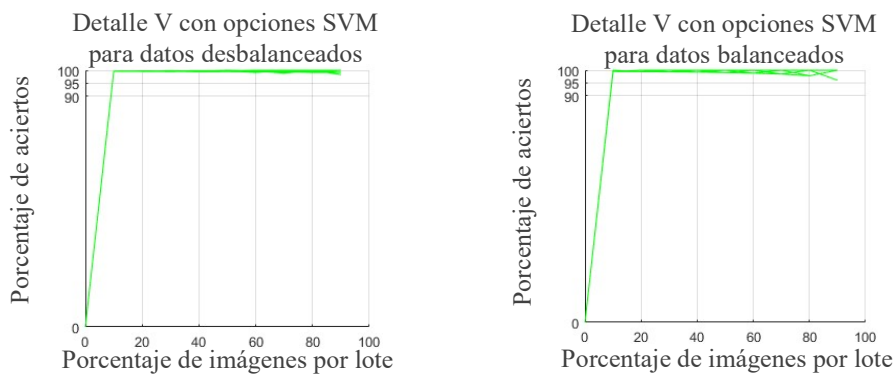


Figura 4.18.- Resultado obtenido al ejecutar 20 veces el Programa 2 para el multclasificador SVM para cinco clases con datos balanceados y desbalanceados.

Análisis de los resultados obtenidos en la Prueba 4

Al comparar la Prueba 4 con la Prueba 2, se concluye que:

- Las opciones c y g de los modelos SVM asociados a los datos BKG, ECR, NBI, STR y CUT pueden ser las mismas. Esto se observó en la Tabla 4.8, donde se comparan los valores para n y d asociados a los multclasificadores SVM para cinco clases, tanto para el conjunto de imágenes balanceado como desbalanceado.
- El porcentaje de aciertos arrojado por el multclasificador SVM para cinco clases al procesar conjuntos de imágenes tanto balanceado como

desbalanceado está por encima del 98% para los detalles A y V. Esto se observa en la Figura 4.17 y se confirma en la Figura 4.18.

- El conjunto mínimo de imágenes por lote para entrenar un multclasificador SVM para cinco clases puede establecerse a partir del 10% y con esto se asegura un alto porcentaje de aciertos, tanto en el detalle A como en el V.

Finalmente, señalar que todo el estudio sobre los modelos SVM ha servido para seleccionar las opciones de la SVM que se utilizarán en el multclasificador SVM de la sección 4.5. Además, se usaran como referencia en los multclasificadores PC e IPC que utilizan como modelo subyacente a la máquina de vectores soporte para determinar las medidas de no conformidad.

Hay que hacer notar que, en el caso de los porcentajes de imágenes obtenidos en los entrenamientos incrementales para realizar el entrenamiento por lote, los mismos se contrastaran con nuevas pruebas incrementales.

4.5 MULTICLASIFICADOR INCREMENTAL CON SVM

A partir de los modelos optimizados de SVM por las opciones SVM encontradas y mostradas en la Tabla 4.8 de la sección 4.4, se presenta en esta sección los resultados obtenidos con el multclasificador SVM donde se analizaran cuatro pruebas de entrenamiento y clasificación incremental, tomando conjuntos balanceados y desbalanceados de imágenes con cuatro y cinco clases, según lo indicado en el diagrama de agrupación de imágenes para entrenamiento propuesto en la Figura 4.1. También puede verse en la Tabla 4.1 el número de muestras dispuestas ante cada prueba.

El objetivo al realizar estos cuatro experimentos se centra en establecer, cual es el conjunto mínimo de muestras que puede ser tomado para entrenar un multclasificador SVM por lote y así conseguir altos porcentajes de aciertos en pruebas de clasificación de imágenes. En la Figura 4.19 se observa el diagrama de bloques general del programa implementado para realizar estos ensayos, apreciándose que la totalidad de las imágenes a observar en cada caso, son divididas en: un conjunto de entrenamiento inicial conformado por cinco muestras por clase de

imágenes para entrenar el *multiclasificador SVM*⁴⁶, y el resto se reserva para pruebas de clasificación.

Cada ensayo comienza tomando el conjunto de muestras para entrenar el multiclasificador SVM. Luego se toma una imagen del conjunto reservado para las pruebas y se clasifica, anotándose el resultado si es un acierto⁴⁷. Seguidamente, la imagen procesada se incorpora al conjunto de muestras inicial usado para el entrenamiento, y nuevamente se entrena el multiclasificador SVM. Se toma una segunda imagen de prueba y se clasifica, se anota su resultado y posteriormente se incorpora también al conjunto de muestras de entrenamiento.

Todo el proceso explicado se repite hasta alcanzar la última imagen de prueba. Es importante destacar que, la incorporación de la imagen predicha al conjunto de entrenamiento, se realiza observando su clase real (etiqueta real) y no su clase predicha (etiqueta predicha).

Con relación al programa implementado para realizar estos ensayos, se resaltan varias características de interés:

- El programa se basa en un conjunto de funciones previamente desarrolladas, donde destacan los códigos para entrenar y probar el multiclasificador SVM. Estas funciones trabajan independientemente y pueden ser llamadas desde un script principal donde se incorporan otros códigos para el pre-procesamiento de los conjuntos de imágenes a tratar.
- El programa se adapta para procesar conjuntos de imágenes con cuatro y cinco clases, definiendo así los multiclasificadores SVM para cuatro y cinco clases.
- El acceso a las imágenes para realizar las pruebas es independiente del tipo de conjunto a tomar, el cual puede ser balanceado o desbalanceado. Para un tipo de conjunto específico, también es posible indicar la cantidad de imágenes.

⁴⁶ Un multiclasificador SVM tiene como núcleo modelos SVM. Para el caso de las imágenes tratadas en esta Tesis, se debe entrenar un modelo SVM por cada clase BKG, ECR, NBI, STR y CUT. Recuérdese que la clase de imágenes CUT no es usada en un multiclasificador con cuatro clases. Sólo se usa en los multiclasificadores con cinco clases.

⁴⁷ Se define como “acierto” en la clasificación de imágenes, cuando una imagen que pertenece a una clase en particular, es predicha correctamente por un multiclasificador.

Todo esto se puede seleccionar en el programa que pre-procesa las imágenes, estableciendo en el script principal, si se tomará un conjunto de imágenes en particular, y si contará con cinco imágenes, seis,..., un porcentaje de la totalidad, etc.

- El programa es independiente del proceso a realizar sobre cualquier detalle AHVD. Puede procesar un detalle en particular o todos. Esto también se selecciona en el script principal.

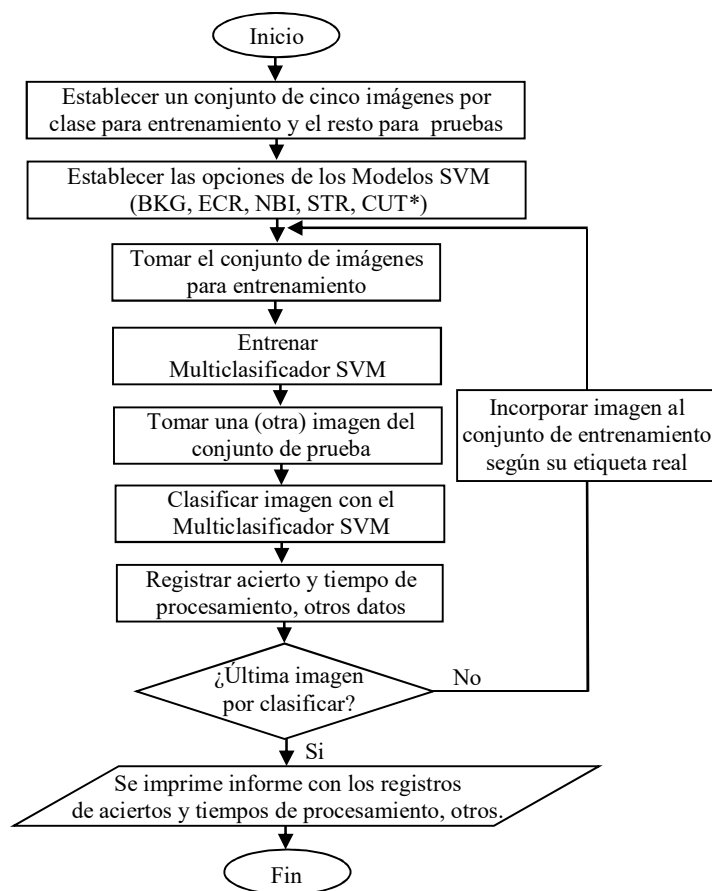


Figura 4.19.- Diagrama de bloques general de la implementación del programa que permite realizar el entrenamiento y la prueba incremental del multclasificador SVM.

- El programa depende de los parámetros (opciones SVM) obtenidos en la sección 4.4.

- El programa registra diferentes tipos de datos que se van generando como resultado del proceso de multclasificación ejecutado. Entre estos datos registrados se tienen: las tasas de aciertos en la clasificación de muestra, el tiempo de procesamiento por muestra clasificada, entre otros.
- El programa tiene incorporado segmentos de código especial para realizar diferentes contrastes según los datos generados, y de esta manera proporcionar un informe adecuado.

Finalmente se destacar que, las características de interés mencionadas con relación al programa implementado y mostrado en la Figura 4.19, son trasladadas al resto de los programas que manejan otros multclasificadores que serán presentados más adelante en esta Tesis.

Prueba 5.- Resultados obtenidos en la clasificación incremental para cuatro-cinco clases balanceado-desbalanceado

A continuación se presentan el desarrollo de cuatro ensayos donde se entrenó y probó el multclasificador SVM en forma incremental, para establecer el conjunto mínimo de imágenes que puede ser tomado para entrenar un multclasificador SVM por lote, y así conseguir altos porcentajes de aciertos en pruebas de clasificación. Como guía, se muestra en la Tabla 4.9 la totalidad de las imágenes usadas en cada caso: el conjunto inicial de entrenamiento al tomar cinco imágenes por cada clase y el conjunto reservado para pruebas de clasificación incrementales.

Tabla 4.9.- Conjuntos de imágenes iniciales para los entrenamientos y pruebas incrementales con el fin de establecer el conjunto mínimo de imágenes para el entrenamiento por lote de un multclasificador SVM.

Prueba 5 Multclasificador SVM	Conjunto total de imágenes	Conjunto inicial de entrenamiento incremental	Conjunto reservado para pruebas incrementales	Conjunto donde puede iniciar el entrenamiento por lote
4 clases desbalanceado	1107	20	1087	Alrededor de 300
5 clases desbalanceado	1149	25	1024	Alrededor de 400
4 clases balanceado	428	20	408	Alrededor de 100
5 clases balanceado	210	25	185	NA

En la Figura 4.20 se muestran los resultados gráficos obtenidos tras examinar cada uno de los conjuntos de imágenes reservados para las pruebas de clasificación incrementales, pudiéndose apreciar tanto la evolución que sufre el porcentaje de aciertos como el tiempo de procesamiento para cada conjunto.

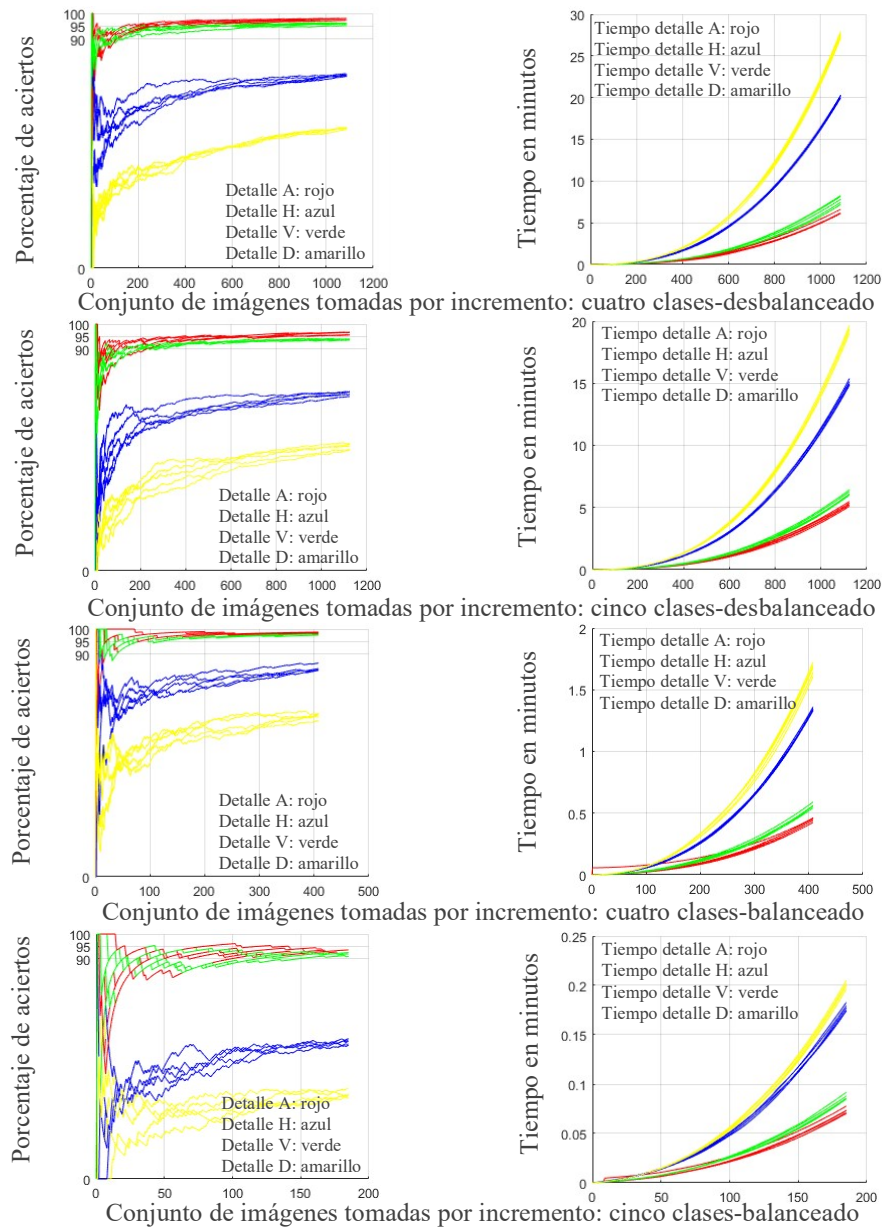


Figura 4.20.- Porcentaje de aciertos y tiempo de procesamiento obtenidos para el multclasificador SVM por incremento con los conjuntos iniciales de imágenes según la Tabla 4.9.

Si se realiza una inspección visual minuciosa, es posible ubicar el conjunto inicial de imágenes a usar en el entrenamiento por lote del multclasificador SVM. Esta información extraída de la Figura 4.20 se muestra al final de la Tabla 4.9 (columna 5) donde se observa que el multclasificador SVM incremental para cinco clases balanceado no converge a un alto valor en el porcentaje aciertos para ninguno de los detalles AHVD examinados, marcándose este hecho con las siglas NA (No Aplica). Para el resto de los experimentos, es evidente el alto porcentaje de acierto reflejado para los detalles A y V.

Prueba 6.- Contraste en la clasificación incremental para cuatro-cinco clases balanceado-desbalanceado para el detalle V

En la Figura 4.21 para las imágenes con detalle V, se observa que los conjuntos desbalanceados presentan un progresivo crecimiento en las tasas de acierto sin oscilaciones, ubicándose finalmente en el 95%. Para los conjuntos balanceados, se puede ver que sólo el de cuatro clases propicia un alto porcentaje de acierto pero con algunos saltos (oscilaciones). Con relación al de cinco clases, finalmente no logra superar el 95%.

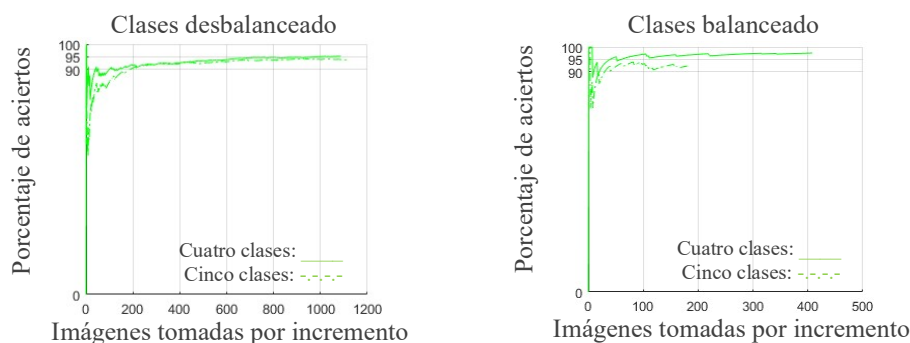


Figura 4.21.- Contraste del porcentaje de aciertos visto entre los cuatro ensayos desarrollados en la Prueba 5 para el detalle V.

Este resultado más que deberse al balanceo o no de los conjuntos de imágenes observados, se obtiene por el número de muestras utilizadas en el momento de hacer la prueba, ya que el multclasificador para cinco clases balanceadas sólo utiliza 210 imágenes, no existiendo las suficientes muestras como para lograr que los modelos

SVM presentes funcionen correctamente, no pudiendo reflejar altos porcentajes de aciertos.

Análisis de los resultados obtenidos en la Prueba 5

- El multclasificador SVM tiende a estabilizar mejor el porcentaje de acierto al procesar conjuntos de imágenes desbalanceados. Esto puede verse en la Figura 4.21 donde se muestra el procesamiento hecho sobre conjuntos balanceados y desbalanceados para cuatro y cinco clases.
- La clasificación con entrenamiento incremental desbalanceado revela que el número de imágenes mínimo para alcanzar altas tasas de aciertos se encuentra por encima de las 600 muestras, pero su tendencia a crecer y estabilizarse aparece por debajo de las 300 observaciones para cuatro clases y alrededor de la muestra 400 para cinco clases, según se observa en la Figura 4.20.
- Si se escogen aleatoriamente diferentes conjuntos de imágenes para realizar el entrenamiento y pruebas de clasificación por incremento, se nota en las diferentes gráficas trazadas para el porcentaje de acierto, una etapa transitoria diferente en cada caso, sin embargo, al final convergen a valores comunes que suelen ser muy cercanos. Esto se demostró al repetir muchas veces cada ensayo asociado, por ello se notan varias curvas del mismo color (según el detalle AHVD) para cada prueba observada en la Figura 4.20, sólo difiriendo un poco en aquellos experimentos incrementales asociados a los conjuntos de imágenes con cuatro clases balanceadas. Lo dicho, deja ver la necesidad de repetir un ensayo más de una vez para poder fijar cualquier valor y/o emitir cualquier valoración.
- Según todo lo anterior, se puede fijar un 27%, 35% y 24% de imágenes para realizar el entrenamiento por lote en los multclasificadores SVM desbalanceados con cuatro y cinco clases, y balanceado con cuatro clases respectivamente. Para los experimentos con cinco clases balanceados no se nota un estable porcentaje de acierto según lo visto en la Figura 4.20 y contrastado en la Figura 4.21.

4.6 MULTICLASIFICADOR POR LOTE CON SVM

Los ensayos presentados a continuación se refieren al entrenamiento y pruebas de clasificación por lote, que a diferencia de los ensayos incrementales no repiten el entrenamiento por cada muestra clasificada, ya que sólo se entrena el multclasificador SVM una única vez y luego realizan la clasificación sobre todo el conjunto de prueba reservado.

En la Figura 4.22 puede verse el diagrama de bloques general del programa implementado para realizar estos ensayos, donde se toma como lote de muestras para entrenamiento, el porcentaje de imágenes fijado en las pruebas de clasificación incremental presentadas en la sección 4.5.

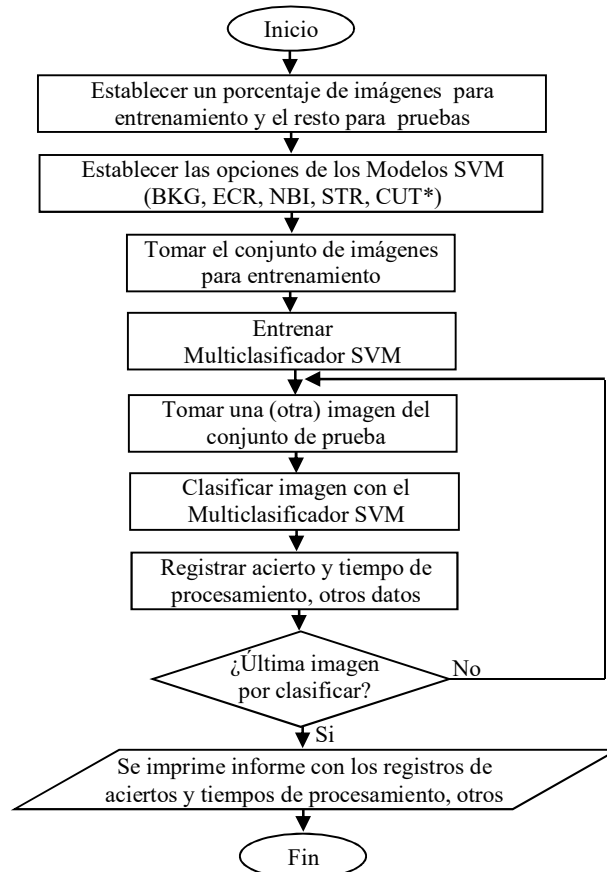


Figura 4.22.- Diagrama de bloques general de la implementación del programa que permite realizar entrenamiento y prueba por lote del Multclasificador SVM.

El objetivo al realizar una prueba por lote se centra en comprobar, si el porcentaje de imágenes fijado por las pruebas incrementales propicia una alta y clara convergencia del porcentaje de aciertos. En otras palabras, con estos ensayos se exploran detalladamente, la evolución que va sufriendo el porcentaje de aciertos por cada imagen examinada del conjunto reservado para pruebas de clasificación por lote, con un mismo conjunto de muestras fijado en los entrenamientos y pruebas incrementales.

Prueba 7.- Resultados obtenidos en la clasificación por lote para cuatro-cinco clases balanceado-desbalanceado

En esta prueba por lote, también se realizan cuatro ensayos al entrenar y probar el Multiclasificador SVM tal y como se muestra en el diagrama de agrupación de imágenes mostrado en la Figura 4.1. De la totalidad de las imágenes dispuestas para la Prueba 7 se toman el 27%, 35% y el 24% para formar los conjuntos de entrenamiento, según lo sugerido por los experimentos incrementales presentados en la sección anterior y cuyos resultados se muestran nuevamente en la columna 3 de la Tabla 4.10. Cabe recordar que, en el anterior experimento incremental no se logró fijar ningún porcentaje para entrenar el multiclasificador por lote para cinco clases balanceado.

Según los resultados obtenidos en los experimentos incrementales (véase columna 3 de la Tabla 4.10), los porcentajes fijados para entrenar un multiclasificador SVM por lote, deberían ser suficientes para propiciar altas tasas de aciertos al procesar un conjunto de imágenes. Sin embargo, como puede verse en la columna 4 de la propia Tabla 4.10, se debe esperar a examinar antes un número considerable de muestras dentro del propio conjunto reservado para las pruebas de clasificación por lote, de tal forma que exista una convergencia clara a altas tasas de aciertos.

Nótese en los multiclasificadores SVM desbalanceados para 4 y 5 clases que deben procesar al menos 50 y 200 muestras respectivamente, antes de alcanzar un 93% en sus tasas de aciertos, para poder decir con certeza que se está cometiendo un error del 7% en la predicción de imágenes. Esto obliga en consecuencia, a realizar un

ajuste manual para algunos multclasificadores y de esta manera lograr altos porcentajes de aciertos de forma inmediata al implementarlos.

Tabla 4.10.- Cantidades de imágenes dispuestas para los entrenamientos y pruebas por lote con el multclasificador SVM y su convergencia hacia altas tasas de aciertos.

Prueba 7 Multclasificador SVM	Conjunto total de imágenes	Conjunto de entrenamiento fijado por pruebas incrementales %	Converge alrededor de la imagen	Ajuste manual del conjunto de entrenamiento %	Converge alrededor de la imagen
4 clases desbalanceado	1107	27	50	27	50
5 clases desbalanceado	1149	35	200	45	50
4 clases balanceado	428	24	150	27	50
5 clases balanceado	210	NA	NA	NA	NA

En la columna 5 de la Tabla 4.10, se puede ver el ajuste manual hecho sobre el conjunto de entrenamiento para los multclasificadores SVM de 5 y 4 clases desbalanceado y balanceado, al fijarse un nuevo porcentaje en 45% (antes estaba en 35) y 27% (antes estaba en 24) respectivamente. Además en la columna 6 de la misma tabla, se observa las actualizaciones del número de imágenes aproximado que deben ahora esperarse antes de que los multclasificadores SVM ajustados converjan a un 93% en su tasa de aciertos respectivamente.

Para comprobar estos datos, se deja en la Figura 4.23 los resultados gráficos obtenidos después de examinar cada conjunto de imágenes reservado para las pruebas de clasificación con los porcentajes señalados en la columna 5 de la Tabla 4.10. Nótese como los detalles A y B propician altos porcentajes de aciertos, siendo más ventajosos los conjuntos de muestras desbalanceados al no mostrar mayores oscilaciones. Observe además, que el multclasificador SVM con cinco clases balanceado no permite una convergencia clara pudiendo deberse a la falta de muestras para entrenar tal multclasificador.

Para finalizar, se comenta un aspecto visto en esta parte del análisis realizado y qué será estudiado más adelante en este capítulo: la importancia de ajustar el número de imágenes fijado por los entrenamientos incrementales para que un

multiclasificador por lote pueda hacer predicciones coherentemente y no reportar porcentajes de aciertos con tantas oscilaciones alrededor de la convergencia.

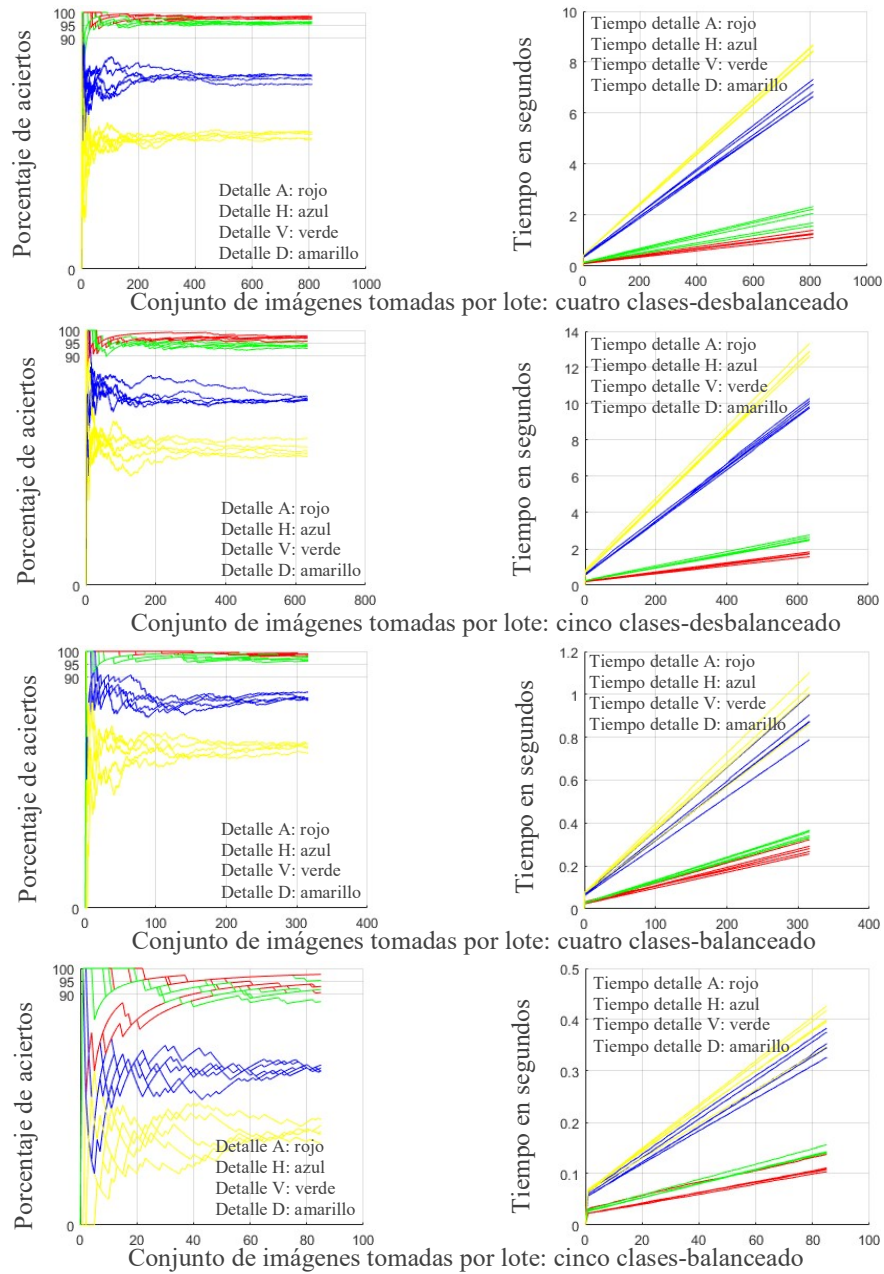


Figura 4.23.- Porcentaje de aciertos y tiempo de procesamiento obtenidos para el multiclasificador SVM por lote, con los conjuntos de imágenes según la Tabla 4.10.

