





## **Agradecimientos**

Agradezco sinceramente a Yosu y Haritz, mis tutores, sus consejos, su preocupación, su interés y su completa disposición para ayudarme y agradezco a Jose Antonio Fernandes, trabajador en AZTI-Tecnalia, por los mismos motivos, con la supervisión que ha hecho del trabajo. También quiero agradecer a Igor Arregi y a Guillermo Boyra, trabajadores en AZTI-Tecnalia, sus ganas de ayudarme y los consejos que me ofrecieron. También tengo en mente, a mis familiares y amistades, y muy en especial a mi madre Begoña, que han soportado mis vicisitudes, y siempre me han animado.



## **Abstract**

In this master thesis the automatic recognition and classification of elements present in sonar images is investigated. The sonar images show the measurement of the bounce of acoustic signals that are emitted by the sonar. In particular it is to automatically identify tuna in long-range sonar images, which is an area of knowledge still very little explored. In this master thesis detect and classify elements of the sonar. In this master thesis a process is built by which elements of the sonar images can be detected and classified.

This master thesis treats the areas of knowledge of image processing and supervised classification which are the areas the process is based on. This process can be divided into stages. The first stage is the image preprocessing where the goal is to improve the image for the next stages. Then comes the stage of segmentation, where the elements are distinguished in the image. The next step is to extract features of each of these elements. Finally, we present the classification stage where each element is labeled.

## Resumen

En esta tesis de máster se investiga acerca del reconocimiento y clasificación automática de elementos presentes dentro de imágenes de sonar. Las imágenes de sonar reflejan la medición del rebote de las señales acústicas que el sonar emite. En concreto se trata de identificar automáticamente atún en imágenes de sonar de largo alcance, que es un área del conocimiento aún muy poco explorada. En esta tesis de máster se crea un proceso mediante el cual se pueden detectar y clasificar elementos en las imágenes de sonar.

En esta tesis de máster se trata las áreas del conocimiento del procesamiento de imagen y de la clasificación supervisada, en las cuales se basa el proceso creado. Dicho proceso se puede dividir en etapas. La primera etapa es el preprocesamiento de la imagen donde se trata de mejorar la imagen de cara a las etapas posteriores. Después viene la etapa de la segmentación, donde se distinguen los elementos presentes en la imagen. La etapa siguiente es la extracción de características de cada uno de esos elementos. Finalmente, se presenta la etapa de clasificación en donde se etiqueta qué es cada elemento.



## **Palabras clave**

Procesamiento de imágenes, imágenes de sonar, preprocesamiento, segmentación, extracción de características, clasificación supervisada





## 1 Índice de contenidos

1	Índice de contenidos .....	10
1.1	Índice de ilustraciones.....	13
1.2	Índice de fórmulas.....	15
1.3	Índice de tablas .....	17
2	Introducción .....	20
2.1	AZTI-Tecnalia .....	21
2.2	Objetivos de la tesis de máster .....	21
2.3	Objetivos últimos de la investigación.....	21
2.4	Productos de la tesis de máster .....	21
3	Procesamiento de imagen.....	24
3.1	La luz.....	24
3.1.1	La naturaleza de la luz .....	24
3.1.2	La descomposición de la luz .....	24
3.1.3	La percepción de la luz .....	24
3.2	La imagen digital.....	25
3.2.1	La imagen digital de dos dimensiones espaciales .....	25
3.2.2	Convenciones en imágenes digitales bidimensionales .....	26
3.2.3	El píxel.....	26
3.2.4	La vecindad de un píxel .....	27
3.2.5	La adyacencia entre pares de píxeles.....	28
3.2.6	El camino entre pares de píxeles.....	29
3.2.7	La conectividad entre píxeles .....	29
3.2.8	La componente conexa .....	30
3.2.9	La región .....	30
3.2.10	La frontera .....	30
3.2.11	Las distancias entre píxeles .....	30
3.2.12	Espacios de color .....	31
3.3	Definición de procesamiento de imagen digital.....	36
3.4	Etapas del procesamiento de imagen digital .....	37
3.5	Adquisición de la imagen.....	37
3.5.1	El ruido.....	38
3.6	Preprocesamiento .....	38
3.6.1	Técnicas de preprocesamiento en el dominio espacial.....	39

# Tesis de máster

---

3.6.2	Técnicas de preprocesamiento en el dominio de frecuencia.....	44
3.7	Segmentación.....	48
3.7.1	Detección de discontinuidades .....	49
3.7.2	Segmentación por identificación de regiones.....	51
3.8	Extracción de características .....	52
3.8.1	Condensación de regiones .....	52
3.8.2	Características para la descripción de la forma.....	53
3.8.3	Características para la descripción de la distribución del color .....	55
3.9	Clasificación e inferencia.....	55
4	Clasificación supervisada.....	58
4.1	El problema del sobreentrenamiento .....	60
4.2	Paradigmas de clasificadores .....	61
4.2.1	Clasificadores K-NN .....	61
4.2.2	Reglas de clasificación .....	62
4.2.3	Árboles de clasificación .....	69
4.2.4	Redes bayesianas .....	76
5	Imágenes de sonar .....	86
5.1	El sonido .....	86
5.2	El sonar .....	86
5.2.1	El sonar empleado.....	87
5.3	Las imágenes de sonar .....	88
6	Diseño de experimentos.....	92
7	Aplicación sonar .....	107
7.1	Diagrama de casos de uso.....	107
7.2	Casos de uso expandidos.....	108
7.3	Dependencias entre paquetes .....	112
7.4	Patrón modelo-vista-controlador.....	112
8	Resultados .....	115
9	Conclusiones.....	131
10	Líneas futuras .....	135
11	Anexos .....	138
11.1	Conceptos básicos de probabilidad.....	138
11.2	La transformada de Fourier.....	140
11.3	Técnicas de validación.....	144

## Tesis de máster

---

11.3.1	Estimación de la bondad con el conjunto de entrenamiento .....	145
11.3.2	Holdout o método H.....	145
11.3.3	Validación cruzada o k-fold cross validation .....	145
11.3.4	Bootstrapping .....	145
11.4	Teoría de la información estadística .....	146
11.5	Calculos matemáticos de resultados.....	148
11.6	Tablas de resultados.....	149
12	Bibliografía.....	164

## 1.1 Índice de ilustraciones

ILUSTRACIÓN 3-1 EL ESPECTRO ELECTROMAGNÉTICO .....	25
ILUSTRACIÓN 3-2 REPRESENTACIÓN DE UNA IMAGEN DIGITAL BIDIMENSIONAL SEGÚN LAS CONVENCIONES .....	26
ILUSTRACIÓN 3-3 REPRESENTACIÓN VISUAL DE LA 4-VECINDAD Y LA 8-VECINDAD .....	28
ILUSTRACIÓN 3-4 IMAGEN ORIGINAL E IMAGEN EN BLANCO-NEGRO .....	33
ILUSTRACIÓN 3-5 IMAGEN ORIGINAL E IMAGEN EN ESCALA DE GRISES .....	33
ILUSTRACIÓN 3-6 MUESTRA DE COLORES DEL ESPACIO RGB.....	34
ILUSTRACIÓN 3-7 MUESTRA DE COLORES DEL ESPACIO HSL.....	35
ILUSTRACIÓN 3-8 MUESTRA DE COLORES DEL ESPACIO HSV .....	36
ILUSTRACIÓN 3-9 ETAPAS SECUENCIALES DEL PROCESAMIENTO DE IMAGEN .....	37
ILUSTRACIÓN 3-10 TRANSFORMACIONES DE NIVELES DE GRIS HABITUALES .....	40
ILUSTRACIÓN 3-11 MUESTRAS DE LA ECUALIZACIÓN DEL HISTOGRAMA.....	41
ILUSTRACIÓN 3-12 FILTRO IDEAL DE SUAVIZADO .....	45
ILUSTRACIÓN 3-13 EJEMPLO DE USO DE UN FILTRO DE SUAVIZADO IDEAL .....	46
ILUSTRACIÓN 3-14 FILTRO IDEAL DE AFILADO .....	47
ILUSTRACIÓN 3-15 EJEMPLO DE USO DE UN FILTRO IDEAL DE AFILADO .....	48
ILUSTRACIÓN 3-16 COEFICIENTES DE LAS MÁSCARAS DE PREWITT Y SOBEL .....	50
ILUSTRACIÓN 4-1 EJEMPLO DE SOBREENRENAMIENTO .....	61
ILUSTRACIÓN 4-2 EL ALGORITMO AQR .....	64
ILUSTRACIÓN 4-3 EL PROCEDIMIENTO STAR .....	64
ILUSTRACIÓN 4-4 ALGORITMO CN2 .....	66
ILUSTRACIÓN 4-5 PROCEDIMIENTO MEJORCOMPLEJO .....	66
ILUSTRACIÓN 4-6 ALGORITMO IREP.....	68
ILUSTRACIÓN 4-7 ALGORITMO TDIDT .....	70
ILUSTRACIÓN 4-8 ALGORITMO ID3 .....	71
ILUSTRACIÓN 4-9 PROCEDIMIENTO DE ID3 PARA LA EXPANSIÓN DE UN ÁRBOL .....	72
ILUSTRACIÓN 4-10 PROCEDIMIENTO DE ID3 PARA LA CONSTRUCCIÓN DE UN ÁRBOL .....	74
ILUSTRACIÓN 4-11 PROCEDIMIENTOS AUXILIARES DE ID3.....	75
ILUSTRACIÓN 5-1 EJEMPLO DE IMAGEN GENERADA POR EL SONAR EMPLEADO.....	88
ILUSTRACIÓN 6-1 IMAGEN TÍPICA DE LA “SECUENCIA CON ATÚN” .....	93
ILUSTRACIÓN 6-2 IMAGEN TÍPICA DE LA “SECUENCIA VACÍA” .....	94
ILUSTRACIÓN 6-3 IMAGEN TÍPICA DE LA “SECUENCIA CON RUIDO DE SUPERFICIE” .....	95
ILUSTRACIÓN 6-4 EJEMPLO DE PREPROCESAMIENTO 1 .....	98
ILUSTRACIÓN 6-5 EJEMPLO DE PREPROCESAMIENTO 2 .....	99
ILUSTRACIÓN 6-6 EJEMPLO DE PREPROCESAMIENTO 3 .....	100
ILUSTRACIÓN 6-7 EJEMPLO DE PREPROCESAMIENTO 4 .....	100
ILUSTRACIÓN 6-8 EJEMPLO DE PREPROCESAMIENTO 5 .....	101
ILUSTRACIÓN 6-9 DIAGRAMA DE LA COMPLEJIDAD RELATIVA DE PREPROCESAMIENTOS .....	101
ILUSTRACIÓN 6-10 DIAGRAMA DE LAS OPERACIONES DE LOS PREPROCESAMIENTOS .....	102
ILUSTRACIÓN 7-1 DIAGRAMA DE CASOS DE USO.....	107
ILUSTRACIÓN 7-2 DEPENDENCIAS ENTRE PAQUETES.....	112
ILUSTRACIÓN 7-3 PATRÓN MODELO-VISTA-CONTROLADOR ESPECÍFICO IMPLEMENTADO.....	113
ILUSTRACIÓN 8-1 EXPERIMENTO 1: BOXPLOT DE PREPROCESAMIENTOS PARA LA PRECISIÓN TENIDOS EN CUENTA TODOS LOS CLASIFICADORES.....	119
ILUSTRACIÓN 8-2 EXPERIMENTO 1: BOXPLOT DE PREPROCESAMIENTOS PARA LA PRECISIÓN TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	119
ILUSTRACIÓN 8-3 EXPERIMENTO 1: BOXPLOT DE PREPROCESAMIENTOS PARA LA SEGMENTACIÓN TENIDOS EN CUENTA TODOS LOS CLASIFICADORES.....	120

ILUSTRACIÓN 8-4 EXPERIMENTO 1: BOXPLOT DE PREPROCESAMIENTOS PARA LA SEGMENTACIÓN TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	120
ILUSTRACIÓN 8-5 EXPERIMENTO 1: BOXPLOT DE PREPROCESAMIENTOS PARA EL F1 SCORE TENIDOS EN CUENTA TODOS LOS CLASIFICADORES .....	121
ILUSTRACIÓN 8-6 EXPERIMENTO 1: BOXPLOT DE PREPROCESAMIENTOS PARA EL F1 SCORE TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	121
ILUSTRACIÓN 8-7 EXPERIMENTO 2: BOXPLOT DE SEGMENTACIONES PARA LA PRECISIÓN TENIDOS EN CUENTA TODOS LOS CLASIFICADORES .....	122
ILUSTRACIÓN 8-8 EXPERIMENTO 2: BOXPLOT DE SEGMENTACIONES PARA LA PRECISIÓN TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	122
ILUSTRACIÓN 8-9 EXPERIMENTO 2: BOXPLOT DE SEGMENTACIONES PARA LA SEGMENTACIÓN TENIDOS EN CUENTA TODOS LOS CLASIFICADORES .....	123
ILUSTRACIÓN 8-10 EXPERIMENTO 2: BOXPLOT DE SEGMENTACIONES PARA LA SEGMENTACIÓN TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	123
ILUSTRACIÓN 8-11 EXPERIMENTO 2: BOXPLOT DE SEGMENTACIONES PARA EL F1 SCORE TENIDOS EN CUENTA TODOS LOS CLASIFICADORES .....	124
ILUSTRACIÓN 8-12 EXPERIMENTO 2: BOXPLOT DE SEGMENTACIONES PARA EL F1 SCORE TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	124
ILUSTRACIÓN 8-13 EXPERIMENTO 3: BOXPLOT DE EXTRACCIONES DE CARACTERÍSTICAS PARA LA PRECISIÓN TENIDOS EN CUENTA TODOS LOS CLASIFICADORES .....	125
ILUSTRACIÓN 8-14 EXPERIMENTO 3: BOXPLOT DE EXTRACCIONES DE CARACTERÍSTICAS PARA LA PRECISIÓN TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	125
ILUSTRACIÓN 8-15 EXPERIMENTO 3: BOXPLOT DE EXTRACCIONES DE CARACTERÍSTICAS PARA LA SEGMENTACIÓN TENIDOS EN CUENTA TODOS LOS CLASIFICADORES .....	126
ILUSTRACIÓN 8-16 EXPERIMENTO 3: BOXPLOT DE EXTRACCIONES DE CARACTERÍSTICAS PARA LA SEGMENTACIÓN TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	126
ILUSTRACIÓN 8-17 EXPERIMENTO 3: BOXPLOT DE EXTRACCIONES DE CARACTERÍSTICAS PARA EL F1 SCORE TENIDOS EN CUENTA TODOS LOS CLASIFICADORES .....	127
ILUSTRACIÓN 8-18 EXPERIMENTO 3: BOXPLOT DE EXTRACCIONES DE CARACTERÍSTICAS PARA EL F1 SCORE TENIDOS EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	127
ILUSTRACIÓN 8-19 EXPERIMENTO 3: BOXPLOT DE CLASIFICADORES PARA LA PRECISIÓN .....	128
ILUSTRACIÓN 8-20 EXPERIMENTO 3: BOXPLOT DE CLASIFICADORES PARA LA SENSITIVIDAD .....	129
ILUSTRACIÓN 8-21 EXPERIMENTO 3: BOXPLOT DE CLASIFICADORES PARA EL F1 SCORE .....	129
ILUSTRACIÓN 11-1 EJEMPLO: IMAGEN E IMAGEN TRANSFORMADA .....	143
ILUSTRACIÓN 11-2 FAVORECIENDO LA PRESENTACIÓN VISUAL MEDIANTE EL INTERCAMBIO DE MITADES .....	144

## 1.2 Índice de fórmulas

FÓRMULA 3-1 DESCRIPCIÓN DE UNA IMAGEN DIGITAL BIDIMENSIONAL.....	25
FÓRMULA 3-2 DESCRIPCIÓN DE UN PÍXEL DE UNA IMAGEN DIGITAL BIDIMENSIONAL .....	27
FÓRMULA 3-3 DESCRIPCIÓN DEL CONJUNTO DE PÍXELES DE UNA IMAGEN DIGITAL BIDIMENSIONAL .....	27
FÓRMULA 3-4 FUNCIÓN DE VECINDAD DE UN PÍXEL EXPRESADA EN SUS COMPONENTES ESPACIALES .....	27
FÓRMULA 3-5 FUNCIÓN DE 4-VECINDAD DE UN PÍXEL DE UNA IMAGEN DIGITAL BIDIMENSIONAL.....	28
FÓRMULA 3-6 FUNCIÓN DE 8-VECINDAD DE UN PÍXEL DE UNA IMAGEN DIGITAL BIDIMENSIONAL.....	28
FÓRMULA 3-7 FUNCIÓN 4-ADYACENTES.....	29
FÓRMULA 3-8 FUNCIÓN 8-ADYACENTES.....	29
FÓRMULA 3-9 FUNCIÓN CAMINO.....	29
FÓRMULA 3-10 FUNCIÓN DE CONECTIVIDAD ENTRE DOS PÍXELES DADO UN CONJUNTO .....	29
FÓRMULA 3-11 DEFINICIÓN DE COMPONENTE CONEXA.....	30
FÓRMULA 3-12 FUNCIÓN QUE DISCRIMINA SI ES REGIÓN O NO .....	30
FÓRMULA 3-13 DEFINICIÓN DE FRONTERA .....	30
FÓRMULA 3-14 DEFINICIÓN DE DISTANCIA .....	31
FÓRMULA 3-15 DEFINICIÓN DE DISTANCIA EUCLÍDEA BIDIMENSIONAL.....	31
FÓRMULA 3-16 DEFINICIÓN DE DISTANCIA MANHATTAN BIDIMENSIONAL.....	31
FÓRMULA 3-17 DESCRIPCIÓN DE LA SATURACIÓN EN EL MODELO HSL .....	35
FÓRMULA 3-18 DESCRIPCIÓN DE LA SATURACIÓN EN EL MODELO HSV.....	35
FÓRMULA 3-19 DESCRIPCIÓN DE LA SATURACIÓN EN EL MODELO HSI.....	36
FÓRMULA 3-20 PROCESO SOBRE UNA IMAGEN COMO CONJUNTO DE FUNCIONES .....	37
FÓRMULA 3-21 TRANSFORMACIÓN DE UNA IMAGEN EN OTRA COMO CONJUNTO DE FUNCIONES.....	37
FÓRMULA 3-22 EJEMPLO: NIVEL DE GRIS TRANSFORMADO.....	43
FÓRMULA 3-23 DERIVADA DE PRIMER ORDEN .....	43
FÓRMULA 3-24 DERIVADA DE SEGUNDO ORDEN .....	43
FÓRMULA 3-25 ECUACIÓN CONVENCIONAL DE LOS FILTROS DE SUAVIZADO EN EL DOMINIO DE LA FRECUENCIA.....	44
FÓRMULA 3-26 FUNCIÓN DE FILTRO DE SUAVIZADO DE TIPO IDEAL .....	44
FÓRMULA 3-27 FUNCIÓN DE FILTRO DE SUAVIZADO DE TIPO GAUSSIANO.....	46
FÓRMULA 3-28 ECUACIÓN CONVENCIONAL DE LOS FILTROS DE AFILADO EN EL DOMINIO DE LA FRECUENCIA.....	47
FÓRMULA 3-29 FUNCIÓN DE FILTRO DE AFILADO DE TIPO IDEAL .....	47
FÓRMULA 3-30 FUNCIÓN DE FILTRO DE AFILADO DE TIPO GAUSSIANO .....	48
FÓRMULA 3-31 GRADIENTE EN EL CASO BIDIMENSIONAL .....	50
FÓRMULA 3-32 MÓDULO DEL GRADIENTE EN EL CASO BIDIMENSIONAL .....	51
FÓRMULA 3-33 COMPACIDAD DE LA REGIÓN .....	54
FÓRMULA 3-34 DESCRIPTOR R DE LA SUAVIDAD .....	55
FÓRMULA 4-1 MATRIZ DE CONFUSIÓN.....	59
FÓRMULA 4-2 MATRIZ DE COSTES.....	59
FÓRMULA 4-3 COSTE .....	60
FÓRMULA 4-4 EJEMPLO DE COSTE.....	60
FÓRMULA 4-5 MATRIZ DE CONFUSIÓN USUAL .....	60
FÓRMULA 4-6 CRITERIO DE MEJORA EN EL CRECIMIENTO DE UN COMPLEJO EN IREP .....	67
FÓRMULA 4-7 CRITERIO DE MEJORA EN LA PODA DE UN COMPLEJO EN IREP .....	67
FÓRMULA 4-8 CRITERIO DE MEJORA EN LA PODA DE UN COMPLEJO EN RIPPER .....	68
FÓRMULA 4-9 GAIN RATIO.....	75
FÓRMULA 4-10 CRITERIO DE PODA EN C4.5 TRAS LA INDUCCIÓN DEL ÁRBOL .....	76
FÓRMULA 4-11 LOGARITMO DE LA VEROSIMILITUD SIN PENALIZACIÓN .....	81
FÓRMULA 5-1 FUNCIÓN DE INTENSIDAD EN COLOR DESCONOCIDA.....	89
FÓRMULA 11-1 PROBABILIDAD CONDICIONADA.....	139
FÓRMULA 11-2 TEOREMA DE LA PROBABILIDAD TOTAL .....	139