Prueba 8.- Contraste en la clasificación por lote para conjuntos con detalle V para cuatro-cinco clases balanceado-desbalanceado

En la Figura 4.24 se muestra el contraste de los resultados obtenidos en la Prueba 7 pero sólo para el detalle V. Obsérvese las altas tasas de aciertos conseguidas, sin mayores oscilaciones, por el multclasificador SVM cuando se procesan los conjuntos desbalanceados para cuatro y cinco clases. Sin embargo, para los conjuntos balanceados, el único que supera las altas tasas de acierto sin oscilaciones es el de cuatro clases. El de cinco clases, pese a que alcanza alrededor de un 90% mantiene oscilaciones. En la Figura también se contrastan los tiempos consumidos para procesar los conjuntos de imágenes según la Tabla 4.10.

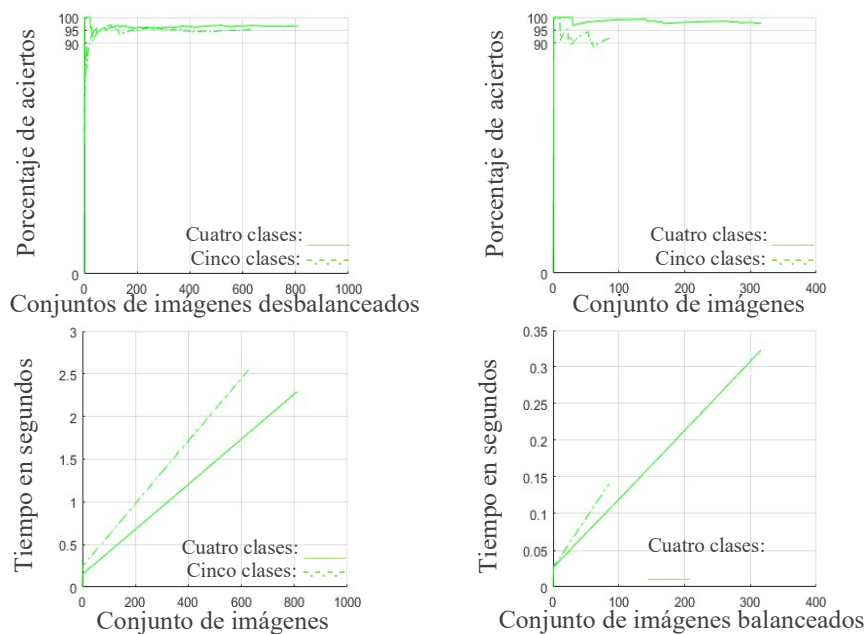


Figura 4.24.- Contraste del porcentaje de aciertos y tiempo de procesamiento visto entre los cuatro ensayos desarrollados en la Prueba 7 para el detalle V. Nota: el eje de las abscisas muestra el progreso por lote de las imágenes tomadas.

Análisis de los resultados obtenidos en la Prueba 7 y 8

- Es claro y notorio que los multclasificadores SVM entrenados con detalles A y V son los que propician altas tasas de aciertos. Para los detalles H y D no existen altas tasas de acierto (véase la Figura 4.23).

- Los multclasificadores SVM tienden a estabilizar mejor el porcentaje de acierto al procesar conjuntos de imágenes desbalanceados para cuatro y cinco clases, ubicándose alrededor del 95%. Específicamente para el detalle V, se nota que lo afirmado también se cumple (véase de nuevo la Figura 4.24).
- De los conjuntos balanceados con detalle V, el multclasificador SVM entrenado con cuatro clases presenta una mejor y alta tasa de acierto sin oscilaciones alrededor de la convergencia, superando el 95%. Esto no sucede para el de cinco clases, el cual tiende a oscilar mucho alrededor del 90%.
- Al comparar las Figuras 4.23 y 4.24, se observa la clara necesidad de repetir varias veces cualquier ensayo. Esto se debe al sorteo aleatorio hecho en momento de escoger las muestras para los conjuntos de entrenamiento y pruebas de clasificación. Especialmente, se tiene la necesidad de repetir las pruebas para poder fijar el punto donde comienza a converger el porcentaje de acierto, ya que estas curvas presentan una parte transitoria que no sigue una trayectoria determinada, aunque después de su estabilización converge a valores parecidos.

Prueba 9.- Contraste entre los resultados obtenidos en la Prueba 5 y 7 para conjuntos desbalanceados con detalle V

En este ensayo se contrastan los resultados obtenidos entre las pruebas 5 y 7 sólo para conjuntos de imágenes desbalanceados con detalle V y permite demostrar que los experimentos incrementales son un buen punto de partida para fijar un porcentaje o conjunto de imágenes para realizar el entrenamiento de un multclasificador SVM por lote y posteriormente poder realizar las pruebas de clasificación correspondientes con la certeza de lograr altas tasas de aciertos.

Además el ensayo, también pretende demostrar que las pruebas incrementales no siempre permiten este cometido comentado líneas atrás, ya que al tantear el punto escogido por incremento en los entrenamientos y pruebas por lote de los multclasificadores SVM, se pudieran estar alcanzado altas tasas de aciertos pero la

convergencia a tales valores, debe esperar por la predicción de un número considerable de muestras.

Esto se comentó en el análisis basado en la Tabla 4.10, pero ahora se reafirma al contrastar los resultados que se muestran en la Figura 4.25. Allí se puede apreciar entre otras cosas, las curvas relacionadas con los porcentajes de aciertos para los experimentos por incremento y por lote con el conjunto de imágenes desbalanceado para cinco clase con detalle V (curvas en color verde y amarilla respectivamente).

Puntualizando, se observa en la Figura 4.25 (a) un alto porcentaje de acierto alrededor del 93% en la muestra 400, señalando esto un 35% de imágenes para realizar el entrenamiento y pruebas de clasificación por lote. Sin embargo, en la misma Figura 4.25 (a) se confirma que al entrenar el multclasificador SVM con tal porcentaje de imágenes escogido (35%), habría que esperar por la predicción de más de 200 muestras en las pruebas de clasificación (véase la curva en color amarillo en la Figura 4.25 (a)) para conseguir un porcentaje de acierto del 93%.

Si se cotejan, las mismas curvas (verde y amarillo) en la Figura 4.25 (b), en especial la curva en amarillo obtenida en este caso para un ajuste del 45% en el conjunto de imágenes para realizar el entrenamiento y pruebas de clasificación por lote del mismo multclasificador SVM, se notará que mucho antes de la muestra 50 ya se perciben altos porcentajes de acierto, alrededor del 93%.

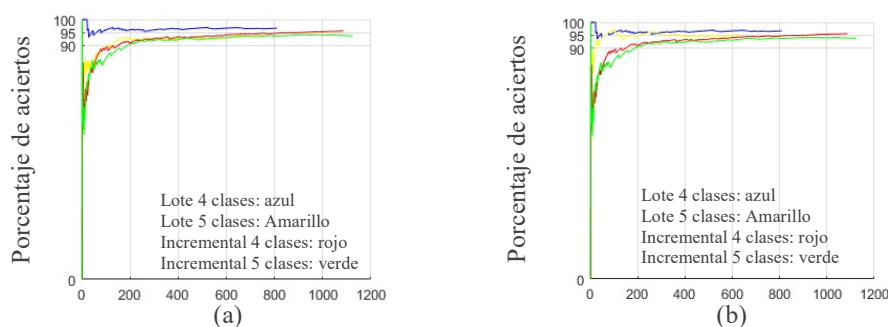


Figura 4.25.- Contraste entre los resultados obtenidos en la Prueba 5 y 7 para conjuntos desbalanceados con detalle V. Porcentaje de acierto en curva amarilla obtenida con un conjunto de imágenes para entrenamiento por lote: (a) Un 35%. (b) Un 45%.

Para finalizar con este análisis, nótese en la misma Figura 4.25, que el porcentaje de imágenes fijado por la prueba incremental del multclasificador SVM para cuatro clases desbalanceado (curva en color rojo), apunta alrededor de las 300 imágenes (como un 27%) siendo evidente que no se requiere de ningún ajuste para tener predicciones muy por encima del 93% (curva en color azul), mucho antes de la muestra 50.

4.7 MULTICLASIFICADOR CON PREDICCIÓN CONFORMAL Y PREDICCIÓN CONFORMAL INDUCTIVA

Hasta el momento se ha analizado el multclasificador SVM y se ha comprobado que tiene capacidad para obtener altas tasas de aciertos en la clasificación de imágenes. Sin embargo, este multclasificador construido no aporta medidas de fiabilidad que indiquen, cuanto de fiable pueden ser las clasificaciones obtenidas.

También fue analizado en teoría y con ejemplos prácticos, en las secciones 3.7 y 3.8, todo el enfoque necesario para construir multclasificadores con la debida implementación de medidas de fiabilidad (confianza y credibilidad). En especial, se analizaron los multclasificadores vistos en las secciones 3.7.3.1 y 3.8, los cuales se basan en la predicción conformal (PC), y su variante, la predicción conformal inductiva (IPC).

En las siguientes secciones se presentan algunas implementaciones para analizar los resultados obtenidos con la predicción conformal y la predicción conformal inductiva cuando se aplican a las imágenes obtenidas en el diagnóstico de esparcimiento Thomson.

4.7.1 SVM como método subyacente y la aproximación modificada uno versus el resto

Los multclasificadores basados en la predicción conformal (se les denominará en adelante: multclasificador PC e IPC) requieren para su implementación de un método subyacente, que en el caso particular de esta

investigación, se ha utilizado la SVM conjuntamente con la aproximación modificada uno versus el resto para generar las medidas de no conformidad, que al ser computadas se resumen en un valor informativo definido como p_valor , proporcionando finalmente la predicción y las medidas de fiabilidad al examinar datos multiclase (véase la sección 3.7.3.1).

Para el caso de la SVM como método subyacente, requiere a su vez del ajuste de sus parámetros (opciones SVM), pudiendo esta tarea ser compleja por los diferentes modelos SVM utilizados según el problema multiclase a resolver. Esta complejidad fue estudiada en secciones anteriores en este trabajo. Adicionalmente, la aproximación modificada uno versus el resto requiere del ajuste del parámetro λ (véase la expresión 3.42).

Como el ajuste de las opciones SVM puede ser un proceso muy complejo pero requerido para que la SVM trabaje como método subyacente en un predictor conformal, se tomaron inicialmente los parámetros establecidos para el multclasificador SVM que fueron vistos en la Tabla 4.8 de la sección 4.4, demostrándose en las pruebas preliminares hechas tanto con la PC como con la IPC, que de haberse mantenido tales valores no hubiesen permitido que el multclasificador PC alcance altas tasas de acierto en la clasificación de imágenes.

Este problema detectado que será presentado en la sección dedicada a la Prueba 10, fue corregido realizando un ajuste sobre los modelos SVM asociados al método subyacente que maneja la predicción conformal y sus resultados serán presentados más adelante. Cabe hacer notar, que este problema detectado no aplica para el multclasificador IPC, tal y como se demostró en la Prueba 11.

Para finalizar, es importante entender que el ajuste hecho sobre las opciones de los modelos SVM usados por el multclasificador PC o IPC no es independiente, ya que se debe realizar en sintonía con el parámetro λ usado en la aproximación modificada uno versus el resto, que también es usada en el multclasificador PC o IPC. Esta sintonía es necesaria para establecer los valores de las opciones SVM y el parámetro λ , y así obtener altas tasas de aciertos y medidas de fiabilidad razonables en un multclasificador PC o IPC.

Prueba 10.- Ensayo con la aproximación modificada uno vs resto, ajuste de las opciones SVM y selección del parámetro λ

Para el ajuste de las opciones SVM y del parámetro λ , se diseñaron las pruebas 10 y 11 apoyadas con la implementación del programa que se muestra en el diagrama de bloques de la Figura 4.26.

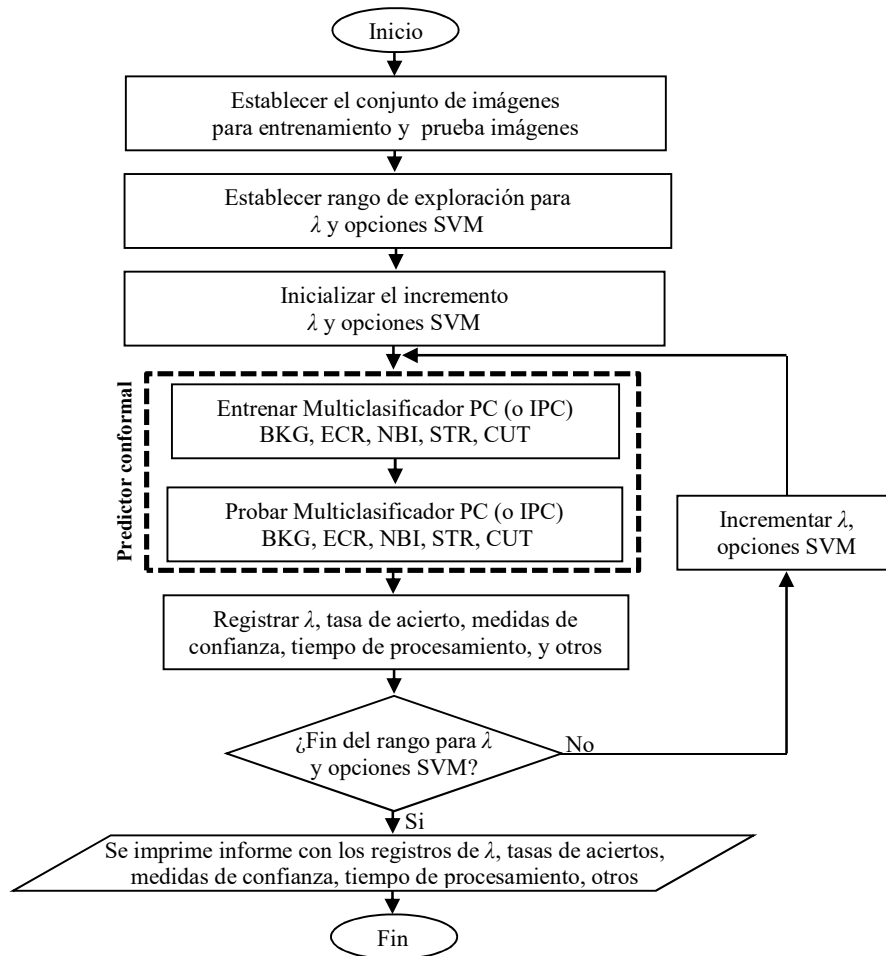


Figura 4.26.- Diagrama de bloques general de la implementación del programa que explora los valores λ y las opciones SVM para un multclasificador PC o IPC.

La Prueba 10 tenía como objetivo, explorar el parámetro λ en sintonía con las opciones SVM necesarias para que los multclasificadores PC e IPC arrojen altas tasas de aciertos y además den medidas de confianza y credibilidad razonables. Tal prueba consumía mucho tiempo durante su ejecución, ya que debía explorar un espacio de valores tanto para λ como para las opciones SVM; apuntando que este

proceso de búsqueda es individual para cada modelo SVM a implementar en un multclasificador PC o IPC.

Por esta razón no se observarán las diferentes variaciones en las siguientes tablas informativas de los valores explorados para las opciones SVM, concentrando el estudio alrededor de los valores de tales opciones que propiciaron los mejores resultados. Al final de este primer estudio, las conclusiones para el multclasificador PC fueron:

- Un cambio no tan significativo en el valor para la opción c del modelo STR que aparece en la Tabla 4.8, escrita como n_4 . Dicho valor pasa de ocho ($n_4=8 \rightarrow c=256$) a nueve ($n_4=9 \rightarrow c=512$).
- Las opciones g cambian rotundamente al pasar de 50.000.000 a 909.000.000, después de hacer una búsqueda exhaustiva.

Con el ajuste comentado anteriormente se obtienen nuevas opciones SVM que propician altas tasas de aciertos con el multclasificador PC, tal y como se observa en la columna 3 (%AC) del informe de la Tabla 4.11. También se pueden apreciar los diferentes valores de λ (columna 1: $IN\lambda$) que han sido variados a una razón de cambio de 0.1. Nótese además, que los altos porcentajes fueron obtenidos como se muestra en la columna 2 (MPR) con un 80% (924 muestras) de las imágenes arbitrariamente tomadas para las pruebas de clasificación. El otro 20% se usó para entrenar el multclasificador PC (e IPC) con las nuevas opciones SVM ajustadas.

Obsérvese que los resultados comentados se contrastaron con los obtenidos al usar las antiguas opciones SVM. Para ello sólo inspeccione la columna 7 (%AC) y aproveche para que vea los tiempos de procesamiento consumidos por ambas opciones SVM establecidas en el multclasificador PC, donde es notorio que el ajuste mejora el cómputo por casi un minuto.

En cuanto a las conclusiones para el multclasificador IPC por el estudio realizado en la Prueba 10, la exploración de las opciones SVM conduce a mantener los antiguos valores usados por el multclasificador SVM, ya que los mismos comparativamente propician mejores tasas de aciertos que las recientes opciones exploradas. Esto se demuestra al contrastar las columnas 5 y 9 de la Tabla 4.11,

donde también se observa una reducción del tiempo de procesamiento (véase las columnas 6 y 10).

Tabla 4.11.- Porcentaje de aciertos y tiempo de procesamiento conseguido mediante la implementación del programa de la Figura 4.19 para un conjunto de imágenes tomando el 20% de muestra para el entrenamiento con las nuevas y antiguas opciones SVM al variar λ .

		Nuevas Opciones_SVM				Antiguas Opciones_SVM			
		PC		IPC		PC		IPC	
IN λ	MPR	%AC	TIE(min)	%AC	TIE(seg)	%AC	TIE(min)	%AC	TIE(seg)
0.1	924	93.07	4.28	93.18	1.86	74.24	3.42	95.56	1.67
0.2	924	93.18	4.26	93.07	1.92	83.55	3.40	94.81	1.64
0.3	924	93.94	4.29	92.32	1.99	85.82	3.42	94.37	1.68
0.4	924	94.26	4.29	93.29	1.93	86.26	3.41	94.26	1.66
0.5	924	93.83	4.29	93.51	1.84	85.39	3.39	94.81	1.67
0.6	924	93.72	4.31	93.61	1.94	85.82	3.40	95.13	1.65
0.7	924	93.94	4.30	94.05	1.84	85.71	3.42	95.67	1.69
0.8	924	94.05	4.27	94.05	1.85	86.15	3.41	95.78	1.70
0.9	924	94.48	4.30	94.48	2.00	86.26	3.46	95.78	1.67
1.0	924	94.26	4.32	94.37	1.87	84.20	3.44	95.78	1.68

IN λ : Incremento de λ
MPR: Muestras Procesadas
%AC: Porcentaje de Aciertos
TIE(min): tiempo de Procesamiento en minutos
TIE(seg): tiempo de Procesamiento en segundos

Prueba 11. Ensayo y contraste con los valores ajustados en las opciones SVM versus antiguas opciones SVM para los diferentes parámetros λ al usar otros lotes de imágenes

La Prueba 11 tenía como objetivo, demostrar que el ajuste conseguido para las opciones SVM con un 20% de imágenes para entrenamiento, era suficiente para replicarlo con otros porcentajes de imágenes. Aprovechando al mismo tiempo para ver el efecto negativo⁴⁸ que introducen las antiguas opciones SVM sobre los altos

⁴⁸ Se recuerda que las antiguas opciones SVM, se determinaron por las pruebas realizadas usando el multclasificador SVM en la sección 4.4 y probadas dando buenos resultados en los ensayos presentados en las secciones 4.5 y 4.6. Sin embargo, tales opciones SVM al ser establecidas en el multclasificador PC, no permitieron predicciones con altas tasas de acierto y de ahí la justificación de su ajuste presentado en la Prueba 10.

porcentaje de acierto sobre el mismo multclasificador PC. Esto no ocurre para el multclasificador IPC.

Con esto en mente, se realizaron otros entrenamientos tomando nuevos conjuntos de muestras por lote entre un 20% a un 80% con una razón de cambio del 10% a la vez. El resto del porcentaje en cada caso quedó para realizar las pruebas de clasificación. Los resultados al aplicar las nuevas opciones SVM se muestran en las Tablas 4.12 y 4.13 para los multclasificadores PC e IPC respectivamente. Para el contraste con las antiguas opciones SVM, se muestran los resultados para los mismos multclasificadores PC e IPC en las Tablas 4.14 y 4.15 respectivamente.

Se indica que, la Tabla 4.11 como las Tablas desde la 4.12 a la 4.15, fueron generadas con la misma implementación del programa visto en la Figura 4.26. Aunque para el caso de los datos arrojados en esta segunda prueba, se incluyó un bucle “for” adicional que controló los diferentes lotes de imágenes mencionados, registrándose a la vez los porcentajes de aciertos asociados junto con las diferentes medidas de confianza acumulada, de credibilidad acumulada y de confianza por la credibilidad acumulada.

Análisis de los resultados obtenidos en las Pruebas 10 y 11

Habiendo ejecutado los experimentos previstos en las Pruebas 10 y 11, y comparando las Tablas 4.11, 4.12, 4.13, 4.14 y 4.15, se observan las siguientes conclusiones:

- Las antiguas opciones SVM usadas en el multclasificador SVM no pueden mantenerse intactas para los diferentes modelos SVM embebidos en el multclasificador PC. Esto se demuestra al comparar los porcentajes de aciertos ofrecidos por los resultados mostrados en la Tabla 11. También se pueden comparar las Tablas 4.12 y 4.14.
- Con relación al multclasificador IPC, se nota que las opciones SVM ajustadas (nuevas opciones) propician altas tasas de aciertos, pero tal predictor funciona mejor con las antiguas opciones SVM. Esto se observa al contrastar las Tablas 4.13 y 4.15.

- Es evidente que al variar λ , se nota una influencia sobre los porcentajes de aciertos. Esto mismo ocurre para las opciones SVM como se ha podido demostrar.

Tabla 4.12.- Informes arrojado por la implementación del programa de la Figura 4.19 para el multclasificador PC con las nuevas opciones SVM.

Confianza acumulada														
λ	20% CON	30% CON	40% CON	50% CON	60% CON	70% CON	80% Con							
0.10	93.05	0.71	93.55	0.72	93.93	0.71	93.92	0.70	93.93	0.72	95.40	0.71	92.67	0.70
0.20	93.81	0.76	93.42	0.77	94.65	0.76	94.27	0.75	94.14	0.77	95.69	0.75	94.40	0.75
0.30	94.03	0.81	94.04	0.81	95.23	0.79	95.14	0.79	95.23	0.81	96.55	0.79	95.69	0.79
0.40	94.46	0.85	93.92	0.85	95.52	0.83	94.97	0.83	95.01	0.85	95.98	0.83	95.26	0.83
0.50	94.03	0.89	93.67	0.90	95.23	0.88	94.97	0.88	94.79	0.90	96.55	0.87	96.55	0.88
0.60	94.46	0.93	93.92	0.94	95.52	0.92	95.49	0.93	95.01	0.94	96.55	0.92	96.98	0.93
0.70	94.35	0.96	94.17	0.96	95.52	0.96	96.35	0.96	95.66	0.97	96.26	0.96	96.55	0.96
0.80	94.68	0.97	94.29	0.98	95.81	0.98	96.35	0.98	96.10	0.99	96.55	0.99	96.55	0.98
0.90	95.44	0.97	94.42	0.98	95.95	0.98	96.70	0.98	96.31	0.99	97.13	0.99	96.55	0.99
1.00	95.55	0.98	94.54	0.99	95.95	0.98	96.88	0.99	95.88	0.99	97.13	0.99	95.69	0.99
Credibilidad Acumulada														
λ	20% CRE	30% CRE	40% CRE	50% CRE	60% CRE	70% CRE	80% CRE							
0.10	93.05	0.53	93.55	0.52	93.93	0.53	93.92	0.53	93.93	0.50	95.40	0.54	92.67	0.54
0.20	93.81	0.51	93.42	0.50	94.65	0.51	94.27	0.51	94.14	0.48	95.69	0.52	94.40	0.51
0.30	94.03	0.49	94.04	0.48	95.23	0.49	95.14	0.49	95.23	0.47	96.55	0.50	95.69	0.50
0.40	94.46	0.48	93.92	0.47	95.52	0.47	94.97	0.47	95.01	0.46	95.98	0.48	95.26	0.48
0.50	94.03	0.46	93.67	0.45	95.23	0.46	94.97	0.46	94.79	0.44	96.55	0.47	96.55	0.46
0.60	94.46	0.45	93.92	0.44	95.52	0.44	95.49	0.45	95.01	0.43	96.55	0.46	96.98	0.45
0.70	94.35	0.44	94.17	0.43	95.52	0.43	96.35	0.44	95.66	0.42	96.26	0.45	96.55	0.44
0.80	94.68	0.43	94.29	0.43	95.81	0.42	96.35	0.43	96.10	0.42	96.55	0.44	96.55	0.43
0.90	95.44	0.43	94.42	0.42	95.95	0.42	96.70	0.42	96.31	0.41	97.13	0.43	96.55	0.43
1.00	95.55	0.42	94.54	0.41	95.95	0.41	96.88	0.41	95.88	0.40	97.13	0.43	95.69	0.42
Credibilidad por confianza acumulada														
λ	20% CPC	30% CPC	40% CPC	50% CPC	60% CPC	70% CPC	80% CPC							
0.10	93.05	0.30	93.55	0.30	93.93	0.30	93.92	0.29	93.93	0.29	95.40	0.30	92.67	0.30
0.20	93.81	0.32	93.42	0.32	94.65	0.31	94.27	0.30	94.14	0.30	95.69	0.31	94.40	0.31
0.30	94.03	0.33	94.04	0.33	95.23	0.32	95.14	0.31	95.23	0.31	96.55	0.32	95.69	0.32
0.40	94.46	0.35	93.92	0.34	95.52	0.33	94.97	0.33	95.01	0.33	95.98	0.33	95.26	0.33
0.50	94.03	0.37	93.67	0.37	95.23	0.35	94.97	0.36	94.79	0.36	96.55	0.36	96.55	0.36
0.60	94.46	0.39	93.92	0.39	95.52	0.38	95.49	0.38	95.01	0.38	96.55	0.38	96.98	0.39
0.70	94.35	0.41	94.17	0.41	95.52	0.40	96.35	0.41	95.66	0.40	96.26	0.41	96.55	0.40
0.80	94.68	0.41	94.29	0.41	95.81	0.41	96.35	0.42	96.10	0.41	96.55	0.43	96.55	0.42
0.90	95.44	0.41	94.42	0.41	95.95	0.40	96.70	0.41	96.31	0.40	97.13	0.43	96.55	0.42
1.00	95.55	0.41	94.54	0.41	95.95	0.40	96.88	0.41	95.88	0.40	97.13	0.42	95.69	0.41

- Se nota un efecto contrario sobre los porcentajes de aciertos presentados por la PC e IPC. En el primer predictor, las tasas aumentan al pasar de las antiguas a las nuevas opciones SVM. Lo contrario ocurre para el predictor IPC.

- La exploración de las opciones SVM para lograr su ajuste sustancial y conseguir altas tasas de aciertos con los multclasificadores PC e IPC no puede realizarse de manera independiente puesto que requiere de la inclusión en la búsqueda del parámetro λ . De esta manera se consigue un contraste de resultados donde se adopta algún criterio de selección. NOTA: Se llega a esta conclusión porque los nuevos valores conseguidos para las opciones SVM y probadas en los multclasificadores PC e IPC, ya se habían usado con anterioridad en los multclasificadores SVM pero arrojaban tasas de aciertos pobres.

Tabla 4.13.- Informes arrojado por la implementación del programa de la Figura 4.19 para el multclasificador IPC con las nuevas opciones SVM.