# Tesis de máster

---

FÓRMULA 11-3 TEOREMA DE BAYES .....	139
FÓRMULA 11-4 INDEPENDENCIA ENTRE DOS VARIABLES .....	139
FÓRMULA 11-5 INDEPENDENCIA CONDICIONAL ENTRE DOS VARIABLES DADA UNA TERCERA.....	140
FÓRMULA 11-6 CONDICIÓN DE INTEGRABILIDAD ABSOLUTA .....	140
FÓRMULA 11-7 TRANSFORMADA DE FOURIER.....	141
FÓRMULA 11-8 TRANSFORMADA INVERSA DE FOURIER.....	141
FÓRMULA 11-9 TRANSFORMADA DE FOURIER PARA EL CASO BIDIMENSIONAL.....	141
FÓRMULA 11-10 TRANSFORMADA INVERSA DE FOURIER PARA EL CASO BIDIMENSIONAL.....	141
FÓRMULA 11-11 TRANSFORMADA DE FOURIER DISCRETA .....	141
FÓRMULA 11-12 TRANSFORMADA INVERSA DE FOURIER DISCRETA.....	142
FÓRMULA 11-13 TRANSFORMADA DE FOURIER DISCRETA PARA EL CASO BIDIMENSIONAL .....	142
FÓRMULA 11-14 TRANSFORMADA INVERSA DE FOURIER DISCRETA PARA EL CASO BIDIMENSIONAL.....	142
FÓRMULA 11-15 RESULTADO DE LA TRANSFORMADA EN COORDENADAS POLARES.....	142
FÓRMULA 11-16 ESCALADO DE LA IMAGEN DESCRITA EN EL DOMINIO DE LA FRECUENCIA .....	144
FÓRMULA 11-17 BONDAD ESTIMADA POR BOOTSTRAPPING .....	146
FÓRMULA 11-18 BONDAD ESTIMADA POR BOOTSTRAPPING REITERADO .....	146
FÓRMULA 11-19 CANTIDAD DE INFORMACIÓN DE UN SUCESO.....	146
FÓRMULA 11-20 ENTROPÍA DE SHANNON.....	147
FÓRMULA 11-21 ENTROPÍA DE LA VARIABLE ALEATORIA BIDIMENSIONAL CONJUNTA.....	147
FÓRMULA 11-22 ENTROPÍA DE UNA VARIABLE ALEATORIA CONDICIONADA A UN VALOR ESPECÍFICO DE OTRA .....	147
FÓRMULA 11-23 ENTROPÍA DE UNA VARIABLE ALEATORIA CONDICIONADA A OTRA.....	147
FÓRMULA 11-24 LEY DE ENTROPÍAS TOTALES.....	147
FÓRMULA 11-25 PROPIEDAD RELATIVA A LA ENTROPÍA DE VARIABLES ALEATORIAS INDEPENDIENTES .....	147
FÓRMULA 11-26 DIVERGENCIA KULLBACK-LEIBLER.....	148
FÓRMULA 11-27 CANTIDAD DE INFORMACIÓN MUTUA.....	148



## 1.3 Índice de tablas

TABLA 4-1 PLANTEAMIENTO DEL PROBLEMA DE LA CLASIFICACIÓN SUPERVISADA .....	58
TABLA 7-1 CU00: PREPROCESAMIENTO .....	108
TABLA 7-2 CU01: SEGMENTACIÓN .....	109
TABLA 7-3 CU02: EXTRACCIÓN DE CARACTERÍSTICAS .....	110
TABLA 7-4 CU03: ETIQUETADO ASISTIDO .....	111
TABLA 11-1 EXPERIMENTO 1: MEDIAS Y DESVIACIONES TÍPICAS DE TP, FP Y FN .....	150
TABLA 11-2 EXPERIMENTO 1: MEDIAS Y DESVIACIONES TÍPICAS DE PRECISIÓN, SENSITIVIDAD Y F1 SCORE .....	151
TABLA 11-3 EXPERIMENTO 1: CONTRASTE DE PREPROCESAMIENTOS SEGÚN PRECISIÓN TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	152
TABLA 11-4 EXPERIMENTO 1: CONTRASTE DE PREPROCESAMIENTOS SEGÚN PRECISIÓN TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	152
TABLA 11-5 EXPERIMENTO 1: CONTRASTE DE PREPROCESAMIENTOS SEGÚN SENSITIVIDAD TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	152
TABLA 11-6 EXPERIMENTO 1: CONTRASTE DE PREPROCESAMIENTOS SEGÚN SENSITIVIDAD TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	152
TABLA 11-7 EXPERIMENTO 1: CONTRASTE DE PREPROCESAMIENTOS SEGÚN F1 SCORE TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	152
TABLA 11-8 EXPERIMENTO 1: CONTRASTE DE PREPROCESAMIENTOS SEGÚN F1 SCORE TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	153
TABLA 11-9 EXPERIMENTO 1: SUPERIORIDAD COMPLETA ENTRE PREPROCESAMIENTOS TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	153
TABLA 11-10 EXPERIMENTO 1: SUPERIORIDAD COMPLETA ENTRE PREPROCESAMIENTOS TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	153
TABLA 11-11 EXPERIMENTO 2: MEDIAS Y DESVIACIONES TÍPICAS DE TP, FP Y FN .....	154
TABLA 11-12 EXPERIMENTO 2: MEDIAS Y DESVIACIONES TÍPICAS DE PRECISIÓN, SENSITIVIDAD Y F1 SCORE .....	155
TABLA 11-13 EXPERIMENTO 2: CONTRASTE DE SEGMENTACIONES SEGÚN PRECISIÓN TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	156
TABLA 11-14 EXPERIMENTO 2: CONTRASTE DE SEGMENTACIONES SEGÚN PRECISIÓN TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	156
TABLA 11-15 EXPERIMENTO 2: CONTRASTE DE SEGMENTACIONES SEGÚN SENSITIVIDAD TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	156
TABLA 11-16 EXPERIMENTO 2: CONTRASTE DE SEGMENTACIONES SEGÚN SENSITIVIDAD TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	156
TABLA 11-17 EXPERIMENTO 2: CONTRASTE DE SEGMENTACIONES SEGÚN F1 SCORE TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	156
TABLA 11-18 EXPERIMENTO 2: CONTRASTE DE SEGMENTACIONES SEGÚN F1 SCORE TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	156
TABLA 11-19 EXPERIMENTO 2: SUPERIORIDAD COMPLETA ENTRE SEGMENTACIONES TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	157
TABLA 11-20 EXPERIMENTO 2: SUPERIORIDAD COMPLETA ENTRE SEGMENTACIONES TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	157
TABLA 11-21 EXPERIMENTO 3: MEDIAS Y DESVIACIONES TÍPICAS DE TP, FP Y FN .....	158
TABLA 11-22 EXPERIMENTO 3: MEDIAS Y DESVIACIONES TÍPICAS DE PRECISIÓN, SENSITIVIDAD Y F1 SCORE .....	159
TABLA 11-23 EXPERIMENTO 3: CONTRASTE DE EXTRACCIONES DE CARACTERÍSTICAS SEGÚN PRECISIÓN TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	160
TABLA 11-24 EXPERIMENTO 3: CONTRASTE DE EXTRACCIONES DE CARACTERÍSTICAS SEGÚN PRECISIÓN TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	160