Confianza acumulada														
λ	20% CON	30% CON	40% CON	50% CON	60% CON	70% CON	80% Con							
0.10	91.56	0.84	93.49	0.79	92.34	0.77	93.47	0.78	94.47	0.78	94.35	0.77	93.70	0.77
0.20	92.42	0.92	93.24	0.87	91.91	0.83	92.78	0.83	94.04	0.83	95.76	0.82	94.96	0.82
0.30	92.75	0.96	93.24	0.92	91.62	0.88	91.75	0.87	93.83	0.87	95.48	0.86	94.96	0.86
0.40	93.07	0.97	93.86	0.95	91.62	0.92	91.58	0.92	94.26	0.92	94.35	0.91	94.96	0.91
0.50	93.18	0.98	94.23	0.97	92.20	0.95	93.13	0.95	94.68	0.96	94.63	0.95	94.96	0.95
0.60	92.64	0.98	94.47	0.98	92.63	0.98	92.78	0.98	95.53	0.99	94.35	0.97	94.54	0.97
0.70	92.32	0.98	94.47	0.98	92.34	0.99	93.47	0.99	95.11	0.99	93.79	0.99	94.54	0.99
0.80	92.64	0.98	94.72	0.99	93.06	0.99	93.30	0.99	95.53	0.99	94.35	0.99	94.96	0.99
0.90	92.97	0.98	94.59	0.99	92.92	0.99	93.64	0.99	96.38	0.99	94.35	0.99	94.96	0.99
1.00	93.07	0.99	94.47	0.99	93.50	0.99	93.47	0.99	96.38	0.99	95.48	0.99	94.96	0.99
Credibilidad Acumulada														
λ	20% CRE	30% CRE	40% CRE	50% CRE	60% CRE	70% CRE	80% CRE							
0.10	91.56	0.56	93.49	0.55	92.34	0.49	93.47	0.48	94.47	0.52	94.35	0.51	93.70	0.49
0.20	92.42	0.50	93.24	0.51	91.91	0.48	92.78	0.45	94.04	0.49	95.76	0.48	94.96	0.47
0.30	92.75	0.46	93.24	0.48	91.62	0.46	91.75	0.42	93.83	0.47	95.48	0.46	94.96	0.46
0.40	93.07	0.43	93.86	0.46	91.62	0.44	91.58	0.41	94.26	0.45	94.35	0.44	94.96	0.44
0.50	93.18	0.42	94.23	0.44	92.20	0.43	93.13	0.40	94.68	0.44	94.63	0.43	94.96	0.43
0.60	92.64	0.41	94.47	0.43	92.63	0.42	92.78	0.39	95.53	0.43	94.35	0.42	94.54	0.43
0.70	92.32	0.40	94.47	0.43	92.34	0.42	93.47	0.38	95.11	0.43	93.79	0.41	94.54	0.43
0.80	92.64	0.40	94.72	0.42	93.06	0.41	93.30	0.38	95.53	0.42	94.35	0.40	94.96	0.42
0.90	92.97	0.40	94.59	0.42	92.92	0.40	93.64	0.37	96.38	0.42	94.35	0.39	94.96	0.42
1.00	93.07	0.40	94.47	0.41	93.50	0.40	93.47	0.37	96.38	0.42	95.48	0.39	94.96	0.41
Credibilidad por confianza acumulada														
λ	20% CPC	30% CPC	40% CPC	50% CPC	60% CPC	70% CPC	80% CPC							
0.10	91.56	0.44	93.49	0.39	92.34	0.31	93.47	0.30	94.47	0.33	94.35	0.32	93.70	0.31
0.20	92.42	0.44	93.24	0.40	91.91	0.33	92.78	0.31	94.04	0.34	95.76	0.33	94.96	0.32
0.30	92.75	0.43	93.24	0.42	91.62	0.35	91.75	0.31	93.83	0.35	95.48	0.34	94.96	0.34
0.40	93.07	0.41	93.86	0.42	91.62	0.38	91.58	0.33	94.26	0.38	94.35	0.36	94.96	0.36
0.50	93.18	0.41	94.23	0.42	92.20	0.39	93.13	0.36	94.68	0.41	94.63	0.38	94.96	0.39
0.60	92.64	0.40	94.47	0.42	92.63	0.40	92.78	0.37	95.53	0.42	94.35	0.39	94.54	0.40
0.70	92.32	0.40	94.47	0.42	92.34	0.41	93.47	0.37	95.11	0.42	93.79	0.40	94.54	0.42
0.80	92.64	0.39	94.72	0.42	93.06	0.40	93.30	0.37	95.53	0.41	94.35	0.40	94.96	0.42
0.90	92.97	0.39	94.59	0.41	92.92	0.40	93.64	0.37	96.38	0.41	94.35	0.39	94.96	0.41
1.00	93.07	0.39	94.47	0.41	93.50	0.40	93.47	0.37	96.38	0.41	95.48	0.38	94.96	0.41

- Si se observa detenidamente las Tablas desde la 4.12 a la 4.15, se notará que los porcentajes de aciertos aunque son cercanos en muchos casos para los diferentes valores de λ , sus correspondientes medidas de fiabilidad varían. Como se verá más adelante, tales patrones de cambios se usarán como criterio de selección del parámetro λ con el fin de conseguir mejores multclasificadores PC e IPC.

Tabla 4.14.- Informes arrojado por la implementación del programa de la Figura 4.19 para el multclasificador PC con las antiguas opciones SVM.

λ	Confianza acumulada													
	20% CON	30% CON	40% CON	50% CON	60% CON	70% CON	80% Con							
0.10	71.88	0.55	83.37	0.54	87.57	0.56	88.37	0.56	89.37	0.55	91.09	0.56	90.52	0.57
0.20	80.89	0.57	91.56	0.56	93.06	0.59	94.44	0.59	94.79	0.57	92.24	0.59	94.83	0.60
0.30	85.78	0.60	92.06	0.58	94.65	0.63	94.79	0.62	95.01	0.59	95.69	0.62	95.69	0.64
0.40	87.40	0.65	91.19	0.61	93.21	0.66	92.36	0.66	93.71	0.62	94.25	0.64	93.10	0.68
0.50	86.86	0.68	89.83	0.64	93.21	0.69	91.15	0.70	93.49	0.65	92.82	0.67	91.81	0.72
0.60	84.47	0.71	89.45	0.67	93.06	0.72	90.28	0.73	93.49	0.68	92.24	0.70	92.24	0.74
0.70	82.95	0.74	88.59	0.70	92.05	0.75	90.28	0.76	93.71	0.72	91.38	0.73	91.38	0.77
0.80	83.28	0.78	87.47	0.75	92.34	0.79	89.93	0.80	92.84	0.75	91.38	0.77	90.52	0.80
0.90	83.17	0.82	85.98	0.80	93.06	0.84	89.06	0.86	91.97	0.82	91.09	0.83	90.95	0.86
1.00	80.67	0.88	87.84	0.91	89.88	0.92	89.58	0.93	91.32	0.93	89.94	0.93	90.09	0.94
λ	Credibilidad Acumulada													
	20% CRE	30% CRE	40% CRE	50% CRE	60% CRE	70% CRE	80% CRE							
0.10	71.88	0.53	83.37	0.52	87.57	0.53	88.37	0.51	89.37	0.52	91.09	0.51	90.52	0.51
0.20	80.89	0.51	91.56	0.51	93.06	0.51	94.44	0.50	94.79	0.51	92.24	0.50	94.83	0.50
0.30	85.78	0.50	92.06	0.50	94.65	0.50	94.79	0.49	95.01	0.50	95.69	0.49	95.69	0.49
0.40	87.40	0.48	91.19	0.49	93.21	0.49	92.36	0.48	93.71	0.49	94.25	0.48	93.10	0.48
0.50	86.86	0.48	89.83	0.49	93.21	0.48	91.15	0.47	93.49	0.48	92.82	0.47	91.81	0.47
0.60	84.47	0.47	89.45	0.48	93.06	0.48	90.28	0.47	93.49	0.48	92.24	0.46	92.24	0.46
0.70	82.95	0.46	88.59	0.48	92.05	0.47	90.28	0.46	93.71	0.47	91.38	0.45	91.38	0.45
0.80	83.28	0.44	87.47	0.47	92.34	0.46	89.93	0.45	92.84	0.47	91.38	0.44	90.52	0.44
0.90	83.17	0.43	85.98	0.46	93.06	0.45	89.06	0.44	91.97	0.46	91.09	0.43	90.95	0.43
1.00	80.67	0.41	87.84	0.44	89.88	0.44	89.58	0.42	91.32	0.44	89.94	0.41	90.09	0.41
λ	Credibilidad por confianza acumulada													
	20% CPC	30% CPC	40% CPC	50% CPC	60% CPC	70% CPC	80% CPC							
0.10	71.88	0.21	83.37	0.19	87.57	0.21	88.37	0.20	89.37	0.20	91.09	0.20	90.52	0.20
0.20	80.89	0.21	91.56	0.20	93.06	0.21	94.44	0.20	94.79	0.20	92.24	0.21	94.83	0.21
0.30	85.78	0.21	92.06	0.20	94.65	0.22	94.79	0.21	95.01	0.21	95.69	0.21	95.69	0.22
0.40	87.40	0.23	91.19	0.21	93.21	0.23	92.36	0.23	93.71	0.22	94.25	0.22	93.10	0.23
0.50	86.86	0.24	89.83	0.23	93.21	0.24	91.15	0.25	93.49	0.23	92.82	0.23	91.81	0.25
0.60	84.47	0.26	89.45	0.24	93.06	0.25	90.28	0.26	93.49	0.24	92.24	0.24	92.24	0.25
0.70	82.95	0.27	88.59	0.26	92.05	0.27	90.28	0.28	93.71	0.26	91.38	0.25	91.38	0.27
0.80	83.28	0.29	87.47	0.29	92.34	0.29	89.93	0.30	92.84	0.29	91.38	0.27	90.52	0.28
0.90	83.17	0.31	85.98	0.33	93.06	0.33	89.06	0.34	91.97	0.33	91.09	0.30	90.95	0.32
1.00	80.67	0.34	87.84	0.39	89.88	0.39	89.58	0.38	91.32	0.40	89.94	0.37	90.09	0.38

- Todo el trabajo reportado referente a la Prueba 10 y 11 se realizó para el detalle V, aunque resultados similares se consiguen también para el detalle A.

Para los detalles H y D se consiguen resultados muy pobres como se verá más adelante.

Tabla 4.15.- Informes arrojado por la implementación del programa de la Figura 4.19 para el multclasificador IPC con las antiguas opciones SVM.

Confianza acumulada														
λ	20% CON	30% CON	40% CON	50% CON	60% CON	70% CON	80% Con							
0.10	94.37	0.76	96.31	0.75	96.10	0.77	96.22	0.76	96.81	0.76	95.76	0.73	98.74	0.78
0.20	94.59	0.82	95.09	0.81	95.52	0.83	95.36	0.81	97.02	0.81	96.05	0.79	97.48	0.83
0.30	93.83	0.88	94.96	0.87	95.52	0.88	95.88	0.86	97.23	0.85	96.61	0.84	97.90	0.86
0.40	94.16	0.94	94.47	0.93	95.23	0.94	95.02	0.92	96.81	0.90	96.33	0.88	97.48	0.90
0.50	93.51	0.97	93.86	0.96	95.38	0.98	94.16	0.96	96.60	0.95	96.33	0.94	97.06	0.93
0.60	93.51	0.98	93.73	0.98	95.52	0.99	94.50	0.98	96.38	0.98	96.33	0.98	97.48	0.97
0.70	93.83	0.98	94.35	0.98	95.38	0.99	95.02	0.99	97.02	0.99	96.05	0.99	97.48	0.98
0.80	94.05	0.98	94.35	0.99	95.38	0.99	94.85	0.99	96.81	0.99	95.76	0.99	98.32	0.98
0.90	94.16	0.98	94.23	0.99	95.52	0.99	95.53	0.99	96.38	0.99	96.05	0.99	98.74	0.99
1.00	93.83	0.99	94.10	0.99	95.23	0.99	95.36	0.99	96.60	0.99	96.05	0.99	98.32	0.99
Credibilidad Acumulada														
λ	20% CRE	30% CRE	40% CRE	50% CRE	60% CRE	70% CRE	80% CRE							
0.10	94.37	0.53	96.31	0.51	96.10	0.48	96.22	0.46	96.81	0.49	95.76	0.48	98.74	0.46
0.20	94.59	0.51	95.09	0.48	95.52	0.46	95.36	0.44	97.02	0.48	96.05	0.47	97.48	0.44
0.30	93.83	0.49	94.96	0.46	95.52	0.45	95.88	0.43	97.23	0.46	96.61	0.46	97.90	0.43
0.40	94.16	0.48	94.47	0.44	95.23	0.44	95.02	0.42	96.81	0.45	96.33	0.45	97.48	0.41
0.50	93.51	0.47	93.86	0.43	95.38	0.43	94.16	0.41	96.60	0.44	96.33	0.44	97.06	0.40
0.60	93.51	0.46	93.73	0.41	95.52	0.42	94.50	0.40	96.38	0.43	96.33	0.43	97.48	0.39
0.70	93.83	0.45	94.35	0.40	95.38	0.42	95.02	0.39	97.02	0.42	96.05	0.42	97.48	0.38
0.80	94.05	0.44	94.35	0.40	95.38	0.42	94.85	0.39	96.81	0.42	95.76	0.41	98.32	0.37
0.90	94.16	0.44	94.23	0.39	95.52	0.41	95.53	0.38	96.38	0.41	96.05	0.40	98.74	0.37
1.00	93.83	0.43	94.10	0.38	95.23	0.41	95.36	0.38	96.60	0.41	96.05	0.39	98.32	0.37
Credibilidad por confianza acumulada														
λ	20% CPC	30% CPC	40% CPC	50% CPC	60% CPC	70% CPC	80% CPC							
0.10	94.37	0.33	96.31	0.31	96.10	0.29	96.22	0.27	96.81	0.30	95.76	0.27	98.74	0.28
0.20	94.59	0.35	95.09	0.33	95.52	0.31	95.36	0.29	97.02	0.31	96.05	0.30	97.48	0.29
0.30	93.83	0.39	94.96	0.35	95.52	0.35	95.88	0.31	97.23	0.33	96.61	0.32	97.90	0.30
0.40	94.16	0.42	94.47	0.38	95.23	0.39	95.02	0.34	96.81	0.36	96.33	0.35	97.48	0.32
0.50	93.51	0.44	93.86	0.39	95.38	0.41	94.16	0.38	96.60	0.40	96.33	0.39	97.06	0.34
0.60	93.51	0.44	93.73	0.40	95.52	0.42	94.50	0.39	96.38	0.42	96.33	0.41	97.48	0.36
0.70	93.83	0.44	94.35	0.39	95.38	0.42	95.02	0.38	97.02	0.42	96.05	0.41	97.48	0.37
0.80	94.05	0.43	94.35	0.39	95.38	0.41	94.85	0.38	96.81	0.41	95.76	0.41	98.32	0.36
0.90	94.16	0.43	94.23	0.38	95.52	0.41	95.53	0.38	96.38	0.41	96.05	0.40	98.74	0.36
1.00	93.83	0.43	94.10	0.38	95.23	0.41	95.36	0.37	96.60	0.40	96.05	0.39	98.32	0.36

- Es muy importante tener claro cómo funciona la predicción conformal y la predicción conformal inductiva. Para ello se pueden revisar las secciones 3.7 y 3.8. Se indica esto porque en la Figura 4.26 se observa que al entrenar ambos predictores (PC e IPC), posteriormente se evalúan. Y realmente estos procesos se ejecutan en dos capas pero internamente dentro del mismo código, tal y como se ilustró en las Figuras 3.18 y 3.22. Más aún, el mecanismo que utilizan ambos predictores (con sus diferencias claramente explicadas en el

Capítulo III), toma un conjunto de muestras para entrenamiento y una observación (muestra) de prueba y con esto determina unas medidas de no conformidad, que al computarse se obtiene un conjunto de *p_valores* de donde finalmente se consigue el pronóstico y las medidas de fiabilidad.

- Finalmente se recalca, que la implementación del programa de la Figura 4.26 es independiente de los detalles AHVD, del porcentaje de imágenes a usar para el entrenamiento y prueba, e incluso de las opciones SVM; ya que estos aspectos pueden ser pre-configurados y el código sólo exploraría los diferentes valores λ , que por lo general se evaluaron en esta investigación a razón de cambio de 0.1 en el rango de cero a uno ($\lambda \in [0,1]$). Más adelante se verá, que el mismo programa permite trabajar con un λ específico tras evaluar sus diferentes valores a partir de un criterio asumido.

4.7.2 Criterio para la selección del parámetro λ

Tras realizar los ensayos relacionados con la Prueba 10 donde se evaluaron las opciones SVM en contraste con el parámetro λ , se debía adoptar un criterio para seleccionar un valor λ entre todos sus valores posibles. En consecuencia y apoyando esta decisión a partir de los resultados observados en las Tablas 4.11 a la 4.15 y entendiendo que su análisis puede conducir a diferentes soluciones, se tomó el criterio de observar sólo los altos porcentajes de acierto con aquellas medidas de confianza que también sean altas. En este sentido, se escoge tanto el valor λ como las opciones SVM que propicien tales condiciones.

Basado en este simple pero laborioso criterio, se fija el parámetro λ en 0.5, entendiendo que su valor no es el único que pudo dar solución al problema multiclase de las imágenes procesadas del TJ-II, pero es uno de los que refleja para un 20% de imágenes con detalle V tomadas para entrenamiento, altos porcentajes de aciertos. Este valor λ como se verá en las secciones 4.8 y 4.9, fue establecido en los experimentos incrementales desarrollados con los multclasificadores PC e IPC para explorar y encontrar el porcentaje de imágenes mínimo necesario para realizar los

experimentos por lote con los multclasificadores PC e IPC y así poder obtener altos porcentajes de aciertos con medidas de fiabilidad razonables.

Como resumen ilustrativo de la importancia en la selección parámetros, en la Tabla 4.16 se muestra un contraste entre los valores para las opciones SVM y el parámetro λ que son usados para construir los multclasificadores SVM, PC e IPC, antes y después del ajuste propuesto en la Prueba 10. En la fila 1 y 2 se observan las opciones SVM antiguas que daban altos porcentajes de aciertos. Para estas mismas opciones, se puede observar en la en las filas 6 y 7 como los porcentajes de aciertos varían, siendo más notorio el efecto en el multclasificador PC que ubica el porcentaje de aciertos en 86%, y para el multclasificador IPC se ubica en 93%.

En las filas 4 y 5 se pueden ver las nuevas opciones SVM con el mismo parámetro λ , notándose como propician un 94% y 93% en la tasa de aciertos para el multclasificador PC e IPC respectivamente. En la fila 3 se puede apreciar como las nuevas opciones SVM al ser establecidas en el multclasificador SVM, no permiten un porcentaje de acierto mejor que las antiguas opciones ya calculadas.

Tabla 4.16.- Selección del parámetro λ y opciones SVM para construir los multclasificadores SVM, PC e IPC para un 20% de muestras para entrenamiento.

λ	n_1	n_2	n_3	n_4	n_5	d_1	d_2	d_3	d_4	d_5	Multclasificador / %Acierto
No aplica	8	10	10	8	7	50	50	50	50	50	SVM / Alrededor de 95%
No aplica	8	10	10	9	7	50	50	50	50	50	SVM / Alrededor del 95%
No aplica	8	10	10	9	7	909	909	909	909	909	SVM / Alrededor de 92%
0.5	8	10	10	9	7	909	909	909	909	909	PC / Alrededor de 94%
0.5	8	10	10	9	7	909	909	909	909	909	IPC / Alrededor de 93%
0.5	8	10	10	8	7	50	50	50	50	50	PC / Alrededor de 86%
0.5	8	10	10	9	7	50	50	50	50	50	IPC / Alrededor del 93%

Nota: los valores para d se expresan en millones

Del resumen mostrado en la Tabla 4.16 se puede concluir que la selección tanto de las opciones SVM como del parámetro λ es un proceso importante para obtener mejores multclasificadores, pero no es directo, ya que se debe realizar una exploración sistemática como se ha demostrado entre las pruebas 10 y 11 en esta sección.

4.8 MULTICLASIFICADOR INCREMENTAL Y POR LOTE CON PC

En este experimento se realizaron pruebas puntuales de clasificación incremental y por lote con la predicción conformal o multclasificador PC. Los ensayos siguieron la agrupación de imágenes propuesta en la Figura 4.1 para cuatro y cinco clases con datos balanceados y desbalanceados según la Tabla 4.1, examinándose únicamente los conjuntos de imágenes con detalle V.

Las pruebas se iniciaron con el entrenamiento y pruebas de clasificación incremental para poder fijar el conjunto inicial mínimo (porcentaje) de imágenes necesario para entrenar el multclasificador PC por lote y así conseguir altas tasas de aciertos y medidas de fiabilidad razonables en la clasificación de imágenes. Cabe acotar que estos ensayos también se repitieron un número considerable de veces⁴⁹.

Antes de presentar los resultados obtenidos con la predicción conformal, se invita al lector a situarse en el contexto del proceso que realiza este modelo de predicción, el cual se ha marcado en un cuadro punteado en las Figuras 4.27 y 4.30, recordando que todo el funcionamiento teórico e incluso práctico asociado a este sistema de aprendizaje automático fue discutido en el Capítulo III.

Prueba 12.- Resultados obtenidos en la clasificación incremental para cuatro-cinco clases balanceado-desbalanceado

Para la implementación de las pruebas con el multclasificador PC incremental se tomaron los valores de las opciones SVM y del parámetro λ según la fila 4 de la Tabla 4.16 vista en la sección 4.7, siendo establecidos en la implementación del programa que se muestra a través del diagrama de flujo de la Figura 4.27 donde se ha embebido el código necesario para ejecutar la predicción conformal por incremento.

⁴⁹ La razón de repetir cada ensayo se debe a la naturaleza aleatoria del sorteo efectuado en el momento de escoger las imágenes que forman los conjuntos para el entrenamiento y pruebas de clasificación incremental. Al ser examinados estos conjuntos y contrastados todos en una misma gráfica, se observa en los resultados gráficos al inicio de cada curva asociada al porcentaje de acierto, diferentes transitorios que no marcan un mismo punto donde pudiera iniciarse el entrenamiento por lote. Aunque después convergen todas las curvas a un lugar común que indica el valor del porcentaje de imágenes que puede tomarse para iniciar el entrenamiento por lote.

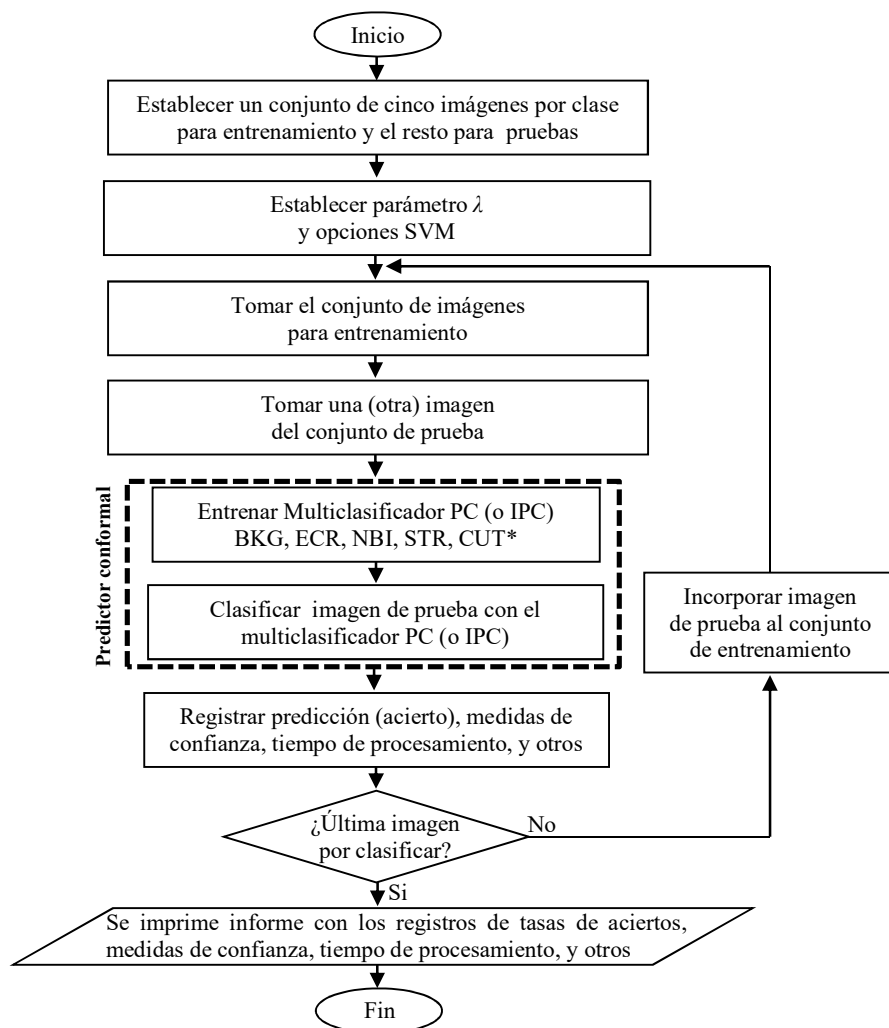


Figura 4.27.- Diagrama de bloques general de la implementación del Programa que clasifica imágenes por incremento a partir del multclasificador PC.

Analizando la implementación del programa, se puede apreciar del proceso ilustrado por el diagrama de flujo, la selección de cinco imágenes por clase para formar el conjunto inicial de entrenamiento. Del resto de las imágenes que quedan para prueba, se va tomando una observación (muestra) a la vez y la misma se incorpora al conjunto inicial de entrenamiento del multclasificador PC incremental. Con esto se entrena el multclasificador PC y se clasifica la muestra al *tomar la*

predicción y las medidas de fiabilidad ⁵⁰. La muestra clasificada, se incorpora al conjunto inicial de entrenamiento según su clase, se toma una segunda muestra del conjunto de prueba, se entrena el multclasificador PC y se clasifica la segunda muestra y después es incorporada según su clase al conjunto de entrenamiento inicial. Este procedimiento se sigue hasta lograr la clasificación completa del conjunto de imágenes reservado para este tipo de ensayos.

Con relación a las cantidades de imágenes dispuestas para iniciar los entrenamientos y pruebas incrementales con el multclasificador PC, se observa en la Tabla 4.17 los conjuntos con cuatro y cinco clases tanto balanceadas como desbalanceadas, sólo con el detalle V. Según esta información, se puede ver también en las Figuras 4.28 y 4.29 los resultados gráficos obtenidos al examinar tales conjuntos.

Adicionalmente se indica que las medidas de fiabilidad (confianza, credibilidad y confianza por credibilidad) para este experimento, así como para el resto de los que se presentaran en adelante, se han colocado en gráficas separadas para una mejor visualización.

Tabla 4.17.- Cantidades de imágenes dispuestas para los entrenamientos y pruebas incrementales con el fin de establecer el conjunto inicial de imágenes de entrenamiento con el multclasificador PC.

Prueba 12 Multclasificador PC	Conjunto total de imágenes	Conjunto inicial de entrenamiento incremental	Conjunto reservado para pruebas incrementales	Conjunto donde puede iniciar el entrenamiento por lote
4 clases desbalanceado	1107	20	1087	Alrededor de 200
5 clases desbalanceado	1149	25	1024	Alrededor de 300
4 clases balanceado	428	20	408	Alrededor de 200
5 clases balanceado	210	25	185	No aplica

⁵⁰ En teoría, la predicción conformal sólo arroja la predicción (clasificación) de una muestra y las medidas de fiabilidad asociada a la confianza y credibilidad. El resto de las variables medidas en el proceso que se está presentando, como el tiempo de procesamiento por muestra o por conjuntos de muestra, medidas de confianza por credibilidad; se miden con el fin de poder realizar un mejor diagnóstico así como los contrastes correspondientes.

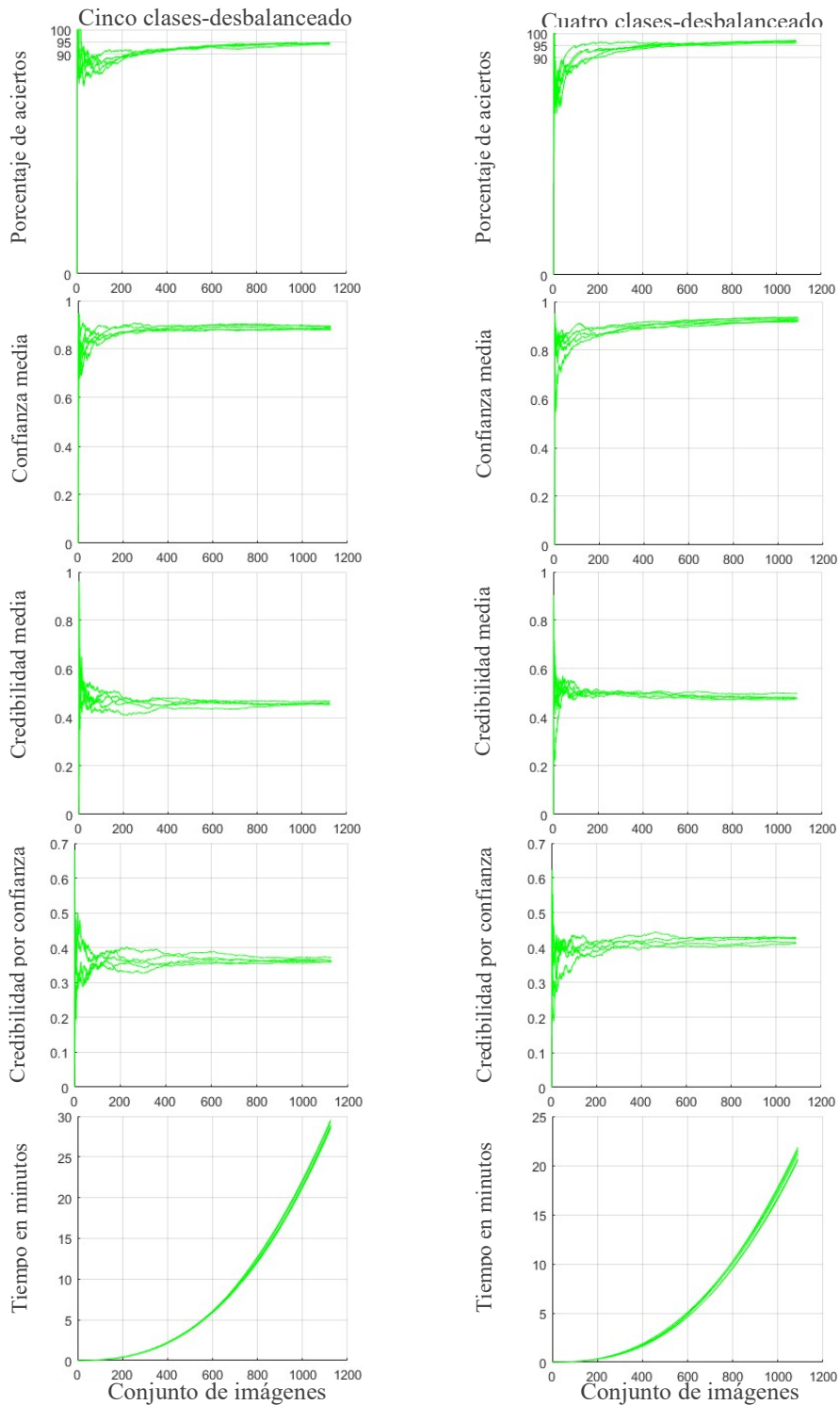


Figura 4.28.- Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador PC incremental para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

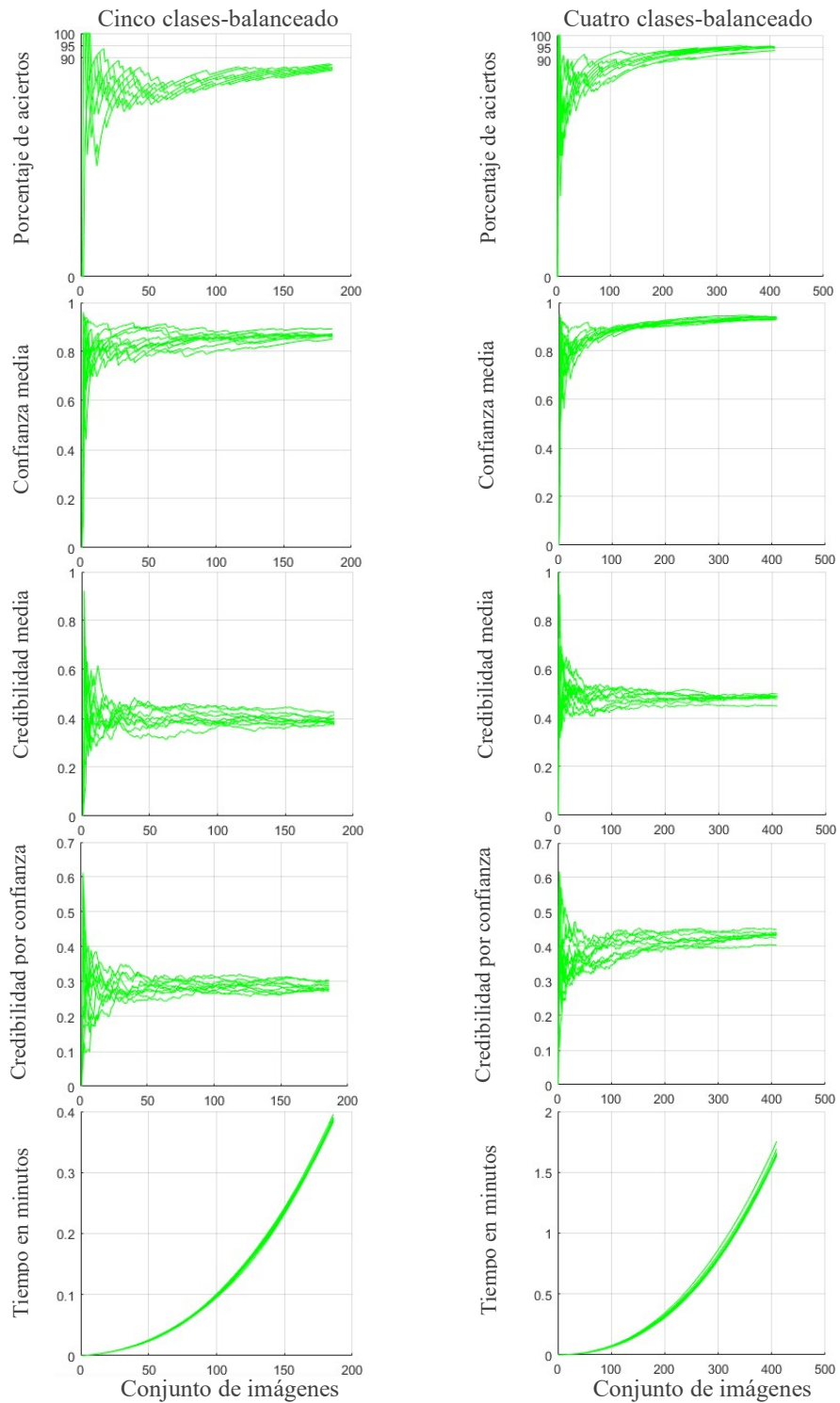


Figura 4.29.- Resultados obtenidos al ejecutar diez veces el programa asociado al multclasificador PC incremental para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

Análisis de los resultados obtenidos en la Prueba 12

- En la Figura 4.28, se observa que los multclasificadores PC incrementales permiten fijar un porcentaje de entrenamiento para los multclasificadores PC por lote para imágenes con cuatro y cinco clases desbalanceadas, alrededor de la muestra 200 y 300 respectivamente. En este punto ya se supera el 90% del porcentaje de aciertos (véase la columna 5 de la Tabla 4.17).
- Por su parte, el multclasificador PC incremental para imágenes con cuatro clases balanceadas supera el 90% alrededor de la muestra 200, indicando con esto el punto para entrenar el multclasificador PC por lote. Sin embargo con cinco clases balanceadas, no se promediaron altos valores en el porcentaje de aciertos (véase la Figura 4.29 y la columna 5 de la Tabla 4.17).
- Se observa además, que la confianza media en todos los casos es elevada, alrededor de 0.9. Por su parte la credibilidad, se ubica alrededor de 0.5 y el producto de ambas medidas se sitúa en torno a 0.4 con sus variaciones particulares, hecho que puede ser notado en las gráficas correspondientes.

Prueba 13.- Resultados obtenidos en la clasificación por lote para cuatro-cinco clases balanceado-desbalanceado

Para la implementación de las pruebas con el multclasificador PC por lote, se mantienen las opciones SVM y el parámetro λ según la fila 4 de la Tabla 4.16. Además de disponer como referencia inicial para tomar los conjuntos de entrenamiento por lote, los porcentajes fijados durante los ensayos con el multclasificador PC por incremento (vea la columna 5 de la Tabla 4.17). Se recuerda que el resto de las imágenes se reserva para la pruebas de clasificación por lote.

Todos estos valores mencionados se establecen en la implementación del programa que tiene embebido el código necesario para ejecutar la predicción conformal por lote, cuyo diagrama de flujo se muestra en la Figura 4.30, y en donde se puede observar que se recibe un porcentaje de imágenes preestablecido para realizar el entrenamiento por lote, el cual no cambia, pero se le debe incorporar desde las imágenes reservadas para realizar pruebas de clasificación; la muestra que será

examinada y de esta manera se clasifica al tomarse la predicción y las medidas de fiabilidad.

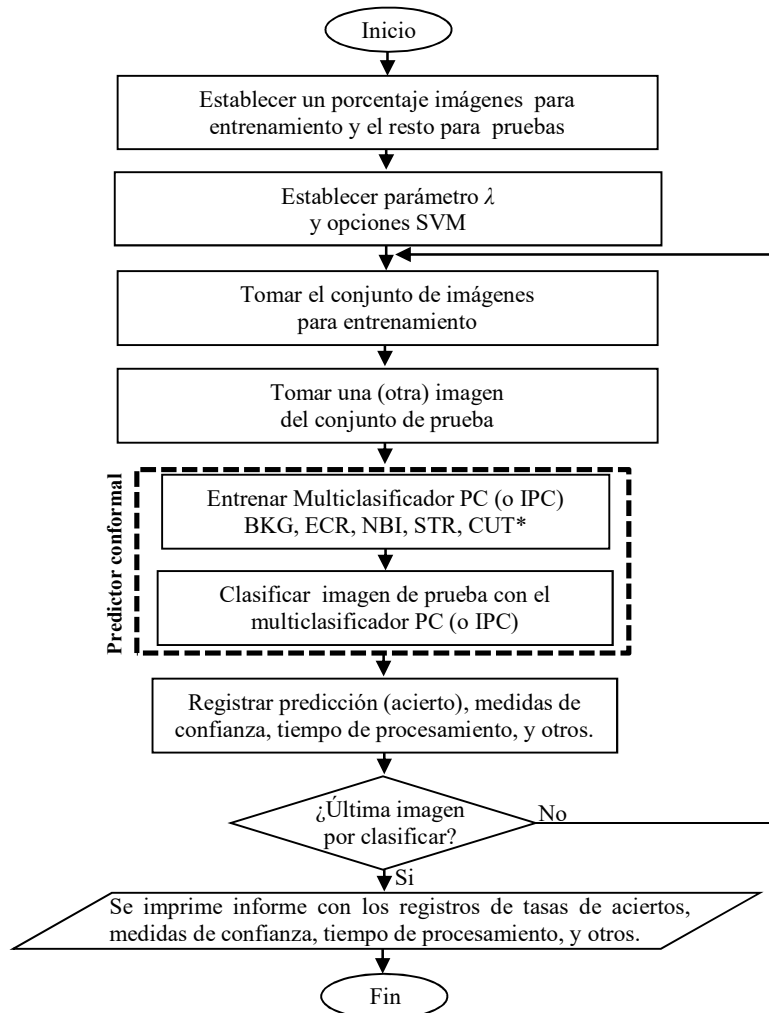


Figura 4.30.- Diagrama de bloques general la implementación del programa que clasifica imágenes por lote a partir del multclasificador PC.

Es importante tener presente las diferencias entre el procedimiento que sigue la clasificación incremental y la clasificación por lote. A grandes rasgos, en la clasificación por lote, el conjunto de imágenes para entrenamiento no cambia puesto que el mismo es prefijado una única vez, examinándose con este todo el conjunto de imágenes reservado para las pruebas de clasificación. Mientras que al usar el procedimiento incremental, el conjunto inicial de imágenes de entrenamiento va

cambiando al terminar de evaluar cada muestra del conjunto de imágenes reservado para pruebas de clasificación. Cada imagen evaluada es incorporada al conjunto inicial (o consecutivo) de entrenamiento fijado (o incrementado). Con cada incorporación de una muestra ya clasificada, se tiene un nuevo conjunto de entrenamiento y de esta manera se examina la siguiente observación.

Continuando con los resultados de la Prueba 13, en columna 2 de la Tabla 4.18 se muestra el resumen con la totalidad de las imágenes dispuestas para este ensayo. Obsérvese en la columna 3, que se han tomado el 19%, 27% y 46% de muestras para realizar los entrenamientos y pruebas por lote según los resultados obtenidos en las pruebas de clasificación incremental.

Tabla 4.18.- Cantidades de imágenes dispuestas para los entrenamientos y pruebas por lote con el multclasificador PC y su convergencia hacia altas tasas de aciertos.

Prueba 13 Multclasificador PC	Conjunto total de imágenes	Conjunto de entrenamiento fijado por pruebas incrementales %	Converge alrededor de la imagen	Ajuste manual del conjunto de entrenamiento %	Converge alrededor de la imagen
4 clases desbalanceado	1107	19	21	20	5
5 clases desbalanceado	1149	27	25	26	50
4 clases balanceado	428	46	31	50	12
5 clases balanceado	210	NA	NA	NA	NA

Nótese, que no se fijó ningún porcentaje para el entrenamiento y prueba del multclasificador PC por lote con cinco clases balanceado, porque en las pruebas con el multclasificador PC incremental no se obtuvo una alta tasa de aciertos al procesar este conjunto de imágenes.

Los resultados obtenidos se muestran en las Figuras 4.31 y 4.32, donde se observan curvas repetidas debido a la ejecución de un número considerable de veces de la implementación del programa asociado al multclasificador PC por lote para los conjuntos de imágenes con cuatro y cinco clases tanto balanceados como desbalanceados, sólo para el detalle V.

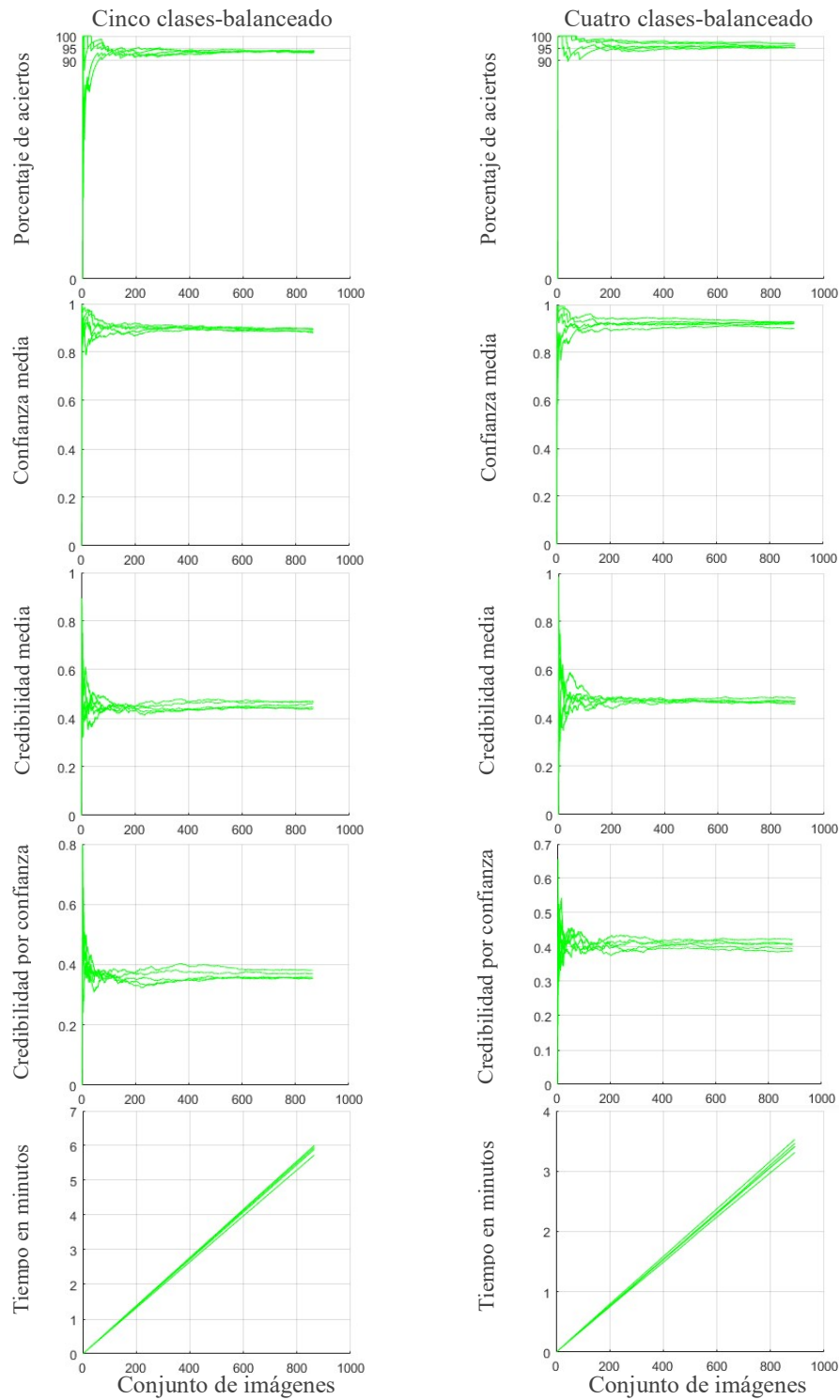


Figura 4.31.- Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador PC por lote para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

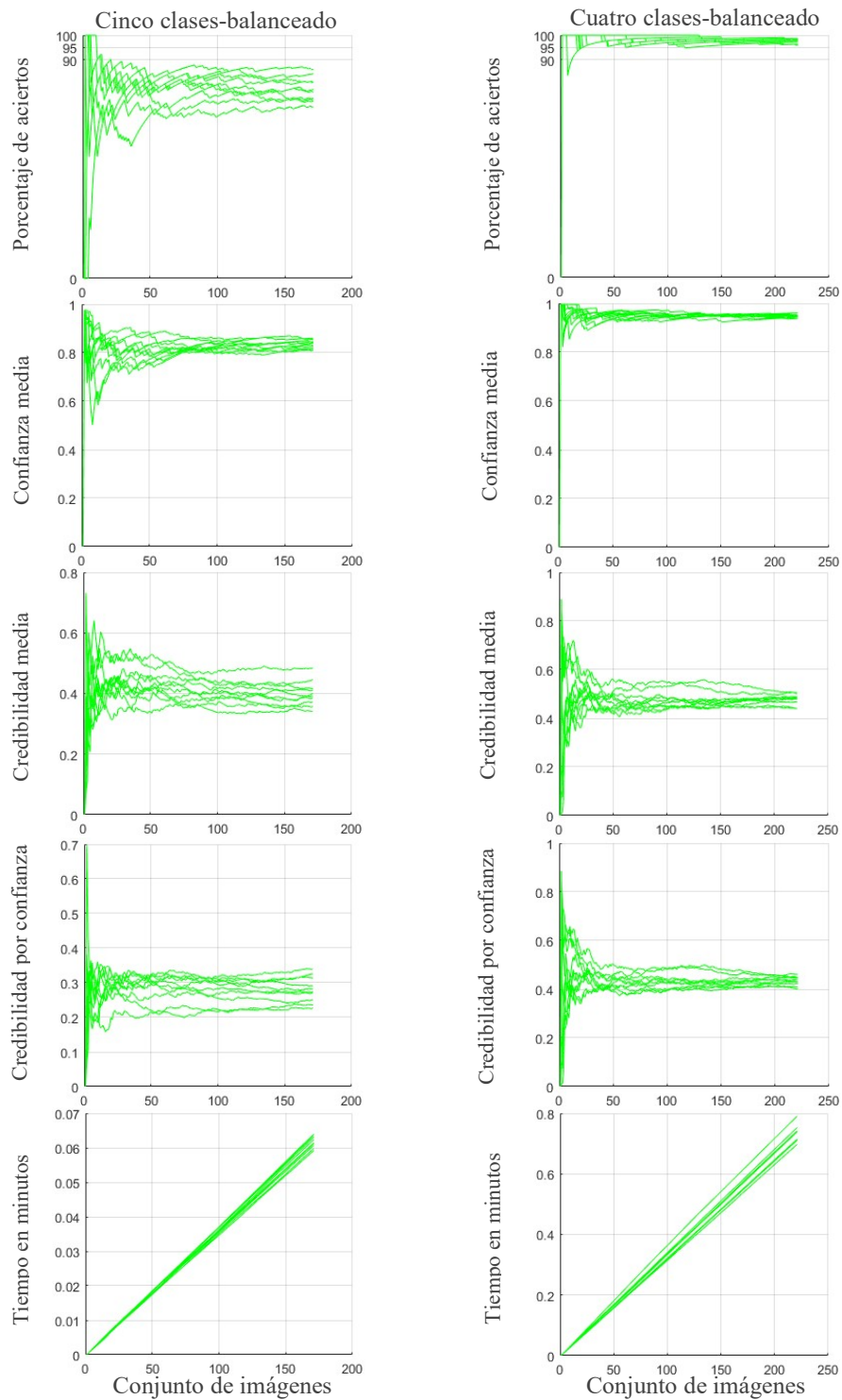


Figura 4.32.- Resultados obtenidos al ejecutar diez veces el programa asociado al multclasificador PC por lote para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

De la inspección realizada sobre los resultados gráficos se puede extraer, el valor asociado al “número de imagen” donde comienza aproximadamente la convergencia hacia altas tasas de acierto en la predicción de imágenes. Estos valores fueron anotados en la columna 4 de la misma Tabla 4.18 para contrastarlos con los conjuntos de entrenamiento (porcentaje de imágenes) fijados por las pruebas incrementales.

También se puede apreciar por la inspección realizada sobre los resultados gráficos, que la convergencia registra picos en las gráficas asociadas al porcentaje de acierto por debajo del 90%, siendo esta la razón por la cual se ajusta manualmente el conjunto de entrenamiento tal y como puede verse en la columna 5 de la Tabla 4.18. Ante el ajuste y posterior ejecución del código, la nueva convergencia se registra en la columna 6 de la misma tabla.

Análisis de los resultados obtenidos en la Prueba 13

- Al observar la Figura 4.31 y el resumen mostrado en la Tabla 4.18 para los multclasificadores PC por lote con cuatro y cinco clases desbalanceados, se puede notar una clara convergencia del porcentaje de acierto alrededor del 95% en la predicción de imágenes, después del ajuste del conjunto de muestras usado para el entrenamiento. Se debe apreciar que esta convergencia comienza a situarse a muy pocas observaciones procesadas por parte del multclasificador PC con cuatro clases, presentando un pico por debajo del 90% en la muestra 50. Para el multclasificador PC con cinco clases, la convergencia se inicia aproximadamente en la muestra 50.
- Para el sistema predictivo balanceado con cuatro clases mostrado en la Figura 4.32, el porcentaje de acierto también ronda el 95% en la predicción de imágenes. Aunque la convergencia comienza a situarse a 12 observaciones procesadas, después del ajuste del conjunto de muestras usado para el entrenamiento.
- Con relación al sistema predictivo balanceado con cinco clases mostrado en la misma Figura 4.32, este jamás supera el 90% de la tasa aciertos para ningún

conjunto de entrenamiento. Por esta razón se ha marcado en la Tabla 4.18 las siglas NA (No Aplica).

- Es importante destacar que los porcentajes de aciertos en la predicción de imágenes obtenidos a partir de los multclasificadores PC por lote, fueron pre visualizados por los altos porcentajes de aciertos explorados en los entrenamientos incrementales, donde los mismos apuntaban a la selección de un conjunto mínimo de imágenes para realizar los entrenamientos por lote.
- También se puede comprobar en la mayoría de los casos, que la selección del conjunto mínimo de imágenes dado por el entrenamiento incremental es meramente orientativa, ya que establece un punto alrededor del número de muestras procesadas (o porcentaje de la misma) que pudiera utilizarse para fijar el conjunto de imágenes a usar en los entrenamientos (por lote).
- Según lo anterior, para fijar el punto asociado con el número de imágenes mínimo y así formar el conjunto de entrenamiento y pruebas por lote, se debe recurrir a un ajuste manual, tal y como ha quedado demostrado hasta ahora. Además, se debe resaltar la importancia del ajuste manual cuando se tenga que entrenar alguno de estos sistemas predictivos con datos masivos bajo el enfoque estudiado en esta Tesis, sobretodo porque este procedimiento permitiría reducir tiempo de procesamiento al seleccionar un conjunto mínimo de imágenes.
- Se observa para todos los multclasificadores presentados, exceptuando el de cinco clases balanceadas, un valor alto en la medida de confianza media situándose alrededor de 0.9, como era de esperar, por los entrenamientos incrementales. Por su parte, la credibilidad se ubica alrededor de 0.45, siendo dispersa para el conjunto con cuatro clases balanceadas.

4.9 MULTICLASIFICADOR INCREMENTAL Y POR LOTE CON IPC

En esta sección se presentaran las pruebas puntuales de clasificación incremental y por lote desarrolladas con la predicción conformal inductiva o multclasificador IPC. De igual forma, los ensayos también siguieron la agrupación

de imágenes propuesta en la Figura 4.1 para cuatro y cinco clases con datos balanceados y desbalanceados según la Tabla 4.1, examinándose sólo conjuntos de imágenes con detalle V.

Las pruebas se iniciaron con el entrenamiento y pruebas de clasificación incremental, sirviendo como en los casos anteriores, para poder fijar el conjunto inicial mínimo (porcentaje) de imágenes necesario para entrenar el multclasificador IPC por lotes y así conseguir altas tasas de aciertos y medidas de fiabilidad razonables en la clasificación de imágenes. Se recuerda que estos ensayos también se repitieron un número considerable de veces para garantizar que el sorteo de los conjuntos de imágenes a usar no afectaba a los resultados.

Finalmente se hace hincapié al lector a situarse en el contexto del proceso que realiza la predicción conformal inductiva y sobre todo a diferenciarla de su predecesora la predicción conformal, aunque en las Figuras 4.27 y 4.30 ambos métodos predictivos se ilustraron con el mismo diagrama de flujo. Sin embargo y ante esto, hay consideraciones que deben ponerse de manifiesto con relación al manejo de los conjuntos de imágenes que son usados por parte de ambos métodos en los entrenamientos y en las pruebas de clasificación. Estas consideraciones no son mencionadas en este capítulo, ya que fueron tratadas rigurosamente en las secciones 3.7 y 3.8 del Capítulo III donde se discute en detalle el funcionamiento tanto de la predicción conformal como de la predicción conformal inductiva respectivamente.

Prueba 14.- Resultados obtenidos en la clasificación incremental para cuatro-cinco clases balanceado-desbalanceado

Para la implementación del multclasificador IPC incremental se tomaron los valores de las opciones SVM y del parámetro λ , escritos en la fila 5 de la Tabla 4.16 en la sección 4.7. Estos valores reportaron resultados favorables en las tasas de aciertos, y por esto son establecidos en la implementación del programa que se ilustra en el diagrama de flujo que fue mostrado en la Figura 4.27.

Comentando nuevamente este programa, pero manejando ahora la predicción conformal inductiva (IPC), se aprecia que el proceso también toma 5 imágenes por

clase para formar el conjunto inicial de entrenamiento. El resto de las imágenes se reserva para pruebas de clasificación. Sin embargo y como consideración especial que no es ilustrada en la propia Figura 4.27 y a diferencia de la predicción conformal, *el código que maneja la predicción conformal inductiva divide el conjunto inicial de entrenamiento en dos nuevos conjunto*⁵¹, los cuales son: el conjunto de entrenamiento propio y el conjunto de calibración. Esto se entiende al ver la Figura 4.33 donde también se ilustra que tales conjuntos inician con 3 y 2 muestras respectivamente.

Del resto de las imágenes reservadas para pruebas de clasificación (en la división original), se va tomando una observación (imagen) a la vez y la misma ya procesada por el multclasificador IPC, es incorporada alternativamente al conjunto inicial de entrenamiento. Esto se simboliza con la línea punteada que se ha trazado en la misma Figura 4.33.

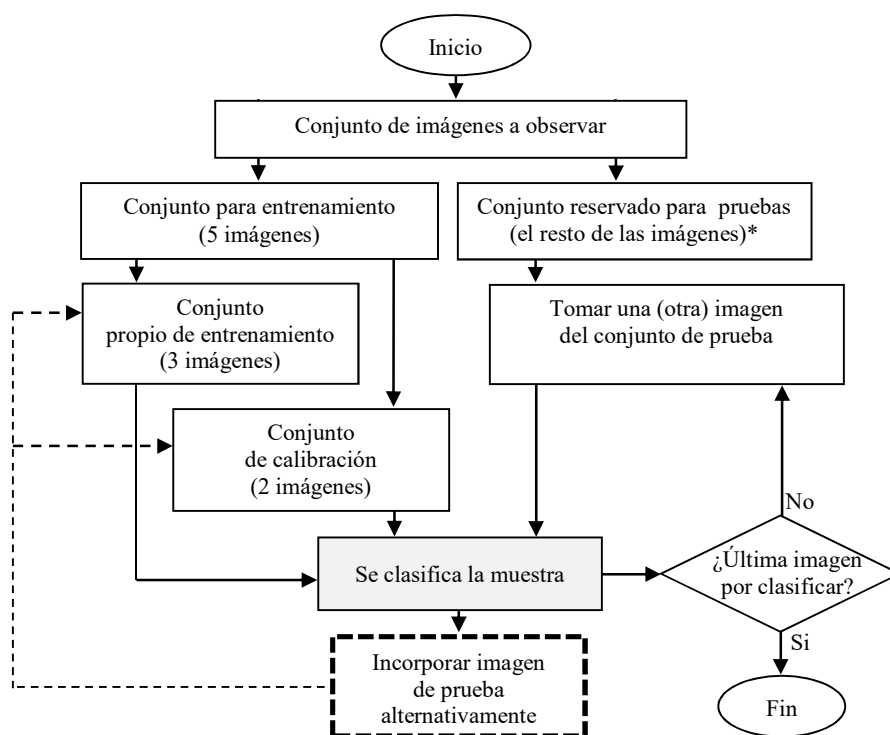


Figura 4.33.- Acceso a las imágenes mediante el algoritmo que implementa la predicción conformal inductiva.