TABLA 11-25 EXPERIMENTO 3: CONTRASTE DE EXTRACCIONES DE CARACTERÍSTICAS SEGÚN SENSITIVIDAD TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	160
TABLA 11-26 EXPERIMENTO 3: CONTRASTE DE EXTRACCIONES DE CARACTERÍSTICAS SEGÚN SENSITIVIDAD TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	160
TABLA 11-27 EXPERIMENTO 3: CONTRASTE DE EXTRACCIONES DE CARACTERÍSTICAS SEGÚN F1 SCORE TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	160
TABLA 11-28 EXPERIMENTO 3: CONTRASTE DE EXTRACCIONES DE CARACTERÍSTICAS SEGÚN F1 SCORE TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	160
TABLA 11-29 EXPERIMENTO 3: SUPERIORIDAD COMPLETA ENTRE EXTRACCIONES DE CARACTERÍSTICAS TENIENDO EN CUENTA TODOS LOS CLASIFICADORES .....	161
TABLA 11-30 EXPERIMENTO 3: SUPERIORIDAD COMPLETA ENTRE EXTRACCIONES DE CARACTERÍSTICAS TENIENDO EN CUENTA LOS CLASIFICADORES “INTELIGENTES” .....	161
TABLA 11-31 EXPERIMENTO 3: CONTRASTE DE CLASIFICADORES SEGÚN PRECISIÓN .....	161
TABLA 11-32 EXPERIMENTO 3: CONTRASTE DE CLASIFICADORES SEGÚN SENSITIVIDAD .....	161
TABLA 11-33 EXPERIMENTO 3: CONTRASTE DE CLASIFICADORES SEGÚN F1 SCORE .....	161
TABLA 11-34 EXPERIMENTO 3: SUPERIORIDAD COMPLETA ENTRE CLASIFICADORES.....	162



## 2 Introducción

En esta tesis de máster se trata la problemática del procesamiento de señales de atún detectadas mediante sonar de largo alcance. La idea es encontrar una solución software que permita el análisis automático de imágenes digitales obtenidas mediante sonar de largo alcance a bordo de buques que pescan atún rojo y atún blanco en el golfo de Bizkaia. Así pues se dispone de un conjunto de secuencias de imágenes de sonar, de la información extraída de la observación directa y juicio experto en imágenes de sonar. El juicio experto permite establecer una correspondencia fiel a la realidad entre elementos gráficos presentes en las imágenes de sonar y aquel objeto o concepto que produce dicho elemento gráfico (como pueden ser los bancos de atún, el ruido o los artefactos).

La memoria de esta tesis de máster está organizada de la siguiente forma. Tras realizar una introducción a la problemática y al entorno de la tesis de máster, se realiza una exposición de conceptos básicos y generales del área de la ciencia del procesamiento de imagen digital. Después se habla de conceptos básicos y generales del área de la ciencia de la clasificación supervisada. Tras hablar de estas dos áreas de la ciencia, se está en condiciones de abordar la temática de las imágenes de sonar, en la que se comentan especificidades inherentes a las mismas. Posteriormente, se describen los ensayos realizados con las secuencias de imágenes de sonar, explicando qué se hace en los mismos. Tras esto se habla de la aplicación software Sonar, que junto con los resultados de los ensayos es el principal producto de esta tesis de máster. Después se documentan los resultados de los ensayos. Posteriormente se extraen conclusiones de los ensayos y de los resultados. Tras esto se comentan algunas de las posibles líneas futuras mediante las cuales se trataría de mejorar lo elaborado en este trabajo. Después figuran los anexos. En el anexo de conceptos básicos de probabilidad se recogen ideas que aparecen a lo largo de la memoria. En el anexo de la transformada de Fourier, se recoge una descripción somera de este operador matemático, para apoyar la comprensión de algunos apartados del capítulo concerniente al procesamiento de imagen digital. En el anexo de técnicas de validación y en el anexo de teoría de la información estadística se recogen conceptos mencionados en el capítulo llamado clasificación supervisada. En el anexo cálculos matemáticos de resultados, se muestran las operaciones matemáticas que permiten resolver algunos detalles que plantean problemas a la hora de obtener los resultados. En el anexo tablas de resultados se recogen las tablas que detallan los resultados que se comentan someramente en el capítulo de resultados. Finalmente se dispone la bibliografía en la que se recogen las fuentes en las que se ha apoyado este trabajo.

Las referencias bibliográficas (Rae) y (Wiki) se han empleado como soporte de conocimiento general. Las referencias (Froufe, 2005), (J. Sánchez, 2005) y (Larman, 2003) se han consultado para el desarrollo de software en Java y para la elaboración de la arquitectura software de la aplicación creada en esta tesis de fin de máster. Las referencias (Hernando, 1999), (Hernández, 1994), (Novoa, 1991) y (M. T. González, 2004) se han empleado como soporte general en ciencias matemáticas. Las referencias (J. F. Velez, 2007), (R. C. González, 2002) y (R. Jain, 1995) se han empleado como soporte en la ciencia del procesamiento de imagen. (S. Russell, 2004) y (Sierra, 2006) se han empleado como soporte en inteligencia artificial y aprendizaje automático.

## 2.1 AZTI-Tecnalia

AZTI-Tecnalia es un centro tecnológico experto en investigación marina y alimentaria. Tiene como objetivo el desarrollo social y la mejora de la competitividad en los sectores del mar y la alimentación mediante la innovación tecnológica. AZTI-Tecnalia dispone actualmente de tres sedes. Una de ellas es la sede central y unidad de investigación marina, está ubicada en Sukarrieta (Bizkaia), Txatxarramendi ugarte z/g. Otra de ellas es unidad de investigación alimentaria y está ubicada en el parque tecnológico de Bizkaia, en Derio, Astondo Bidea, edificio 609. La otra sede es una unidad de investigación marina y está situada en Pasaia (Guipuzkoa), Herrera Kaia, Portualdea z/g. Es en comunión con esta última sede con la que se ha llevado a cabo esta tesis de máster. Concretamente se ha realizado un contrato de prácticas en AZTI-Tecnalia, siendo Haritz Arrizabalaga el tutor en el centro tecnológico AZTI-Tecnalia, siendo Yosu Yurramendi el tutor en la UPV/EHU y siendo Ari Urkullu el becario.

## 2.2 Objetivos de la tesis de máster

El objetivo de la tesis de máster es el de construir un proceso automatizado de análisis de imágenes de sonar de largo alcance. Mediante dicho proceso, en una imagen de sonar se podrá discernir los componentes existentes registrados por el sonar, para poder describirlos en función de un conjunto de características medibles para los objetos identificados. Finalmente mediante ayuda de juicio experto se identifica qué es cada objeto, para crear una base de datos para realizar un aprendizaje supervisado, para obtener un modelo de clasificación con el cual clasificar objetos entrantes que se desconoce qué son. En definitiva el deseo es dar un primer paso en la investigación de procesamiento automático de imágenes de sonar de largo alcance.

## 2.3 Objetivos últimos de la investigación

En este apartado se recoge la motivación que impulsa la investigación en el campo tratado en esta tesis de máster. El deseo final es, realizar una estimación en tiempo real de la cuantía y ubicación de atún blanco y atún rojo en las imágenes de sonar que resulte lo suficientemente próxima a lo real. Sin embargo la complejidad del problema planteado requiere una investigación mayor de la que se espera abarcar en una tesis de fin de máster. Esto unido al hecho de que no se ha conseguido localizar bibliografía accesible de una temática tan específica como es el procesamiento automático de señales de atún obtenidas mediante sonar de largo alcance, hace que en la tesis de máster se parta desde la nada hasta los tratamientos más sencillos. No obstante en el apartado líneas futuras se recogen un conjunto de ideas por donde la investigación en esta área del conocimiento podría avanzar, aunque no sean más que unas directrices someramente descritas.

## 2.4 Productos de la tesis de máster

Existen dos principales productos en esta tesis de máster. Uno de ellos es la aplicación software Sonar desarrollada, que permite procesar imágenes de sonar entrantes. En concreto, permite preprocesarlas ensalzando su calidad para el proceso de segmentación, en el cual una imagen se descompone en los objetos identificados en la misma. La aplicación Sonar también permite segmentar imágenes de sonar previamente preprocesadas con la aplicación Sonar. Las imágenes

## Tesis de máster

---

resultantes de las segmentaciones de la aplicación Sonar pueden procesarse nuevamente con la aplicación Sonar para extraer características de las mismas, generando una base de datos para aprendizaje supervisado cuasi-completa, en la que falta exclusivamente indicar qué es cada objeto descrito. Para esta etiquetación de los objetos, la aplicación Sonar ofrece una funcionalidad añadida que asiste a la etiquetación manual, abreviando en gran cuantía el tiempo que se invierte en este etiquetado. Tras este etiquetado asistido, la base de datos generada es apta para ser empleada con el software de libre distribución Weka bajo licencia GNU. El otro producto principal de esta tesis de máster son los ensayos realizados y los resultados obtenidos, con los distintos procesos empleados, en el procesamiento de imágenes de sonar.