⁵¹ La división del conjunto inicial de entrenamiento que realiza el multclasificador IPC se explica en profundidad en la sección 3.8 del Capítulo III.

Cuando se dice que la incorporación de la imagen se hace alternativamente, se está indicando que luego de ser examinada la muestra actual, se deposita en el conjunto propio de entrenamiento; y la siguiente muestra a examinar, se colocará en el conjunto de calibración. Todo este proceso se repetirá alternativamente, por lo tanto, el conjunto propio de entrenamiento y el de calibración, irán incrementándose alternativamente hasta predecir la totalidad de las imágenes reservadas para las pruebas de clasificación.

Con relación a las cantidades de imágenes dispuestas para realizar los entrenamientos y pruebas incrementales con el multclasificador IPC, se observa en la Tabla 4.19 los conjuntos con cuatro y cinco clases tanto balanceadas como desbalanceadas, sólo con el detalle V. Según esta información, se puede ver también en las Figuras 4.34 y 4.35 los resultados gráficos obtenidos al examinar tales conjuntos.

Tabla 4.19.- Cantidades de imágenes dispuestas para los entrenamientos y pruebas incrementales con el fin de establecer el conjunto inicial de imágenes de entrenamiento con el multclasificador IPC.

Prueba 14 Multclasificador IPC	Conjunto total de imágenes	Conjunto inicial de entrenamiento incremental	Conjunto reservado para pruebas incrementales	Conjunto donde puede iniciar el entrenamiento por lote
4 clases desbalanceado	1107	20	1087	Alrededor de 150
5 clases desbalanceado	1149	25	1024	Alrededor de 350
4 clases balanceado	428	20	408	Alrededor de 125
5 clases balanceado	210	25	185	No aplica

Análisis de los resultados obtenidos en la Prueba 14

- Mediante la observación de la Figura 4.34, se concluye que los multclasificadores IPC incrementales permiten fijar un porcentaje de entrenamiento para los multclasificadores IPC por lote para imágenes con cuatro y cinco clases desbalanceadas, alrededor de la muestra 150 y 350 respectivamente (véase la columna 5 de la Tabla 4.19). En este punto ya se supera el 90% del porcentaje de acierto.

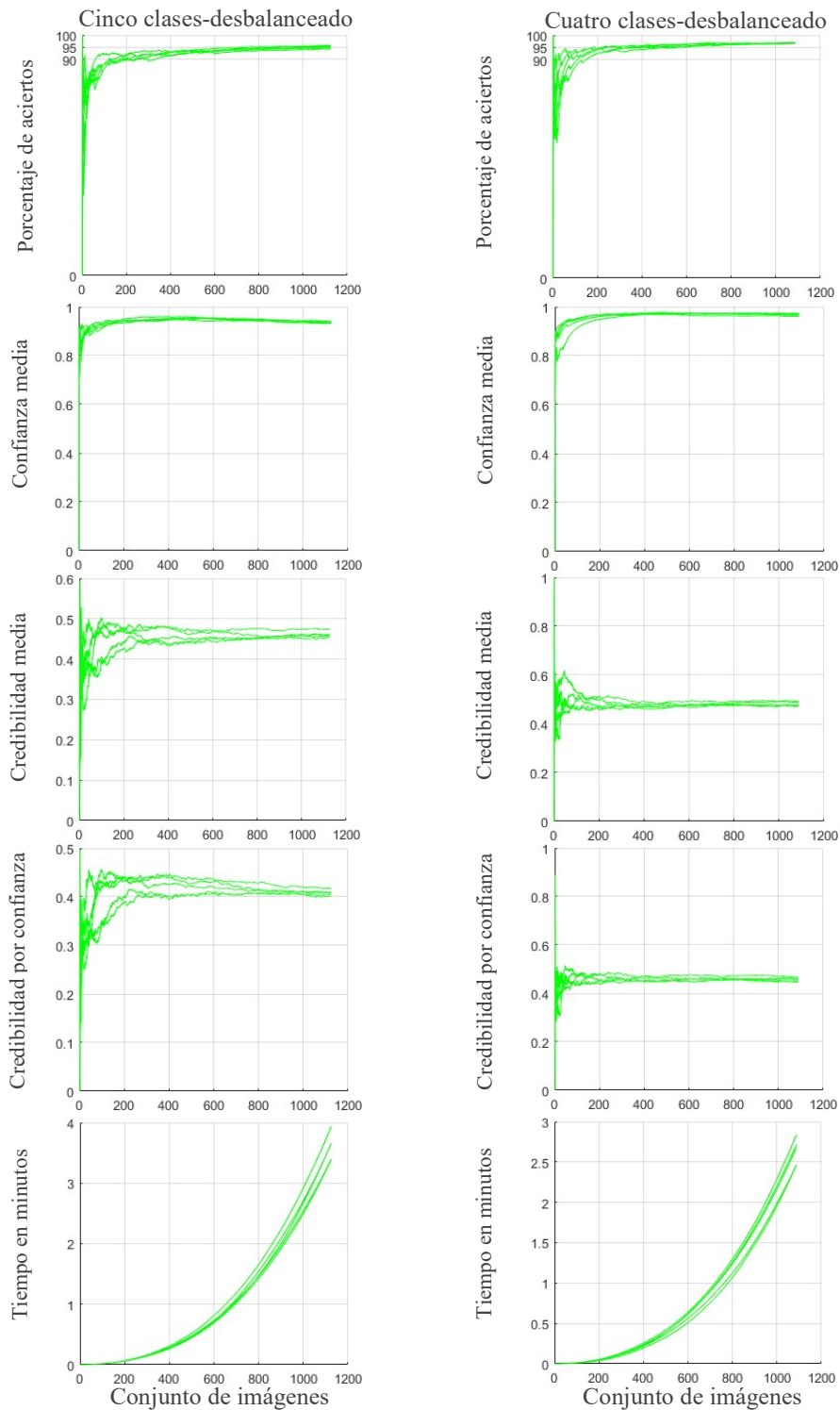


Figura 4.34.- Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador IPC incremental para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

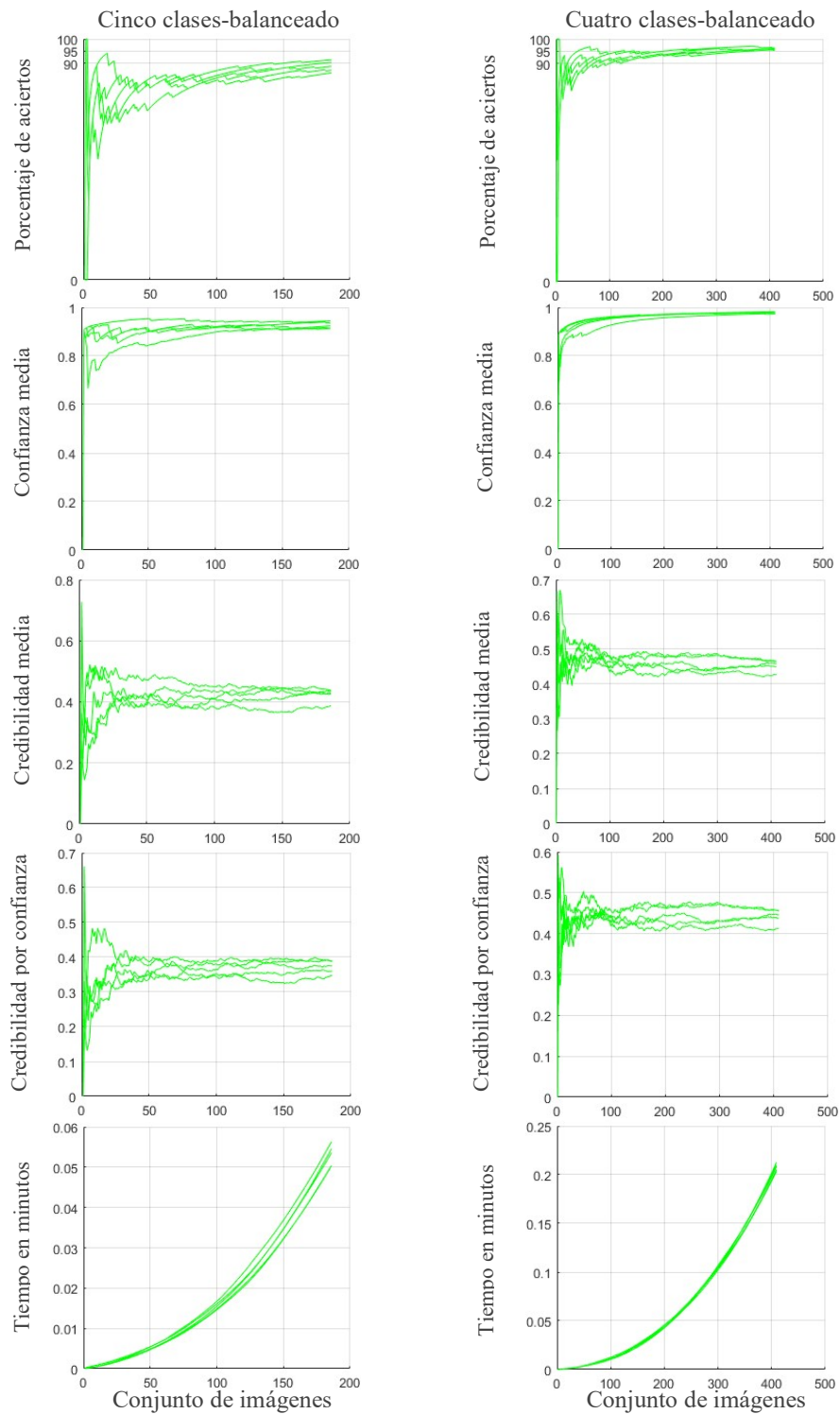


Figura 4.35.- Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador IPC incremental para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

- En condiciones similares, el multclasificador IPC incremental para imágenes con cuatro clases balanceadas supera el 90% alrededor de la muestra 125 (véase la columna 5 de la Tabla 4.19), aunque el de cinco clases no promedia altos valores en el porcentaje de aciertos (véase la Figura 4.35).
- Con relación a las medidas de fiabilidad, las mismas presentan valores similares a los ya obtenidos por los multclasificadores PC incrementales.

Prueba 15.- Resultados obtenidos en la clasificación por lote para cuatro-cinco clases balanceado-desbalanceado

Para analizar las pruebas realizadas con el multclasificador IPC por lote, se mantuvieron las opciones SVM y el parámetro λ mostrados en la fila 5 de la Tabla 4.16. Además y como era de esperar, se dispuso de los porcentajes fijados mediante los ensayos con el multclasificador IPC por incremento que son reflejados en la columna 5 de la Tabla 4.19, sirviendo esto para seleccionar los conjuntos de imágenes a ser usados en los entrenamientos y pruebas de clasificación por lote.

Todos estos valores mencionados se establecen en la implementación del programa mostrado a través del diagrama de flujo de la Figura 4.30, que además del código para ejecutar la predicción conformal, permite a través de un “if” ejecutar la predicción conformal inductiva por lote.

Conjuntamente, se debe recordar la consideración hecha en el preámbulo de la Prueba 14, donde se planteó una de las diferencias fundamentales que existe entre un multclasificador PC y un multclasificador IPC. Esta diferencia discutida, también tiene su implicación al examinar las imágenes en esta Prueba 15 a partir de la implementación del programa ilustrado, ya presentado en la Figura 4.30. Específicamente sobre el conjunto de entrenamiento, ya que el mismo también se debe dividir en el conjunto propio de entrenamiento y en el conjunto de calibración.

Asimismo se debe recordar, la diferencia entre una prueba incremental y una prueba por lote. Como ya se sabe, la clasificación por incremento sugiere un cambio progresivo del conjunto de entrenamiento por cada muestra clasificada. Mientras que

en la clasificación por lote, el conjunto de entrenamiento permanece constante ante todo el espacio de muestras a ser clasificadas.

Una vez aclarados todos estos aspectos de interés para la comprensión de lo que se está presentado, se analiza la Tabla 4.20. Ahí se puede apreciar la totalidad de las imágenes dispuestas para este otro ensayo, de donde se tomó el 14%, 31% y 30% de muestras para realizar los entrenamientos y pruebas por lote según lo fijado por las pruebas incrementales (columna 3). Véase además, que no se logró fijar ningún porcentaje para el entrenamiento y prueba del multclasificador IPC con cinco clases balanceado.

Tabla 4.20.- Cantidades de imágenes dispuestas para los entrenamientos y pruebas por lote con el multclasificador IPC y su convergencia hacia altas tasas de aciertos en la Prueba 15.

Prueba 15 Multclasificador IPC	Conjunto total de imágenes	Conjunto de entrenamiento fijado por pruebas incrementales %	Converge alrededor de la imagen	Ajuste manual del conjunto de entrenamiento %	Converge alrededor de la imagen
4 clases desbalanceado	1107	14	24	15	100
5 clases desbalanceado	1149	31	30	35	50
4 clases balanceado	428	30	32	35	39
5 clases balanceado	210	NA	NA	NA	NA

Ante esto, se muestran los resultados gráficos obtenidos en las Figuras 4.36 y 4.37, recordando que las diferentes curvas presentadas se han trazado un número considerable de veces para los conjuntos de imágenes con cuatro y cinco clases tanto balanceados como desbalanceados, sólo para el detalle V.

De la inspección realizada sobre los resultados gráficos se pudo extraer el valor asociado al número de imagen donde comienza aproximadamente la convergencia hacia altas tasas de acierto en la predicción de imágenes. Estos valores fueron anotados en la columna 4 de la misma Tabla 4.20 para contrastarlo con los conjuntos de entrenamiento (porcentaje de imágenes) fijados por las pruebas incrementales.

Apréciase en esta oportunidad, que la convergencia inicial en los multclasificadores IPC no acepta ajustes adicionales en sus conjuntos de entrenamiento, siendo suficiente los valores fijados por incremento para que estos predictores alcancen porcentajes de acierto a muy tempranas muestras por encima del 90%.

Análisis de los resultados obtenidos en la Prueba 15

- Al observar las Figuras 4.36 y 4.37 y el resumen mostrado en la Tabla 4.20 para los multclasificadores IPC por lote con cuatro y cinco clases desbalanceados y cuatro clases balanceado, se puede notar una clara convergencia del porcentaje de acierto alrededor del 90% a muy tempranas muestras con una tendencia final alrededor del 95%.
- Se puede corroborar en la propia Tabla 4.20 (columnas 5 y 6) para estos tres multclasificadores IPC presentados, que los porcentajes de imágenes por lote fijados inicialmente en los entrenamientos incrementales no permiten ajustes manuales posteriores para mejorar el número de muestras que deben ser procesadas antes de alcanzar una convergencia por encima del 90%. Por ejemplo si se observa la primera fila (Tabla 4.20), se puede notar que al ajustar el porcentaje inicial de 14% a 15% la convergencia pasa de esperar 24 a 100 muestras, indicando esto una desmejora del predictor. Para los otros dos multclasificadores (fila 2 y 3), si se propicia un ajuste de 31% y 30% a 35% (en ambos), se observa un retraso de 30 y 32 a 50 y 39 muestras respectivamente.
- Para el multclasificador IPC por lote con cinco clases balanceado, no se registra convergencia del porcentaje de acierto por encima del 90%.
- Se observa para todos los multclasificadores IPC presentados exceptuando el de cinco clases balanceadas, que la medida de confianza media aumenta situándose muy próxima al valor de uno. Por su parte, la credibilidad sigue ubicada alrededor de 0.45, y sigue siendo dispersa para el conjunto con cuatro clases balanceadas.

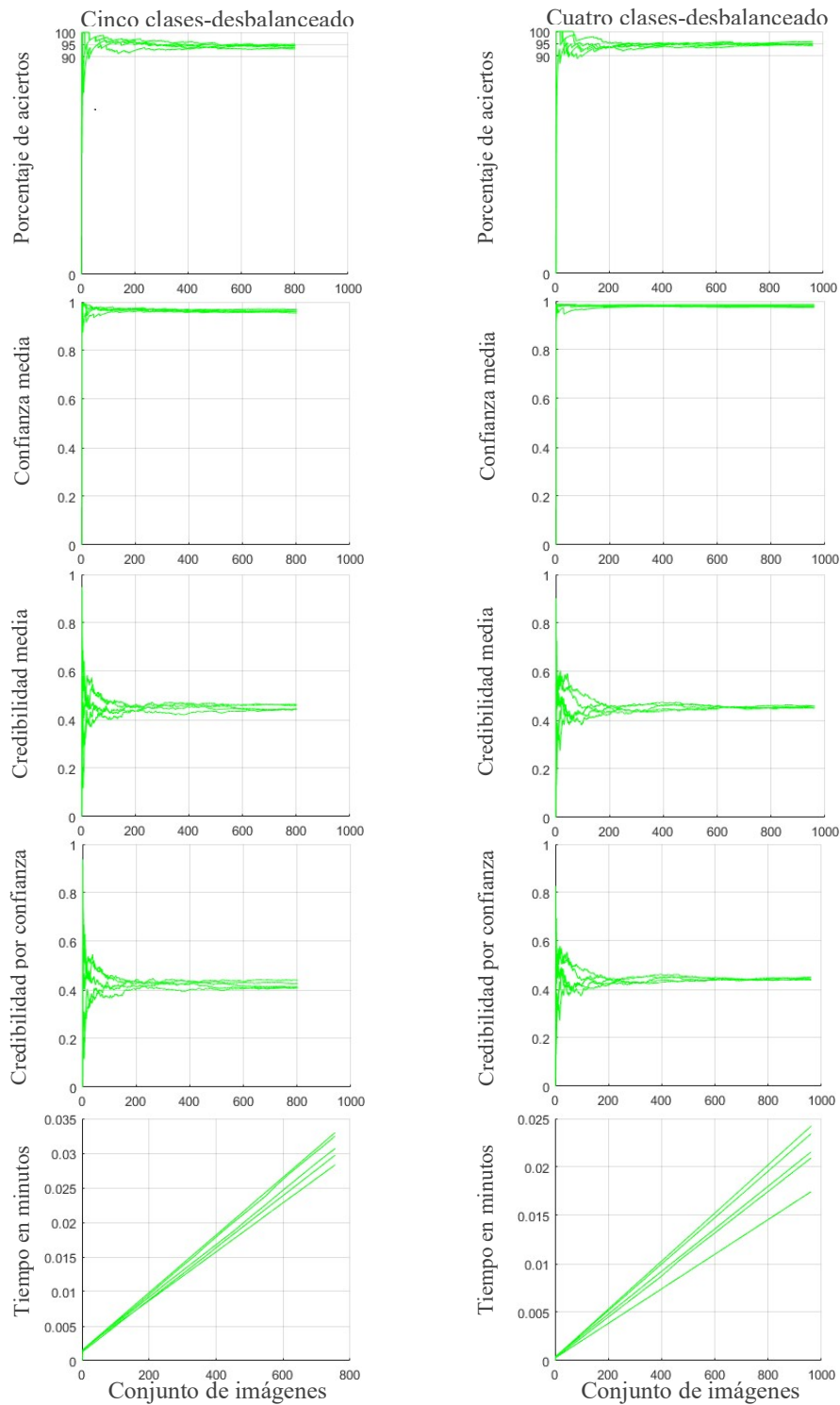


Figura 4.36.- Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador IPC por lote para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

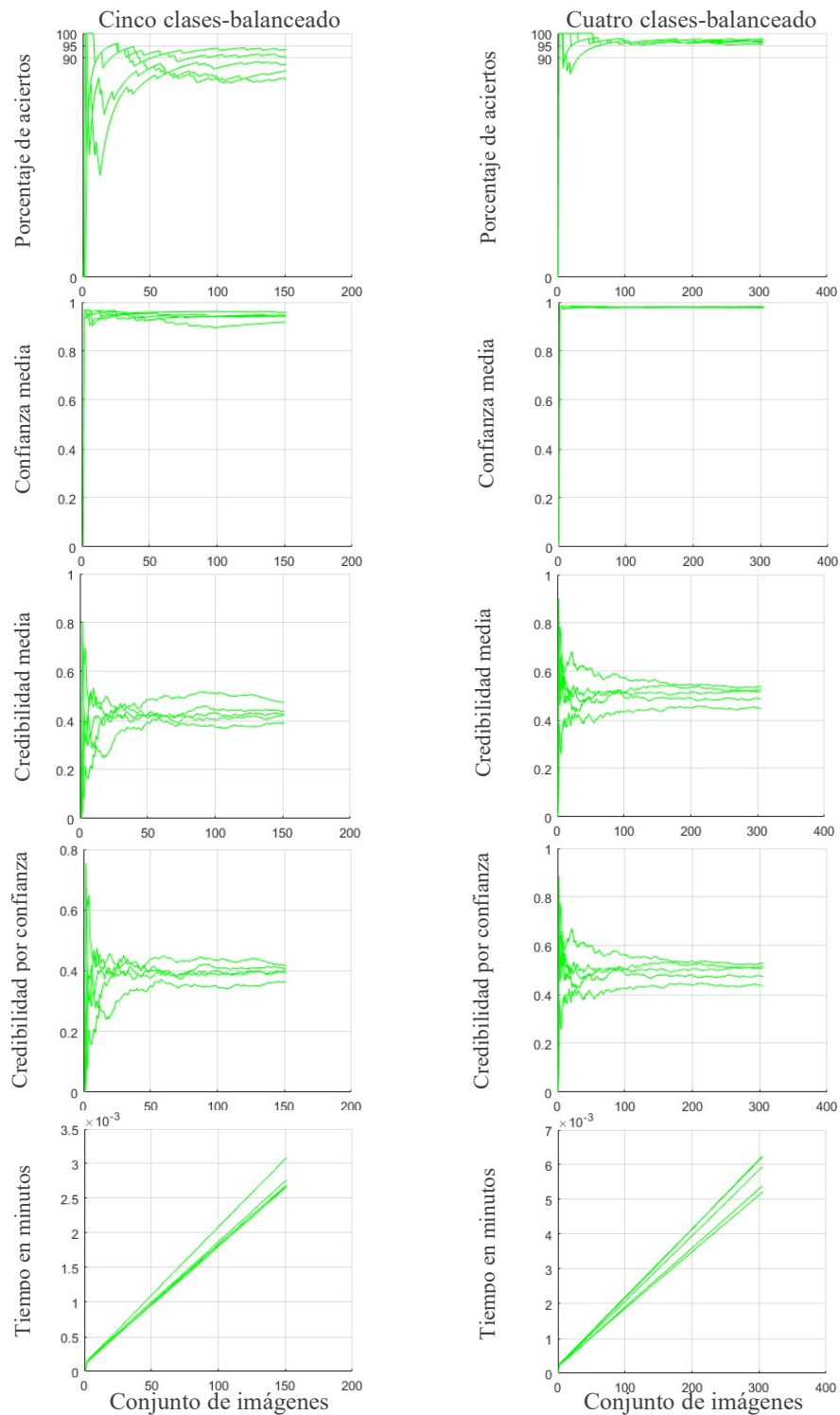


Figura 4.37.- Resultados obtenidos al ejecutar cinco veces el programa asociado al multclasificador IPC por lote para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

4.10 MULTICLASIFICADOR CON PREDICCIÓN VENN

En esta sección se presentarán los experimentos realizados y los resultados obtenidos con la predicción VENN o multclasificador PV. Estos ensayos se dividirán en dos partes por el uso de dos tipos de taxonomías, probadas sobre el conjunto de imágenes provenientes del dispositivo de fusión termonuclear TJ-II ubicado en el CIEMAT-España. Las taxonomías se refieren a los métodos de agrupamiento utilizados: k vecinos más cercanos (KNN) y el centroide más cercano (CNN).

Los ensayos con el multclasificador PV también siguieron la agrupación de imágenes propuesta en la Figura 4.1 para cuatro y cinco clases con datos balanceados y desbalanceados según la Tabla 4.1, examinándose sólo conjuntos de imágenes con detalle V. Al mismo tiempo se recuerda que, el funcionamiento de todo el sistema predictivo VENN así como sus taxonomías fueron discutidas en profundidad en la sección 3.9 del Capítulo III.

4.10.1 Multclasificador incremental y por lote con PV-KNN

Como en los sistemas predictivos ya analizados, los ensayos con el multclasificador PV se inician con el entrenamiento y pruebas de clasificación incremental para poder fijar el conjunto inicial mínimo de imágenes (porcentaje) necesarias para entrenar el multclasificador PV por lote y así conseguir altas tasas de aciertos en las pruebas de clasificación. Además de verificar buenos intervalos de probabilidades.

Prueba 16.- Resultados obtenidos en la clasificación incremental para cuatro-cinco clases balanceado-desbalanceado

En la Tabla 4.21 se pueden ver los conjuntos iniciales de imágenes de entrenamiento incremental y los conjuntos el conjunto reservado para pruebas incrementales que son procesador por la implementación del programa mostrado en la Figura 4.38, colocando los resultados gráficos en las Figuras 4.39 y 4.40. Se aclara que no fue ejecutado el programa asociado al multclasificador PV-KNN con cuatro y

cinco clases desbalanceadas más de una vez por el costo computacional que esto sugiere.

Para ver dicho costo computacional, en la Figura 4.39 se pueden observar los registros de tiempo para una ejecución alcanzando más de 800 y 600 minutos, representando esto más de 13 y 10 horas de cálculo realizado por cada predictor respectivamente. En consecuencia, los puntos fijados por incremento para encontrar el porcentaje de imágenes a usar en los entrenamientos y pruebas por lote sólo se usan como referencia, entendiéndose que luego se debe realizar un ajuste manual.

Tabla 4.21.- Cantidades de imágenes dispuestas para los entrenamientos y pruebas incrementales con el fin de establecer el conjunto inicial de imágenes de entrenamiento con el multclasificador PV-KNN.

Prueba 16 Multclasificador PV-KNN	Conjunto total de imágenes	Conjunto inicial de entrenamiento incremental	Conjunto reservado para pruebas incrementales	Conjunto donde puede iniciar el entrenamiento por lote
4 clases desbalanceado	1107	20	1087	Alrededor 5
5 clases desbalanceado	1149	25	1024	Alrededor de 44
4 clases balanceado	428	20	408	Alrededor de 125
5 clases balanceado	210	25	185	Alrededor de 125

Análisis de los resultados obtenidos en la Prueba 16

- En la Figura 4.39 se observa para los multclasificadores PV-KNN con cuatro y cinco clases desbalanceadas, un alto porcentaje de acierto que va más allá del 95% con un intervalo de probabilidades por el orden del 90%. Sin embargo y en detrimento de esto, se observa un lento proceso al computarse tiempos excesivos como los ya comentados en párrafos anteriores.
- En la Figura 4.40 se observan los resultados para los multclasificadores PV-KNN con cuatro y cinco clases balanceadas. Particularmente, se puede notar en los transitorios que se originan al procesar los conjuntos de imágenes para cinco clases balanceados que si convergen hacia el 90% del porcentaje de acierto, diferenciándose de los multclasificadores PC e IPC con cinco clases balanceadas porque no habían conseguido altos valores.

- Como se verá en la Prueba 17, los porcentajes fijados por los ensayos incrementales para los multclasificadores PV-KNN desbalanceados, no son suficientes para conseguir altas tasas de aciertos en los entrenamientos y pruebas de clasificación por lote. Sin embargo, esta incapacidad no se le debe atribuir al entrenamiento incremental en sí, ya que no se pudieron realizar pruebas suficientes por los tiempos excesivos que necesita el método para alcanzar un resultado razonable.

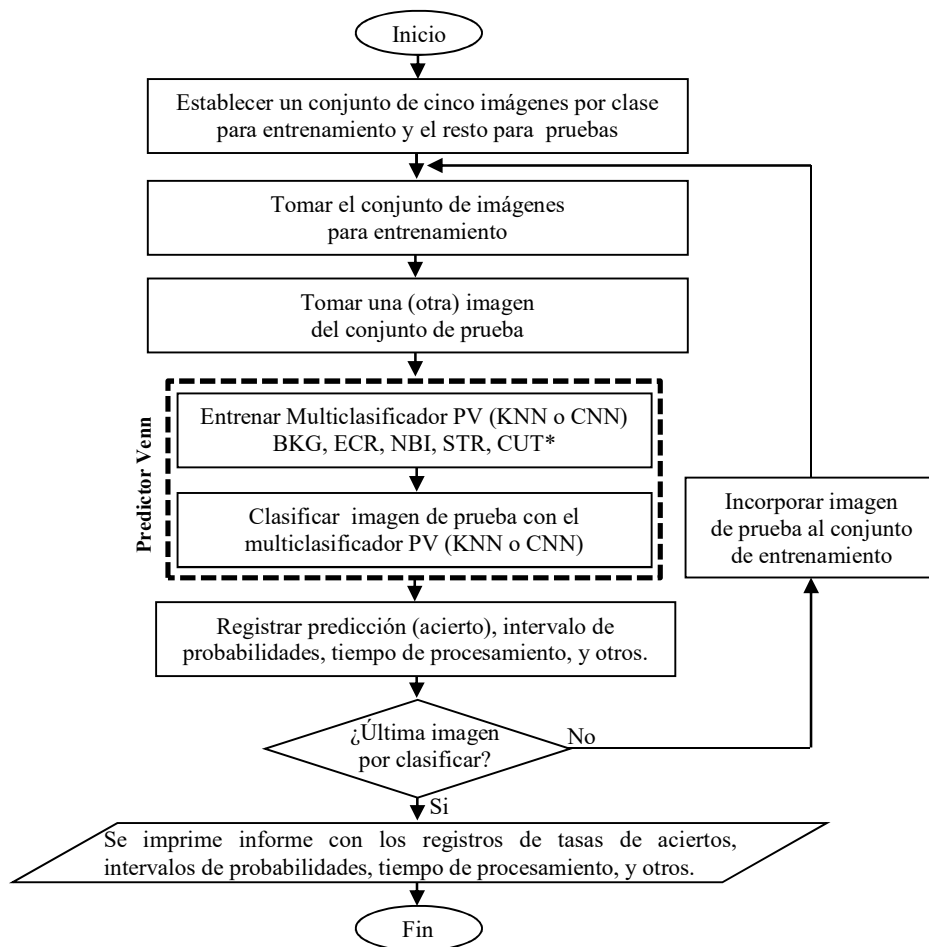


Figura 4.38.- Diagrama de bloques general de la implementación del programa que clasifica imágenes por incremento a partir del multclasificador PV con taxonomía KNN y CNN.

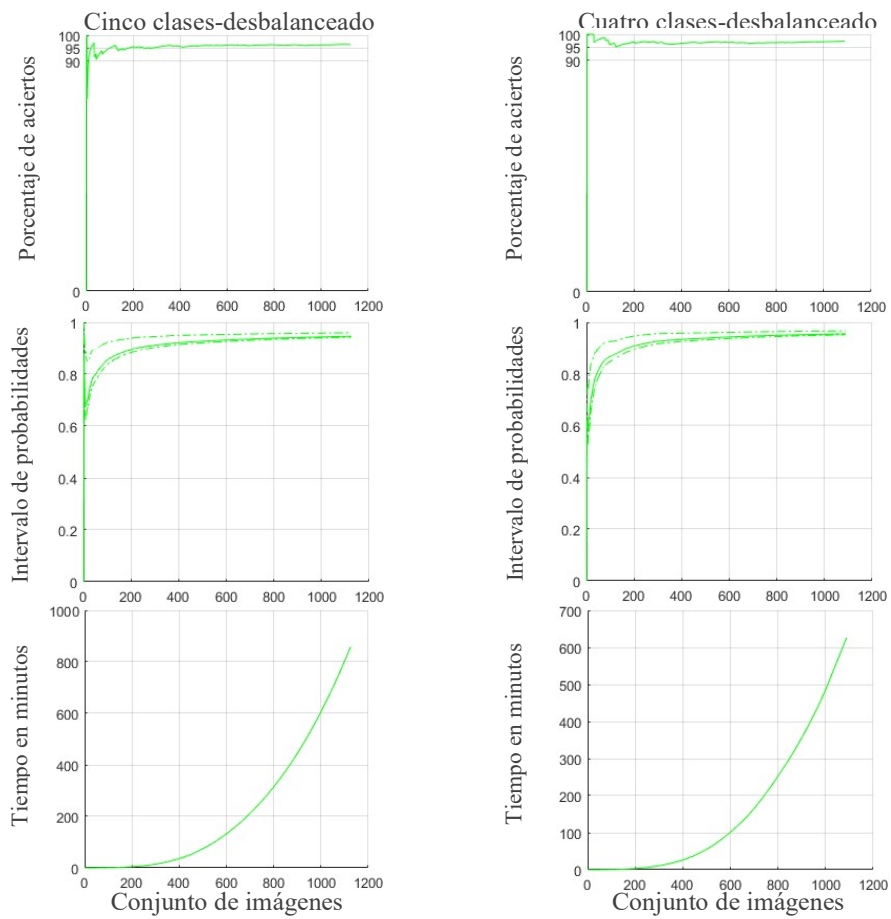


Figura 4.39.- Resultados obtenidos al ejecutar la implementación del programa asociado al multclasificador PV-KNN incremental para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V.

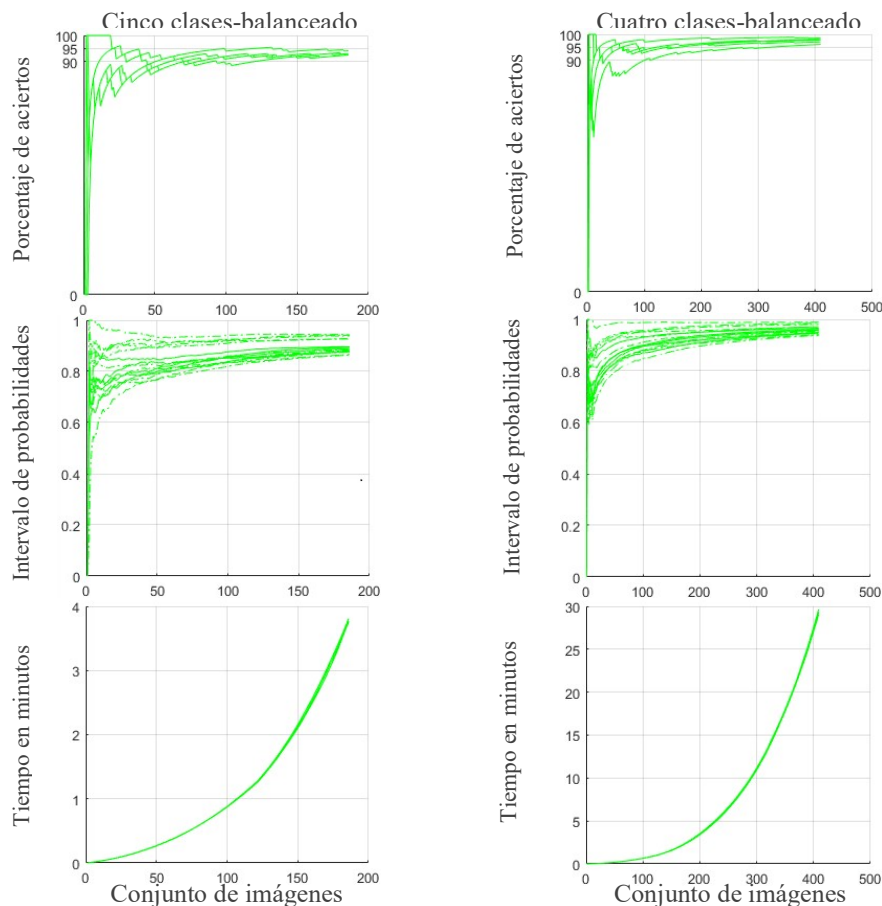


Figura 4.40.- Resultados obtenidos al ejecutar cinco veces la implementación del programa asociado al multclasificador PV-KNN incremental para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V.

Prueba 17.- Resultados obtenidos en la clasificación por lote para cuatro-cinco clases balanceado-desbalanceado

En esta prueba se presentan los resultados obtenidos con el multclasificador PV-KNN por lote, mediante la implementación del programa mostrado a través del diagrama de flujo de la Figura 4.41. En esta implementación se establecen en principio, los conjuntos de imágenes fijados por los entrenamientos incrementales según lo indicado en la Tabla 4.22. Se puede apreciar en la columna 4 de la mencionada tabla, que sólo el conjunto con 4 clases balanceadas converge a un

porcentaje de acierto por encima del 90%. Esto obliga a realizar ajustes consecuentes en los conjuntos iniciales de entrenamiento, mostrándose los mejores resultados en la columna 5. Correspondientemente, en la columna 6 se observa el número de imágenes que deben ser procesadas por el multclasificador PV por lote para alcanzar la convergencia a altas tasas de aciertos. Estos resultados favorables propiciados por el ajustes realizado pueden ser comprobados observando las Figuras 4.42 y 4.43.

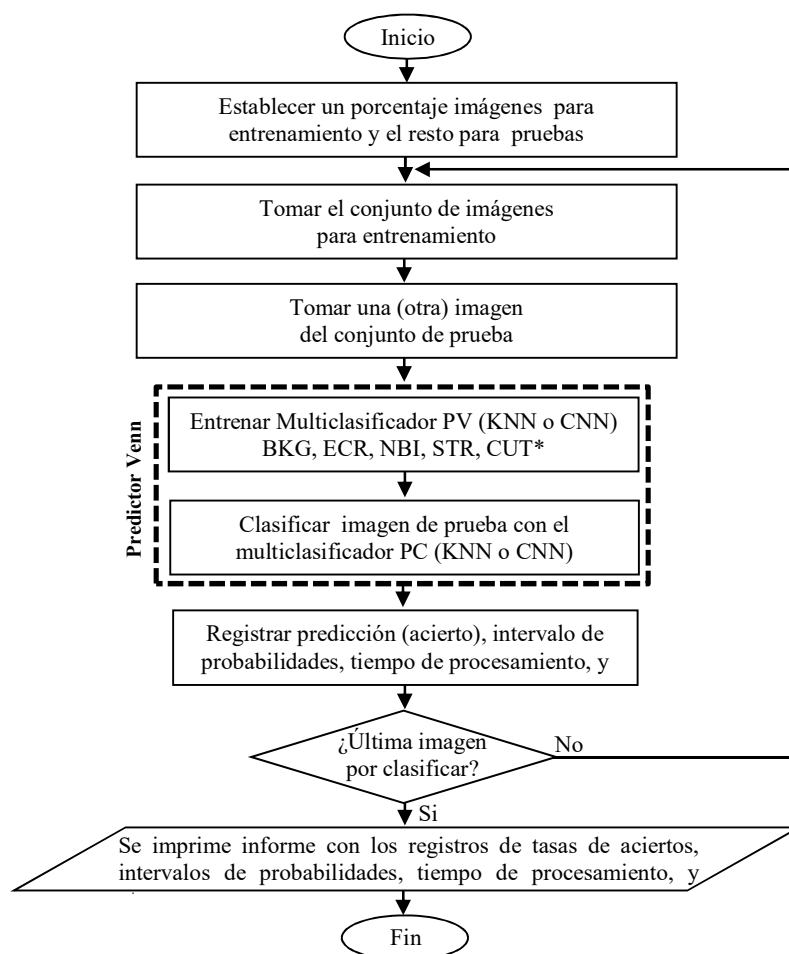


Figura 4.41.- Diagrama de bloques general de la implementación del programa que clasifica imágenes por lote a partir del multclasificador PV con taxonomía KNN y CNN.

Tabla 4.22.- Cantidades de imágenes dispuestas para los entrenamientos y pruebas por lote con el multclasificador PV-KNN y su convergencia hacia altas tasas de aciertos en la Prueba 17.

Prueba 17 Multclasificador PV-KNN	Conjunto total de imágenes	Conjunto de entrenamiento fijado por pruebas incrementales %	Converge alrededor de la imagen	Ajuste manual del conjunto de entrenamiento %	Converge alrededor de la imagen
4 clases desbalanceado	1107	Menos del 5	NA	15	53
5 clases desbalanceado	1149	4 a 6	NA	20	40
4 clases balanceado	428	30	22	32	21
5 clases balanceado	210	60	NA	65	NA

Análisis de los resultados obtenidos en la Prueba 17

- Al observar la Figura 4.42 y el resumen mostrado en la Tabla 4.22 para el multclasificador PV-KNN por lote con cuatro y cinco clases desbalanceadas, se nota un porcentaje de acierto alrededor del 95%, con una convergencia inicial que se alcanza a 40 y 53 muestras procesadas respectivamente.
- El resultado presentado anteriormente, cuenta con una alta probabilidad de ocurrencia alrededor del 96%, con un estrecho intervalo de probabilidades en detrimento de un largo tiempo de procesamiento que alcanza 23 y 60 minutos aproximadamente para los predictores con cuatro y cinco clases respectivamente.
- Para el multclasificador con cuatro clases balanceado, también se muestra un porcentaje de acierto alto alrededor del 95%, y similarmente una probabilidad alta. Por las pocas muestras procesadas, se alcanza un tiempo de procesamiento de 4 minutos. En cambio, para el sistema predictivo balanceado con cinco clases no se tiene una convergencia clara hacia altas tasas de aciertos (por encima del 90%), pero el registro de probabilidades es alto confirmando tal resultado (véase la Figura 4.43)
- Se concluye que los multclasificadores PV-KNN desbalanceados son buenos predictores pero poco prácticos en su implementación, al alcanzar tiempos de

procesamiento excesivos que van en función del número de imágenes que se incorporan al conjunto de entrenamiento.

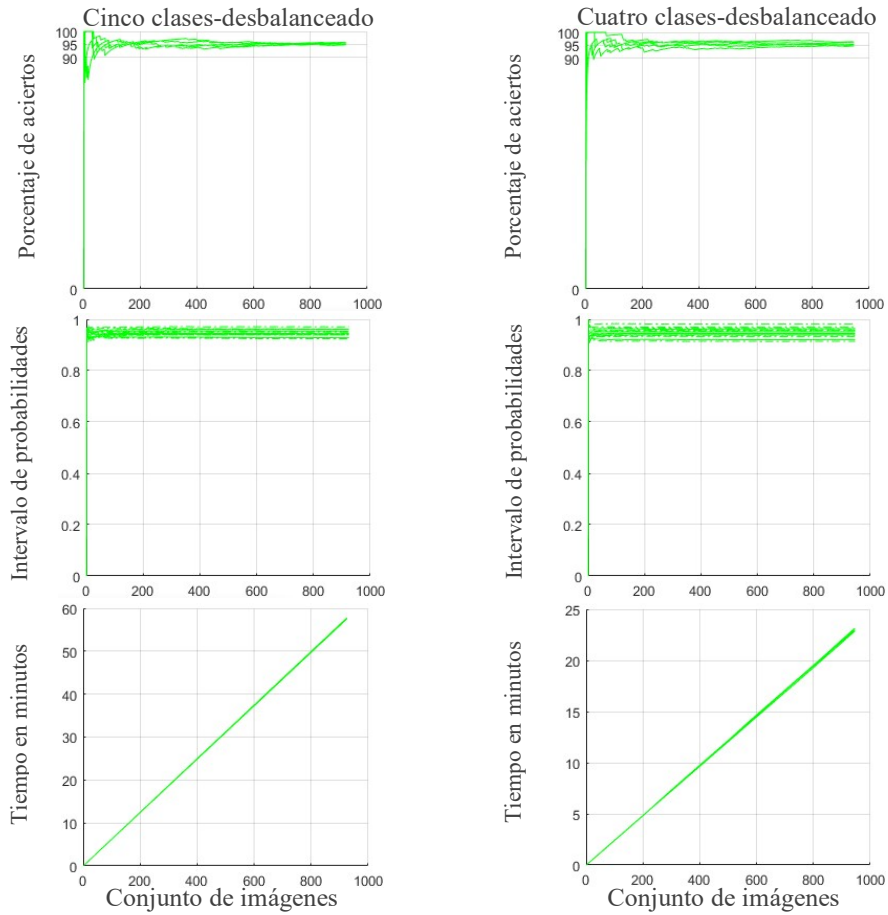


Figura 4.42.- Resultados obtenidos al ejecutar 5 veces el programa asociado al multclasificador PV-KNN por lote para los conjuntos de imágenes con cuatro y cinco clases desbalanceadas usando sólo el detalle V.

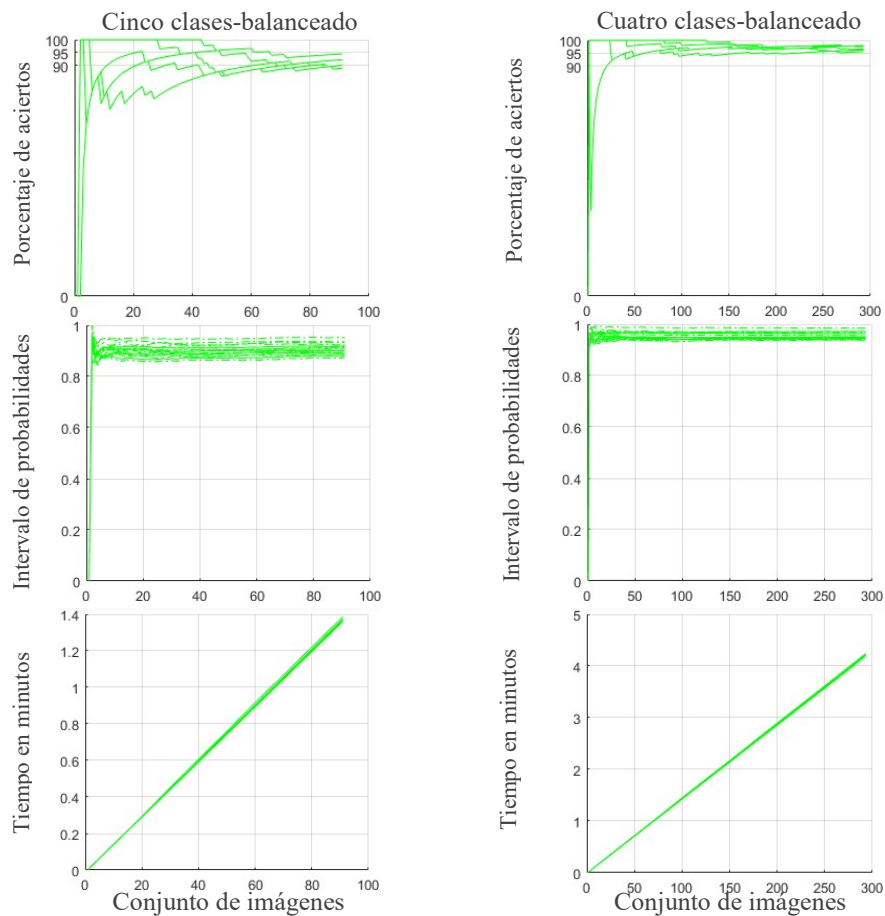


Figura 4.43.- Resultados obtenidos al ejecutar 5 veces el Programa asociado al multclasificador PV-KNN por lote para los conjuntos de imágenes con cuatro y cinco clases balanceadas usando sólo el detalle V.

4.10.2 Multclasificador incremental y por lote con PV-CNN

Para el multclasificador PV-CNN, los experimentos realizados revelan que no se consiguen altos porcentajes de aciertos en ninguna de sus cuatro presentaciones incrementales. Esto se afirma con un 70% de probabilidades obtenidas con un intervalo de confianza del 10% aproximadamente para los sistemas balanceados y que se reduce a un 5% para los sistemas desbalanceados.

Estos resultados obtenidos demuestran que la versión del predictor Venn usando la taxonomía del centroide más cercano, no es propicia frente a la taxonomía del vecino más cercano.

CAPÍTULO V

5 ANÁLISIS DE LOS RESULTADOS

5.1 INTRODUCCIÓN

En este capítulo se presenta el contraste de los resultados obtenidos en el Capítulo 4 al clasificar un conjunto de imágenes provenientes del diagnóstico de esparcimiento Thomson (que opera en el dispositivo de fusión termonuclear TJ-II ubicado en el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT) de España)), mediante la aplicación de los multclasificadores desarrollados, relacionados con los cuatro métodos referidos a: la máquina de vectores soporte (SVM) que permitió la construcción de un multclasificador “clásico” donde sólo se obtienen predicciones; la predicción conformal (PC), la predicción conformal inductiva (IPC) y la predicción Venn (PV), que permitieron construir “predictores de confianza” donde además de la predicción se obtienen medidas de fiabilidad.

A partir de los resultados obtenidos por el contraste, se discutirán varios aspectos que surgen de los propios hechos experimentales y considerados importantes como punto de referencia para futuras investigaciones. Entre los aspectos a considerar se destacan los siguientes: importancia del procesamiento aleatorio, importancia de repetir un experimento más de una vez, importancia de ajustar manualmente el porcentaje fijado por las pruebas incrementales para seleccionar el conjunto de imágenes usado en los entrenamientos por lote con el objetivo de lograr una efectiva convergencia hacia altas tasas de aciertos en un multclasificador, y el efecto del sorteo aleatorio sobre las imágenes.

Finalmente se recuerda que, este capítulo no se dedicará a dar detalles sobre los experimentos desarrollados, ya que estos se pueden consultar a lo largo del Capítulo IV, así como no se darán explicaciones teóricas sobre los multclasificadores, ni de la base teórica de los modelos en lo que se fundamentan los

mismos, puesto que en el Capítulo III se puede encontrar todo un estudio amplio con ejemplos prácticos.

5.2 CONTRASTE DE LOS RESULTADOS OBTENIDOS EN LOS EXPERIMENTOS CON IMÁGENES PROVENIENTES DEL DIAGNÓSTICO DE ESPARCIMIENTO THOMSON

El contraste se realiza en base a los resultados de aplicar los multclasificadores SVM, PC, IPC y PV sobre un conjunto de 1149 imágenes pertenecientes a cinco clases con el fin de clasificarlas. Las imágenes se agruparon en cuatro y cinco clases formando *grupos balanceados y desbalanceados*⁵², para realizar experimentos incrementales y por lote.

De manera general, un experimento incremental⁵³ permite entrenar y probar a un multclasificador, con el objetivo de observar la evolución que va sufriendo el porcentaje de acierto versus el número de imágenes. Con esto se logra conocer el punto donde se tiene una alta convergencia hacia un determinado porcentaje de acierto y el número mínimo de imágenes necesarias para provocar esta condición favorable.

Hay varias consideraciones que deben ser tomadas en cuenta ante un entrenamiento y prueba incremental, estas son:

- El punto de alta convergencia, es fijado por el usuario y “seguramente” lo determina el tipo de aplicación a resolver. Es decir, si se quieren predicciones por encima del 90%, entonces se debe rastrear un punto cuando los porcentajes de aciertos se sitúen en ese valor. Esto se visualiza el futuro desarrollo de un multclasificador por lote, para que no cometa errores mayores al 10% en la predicción de imágenes.
- El conjunto inicial mínimo de imágenes para entrenar un multclasificador por lote y así resolver una aplicación de clasificación en particular, es

⁵² Véase la Figura 4.1 en la sección 1 del Capítulo IV para conocer la agrupación de imágenes y la Tabla 4.1 para verificar sus cantidades.

⁵³ Se puede encontrar información detallada sobre los experimentos incrementales y por lote en el Capítulo IV.

establecido por la convergencia hacia el valor del porcentaje de acierto prefijado. Por ejemplo, si la convergencia al 90% ocurre en la muestra 50 en un entrenamiento incremental, entonces se debe escoger como conjunto inicial mínimo de imágenes para entrenar tal multclasificador por lote, un conjunto con 50 imágenes. Esto pre visualiza un error en la predicción de imágenes menor o igual al 10%.

- El punto que establece un entrenamiento incremental para encontrar un conjunto mínimo de imágenes para el entrenamiento por lote, en ocasiones es orientativo y debe ser ajustado (por eso se indica que pre visualiza). Por lo tanto, todo entrenamiento incremental conduce a encontrar un multclasificador óptimo en dos etapas: una primera etapa que se denomina ajuste grueso, y es establecida por la convergencia hacia un alto valor del porcentaje de acierto, tomándose el número de imágenes coincidente como conjunto inicial mínimo para realizar un entrenamiento por lote. Una segunda etapa que se denomina ajuste fino, y simplemente es el compromiso consecuente que debe propiciarse sobre el punto establecido por el ajuste grueso para garantizar un mejor conjunto de imágenes para realizar el entrenamiento por lote. Esto indica que no todo conjunto de imágenes para realizar un entrenamiento y pruebas por lote prefijado por un entrenamiento incremental es válido, sino más bien orientativo.
- Para los casos de los multclasificadores PC, IPC y PV, las medidas de fiabilidad registradas en un entrenamiento incremental, se mantienen cercanas a las conseguidas cuando se realiza el entrenamiento por lote. Esto indica que el entrenamiento incremental también orienta en los valores que pueden esperarse en las medidas de fiabilidad.
- No se emite ninguna valoración para el registro de tiempo dado por los entrenamientos incrementales, salvo que consumen tiempos excesivos de procesamiento como es el caso de la PC y la PV-KNN.
- Finalmente, se puede decir que un entrenamiento incremental pre visualizará la construcción de un multclasificador óptimo. Atribuyendo el concepto de

óptimo al clasificador implementado con un mínimo número de muestras pero que alcance altas tasas de aciertos y tiempo de procesamiento mínimo. Esto sin la inclusión de medidas de fiabilidad.

Vistas las consideraciones sobre el entrenamiento incremental, se presentan a continuación, los contrastes de los resultados obtenidos en los entrenamientos y pruebas por lote, recogiendo para ello datos numéricos de los multclasificadores SVM, PC, IPC y PV-KNN obtenidos en el Capítulo IV para su discusión.

De esta manera se puede ver en la Tabla 5.1 para los multclasificadores SVM, PC, IPC y PV-KNN por lote, el contraste entre: el porcentaje o conjuntos de imágenes usados para su entrenamiento y prueba, el número de imágenes donde comienza la *convergencia a un alto valor del porcentaje de acierto*⁵⁴, el porcentaje de acierto final y el tiempo de procesamiento aproximado.

Tabla 5.1.- Resultados obtenidos por los experimentos con los multclasificadores SVM, PC, IPC y PV-KNN en el Capítulo IV-Parte 1.

Multclasificador	%Imágenes para Entrenamiento	#Imágenes para convergencia	%Acierto	Tiempo de Procesamiento Aprox	%Imágenes para Entrenamiento	#Imágenes para convergencia	%Acierto	Tiempo de Procesamiento Aprox	%Imágenes para Entrenamiento	#Imágenes para convergencia	%Acierto	Tiempo de Procesamiento Aprox	%Imágenes para Entrenamiento	#Imágenes para convergencia	%Acierto	Tiempo de Procesamiento Aprox
4CD	27	50	96	2.3	20	5	96	3.3	14	24	95	0.02	15	53	96	23
5CD	45	50	94	3.0	26	50	94	6.0	31	30	94	0.03	20	40	95	58
4CB	27	50	95	0.4	50	12	97	0.7	30	32	96	0.01	32	21	96	4.3
5CB	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
MC	SVM				PC				IPC				PV-KNN			

Del análisis de la información suministrada por la Tabla 5.1, se concluye lo siguiente:

- Todos los multclasificadores obtienen altos porcentajes de acierto sobre el

⁵⁴ Recuérdese que es el punto donde el porcentaje de acierto supera irreversiblemente el 90%.

conjunto de imágenes tratado, existiendo en promedio un 95.3% en las estimaciones. Siendo la diferencia más acentuada con relación a la media para el multclasificador PC con cuatro clases balanceadas (4CB). Aunque para los conjuntos completos (5CD: cinco clases desbalanceadas) el predictor PV-KNN resalta con un 1% por encima de sus similares. Véase las columnas 4, 8, 12 y 16 de la Tabla 5.1.

- Con relación al número de imágenes empleadas para el entrenamiento de los multclasificadores por lote (porcentaje del *conjunto total*⁵⁵), se nota que al procesar los conjuntos de imágenes completos (4CD y 5CD), el multclasificador SVM requiere siempre de más muestras. Esto coloca a este predictor en desventaja con relación al resto de sus similares (PC, IPC y PV-KNN). Véase las columnas 2, 6, 10 y 14 de la Tabla 5.1.
- Se observa para la conversión hacia altas tasas de aciertos, que el multclasificador IPC 5CD es el que menos muestras debe dejar pasar para comenzar a realizar predicciones con un error menor o igual al 10%. Es importante puntualizar, que esto es relativo al ajuste efectuado sobre el conjunto de muestras para entrenamiento, establecido en los entrenamientos incrementales tal y como se estudió en el Capítulo IV. Véase las columnas 3, 7, 11 y 15 de la Tabla 5.1.
- Según los datos extraídos de los experimentos realizados en el Capítulo IV con las imágenes tratadas, el multclasificador IPC se convierte en el más rápido para los diferentes conjuntos de imágenes (4CD, 5CD y 4CB). Esto lo cataloga como el más viable a la hora de implementar un predictor para las muestras observadas, ya que no sólo en esta característica supera a sus similares sino que mantiene ventajas en las características comentadas anteriormente. Además se puede decir que el multclasificador PV-KNN, aunque se comporta muy cercano al predictor IPC, su implementación sería poco práctica por ser el que más tiempo de proceso consume. Compare las

⁵⁵ Véase la tabla 4.1 del Capítulo IV para conocer las cantidades de imágenes por clase a ser usadas según el tipo de multclasificador a entrenar.

columnas 5, 9, 13 y 17 de la Tabla 5.1 para corroborar esto.