## 3 Procesamiento de imagen

### 3.1 La luz

#### 3.1.1 La naturaleza de la luz

La luz es de naturaleza dual, pues su propagación se rige por sus propiedades ondulatorias, mientras que el intercambio de energía entre luz y materia se guía por las propiedades corpusculares de la luz. La naturaleza ondulatoria de la luz (su comportamiento como onda electromagnética) fue demostrada por Thomas Young, al iluminar con una fuente de luz un par de rendijas paralelas tras las cuales había una pared, y observar que en la pared se mostraba una imagen propia de la interferencia de dos fuentes luminosas. La naturaleza corpuscular de la luz (su comportamiento como conjunto de fotones) fue propuesta por Albert Einstein para la explicación del efecto fotoeléctrico (explicación por la cual recibiría el premio nobel), en el cual un haz de luz incide sobre un material, y éste, libera electrones.

#### 3.1.2 La descomposición de la luz

La luz puede descomponerse en un espectro continuo separándose en sus distintas longitudes de onda. La longitud de onda es el espacio recorrido por un punto mientras éste realiza una oscilación completa, sin incluir dentro de ese espacio recorrido el espacio recorrido provocado por el movimiento de oscilación. El primer hallazgo de la descomposición de la luz fue realizado por Newton al pasar un rayo de sol a través de un prisma de vidrio. El ángulo de la luz refractada en el prisma de vidrio depende de la longitud de onda, por lo que en la imagen de la luz refractada pudo observar la luz separada en sus diferentes longitudes de onda.

#### 3.1.3 La percepción de la luz

El espectro de la luz, el humano lo percibe a través de sus ojos como colores. Los colores están asociados con las longitudes de onda de la luz. Pero no se puede hablar de la existencia de una aplicación biyectiva de los colores percibidos por humanos en longitudes de onda de la luz. Primero, porque no sería suprayectiva, ya que el ojo humano sólo percibe luz en el entorno de longitudes de onda de [400,700] nm (según se puede ver en la figura 1). Segundo porque no sería inyectiva por fenómenos como el que el cerebro interpreta la mezcla de luz verde y luz roja como amarillo, a pesar de la ausencia de luz amarilla. Esto implica que podría suceder que el mismo color percibido tenga correspondencias distintas en longitudes de onda del espectro electromagnético. Finalmente, ni siquiera sería una aplicación, puesto que algunos colores que el ojo puede percibir se corresponden con la adición de luces de distintas longitudes de onda en el espectro, como ocurre por ejemplo, con el color blanco.



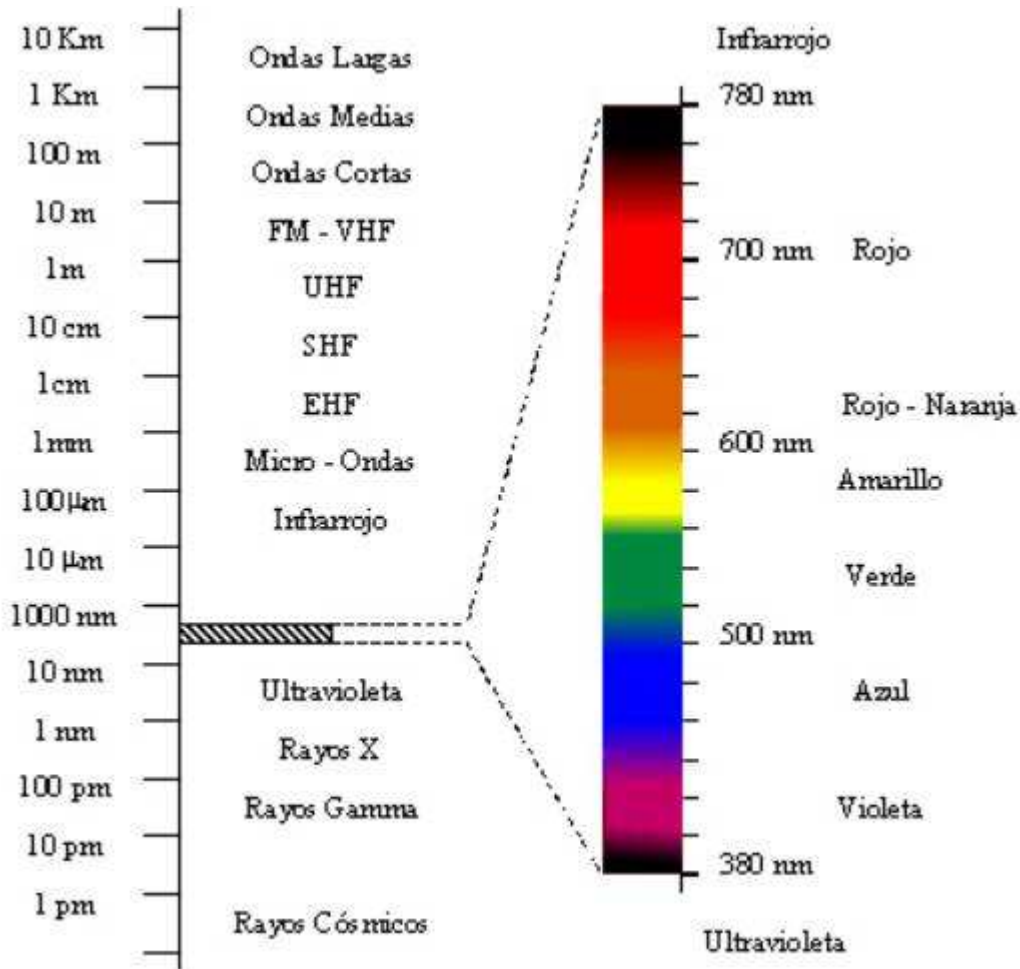


Ilustración 3-1 El espectro electromagnético

### 3.2 La imagen digital

#### 3.2.1 La imagen digital de dos dimensiones espaciales

Una imagen digital de dos dimensiones espaciales se puede describir como un conjunto de aplicaciones donde cada una de ellas toma siempre como variables independientes las dos variables que describen la posición en el plano de la imagen. El conjunto X representa un conjunto discreto, finito y ordenable de elementos que representan la posición espacial en uno de los ejes del plano espacial en el que se describe la imagen, mientras que Y representa un conjunto discreto, finito y ordenable de elementos que representan la posición espacial en el otro eje del par de ejes que conforman el plano espacial de la imagen.

$$f_1: X \times Y \rightarrow D_1, \dots, f_n: X \times Y \rightarrow D_n$$

$$I = \{f_1, \dots, f_n\}$$

Fórmula 3-1 Descripción de una imagen digital bidimensional

Para que se considere una imagen digital, el conjunto de origen y el conjunto de destino de cada función, ambos, deben ser discretos y finitos. De ahora en adelante, siempre que se mencione el concepto de imagen se asumirá que se está hablando de imagen digital bidimensional a no ser

que se denote específicamente que se está hablando de otro tipo de imagen. Adicionalmente, se asumirá que la imagen respeta las convenciones que en el siguiente apartado se exponen.

## 3.2.2 Convenciones en imágenes digitales bidimensionales

En la descripción de imágenes existen una serie de convenciones. Una de ellas, es que el eje horizontal se denomina Y y el eje vertical se llama X. Otra convención es que los valores en el eje X crecen según se desciende en él, y en el eje Y los valores crecen según se avanza hacia la derecha. Cabe mencionar también, la convención por la cual se emplea el conjunto de los números naturales más el cero como conjunto de valores que puede tomar cada dimensión espacial. También se presenta la convención de que entre el valor mínimo y el máximo en una dimensión, se presentan todos los valores que describen un punto de la imagen. Por ejemplo, en una dimensión espacial entre el cero y un valor máximo n que es un número natural, se presentan en la dimensión todos los valores naturales entre el cero y dicho valor. Finalmente, existe la convención de que el llamado punto de origen (el que tiene el menor valor en cada componente espacial), tiene un valor igual a cero en cada componente espacial. Indicar que una imagen digital bidimensional tal y como se ha descrito, que además respete las convenciones, será una imagen rectangular.

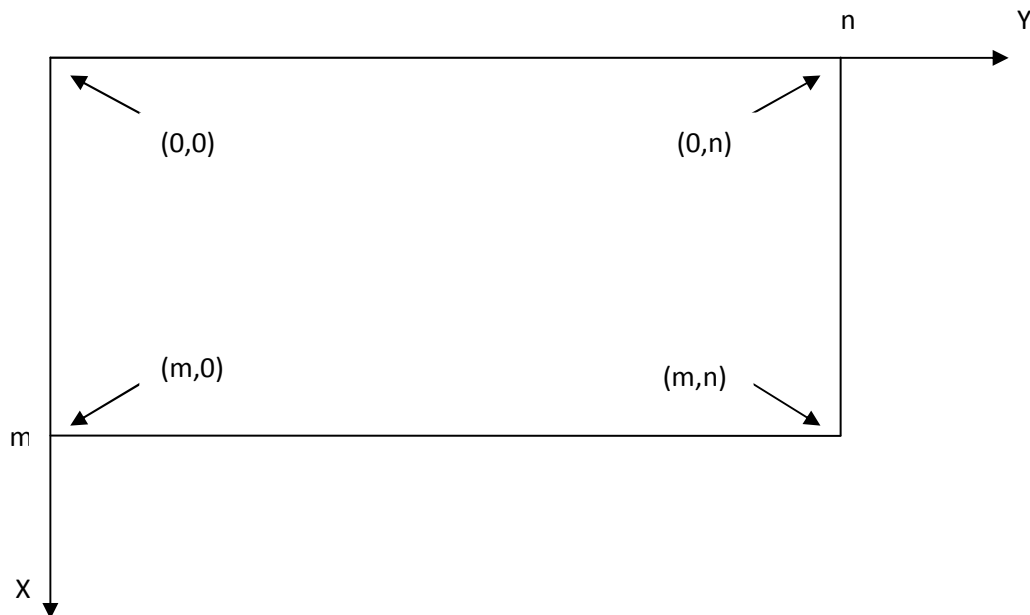


Ilustración 3-2 Representación de una imagen digital bidimensional según las convenciones