También se puede ver en la Tabla 5.2 para los multclasificadores PC e IPC por lote: la confianza media, la credibilidad media y la confianza media por la credibilidad media. Al mismo tiempo se observa para el multclasificador PV-KNN por lote: la probabilidad y los intervalos de probabilidades Alto y Bajo. Estas variables se contrastan recordando que el multclasificador SVM no registra ninguna de estas medidas de fiabilidad.

Tabla 5.2.- Resultados obtenidos por los experimentos con los multclasificadores SVM, PC, IPC y PV-KNN en el Capítulo IV-Parte 2.

Multclasificador	SVM no presenta medidas de fiabilidad	Confianza media	Credibilidad media	Conf. por cred.	Confianza media	Credibilidad media	Conf. por cred.	Intervalo-Alto	Probabilidad	Intervalo-Bajo
4CD		0.90	0.50	0.40	0.99	0.43	0.43	0.98	0.95	0.92
5CD		0.90	0.43	0.38	0.97	0.43	0.41	0.97	0.95	0.93
4CB		0.95	0.50	0.43	0.97	0.50	0.50	0.95	0.97	0.99
5CB		NA	NA	NA	NA	NA	NA	NA	NA	NA
MC	SVM	PC			IPC			PV-KNN		

Del análisis de la información suministrada por la Tabla 5.2, se concluye que:

- SVM no tiene ventajas con relación al aporte de medidas de fiabilidad, que verifiquen la calidad de las estimaciones hechas por los multclasificadores desarrollados en base a este método predictivo.
- Se observa para los multclasificador IPC con datos completos (4CD y 5CD), una confianza mayor que para los predictores PC. Esto indica que la IPC es mucho más selectiva al descartar muestras poco informativas. Compárese las columnas 5 y 6 de la Tabla 5.2.
- Con relación a la credibilidad, sólo se observa diferencias en los multclasificadores con cinco clases desbalanceadas (5CD). Notándose para este conjunto completo, que la PC indica una mayor conformidad de cada

objeto predicho con relación al conjunto usado para hacer las estimaciones.

- Por su parte, la medida de confianza media por credibilidad media, indica que el mejor comportamiento al realizar predicciones confiables lo tienen los multclasificadores IPC.
- Estos tres últimos comentarios (6, 7, 8) dejan ver que, el procedimiento aplicado por la predicción conformal inductiva al dividir los datos usados en dos conjuntos para la etapa de entrenamiento, no afecta su rendimiento en las predicciones hechas; pese a que tal procedimiento, atenúa la utilización propiamente de un conjunto menor de información, como se vio en los Capítulos III y IV.
- En cuanto al multclasificador PV-KNN, sus medidas de fiabilidad son muy buenas al indicar altas probabilidades de ocurrencia en las estimaciones de imágenes. Aunque sólo ocupa un intervalo estrecho, los referidos a los conjuntos completos 5CD y 4CB. Sin embargo, en cualquier caso el error cometido nunca supera el 8%.

Conclusiones según el contraste realizado

Para finalizar, y basándose en el contraste realizado por los resultados obtenidos en los experimentos con las imágenes provenientes del diagnóstico de esparcimiento Thomson mediante el análisis de cuatro multclasificadores analizados en el marco del estudio de los “modelos de clasificación con medidas de fiabilidad basados en predictores conformales y aplicados a” cinco tipos de “imágenes de fusión nuclear”, que planteó el objetivo de “cómo obtener un multclasificador que con el menor tiempo de procesamiento en la fase de entrenamiento, proporcione el mejor rendimiento en términos de tasa de aciertos y fiabilidad”, se concluye que:

- Los multclasificadores IPC contruidos a partir de la predicción conformal inductiva, permiten altas tasas de aciertos en la clasificación de imágenes con buena calidad de las estimaciones obtenidas. En general se puede decir que: el multclasificador IPC permite altas tasas de aciertos, con un menor número de muestras para su entrenamiento. Alcanza mucho más rápido la convergencia hacia altas tasas de acierto. El tiempo de procesamiento de imágenes en

comparación con otros métodos, es inmediato. Si se observan las medidas de fiabilidad, los multclasificadores IPC registran mejores tasas de confianza, con una credibilidad aceptable, poniéndose de manifiesto que son buenos predictores. Véase los comentarios 1, 2, 3, 4 y 6, 7, 8 y 10 según el análisis realizado sobre las Tablas 5.1 y 5.2 respectivamente.

- Los multclasificadores IPC no presentan el inconveniente del coste computacional. Esto se demuestra al comparar los resultados presentados en la columna 13 con las columnas 5, 9 y 17 de la Tabla 5.1, notándose indiscutiblemente que los multclasificadores IPC desarrollados en base a la predicción conformal inductiva, registran los tiempos (en minutos) más bajos.
- Mediante las pruebas incrementales y por lote se puede establecer un procedimiento para obtener multclasificadores con un tiempo de procesamiento mínimo en la fase de entrenamiento, garantizando un buen rendimiento en términos de tasa de aciertos y fiabilidad.

5.3 IMPORTANCIA DEL PROCESAMIENTO ALEATORIO

Un pre-requisito para el procesamiento que se realiza sobre los datos que manejan los métodos predictivos asociados a la predicción conformal, predicción conformal inductiva y predicción Venn, es que los datos tienen que ser del tipo IID (independientes e idénticamente distribuidos) [Vovk, Gammerman y Shafer, 2005]. Para garantizar esto en esta investigación, las imágenes tratadas se obtienen del banco de imágenes disponibles (véase la Figura 4.4 en el Capítulo IV) de forma aleatoria, incluso para el procesamiento que realiza el multclasificador SVM.

Sin embargo, después del pre-procesamiento realizado por la transformada wavelet sobre las imágenes, las observaciones se etiquetan numéricamente según la clase (BKG, ECR, NBI, STR o CUT) a la que pertenecen y de esta manera se controlan, pudiendo conocer con exactitud a que clase pertenece una imagen predicha. Este procedimiento es importante sobre todo porque permite medir la tasa de aciertos en las clasificaciones de imágenes.

Después del etiquetado, las imágenes entran a un vector ordenadas por su etiqueta, es decir, las primeras posiciones del vector son ocupadas por la clase BKG, luego sigue las ECR, y así sucesivamente con el resto de clases NBI, STR y CUT. Seguidamente, estos vectores compuestos por imágenes (en general dos tipos de vectores: uno para entrenamiento y otro para pruebas) son usados finalmente por cada multclasificador SVM, PC, IPC y PV, tanto para realizar los experimentos por incremento como por lote.

Al inicio de esta investigación se creía que era suficiente este tipo de procedimiento⁵⁶ que selecciona el conjunto de imágenes a ser usado, sobre todo, porque se obtenían altas tasas de aciertos en los entrenamientos por lote; aunque el conjunto de imágenes usado para realizar el entrenamiento por lote, no coincidía con el conjunto de imágenes que se establecía en el entrenamiento incremental.

Algo incluso más grave y que posteriormente se analizó, pese a que se obtenían finalmente altas tasas de aciertos con los multclasificadores por lote, era su incapacidad en converger rápidamente a altas tasas de acierto; ya que para cualquier predictor entrenado por lote, al ser evaluado, se debía esperar un número de muestras elevado (*velocidad de aprendizaje*⁵⁷ lenta) para lograr predicciones con errores menores al 10%, aunque los entrenamientos incrementales registraban en algunos casos, conjuntos mínimos de datos para lograr la convergencia rápida a altos porcentaje de aciertos. Este problema adicionalmente introduce *retardos*⁵⁸ en los multclasificadores.

En este punto de la investigación donde se conseguían altas tasas de acierto, pero con unas convergencias lentas (había que esperar muchas muestras), se implementó un programa que por referencia se denominará *ordenador_de_datos_IID*. Este programa toma los vectores etiquetados de imágenes, y vuelve a partir de ese

⁵⁶ El procedimiento comentado, en realidad es algo más complejo, en este punto únicamente es necesario tener una cierta idea del procedimiento.

⁵⁷ Se recuerda que, la velocidad de aprendizaje está relacionada con el número de muestras que se necesitan para entrenar un determinado modelo de clasificación y así obtener altas tasas de acierto.

⁵⁸ Se define como retardo en esta investigación, para un multclasificador entrenado, al tiempo consumido por el número de imágenes que se deben dejar pasar antes de conseguir estimaciones con errores menores al punto de convergencia.

punto, a re-seleccionar aleatoriamente las muestras para realizar el entrenamiento incremental y el entrenamiento por lote del multclasificador a ser analizado. De esta forma, el *ordenador_de_datos_IID* corrige en buena parte, los inconvenientes que son introducidos sobre el porcentaje de aciertos debido a la selección imágenes al finalizar la etapa de pre procesamiento.

En la Figura 5.1 se pueden ver gráficas del procesamiento de imágenes realizado por un multclasificador SVM para cuatro clases desbalanceadas, entrenado tanto por incremento (Figuras 5.1 (a) y (b)) como por lote (Figura 5.1 (c) y (d)), con y sin el *ordenador_de_datos_IID*. Demostrándose con esto, que la implementación propuesta para la selección aleatoria de las muestras, controla en buena parte la distorsión que presentan los transitorios en las gráficas del porcentaje de acierto en la clasificación de imágenes, incluyendo las curvas generadas con el multclasificador SVM.

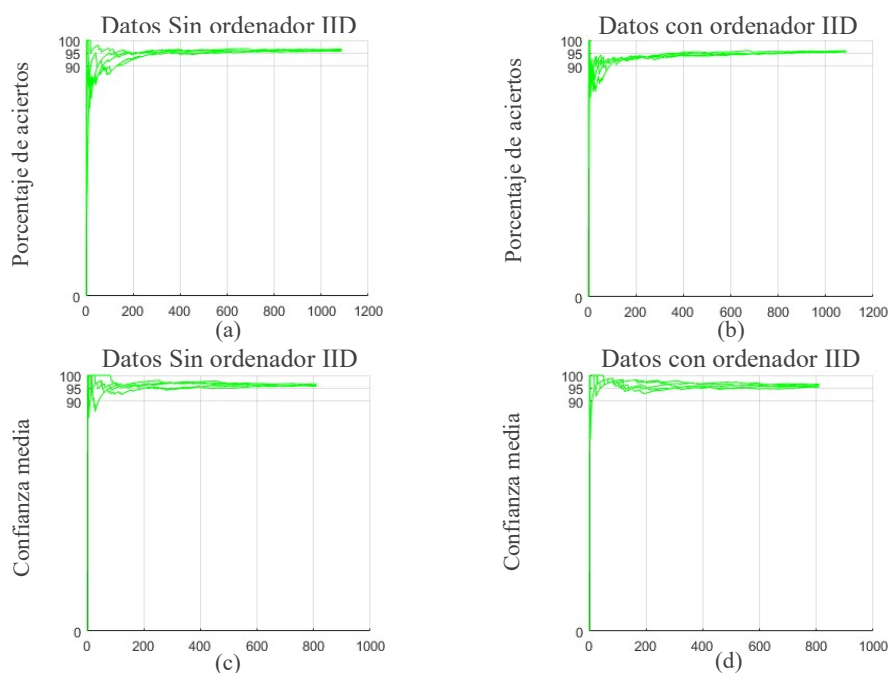


Figura 5.1.- Graficas del porcentaje de acierto obtenido para un Multclasificador SVM sin incluir e incluyendo el ordenador de imágenes IID tanto para el procesamiento de muestras: (a), (b) por incremento, (c), (d) por lote.

En general, el *ordenador_de_datos_IID* al ser insertado en el proceso que forma parte del pre-procesamiento final previo antes de tratar las imágenes, permite:

- Que en los entrenamientos y pruebas de clasificación incremental se pueda realmente fijar un conjunto inicial mínimo de imágenes con el cual se realice el entrenamiento por lote de un multclasificador. Esto se logra porque los transitorios vistos en las curvas de los porcentajes de aciertos, en la mayoría de las figuras presentadas en el Capítulo IV, tienden a convergen más rápido a un punto común (punto de convergencia hacia altas tasas de aciertos).
- Que se pueda ajustar realmente el conjunto mínimo de muestras usadas en el entrenamiento y pruebas por lote, establecido en los entrenamientos y pruebas incrementales. Sin la implementación del programa *ordenador_de_datos_IID*, en la mayoría de los casos el punto se alejaba. Esto se explicará en detalle en la siguiente sección, donde se discute la importancia de repetir un experimento más de una vez, debido a los problemas detectados en la etapa transitoria de las curvas de los porcentajes de aciertos.
- Que en los entrenamientos y pruebas por lote, la velocidad de aprendizaje sea mayor y así alcanzar más rápido el punto de convergencia a altas tasas de aciertos. Sin la implementación del programa *ordenador_de_datos_IID*, los multclasificadores SVM, PC, IPC y PV presentaban una velocidad de aprendizaje lenta, lo que se traduce en dejar pasar muchas muestras antes de realizar estimaciones con errores menores al 10%.
- Los experimentos confirman que los multclasificadores SVM son menos propensos a sufrir estos problemas, pero como se observó en la Figura 5.1, no son inmunes.

5.4 IMPORTANCIA DE REPETIR UN EXPERIMENTO MÁS DE UNA VEZ

Durante la práctica experimental se pudo detectar la necesidad de repetir los experimentos incrementales y por lote más de una vez, debido al sorteo de imágenes para seleccionar los conjuntos de entrenamiento y prueba a ser usados. Este problema

se detectó inicialmente, cuando se observaban los porcentajes de aciertos arrojados en las pruebas incrementales realizadas con conjuntos diferentes de imágenes que se obtenían por el sorteo que da acceso a las mismas en la etapa de pre-procesamiento.

Específicamente en las gráficas de los porcentajes de aciertos, se pudo ver en la etapa transitoria, distintas trayectorias tal y como se observa en la Figura 5.2 extraída de la Figura 4.34 referente al multclasificador IPC incremental con cuatro clases desbalanceadas, que se ha vuelto a colocar aquí para una mejor comprensión de lo que se está discutiendo⁵⁹.

En este ejemplo de la Figura 5.2, se puede verificar para las diferentes curvas, que existen diferentes transitorios donde cada trayectoria cruza la línea de convergencia hacia altas tasas de acierto (situada al 90%) en diferentes puntos. Con cada cruce se fija un punto sobre el eje de las imágenes (abscisas) marcando de esta manera el conjunto de imágenes mínimo que puede ser tomado en un entrenamiento por lote. Esta información se registra en la columna 1 y 2 de la Tabla 5.3.

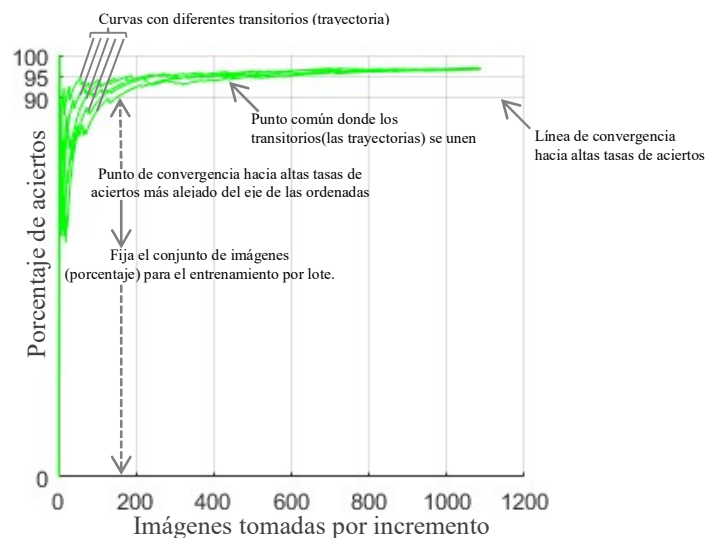


Figura 5.2.- Resultados obtenidos al ejecutar 5 veces el Programa asociado al multclasificador IPC incremental para los conjuntos de imágenes con cuatro clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$.

⁵⁹ Nota: Todos los multclasificadores presentados, sufren este tipo de problemas en mayor o menor proporción.

Tabla 5.3.- Conjuntos de imágenes para entrenar un multclasificador IPC por lote marcado por el cruce del transitorio en las gráficas del porcentaje de acierto de una prueba incremental para cinco sorteos diferentes de imágenes.

Datos para las pruebas por incremento	Datos para las pruebas por lote				
#imagen donde cruzan el porcentaje de acierto y la línea de convergencia	%Imágenes mínimo a tomar para realizar el entrenamiento por lote	#Imágenes convergencia hacia altas tasas de acierto	%Acierto final	Tiempo de procesamiento aproximado (en minutos)	%Confianza por credibilidad
150	14	24	95	0.019	0.46
110	10	240	94	0.015	0.45
50	7	NA	95	0.014	0.45
75	5	No se realiza por el tipo de multclasificador			
35	2				

Lógicamente, si sólo se hiciera una única prueba incremental y no cinco como las que se han mostrado, y luego se toma el porcentaje de acierto que marca en el cruce con la línea de alta convergencia (columna 2 de la Tabla 5.23), se pudiera estar cometiendo un error que tiene una consecuencia inmediata en las pruebas por lote.

Esto puede entenderse por la explicación que se dará a continuación basada en la Figura 5.3 donde se tienen graficas como resultado de usar los diferentes conjuntos de imágenes que se han marcado por el entrenamiento incremental visto en la Figura 5.2, además de justificar que en los entrenamientos por lote, también existe la necesidad de repetir las pruebas más de una vez.

Como puede verse en la columna 2 de la Tabla 5.3, para cada trayectoria generada en el experimento incremental, se toma un conjunto de imágenes (porcentajes: 2%, 5%, 7%, 10% y 14%) para realizar el entrenamiento (y prueba) por lote. Si ahora con el mismo porcentaje fijado por incremento se repiten las pruebas por lote, se podrá ver que siempre las gráficas generadas son diferentes como se refleja en la Figura 5.3, por lo menos en la etapa transitoria aunque luego convergen a un punto común y de allí siguen una trayectoria hacia las altas tasas de acierto más o menos similar.

De hecho, muchas pruebas repetidas por lote siempre confirman cruces de convergencia a muy pocas muestras procesadas, indicando altos porcentajes de acierto finalmente (velocidad de aprendizaje elevado), pero como se observa en la

Figura 5.3 (a), en ocasiones tales porcentajes de acierto final ni siquiera tocan la línea de convergencia hacia altas tasas de aciertos. De ocurrir esto, sólo se tiene que ajustar el porcentaje de imágenes que se fija en el procedimiento incremental para alcanzar mejoras parciales o definitivas como se muestra en las Figuras 5.3 (b) y (c) respectivamente. Sin embargo, este fenómeno visto en las pruebas por lote, sólo se observa si tal prueba se repite más de una vez para diferentes sorteos de imágenes.

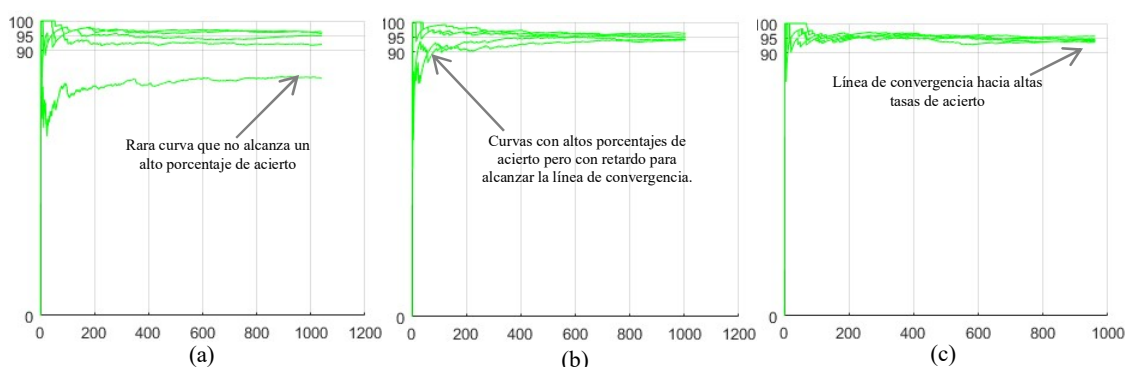


Figura 5.3.- Resultados obtenidos del multclasificador IPC por lote para diferentes conjuntos (porcentajes) de imágenes con cuatro clases desbalanceadas usando sólo el detalle V con opciones SVM antiguas y $\lambda=0.5$. Porcentajes: (a) 7%, (b) 10%, (c) 14%.

En general, los inconvenientes durante el inicio de esta investigación, que fueron registrados y corregidos con sólo repetir las pruebas, son comentados a continuación:

- Encontrar un porcentaje que propiciara buenas tasas de aciertos en un número de *iteraciones*⁶⁰ menores posibles, lo que se traduce en un tiempo excesivo de trabajo por parte del diseñador.
- Encontrar un conjunto (porcentaje) de imágenes mínimo final. Sin la repetición de pruebas, muchas veces se fijaban conjuntos que no eran óptimos.
- También se podían llegar a grandes conjuntos de datos que arrojaban buenas tasas de aciertos, pero al no ser mínimos introducían tiempos de procesamiento excesivos para muchos de los multclasificadores estudiados.

⁶⁰ Se entiende por iteración, el número de veces que se repite un experimento por ajuste de un parámetro.

- Aunque el ajuste del porcentaje de imágenes dado por un experimento incremental se conseguía, muchas veces la convergencia hacia altas tasas de acierto en un experimento por lote era lenta como se observa en la Figura 5.3 (b). No presentan una buena velocidad de aprendizaje.
- Aunque la realización de los experimentos con el acceso a las imágenes se haga en forma aleatoria, para suavizar los transitorios como se vio en la sección 5.3, no se logra un pliegue de las trayectorias que marcan las diferentes curvas como se aprecia en la Figura 5.3 (b).
- Los conjuntos (porcentajes) de imágenes señalados por los experimentos incrementales, no pre-visualizan los porcentajes de aciertos a obtener en los entrenamientos y pruebas de clasificación por lote.

En definitiva y como resumen, las pruebas preliminares dieron a entender que la selección de un conjunto mínimo de imágenes para realizar un experimento por lote no puede ser pre-visualizado a partir de un experimento incremental, sino se repite el ensayo más de una vez.

Se puntualiza en que, esta necesidad de repetir un experimento un número considerable de veces, permite determinar con gran precisión el cruce de convergencia hacia altas tasas de acierto. Además de permitir a partir de los diferentes transitorios, la selección bien sea por el promedio de los cruces, o como se optó en este trabajo, seleccionar el punto más alejado del eje de las ordenadas. En cualquiera de los dos casos, se podrá establecer el conjunto mínimo de imágenes para entrenar un multclasificador por lote, mejorando realmente su velocidad de aprendizaje.

Para comprobar lo anteriormente dicho, véase nuevamente la Figura 5.2 donde se observan diferentes transitorios en las curvas de los porcentajes de aciertos del multclasificador IPC que se ha entrenado por incremento. Cada curva marca una trayectoria diferente, que cruza la línea de convergencia hacia altas tasas de aciertos en puntos diferentes.

Nótese que, se ha seleccionado el punto de cruce más alejado del eje de las ordenadas. Este punto establece en el eje de las abscisas, 150 imágenes que se han

tenido que procesar antes de la convergencia definitiva hacia las altas tasas de aciertos, y a su vez establece un 14% como conjunto mínimo de imágenes que se deben tomar para realizar el entrenamiento por lote (véase las columnas 1 y 2, fila 3 de la Tabla 5.3).

En la Figura 5.3 (c) se puede observar para el 14% del conjunto de imágenes establecido por el experimento incremental, como se consigue un elevado porcentaje de acierto, alrededor del 95% con una convergencia inmediata de apenas 24 muestras procesadas, alcanzándose una buena velocidad de aprendizaje. Esta información también se puede ver en la columna 3 de la fila 3 en la Tabla 5.3.

Además en la misma Tabla 5.3, se puede contrastar esta selección óptima propiciada por el punto de cruce en la línea de convergencia hacia altas tasas de aciertos, con otros cruces originados por los porcentajes de aciertos en los experimentos incrementales, para comprobar que no siempre se obtiene buenos resultados en los experimentos por lote.

Por ejemplo, en la fila 4 se puede observar que si se toma un 10% de las imágenes para realizar el entrenamiento por lote, la convergencia hacia altas tasas de aciertos es lenta, ya que se debe esperar a procesar 240 muestras. Incluso la selección de otros conjuntos de imágenes (porcentajes) para realizar el entrenamiento por lote por debajo del 10%, no permiten una convergencia real hacia altas tasas de aciertos, lo que indica que el multclasificador no tiene una velocidad de aprendizaje clara.

Finalizando con esta discusión y basándose en todo el análisis realizado, se concluye diciendo que, siempre se tendrá la necesidad de repetir un experimento más de una vez. Debido a esto, se observa en la gran mayoría de las imágenes mostradas en el Capítulo IV, que las curvas se repiten más de una vez.

5.5 IMPORTANCIA DE AJUSTAR MANUALMENTE EL PORCENTAJE FIJADO POR LAS PRUEBAS INCREMENTALES

Como se observó en la mayoría de los experimentos vistos en el Capítulo IV, por lo discutido especialmente en la Prueba 9 de la sección 4.6 del Capítulo IV, y como consecuencia del acceso aleatorio que se realiza para obtener las imágenes y

poder realizar el respectivo procesamiento discutido en la sección 5.3 del Capítulo V; se le da una gran importancia al ajuste manual que debe ser realizado sobre el porcentaje fijado por las pruebas incrementales para seleccionar el conjunto de imágenes usado en los entrenamientos por lote con el objetivo de lograr una efectiva convergencia hacia altas tasas de aciertos.

Es evidente que no sólo es uno de los motivos para propiciar el ajuste y por esto se comenta en rasgos generales los beneficios introducidos por el mencionado ajuste, entre los cuales se tiene:

- Alcanzar conjuntos óptimos de datos para realizar los entrenamientos y pruebas por lote, propiciando altas tasas de acierto y minimizando el tiempo de procesamiento en los multclasificadores.
- Calibrar el error cometido en el procesamiento inicial que realiza un multclasificador al entrenarse, garantizando que para un número de muestras visto se tendrán estimaciones con errores menores al 10% de las predicciones obtenidas, con una línea de convergencia hacia altas tasas de aciertos del 90%.
- Conseguir menos retardo en los multclasificadores entrenados. Esto es una consecuencia de lo anterior. Como ya se ha estudiado, los multclasificadores presentan un retardo que está asociado con el tiempo consumido por el número de muestras que deben dejar pasar para poder ver altas tasas de acierto.
- Aumentar la velocidad de aprendizaje, que está relacionada con el número de muestras necesarias para entrenar un multclasificador y así obtener altas tasas de aciertos. Siendo evidente que, al optimizar los conjuntos de entrenamiento se aumenta la velocidad de aprendizaje. Claro está que la optimización de un conjunto de imágenes, también sugiere que sea mínimo.
- Todo lo anterior es consecuencia inmediata de reducir aún más por el ajuste sugerido, los transitorios en las gráficas del porcentaje de acierto.

Es evidente que cada clasificación es individual, sobre todo para los multclasificadores PC, IPC y PV que no sólo presentan la predicción sino que aportan medidas de fiabilidad. Sin embargo, el porcentaje de aciertos sobre el

conjunto de muestra en general, permite medir la efectividad final del predictor con relación a la totalidad de las imágenes que se procesan en esta investigación, y no una simple muestra. Además, esta medida del porcentaje de acierto sirve para validar el multclasificador SVM que no emite ninguna medida adicional de fiabilidad.

5.6 EL EFECTO DEL SORTEO ALEATORIO SOBRE LAS IMAGENES

Luego de los resultados obtenidos sobre el examen hecho por los diferentes multclasificadores SVM, PC, IPC y PV-KNN al procesar las imágenes provenientes del dispositivo de fusión termonuclear TJ-II, y en especial del efecto que provoca el sorteo aleatorio de las imágenes al momento de seleccionar los conjuntos de muestras para realizar los entrenamientos y pruebas de clasificación, tanto por incremento como por lote; se deja en evidencia la clara distorsión que provoca el orden aparente que finalmente ocupa cada observación (imagen) tomada. Lo que puede traducirse en que, para ciertas condiciones al momento de captar una imagen de fusión, la misma no retiene la suficiente información haciéndose irreconocible⁶¹.

Si se analiza bien este fenómeno observado, más que ser un inconveniente, podría resultar beneficioso para el proceso que se estudia de fondo que no es más que la fusión nuclear. Sólo bastaría con realizar varios diagnósticos con datos completos (secuencias de imágenes completas tomadas directamente de los frames producidos por videos), que se obtengan desde el momento que se inicia un experimento (descarga) hasta el instante donde debería darse la ignición del plasma, ya que durante este tránsito ocurren fenómenos dentro del plasma que terminan en un evento crítico como la pérdida de confinamiento o las disrupciones, dando fin a una descarga. Después, para una clase de imágenes determinada, se debe crear un multclasificador que las inspeccione para ver hasta qué punto cambian las características retenidas en cada frame, viendo al mismo tiempo las condiciones donde se hacen irreconocibles.