## 3.2.3 El píxel

Un píxel de una imagen queda definido por un par de coordenadas específicas pertenecientes al plano de la imagen, y el conjunto de resultados obtenidos de las aplicaciones definidas por la imagen para dichas coordenadas específicas. En la siguiente fórmula se puede observar el píxel *i*-ésimo.

$$p_i = \{x_i, y_i, a_i^1 = f_1(x_i^1, y_i^1), \dots, a_i^n = f_n(x_i^n, y_i^n)\}$$

### Fórmula 3-2 Descripción de un píxel de una imagen digital bidimensional

Denotar que para una imagen dada un píxel puede ser aludido tan sólo con su par de coordenadas espaciales dentro de la imagen misma. Es decir, en una imagen, un par de coordenadas específicas aluden de forma unívoca a un píxel en concreto. Mediante el producto cartesiano de los conjuntos que describen cada una de las componentes espaciales de la imagen, se pueden sacar todos los pares de coordenadas específicas. Finalmente, con esos pares y las aplicaciones que definen la imagen, se pueden definir completamente todos los píxeles de la imagen. Decir que para un píxel dado, al par que conforman sus dos componentes que definen su ubicación espacial se le llama posición, y al conjunto que forman el resto de valores que describen el píxel se le llama valor. Por último, se define el conjunto de píxeles de una imagen,  $P$ , resultado de la unión elemento a elemento, del producto cartesiano de los conjuntos de coordenadas espaciales con los conjuntos resultantes de emplear las aplicaciones que definen la imagen sobre el conjunto resultante del mencionado producto cartesiano.

$${}^I P = \text{píxeles}(I) = \{p_1, \dots, p_m\} /: (x_i, y_i) \in D_f(f_j) a_i^j = f_j(x_i, y_i), \forall i \in [1, m] \forall j \in [1, n]$$

### Fórmula 3-3 Descripción del conjunto de píxeles de una imagen digital bidimensional

Indicar que, es usual decir que un píxel pertenece a una imagen, si dicho píxel pertenece al conjunto de píxeles de la imagen.

#### 3.2.4 La vecindad de un píxel

La vecindad de un píxel alude a un conjunto de píxeles espacialmente próximos al mismo en una imagen. Hay que tomar una decisión sobre qué píxel es próximo y qué píxel no es próximo. La vecindad se puede entender como una función que se aplica a un píxel y el conjunto de píxeles de una imagen y devuelve un conjunto de  $n$  píxeles que son vecinos del primer píxel dentro del conjunto antes citado.

$$\text{vecindad}(p, C \subset {}^I P) = \{p_1, \dots, p_n\} /: p \in C \ p_i \in C, \forall i \in [1, n]$$

### Fórmula 3-4 Función de vecindad de un píxel expresada en sus componentes espaciales

Existen un conjunto de vecindades de píxeles de uso muy extendido, si bien son dos las más conocidas. La primera vecindad de esas dos, es la llamada 4-vecindad, donde el conjunto de vecinos para un píxel y un conjunto de píxeles dados, contiene a los píxeles que pertenezcan a ese conjunto y que además difieren en los valores de las componentes espaciales respecto al píxel para el cual se está definiendo la vecindad en una única componente, y además en dicha componente deben diferir en una única unidad. La segunda vecindad de las dos aludidas, es la llamada 8-vecindad, en la cual cada elemento del conjunto de vecinos para un píxel y un conjunto de píxeles dados, además de pertenecer a ese conjunto dado, cumple con la propiedad de diferir en cada componente espacial en una unidad respecto a las coordenadas espaciales del píxel cuya vecindad se está calculando. Esto se puede apreciar visualmente en la siguiente ilustración

(respecto al píxel central, los píxeles que son vecinos contienen un uno, y los que no lo son, un cero).

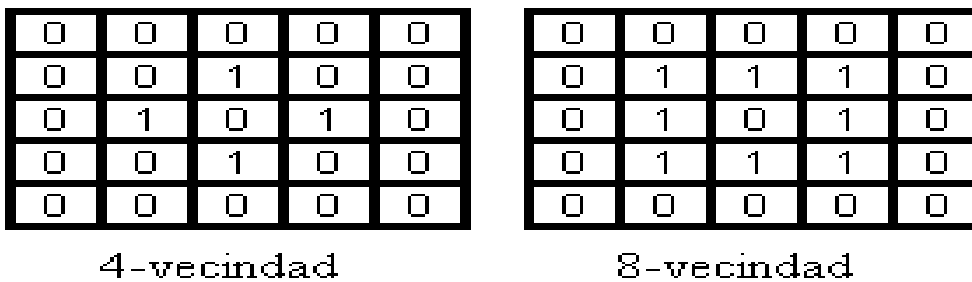


Ilustración 3-3 Representación visual de la 4-vecindad y la 8-vecindad

$$4 - \text{vecindad}(p_i, C \subset {}^1P) = v / : p \in v \Leftrightarrow p_i \in C p \in C($$

$$(x_i = x + 1 y_i = y)(x_i = x - 1 y_i = y)$$

$$(x_i = x y_i = y + 1)(x_i = x y_i = y - 1))$$

Fórmula 3-5 Función de 4-vecindad de un píxel de una imagen digital bidimensional

$$8 - \text{vecindad}(p_i, C \subset {}^1P) = v / : p \in v \Leftrightarrow p_i \in C p \in C($$

$$(x_i = x + 1 y_i = y + 1)(x_i = x + 1 y_i = y)(x_i = x + 1 y_i = y - 1)$$

$$(x_i = x y_i = y + 1)(x_i = x y_i = y)(x_i = x y_i = y - 1)$$

$$(x_i = x - 1 y_i = y + 1)(x_i = x - 1 y_i = y)(x_i = x - 1 y_i = y - 1))$$

Fórmula 3-6 Función de 8-vecindad de un píxel de una imagen digital bidimensional

### 3.2.5 La adyacencia entre pares de píxeles

Una función de adyacencia, toma dos píxeles y un conjunto de píxeles y determina si los dos píxeles son adyacentes o no. Existen múltiples funciones de adyacencia que operan no sólo basándose en las posiciones de los pares de píxeles que evalúan, sino también teniendo en cuenta los valores que toman los pares de píxeles. Sin embargo, aquí se exponen dos funciones de adyacencia que no tienen en cuenta más que la relación espacial entre los pares de píxeles que evalúan. Así pues, se presentan aquí las funciones 4-adyacentes y la 8-adyacentes. En la función 4-adyacentes se dice que dos píxeles son adyacentes si el primero es vecino del segundo y viceversa según el criterio de vecindad 4-vecindad, dado un conjunto de píxeles. En la función 8-adyacentes, dos píxeles son adyacentes si el primero es vecino del segundo y viceversa según el criterio de vecindad 8-vecindad, dado un conjunto de píxeles.

$$4 - \text{adyacentes}(p_1, p_2, C \subset {}^I P) \begin{cases} 1 \Leftrightarrow \begin{matrix} p_2 \in 4 - \text{vecindad}(p_1, C) \\ p_1 \in 4 - \text{vecindad}(p_2, C) \end{matrix} \\ 0 \Leftrightarrow \neg \begin{pmatrix} p_2 \in 4 - \text{vecindad}(p_1, C) \\ p_1 \in 4 - \text{vecindad}(p_2, C) \end{pmatrix} \end{cases}$$

Fórmula 3-7 Función 4-adyacentes

$$8 - \text{adyacentes}(p_1, p_2, C \subset {}^I P) \begin{cases} 1 \Leftrightarrow \begin{matrix} p_2 \in 8 - \text{vecindad}(p_1, C) \\ p_1 \in 8 - \text{vecindad}(p_2, C) \end{matrix} \\ 0 \Leftrightarrow \neg \begin{pmatrix} p_2 \in 8 - \text{vecindad}(p_1, C) \\ p_1 \in 8 - \text{vecindad}(p_2, C) \end{pmatrix} \end{cases}$$

Fórmula 3-8 Función 8-adyacentes

### 3.2.6 El camino entre pares de píxeles

El camino entre un par de píxeles de una imagen es una sucesión finita de píxeles cuyas posiciones verifican una propiedad. Tener en cuenta que el primer elemento de la sucesión es el primer elemento del par de píxeles cuyo camino se está describiendo, y el último elemento de la sucesión es el segundo elemento del mencionado par. La propiedad que se debe verificar es que todo elemento de la sucesión es adyacente del elemento precedente y del siguiente, dado un conjunto de píxeles. Por tanto, la decisión de si una sucesión de píxeles de una imagen es un camino depende de la definición de adyacencia en la que se basen, dado un conjunto de píxeles.

$$\text{camino}(\{p_0, p_n\}, C \subset {}^I P) \begin{cases} 1 \Leftrightarrow \begin{matrix} \text{adyacentes}(p_{i-1}, p_i, C) = 1 \\ \text{adyacentes}(p_i, p_{i+1}, C) = 1, \forall i \in [1, n-1] \end{matrix} \\ 0 \Leftrightarrow \neg \begin{pmatrix} \text{adyacentes}(p_{i-1}, p_i, C) = 1 \\ \text{adyacentes}(p_i, p_{i+1}, C) = 1, \forall i \in [1, n-1] \end{pmatrix} \end{cases}$$

Fórmula 3-9 Función camino

Denotar que dada la fórmula anterior, la longitud del camino será igual a  $n$ . Cuando ambos elementos del par de píxeles cuyo camino se está describiendo son el mismo, se dice que el camino es cerrado.

### 3.2.7 La conectividad entre píxeles

Dado un conjunto de píxeles de una imagen, se dice que dos píxeles están conectados en ese conjunto si existe un camino entre ambos para el cual se verifica que todas las coordenadas de posiciones que describen el camino pertenecen al conjunto mencionado.

$$\text{conectados}(p_i, p_j, C \subset {}^I P) \begin{cases} 1 \Leftrightarrow \begin{matrix} \exists \{p_1, \dots, p_n\} /: p_k = \{x_k, y_k, a_k^1, \dots, a_k^n\} \\ p_k \in C \\ \text{esCamino}((x_0, y_0), \dots, (x_n, y_n)), \forall k \in [0, n] \end{matrix} \\ 0 \Leftrightarrow \neg \begin{pmatrix} \exists \{p_1, \dots, p_n\} /: p_k = \{x_k, y_k, a_k^1, \dots, a_k^n\} \\ p_k \in C \\ \text{esCamino}((x_0, y_0), \dots, (x_n, y_n)), \forall k \in [0, n] \end{pmatrix} \end{cases}$$

Fórmula 3-10 Función de conectividad entre dos píxeles dado un conjunto



$$d: C \times C \rightarrow R$$

$$I. \quad d(c_1, c_2) \geq 0, \forall c_1, c_2 \in C$$

$$II. \quad d(c_1, c_1) = 0, \forall c_1 \in C$$

$$III. \quad d(c_1, c_2) = d(c_2, c_1), \forall c_1, c_2 \in C$$

$$IV. \quad d(c_1, c_2) \leq d(c_1, c_3) + d(c_3, c_2), \forall c_1, c_2, c_3 \in C$$

#### Fórmula 3-14 Definición de distancia

Para la medición de distancias entre píxeles de una imagen dada se suelen usar dos distancias principalmente. Una de ellas es la distancia Euclídea bidimensional y la otra es la distancia Manhattan bidimensional. Ambas se aplican a la posición del par de píxeles cuya distancia se desea medir.

$$d_e: \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$d_e((x_1, x_2), (y_1, y_2)) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

#### Fórmula 3-15 Definición de distancia Euclídea bidimensional

$$d_m: \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$$

$$d_m((x_1, x_2), (y_1, y_2)) = |x_1 - y_1| + |x_2 - y_2|$$

#### Fórmula 3-16 Definición de distancia Manhattan bidimensional

### 3.2.12 Espacios de color

Un espacio de color es un modelo matemático que permite representar colores como un conjunto de valores numéricos, donde cada uno de los valores numéricos representa una propiedad del color a describir. Dicho modelo se compone de un sistema de coordenadas donde cada componente aludirá a una propiedad de los colores, y de un par de conjuntos de aplicaciones. El sistema de coordenadas tiene un número de componentes discreto y finito, y cada componente toma valores de un conjunto ordenable y de valores acotados. Ambos conjuntos de aplicaciones tienen un número de elementos igual al número de componentes del sistema de coordenadas. El objetivo del par de conjuntos de aplicaciones es permitir la traducción de coordenadas específicas en intervalos y uniones de intervalos del espectro electromagnético, y viceversa. Esto puede hacerse directamente o indirectamente.

En caso de hacerse directamente cada aplicación del primer conjunto de aplicaciones es del conjunto de todos los intervalos y uniones de intervalos del espectro electromagnético en el conjunto de valores de una de las componentes del sistema de coordenadas (cada aplicación de dicho conjunto de aplicaciones en una componente distinta). Cada aplicación del segundo conjunto de aplicaciones es del conjunto de valores de una de las componentes del sistema de coordenadas (cada aplicación de dicho conjunto de aplicaciones en una componente distinta) en el conjunto de todos los intervalos y uniones de intervalos del espectro electromagnético. Cada

aplicación del cualquiera de los dos conjuntos de aplicaciones, no tiene por qué ser ni suprayectiva ni inyectiva. Mediante este par de conjuntos de aplicaciones se puede traducir el intervalo o unión de intervalos cubiertos del espectro electromagnético por un haz de luz al sistema de coordenadas del espacio de color, y traducir unas coordenadas específicas del sistema de coordenadas a un intervalo o unión de intervalos cubiertos del espectro electromagnético por un haz de luz.

En caso de hacerse indirectamente, el espacio de color en lugar de componerse del par de conjuntos de aplicaciones antes descrito, se compone de un par de conjuntos de aplicaciones (donde igualmente, ninguna aplicación tiene por qué ser ni suprayectiva ni inyectiva), que en lugar de tomar intervalos y uniones de intervalos del espectro electromagnético tomen coordenadas de otro espacio de color.

Los valores de los píxeles de una imagen, están expresados en un espacio de color discreto, o en una discretización específica de un espacio de color continuo. Una discretización de un espacio de color continuo se especifica mediante la declaración de una cantidad básica indivisible para cada componente. Lo usual es crear espacios de color que modelen la percepción humana del color (con mayor o menor precisión) y aquí se habla de algunos de los más empleados.

### ***3.2.12.1 Espacio blanco-negro***

El espacio blanco-negro es un espacio compuesto por un sistema de coordenadas de una única componente discreta de tan sólo dos valores, en el que uno de los valores representa el negro (el cero), y el otro de los valores representa el blanco (el uno). En la transformación a este sistema de coordenadas, bien sea desde otro sistema de coordenadas, o bien sea desde un intervalo o unión de intervalos del espectro electromagnético, queda por decidir en qué coordenada específica se transforma. Esta decisión normalmente se basa en un umbral a prefijar sobre una característica calculable desde el conjunto de origen de la aplicación. Es decir si un elemento del conjunto original de la aplicación tiene una característica calculable con un valor mayor o igual que un cierto valor se le asigna en el sistema de coordenadas del espacio blanco-negro un valor de uno de los dos posibles del espacio de blanco-negro, y en caso de que el valor sea menor se le asigna el otro.

Como ejemplo el siguiente par de imágenes, donde el criterio decisional ha sido aplicar un umbral sobre la característica calculable de la imagen original llamada brillo (se verá más adelante). En este caso, se ha decidido que si para un píxel en la imagen original el brillo es mayor que un cierto valor, en la imagen con los valores de los píxeles en el espacio blanco-negro, su valor debe ser blanco, y en otro caso debe ser negro.



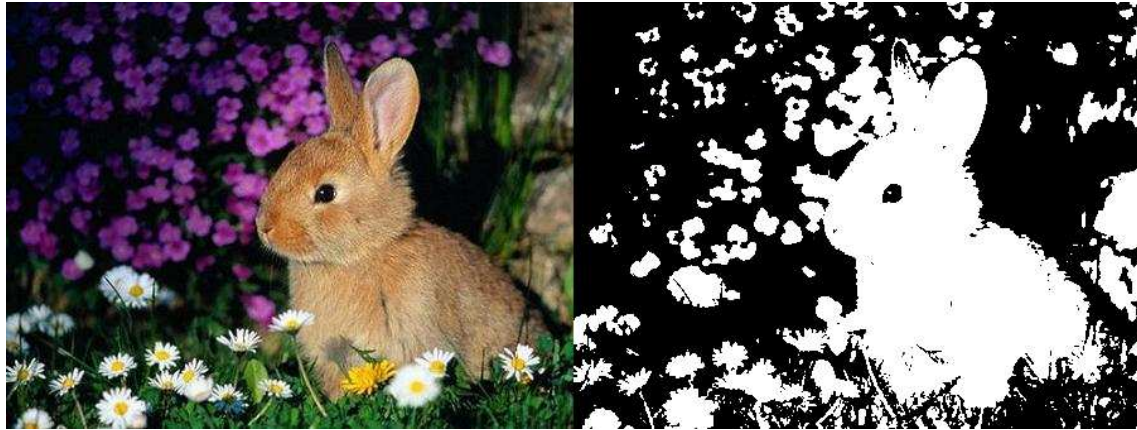


Ilustración 3-4 Imagen original e imagen en blanco-negro

### 3.2.12.2 Espacio escala de grises

El espacio escala de grises es un espacio compuesto por una única componente continua en un sistema de coordenadas cartesianas. Los valores de dicha componente están acotados en el intervalo  $[0,1]$ . Cuando dicha componente tiene un valor igual a cero se indica la ausencia de toda luz visible, el negro, y cuando dicha componente tiene un valor igual a uno se indica la presencia de toda luz visible, el blanco. Todos los valores intermedios expresan grises más o menos próximos al negro o al blanco en función de la proximidad de su valor al cero o al uno. A continuación se muestra un ejemplo visual de la transformación de una imagen a una imagen cuyos píxeles toman valores en una discretización del espacio escala de grises.