Posteriormente se debe contrastar tal anomalía con el resto de las condiciones

⁶¹ Otras interpretaciones sobre lo irreconocible que puede ser una imagen de fusión para un clasificador entrenado, pudieran comentarse.

dadas y se debe verificar si coinciden con los instantes en la cual ocurre un evento crítico que da fin a una descarga. De coincidir, se estaría en presencia de condiciones donde el clasificador entrenado no logra reconocer las imágenes obtenidas, pudiendo deberse a la falta de información que debe ser atrapada, pero por las condiciones de operación durante el evento crítico no pueden ser captadas, dejando irreconocibles tales imágenes.

Para el caso del TJ-II, muchos eventos pudieran coincidir, así el más importante a describir por cualquier medio para avanzar al estado de ignición del plasma, es el denominado colapso radiactivo por calentamiento (véase el Capítulo II) y se da por el hecho de calentar el plasma con ondas y al momento de alcanzar cierto estado energético donde se tiene un nivel de densidad y temperatura, el plasma no puede aumentar más su temperatura, lo cual hace que se pierda el confinamiento magnético.

Claro está que, desde el inicio de una descarga hasta el fin de la misma por un evento como el colapso radiactivo por calentamiento, el análisis de las imágenes del TJ-II revela el lógico cambio que en el tiempo sufren las condiciones del proceso. Tales condiciones quedan almacenadas por los diagnósticos y en especial aquellos que son capaces de medir la luz emitida por el plasma tanto en el rango visible como el no visible.

Estas condiciones dan información que son atrapadas por las imágenes que se procesan y en particular originan algunas cuestiones:

- ¿Por qué la información detectada y atrapada en una imagen llega a tal punto de ser irreconocible por los clasificadores?.
- ¿Pueden desarrollarse en torno a este fenómeno observado, multclasificadores específicos que permitan por contraste de todas las variables involucradas conocer condiciones específicas de los plasmas estudiados?.
- ¿En especial, demostrar esto serviría para la predicción de eventos como el colapso radiactivo o las disrupciones?.

Es evidente, que esta investigación no responderá estas preguntas pero quedan

abiertas para futuras investigaciones, recordando por todo lo comentado y por los análisis realizados, que existe por lo menos en el banco de datos manejados, hasta un 4% de imágenes que son irreconocibles según los resultados comparados en la Tabla 5.1 donde se observa que el método de la PV-KNN como el mejor de los predictores estudiados para un conjunto de datos completo.

CONCLUSIONES

En esta investigación se estudiaron modelos de clasificación con medidas de confianza basados en predictores conformales y aplicados a imágenes relacionadas con mediciones de densidad y temperatura de los plasmas de fusión nuclear, captadas a partir del diagnóstico de esparcimiento Thomson, que opera en el dispositivo de fusión termonuclear TJ-II ubicado en el Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT).

Para el estudio se desarrollaron varios multclasificadores relacionados con cuatro métodos referidos a: la máquina de vectores soporte (SVM) que permitió la construcción de un multclasificador “clásico” donde sólo se obtienen predicciones; la predicción conformal (PC), la predicción conformal inductiva (IPC) y la predicción Venn (PV), que permitieron construir “predictores de confianza” donde además de la predicción se obtienen medidas de fiabilidad.

Los multclasificadores SVM, PC, IPC y PV se analizaron con el objetivo de “cómo obtener un multclasificador que con el menor tiempo de procesamiento en la fase de entrenamiento, proporcione el mejor rendimiento en términos de tasa de aciertos y fiabilidad”.

Todo el análisis se logró mediante la implementación de un conjunto de programas en el entorno de programación de MATLAB con la ayuda de la librería LIBSVM, permitiendo la realización de diferentes ensayos con los multclasificadores mencionados, proporcionando resultados referidos a: las tasas de aciertos en la clasificación de imágenes, así como las medidas de fiabilidad (confianza, credibilidad, confianza por credibilidad medidos en los multclasificadores PC e IPC; probabilidad e intervalos de probabilidades medidos en los multclasificadores PV), tiempos de procesamiento tanto en entrenamiento como en pruebas de clasificación, número de imágenes empleado para el entrenamiento y el número de imágenes que deben procesar para alcanzar altas tasas de aciertos.

Los ensayos realizados se enfocaron como experimentos incrementales y experimentos por lote, utilizándose conjuntos de imágenes balanceados y conjuntos

de imágenes desbalanceados. El acceso y pre procesamiento de las imágenes se logró con la implementación de programas donde se utilizó la transformada wavelet para la compresión de imágenes hasta un nivel cuatro, representando una reducción importante de la cantidad de información a utilizar en los multclasificadores, garantizando a su vez, una resolución apropiada para la selección de características representativas al permitir cuatro diferentes niveles de detalles.

Con los resultados proporcionados por los ensayos, se realizaron contrastes individuales para cada multclasificador al confrontar los tiempos de procesamiento frente al rendimiento en términos de tasa de aciertos y fiabilidad. Además de contrastar todos los resultados individuales obtenidos. Toda esta información generada por los diferentes contrastes fue presentada en tablas y gráficos, permitiendo un análisis global de toda la investigación realizada con la cual se llega a las siguientes conclusiones:

1. Todos los multclasificadores obtienen altos porcentajes de acierto sobre el conjunto de imágenes tratado, existiendo en promedio un 95.3% en las estimaciones que realizan.
2. El multclasificador SVM requiere siempre de más muestras para su entrenamiento, quedando de esta manera en desventaja con relación al resto de los predictores (PC, IPC y PV-KNN). Por lo tanto presenta una baja velocidad de aprendizaje.
3. El multclasificador IPC es el que menos muestras debe dejar pasar para comenzar a realizar predicciones con un error menor o igual al 10%. Esto lo cataloga como el predictor que más rápido converge hacia altas tasas de aciertos. Presentando una alta velocidad de aprendizaje.
4. Con relación a las medidas de confianza y credibilidad, el multclasificador IPC mantiene valores muy cercanos con relación al multclasificador PC, pese a que utiliza un procedimiento que divide los datos usados para el entrenamiento en dos sus conjuntos, pero siendo evidente que esto no afecta su rendimiento en las predicciones hechas. Se resalta además, que la medida de confianza por credibilidad apunta a la IPC como mejor predictor.

5. De los detalles AHVD obtenido tras el pre procesamiento realizado con la transformada wavelet, el coeficiente de aproximación (A) y el detalle vertical (V), son la fuente más representativa para propiciar altas tasas de aciertos en los multclasificadores estudiados.
6. El procesamiento aleatorio tiene una influencia positiva sobre los procedimientos incrementales y por lote, permitiendo una mejor obtención de los conjuntos de entrenamiento por lote y su consecuente obtención de una rápida convergencia hacia altas tasas de aciertos. Esto se debe a la regulación que la misma tiene sobre los transitorios de las curvas del porcentaje de acierto. En general, el procesamiento aleatorio mejora la velocidad de aprendizaje, permitiendo construir mejores multclasificadores.
7. Los experimentos realizados ponen de manifiesto, la importancia de repetir un experimento tanto incremental como por lote más de una vez; de lo contrario se pudieran estar cometiendo errores en la medición de algunas variables como son las tasas de acierto y las medidas de fiabilidad, traduciéndose esto en la obtención de *pobres multclasificadores*⁶². Este aspecto también influye positivamente sobre la velocidad de aprendizaje de los multclasificadores estudiados.
8. Los experimentos realizados ponen de manifiesto, la importancia de realizar un ajuste manual sobre el porcentaje fijado por las pruebas incrementales para seleccionar el conjunto de imágenes usado en los entrenamientos por lote, con el objetivo de lograr una efectiva convergencia hacia altas tasas de aciertos para un multclasificador entrenado. Este aspecto tiene una consecuencia inmediata sobre la velocidad de aprendizaje de los multclasificadores estudiados.

En general, se concluye indicando que, la predicción conformal inductiva permite obtener un multclasificador con un tiempo de procesamiento en la fase de entrenamiento mínimo, proporcionando un buen rendimiento en términos de tasa de

⁶² Es importante entender que la denominación de pobre se le atribuye a un multclasificador en este trabajo, al predictor que no arroja altas tasas de aciertos con tiempos de procesamiento excesivo. Además de una lenta convergencia hacia altas tasas de aciertos.

aciertos y fiabilidad.

Además se indica que, las pruebas incrementales y por lote, permiten establecer un procedimiento para obtener multclasificadores con un tiempo de procesamiento mínimo en la fase de entrenamiento, garantizando un buen rendimiento en términos de tasa de aciertos y fiabilidad.

Para finalizar, se analiza el efecto que provoca la selección aleatoria de imágenes para elegir los conjuntos de muestras de entrenamiento y prueba, tanto por incremento como por lote. El procedimiento aplicado en esta investigación para entrenar y probar los multclasificadores analizados, siempre permite obtener una convergencia hacia altas tasas de acierto, dejando en evidencia su robustez.

LINEAS FUTURAS DE TRABAJO

Las líneas lógicas de continuación de esta investigación son:

- Realizar un estudio de la metodología aplicada en esta Tesis para entrenar y probar los multclasificadores analizados con imágenes del diagnóstico de esparcimiento Thomson, pero utilizando datos de otros diagnósticos de imagen que operan en el dispositivo de fusión TJ-II del CIEMAT.
- Realizar un estudio sobre los algoritmos implementados en esta Tesis con el fin de paralelizarlos en entornos de supercomputación, reduciendo de esta manera el costo computacional sin perder la calidad de las predicciones conseguidas.
- Aplicar la metodología estudiada en esta Tesis para entrenar y probar los predictores Venn ABERS con las imágenes obtenidas en el diagnóstico de esparcimiento Thomson, verificando que se pueden obtener mejores predicciones probabilísticas.
- Realizar un estudio que permita establecer una metodología de entrenamiento activo, con la cual se puedan elegir las mejores muestras para formar un sólo conjunto de entrenamiento para los multclasificadores analizados en esta Tesis.

BIBLIOGRAFIA

- 1 Acosta B. Maria I. y Zuluaga M. Camilo A. (2000). Tutorial Sobre Redes Neuronales Aplicadas en Ingeniería Eléctrica y su Implementación en un Sitio Web. Tesis. Universidad Tecnológica de Pereira. Pereira. Disponible en: <http://ohm.utp.edu.co/neuronales/Download/Contenido.Pdf>.
- 2 Alejaldre, C., Almoguera, L., Alonso, J., Ascasíbar, E., Baciero, A., Balbin, R., y Cappa, A. (2001). Review of confinement and transport studies in the TJ-II flexible heliac. *Nuclear Fusion*, 41(10), 1449.
- 3 Alejaldre, C., Alonso, J., Almoguera, L., Ascasíbar, E., Baciero, A., Balbín, R., y Cappa, A. (1999). First plasmas in the TJ-II flexible heliac. *Plasma physics and controlled fusion*, 41(3A), A539.
- 4 Alejaldre, C., Gozalo, J. J. A., Perez, J. B., Magaña, F. C., Diaz, J. R. C., Perez, J. G., y Navarro, A. P. (1990). TJ-II project: a flexible heliac stellarator. *Fusion Technology*, 17(1), 131-139.
- 5 Ariola, M., y Pironti, A. (2008). *Magnetic control of tokamak plasmas* (Vol. 187). London: Springer.
- 6 Armbrust, M., Foz, A., Griffith, R., Katz, R., Konwinski, G., Patterson, D., Rabkin, A. y Zaharia, M. (2010). A View of Cloud Computing. *The Magazine Communications of the ACM*, 53(4), 50-58.
- 7 Aymerich, O. B., Miguel, J., Cordero, C., Carpio, G. N., y Umaña, G. N. (2013). Revisión repercusiones médico legales de los desastres nucleares, 30(1), 45–61.
- 8 Azarang, M. (1996). Simulación y análisis de modelos estocásticos/por Mohammad R. Azarang y Eduardo García Dunna (No. 519.1 A9).
- 9 Barth, C., Pijper, F., Meiden, H., Herranz, J., y Pastor, I. (1999). High-resolution multiposition Thomson scattering for the TJ-II stellarator. *Review of scientific instruments*, 70(1), 763-767.
- 10 Bellotti, T., Luo, Z., Gammerman, A., Van Delft, F., y Saha, V. (2005). Qualified Predictions for Microarray and Proteomics Pattern Diagnostics with Confidence Machines. *International Journal of Neural Systems*, vol. 15, no. 4, pp. 247 - 258, 2005.

- 11 Boozer, A. (2012). Theory of tokamak disruptions. *Physics of plasmas*, 19(5), 058101.
- 12 Boser, B., Guyon, I., y Vapnik, V. (1992, July). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory* (pp. 144-152).
- 13 Braams, C., & Stott, P. (2002). *Nuclear fusion: half a century of magnetic confinement fusion research*. CRC Press.
- 14 Cairns, R. (1993). Radio-frequency plasma heating. In *Plasma physics: an introductory course*.
- 15 Carmona S. (2014). Tutorial sobre Máquinas de Vectores Soporte (SVM). Dpto. de Inteligencia Artificial, ETS de Ingeniería Informática, Universidad Nacional de Educación a Distancia (UNED), Madrid.
- 16 Castro, R. (2010). PAPI como infraestructura de seguridad distribuida aplicada a entornos de fusión termonuclear. Tesis Doctoral. UNED, España.
- 17 Cavero, J. (1989). *Historia de la Energía Nuclear*.
- 18 Chang C., y Lin C. (2018). LIBSVM: a library for support vector machines [Página web]. Disponible: <https://www.csie.ntu.edu.tw/~cjlin/libsvm/> [Consulta: 2019, julio 15]
- 19 Chang, O., y Olmedo, A. (2010). Propuesta de implementación y simulación de modelos basados en sinapsis dinámica. *Universidad, Ciencia y Tecnología*. Vol. 14, N° 55. pp.119-127.
- 20 Chang, Y. W., Hsieh, C. J., Chang, K. W., Ringgaard, M., & Lin, C. J. (2010). Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research*, 11(Apr), 1471-1490.
- 21 Chapra S. y Canale R. (2003). *Métodos Numéricos para Ingenieros*. Cuarta Edición. Editorial Mc Graw Hill.
- 22 Cherkassky V. y Ma Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks* 17.113–126.
- 23 Cherkassky V. y Mulier F. (2007). *Learning From Data. Concepts, Theory, and Methods*. 2nd Ed.

- 24 Cortes, C., y Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3), 273-297.
- 25 Cox, D. (1958). Some problems connected with statistical inference. *Annals of Mathematical Statistics*, 29:357-372.
- 26 Dhar, V. (2013). "Data science and prediction". *Communications of the ACM*. 56 (12): 64–73.
- 27 Dormido C., Vega J., Ramírez J.M., Murari A., Moreno R., López J.M., Pereira A., y JET-EFDA Contributors. (2013). Development of an efficient real-time disruption predictor from scratch on JET and implications for ITER. *Nucl. Fusion* 53 113001 (8pp).
- 28 Dormido-Canto S., Vega J., Ramírez J.M., Murari A., Moreno R., López J.M., Pereira A. and JET-EFDA Contributors. (2013). Development of an efficient real-time disruption predictor from scratch on jet and implications for ITER. *Nuclear Fusion*, 53, Issue 11, 8pp, November 2013.
- 29 Farias, G., Fabregas, E., Dormido-Canto, S., Vega, J., Vergara, S., Bencomo, S.D., Pastor, I., y Olmedo, A. (2018). Applying deep learning for improving image classification in nuclear fusion devices. *IEEE Access*. Vol. 6. Pp. 72345-72356.
- 30 Farias, G., Santos, M., Marrón, J. L. y Dormido-Canto, S. (2004). Determinación de los Parámetros de la Transformada Wavelet para la Clasificación de Señales del Diagnóstico Scattering Thomson. XXV Jornadas de Automática, Ciudad Real, España.
- 31 Fayyad, U., Haussler, D., y Stolorz, P. (1996). "KDD for Science Data Analysis: Issues and Examples". *Proceedings of the 2da. International Conference on Knowledge Discovery and Data Mining*. Portland, Oregon, USA, pp. 50 - 56.
- 32 Fedorova V., Nourtdinov I., Gammerman A., (2012). Testing the Gauss linear assumption for on-line predictions. *Prog Artif Intell* 1:205–213.
- 33 Fernández, Á., Likin, K., Turullols, P., Teniente, J., Gonzalo, R., del Río, C., y Martín, R. (2000). Quasioptical transmission lines for ECRH at TJ-II stellarator. *International Journal of Infrared and Millimeter Waves*, 21(12), 1945-1957.
- 34 Freeman J. y Skapura D. (1993). *Redes Neuronales. Algoritmos, aplicaciones y técnicas de programación*. Editorial Addison-Wesley/Diaz de Santos.

- 35 Fuentes, C., Liniers, M., Wolfers, G., Alonso, J., Marcon, G., Carrasco, R., y García, A. (2005). Neutral beam injection optimization at TJ-II. *Fusion engineering and design*, 74(1-4), 249-253.
- 36 Gammernan, A., y Vovk, V. (2007). Hedging predictions in machine learning. *The Computer Journal*, 50(2), 151-163.
- 37 García, J., Rodríguez, Vidal, J. (2005). *Aprenda MATLAB como si estuviera en primero*. Escuela Técnica Superior de Ingenieros Industriales. UPM. Madrid-España.
- 38 González, J., Ruiz, M., Vega, J., Barrera, E., Arcas, G., y López, J. M. (2010). Event recognition using signal spectrograms in long pulse experiments. *Review of Scientific Instruments*, 81(10), 10E126.
- 39 González, R. (2012). *Estadística Inferencial II*. Instituto Tecnológico de Ensenada [Libro en Línea]. Disponible: https://www.academia.edu/8137314/Estad%C3%ADstica_Inferencia_II. [Consulta: 2015, marzo 25].
- 40 Harris, J., Cantrell, J., Hender, T., Carreras, B. A., y Morris, R. (1985). A flexible heliac configuration. *Nuclear fusion*, 25(5), 623.
- 41 Hender, T., Cantrell, J., Harris, J., Carreras, B., Lynch, V., Lyon, J., y Navarro, A. (1987). Studies of a flexible heliac configuration (No. ORNL/TM-10374). Junta de Energía Nuclear, Madrid (Spain). Div. de Fusion.
- 42 Hernández, B. (2007). Estudio de la dinámica de iones por métodos espectroscópicos en el TJ II. Tesis Doctoral. Universidad Complutense de Madrid, España.
- 43 Herranz, J., Castejón, F., Pastor, I., y McCarthy, K. (2003). The spectrometer of the high-resolution multiposition Thomson scattering diagnostic for TJ-II. *Fusion Engineering and design*, 65(4), 525-536.
- 44 Johansson, U., Löfström, T., y Sundell, H. (2018). Venn Predictors Using Lazy Learners. In *International Conference Data Science ICDATA* (pp. 220-226).
- 45 Kosko B. (1992). *Neural Networks For Signal Processing*. Editorial Prentice Hall International Editions.

- 46 Laboratorio Nacional de Fusión. (2019, Febrero). Glossary of terms for fusion (Spanish) [Página Web]. Disponible: http://www-fusion.ciemat.es/New_fusion/en/Fusion/glosario_es.pdf.
- 47 Léxico Powered by Oxford. [Versión en línea]. <<https://www.lexico.com>> [2020, Abril].
- 48 Makili, L. (2014). Sistemas de clasificación automáticos con confianza y credibilidad en fusión termonuclear. Tesis Doctoral. UNED, España.
- 49 Makili, L., Dormido-Canto, S., y Sánchez, J. V. (2010). Clasificador de Imágenes del Diagnóstico Thomson Scattering del TJ II Basado en Template Matching. Revista Científica de Ingeniería Electrónica, Automática y Comunicaciones ISSN: 1815-5928, 29(1), 14-19.
- 50 Makili, L., Vega, J., y Dormido-Canto, S. (2013). Incremental support vector machines for fast reliable image recognition. Fusion Engineering and Design, 88(6-8), 1170-1173.
- 51 Martin, Y y Takizuka, T. (2008). Power requirement for accessing the H-mode in ITER. In Journal of Physics: Conference Series (Vol. 123, No. 1, p. 012033). IOP Publishing.
- 52 Martínez, F. (2017). Supercomputación aplicada para la identificación de patrones mediante clasificadores probabilísticos en diagnósticos de fusión. Tesis Doctoral. UNED, España.
- 53 MathWorks Documentation. Support Vector Machines (SVM). Disponible: http://www.mathworks.com/help/stats/support-vector-machines-svm.html#bss0s6_-1 [Consulta: 2015, julio 14].
- 54 McCracken G., y Stott P. (2005). Fusion The Energy of the Universe. Burlington, MA, USA: Elsevier Academic Press.
- 55 Melo, S., y García, R.. (2005). Compresión de imágenes con Wavelets y Multiwavelets. Ingeniería, 10(1), 48-54.
- 56 Merino, M. (2015). Técnicas clásicas de optimización. Parte I: programación lineal y no lineal [Documento en Línea]. Disponible: http://www.ehu.eus/mae/html/prof/Maria_archivos/plnlapuntes.pdf. [Consulta: 2019, Julio 27]
- 57 Misiti, M., Misiti, Y., Oppenheim, G., y Poggi, J. M. (1997-2015). Wavelet toolbox user's guide. The Math Works Ins.

- 58 Montgomery D., y Runger G. (2008). Probabilidades y Estadística aplicadas a la Ingeniería. Segunda Edición. Editorial Limusa Wiley.
- 59 Moraleda, A. y Villalba, C. (2016, Marzo). Introducción al análisis de datos con R. Dpto. Informática y Automática, ETS Ingeniería Informática, UNED, Madrid, España.
- 60 Morones, I. (2012). El accidente nuclear de Fukushima. Ingenierías, 15(56), 38-47.
- 61 Murari, A. (2015, 30 de Abril). Caracterización de plasmas termonucleares en reactores de fusión: Diagnósticos en JET y herramientas de análisis. Ciclo de Conferencias del Master y del Programa de Doctorado en “Ingeniería de Sistemas y de Control” [Vídeo]. Disponible en: <https://canal.uned.es/video/5a6f5c0cb1111f967a8b461f>.
- 62 Naciones Unidas, Departamento de Asuntos Económicos y Sociales, División de Población (2019). World Population Prospects 2019: Metodología de las estimaciones y proyecciones de población de las Naciones Unidas. [Página Web]. Disponible: <https://population.un.org/wpp/Publications/>
- 63 Nieto, M (2001). Procesos de ignición en fusión por confinamiento inercial (Vol. 112). Universidad de Castilla La Mancha, España.
- 64 Nouretdinov, I., Devetyarov, D., Burford, B., Camuzeaux, S., Gentry-Maharaj, A., Tiss, A. y Vovk, V. (2012, September). Multiprobabilistic Venn predictors with logistic regression. In IFIP International Conference on Artificial Intelligence Applications and Innovations (pp. 224-233). Springer, Berlin, Heidelberg.
- 65 Ogata K. (2010). Ingeniería de Control Moderna. Editorial PEARSON EDUCACIÓN, S.A., Madrid.
- 66 Olmedo A. (2015). Implementación de Código Script para Máquinas de Soporte Vectorial con MATLAB R2009a. Actividad del plan de Investigación (Primera parte). ETS de Ingeniería Informática. UNED. Madrid-España. Documento No Publicado.
- 67 Olmedo A. (2015). Implementación de Código Script para Máquinas de Soporte Vectorial con MATLAB R2009a. Actividad del plan de Investigación (Segunda parte). ETS de Ingeniería Informática. UNED. Madrid-España. Documento No Publicado.

- 68 Olmedo A. (2015). Implementación de Código Script para Máquinas de Soporte Vectorial con MATLAB R2009a. Actividad del plan de Investigación (Tercera parte). ETS de Ingeniería Informática. UNED. Madrid-España. Documento No Publicado.
- 69 Olmedo A. (2016). Análisis Estadístico en Señales de Fusión. MÓDULO III de las actividades formativas transversales: Gestión y Análisis de Datos Científicos. ETS de Ingeniería Informática. UNED. Madrid-España. Documento No Publicado.
- 70 Olmedo, A. (2009). Diseño y entrenamiento de una sinapsis dinámicas para el reconocimiento de la voz. Tesis de Maestría. Universidad Central de Venezuela, Venezuela.
- 71 Papadopoulos H. (2008). Inductive Conformal Prediction: Theory and Application to Neural. Tools in Artificial Intelligence, Paula Fritzsche, IntechOpen, DOI: 10.5772/6078. [Documento en línea]. Disponible: https://www.intechopen.com/books/tools_in_artificial_intelligence/inductive_conformal_prediction_theory_and_application_to_neural_networks [Consulta: 2019, febrero 15].
- 72 Pereira, G. (2015). Selección de Características para el Reconocimiento de Patrones con Datos de Alta Dimensionalidad en Fusión Nuclear. Tesis Doctoral. UNED, España.
- 73 Pérez, J. (2014). Máquinas de vectores soporte en entornos de supercomputación: aplicación a fusión nuclear. Tesis Doctoral. UNED, España.
- 74 Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3), 61-74.
- 75 Proakis J., y Manolakis, D. (1998). Tratamiento Digital de Señales. Tercera edición. Editorial Prentice Hall.
- 76 REAL ACADEMIA ESPAÑOLA: Diccionario de la lengua española, 23.^a ed., [versión 23.3 en línea]. <<https://dle.rae.es>> [2020, Abril].
- 77 Reux, C. (2010). Study of a disruption mitigation method for tokamak plasmas.
- 78 Rusell S. y Norving P. (2003). Inteligencia Artificial Un Enfoque Moderno. Segunda Edición. Editorial Prentice Hall. 2003.

- 79 Salkin, N. (1997). *Métodos de Investigación*. Editorial Prentice Hall. Tercera edición.
- 80 Sánchez, J., Acedo, M., Alegre, D., Alonso, A., Alonso, J., Álvarez, P., y Barrera, L. (2011). Overview of TJ-II experiments. *Nuclear Fusion*, 51(9), 094022.
- 81 Sánchez, J., Ascasibar, E., y Acedo, P. (1998). Diagnostic systems for the TJ-II flexible heliac. In *Helical system research*.
- 82 Saunders, C., Gammerman, A. y Vovk, V. (1999). Transduction with confidence and credibility. *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Vol. 2, pp. 722–726, Morgan Kaufmann, Los Altos, CA.
- 83 Shafer G., y Vovk V. (2008). A Tutorial on Conformal Prediction. *Journal of Machine Learning Research* 9, 371-421.
- 84 Sheffield, J. (1994). The physics of magnetic fusion-reactors, *Rev. Mod. Phys.* 66, 1015.
- 85 Soria, A. (2015). *Física nuclear y de partículas* (Vol. 62). Universitat de València.
- 86 Spiegel, M. (1991). *Estadística*. Segunda edición. Editorial Mc. Graw Hill.
- 87 Thornton, A. (2011). The impact of transient mitigation schemes on the MAST edge plasma. *Doctoral Dissertation*, University of York
- 88 Valiant, L. (1984). A theory of the learnable. *Communications of the ACM*, 27(11), 1134-1142.
- 89 Vapnik, V. (2000). *The nature of statistical learning theory*. Second edition. Springer. New york.
- 90 Vega J., Murari A., Pereira A., Portas A., Castro R. and JET-EFDA Contributors. (2009). Overview of intelligent data retrieval methods for waveforms and images in massive fusion databases. *Proceedings of the 25th Symposium on Fusion Technology - SOFT-25*. Rostock, Germany 15–19 September. *Fusion Engineering and Design*. Volume 84, Issues 7-11, June 2009, Pages 1916-1919.
- 91 Vega, J., Dormido-Canto, S., López, J. M., Murari, A., Ramírez, J. M., Moreno, R., y JET-EFDA Contributors. (2013). Results of the JET real-time disruption predictor in the ITER-like wall campaigns. *Fusion Engineering and Design*, 88(6-8), 1228-1231.

- 92 Vega, J., Pastor, I., Cereceda, J., Pereira, A., Herranz, J., Pérez, D., y Dormido, R. (2005, June). Application of intelligent classification techniques to the TJ-II Thomson Scattering diagnostic. In Proceedings of the 32nd EPS conference on plasma Phys. ECA (Vol. 29).
- 93 Vert, J., Tsuda, K., y Schölkopf, B. (2004). A primer on kernel methods. *Kernel methods in computational biology*, 47, 35-70.
- 94 Vovk V., Gammerman A., y Shafer G. (2005). *Algorithmic Learning in a Random World*. Springer, New York.
- 95 Vovk V., Nouretdinov I., y Gammerman A. (2009) On-Line Predictive Linear Regression. *The Annals of Statistics*, Vol. 37, No. 3, pp. 1566-1590. Institute of Mathematical Statistics.
- 96 Wakatani, M. (1998). *Stellarator and heliotron devices* (Vol. 95). Oxford University Press on Demand.
- 97 Yela, F. (2007). *Caracterización de los plasmas de TJ-II y sus poblaciones supratérmicas con diagnósticos de rayos X*. Universidad Complutense de Madrid, Servicio de Publicaciones.
- 98 Yoshikawa, S. (1983). Design of a helical-axis stellarator. *Nuclear Fusion*, 23(5), 667.