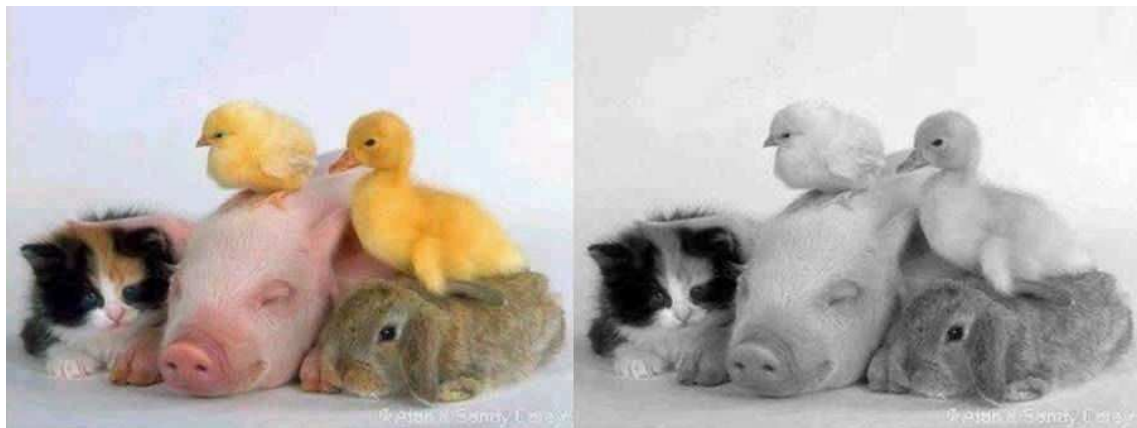


Ilustración 3-5 Imagen original e imagen en escala de grises

### 3.2.12.3 Espacio RGB

El espacio RGB es un espacio compuesto por tres componentes continuas en un sistema de coordenadas cartesianas. Los valores de las tres componentes están acotados en el intervalo  $[0,1]$ . Cuando el valor de una componente es cero indica la ausencia del color al que alude, y cuando el valor de esa componente es uno, indica la presencia del color al que alude. En una componente, todos los valores intermedios expresan puntos intermedios entre la presencia o ausencia del color al que alude la componente. La primera componente R, referencia al color rojo, la segunda componente G, alude al color verde, y la tercera componente B, es la del color azul.

En la siguiente imagen se puede observar el espacio RGB, donde cada punto se ha colorado del color al que dicho punto pertenece al espacio RGB hace referencia.

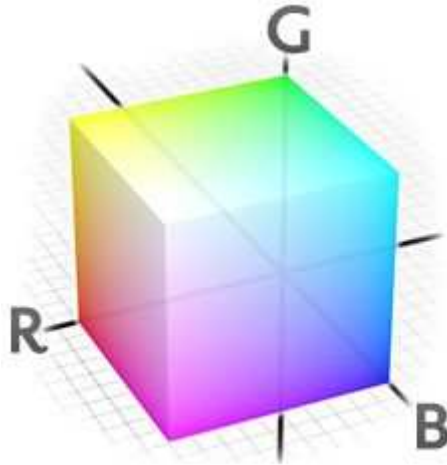


Ilustración 3-6 Muestra de colores del espacio RGB

### 3.2.12.4 Espacio HSL

El espacio HSL es un espacio compuesto por tres componentes continuas en un sistema de coordenadas cilíndricas. La primera componente H, es el matiz (Hue), es el ángulo del cilindro en las coordenadas cilíndricas, por tanto acotado en el intervalo  $[0,360)$ . El matiz se encarga de definir el tipo de color. La segunda componente S, es la saturación (Saturation), es una distancia, acotada en el intervalo  $[0,1]$ . La saturación indica cómo de puro es el color, y cuanto más baja sea la saturación más decolorado estará, tomando matices más grisáceos. La tercera componente L, es la luminosidad (lightness), es una distancia, acotada en el intervalo  $[0,1]$ .

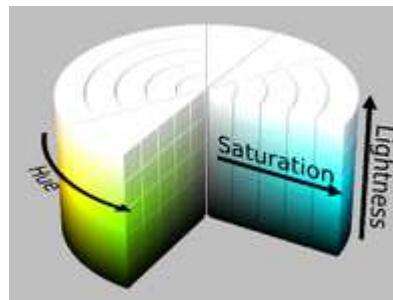
Haciendo una transformación desde el espacio RGB, se pueden entender las tres componentes del espacio HSL de la siguiente forma. Dado el cubo de la anterior ilustración, para entender el matiz, tener en mente un cubo como el de la ilustración anterior, e imaginar que éste se sitúa sobre un plano, al cual se le llama plano cromático, apoyado sobre el vértice que en el cubo de la ilustración está situado en el origen de coordenadas, y considerar que dicho punto de apoyo es el origen de coordenadas del plano. Además debe apoyarse de tal forma que la proyección del vértice opuesto del cubo, sobre el plano en el que se está apoyando, coincida en posición con el vértice sobre el que está apoyado. Una vez situado el cubo, se realiza una proyección de sus vértices sobre el plano, quedando seis de las proyecciones a una distancia mayor que cero del origen de coordenadas del plano. A la hora de definir un vector director en el plano cromático en coordenadas polares, se considera que la componente del ángulo tiene valor igual a  $0^\circ$  cuando el vector director apunta a la dirección en la que está situada la proyección del vértice de color rojo del cubo. Finalmente al traducir un punto del espacio RGB, se ubica en el cubo y se proyecta en el plano cromático, y el matiz será el ángulo que forme el vector director que va del origen de coordenadas del plano, a la proyección del punto. La luminosidad, se define como el promedio entre la componente de mayor valor y de menor valor del cubo RGB. La saturación, se define a partir de la distancia existente entre el origen de coordenadas del plano cromático y el punto resultante de la proyección del punto RGB que se desea traducir y que se ha ubicado en el cubo. Finalmente en la siguiente fórmula se expone cómo calcular la saturación (en la siguiente fórmula

S) para el espacio HSL dada la luminosidad (en la siguiente fórmula L), y dada la distancia mencionada del origen de coordenadas del plano cromático al punto de proyección del punto RGB (en la siguiente fórmula, dicha distancia será C).

$$S \begin{cases} 0 \Leftrightarrow C = 0 \\ C/2L \Leftrightarrow L \leq 1/2 \\ C/(2 - 2L) \Leftrightarrow L > 1/2 \end{cases}$$

**Fórmula 3-17 Descripción de la saturación en el modelo HSL**

Este último paso se realiza para ubicar la saturación en el intervalo [0,1]. En la siguiente imagen se puede observar el espacio HSL, donde cada punto se ha colorado del color al que dicho punto perteneciente al espacio HSL hace referencia.



**Ilustración 3-7 Muestra de colores del espacio HSL**

### 3.2.12.5 Espacio HSV

El espacio HSV es un espacio compuesto por tres componentes continuas en un sistema de coordenadas cilíndricas. La primera componente H, es la tonalidad (Hue), es el ángulo del cilindro en las coordenadas cilíndricas, por tanto acotado en el intervalo [0,360). La segunda componente S, es la saturación (Saturation), es una distancia, acotada en el intervalo [0,1]. La tercera componente V, se le llama valor (value), es una distancia, acotada en el intervalo [0,1]. Al espacio HSV también se le conoce como espacio HSB, en donde la última componente se llama brillo (brightness) en lugar de valor.

Haciendo una transformación desde el espacio RGB, se pueden entender las tres componentes del espacio HSV de la siguiente forma. La transformación del matiz, es igual que en el modelo HSL. El valor o brillo, tomará como valor el mayor valor de entre los tres del punto RGB. La saturación, se define a partir de la distancia existente entre el origen de coordenadas del plano cromático y el punto resultante de la proyección del punto RGB que se desea traducir y que se ha ubicado en el cubo. Finalmente en la siguiente fórmula se expone cómo calcular la saturación (en la siguiente fórmula S) para el espacio HSV dado el valor o brillo (en la siguiente fórmula V), y dada la distancia mencionada del origen de coordenadas del plano cromático al punto de proyección del punto RGB (en la siguiente fórmula, dicha distancia será C).

$$S \begin{cases} 0 \Leftrightarrow C = 0 \\ C/V \Leftrightarrow C \neq 0 \end{cases}$$

**Fórmula 3-18 Descripción de la saturación en el modelo HSV**

Éste último paso se realiza para ubicar la saturación en el intervalo [0,1]. En la siguiente imagen se puede observar el espacio HSV, donde cada punto se ha colorado del color al que dicho punto perteneciente al espacio HSV hace referencia.

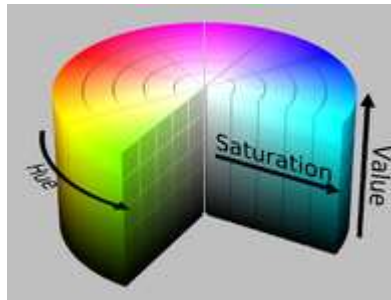


Ilustración 3-8 Muestra de colores del espacio HSV

### 3.2.12.6 Espacio HSI

El espacio HSI es un espacio compuesto por tres componentes continuas en un sistema de coordenadas cilíndricas. La primera componente H, es la tonalidad (Hue), es el ángulo del cilindro en las coordenadas cilíndricas, por tanto acotado en el intervalo [0,360). La segunda componente S, es la saturación (Saturation), es una distancia, acotada en el intervalo [0,1]. La tercera componente I, es la intensidad (intensity), es una distancia, acotada en el intervalo [0,1]. La intensidad refleja la cantidad de luz. La transformación del matiz, es igual que en el modelo HSI. La intensidad es igual a la altura a la que está situado el punto RGB en el cubo RGB inclinado, que es igual a un tercio de la suma de las tres componentes RGB. La saturación, se define a partir de la distancia existente entre el origen de coordenadas del plano cromático y el punto resultante de la proyección del punto RGB que se desea traducir y que se ha ubicado en el cubo. Por último, en la siguiente fórmula se expone como calcular la saturación (en la siguiente fórmula S) para el espacio HSI dada la intensidad (en la siguiente fórmula V), el valor de la componente RGB de menor valor (en la siguiente fórmula mínima sub RGB) y dada la distancia mencionada del origen de coordenadas del plano cromático al punto de proyección del punto RGB (en la siguiente fórmula, dicha distancia será C).

$$S \begin{cases} 0 \Leftrightarrow C = 0 \\ 1 - (\text{mínima}_{RGB}/I) \Leftrightarrow C \neq 0 \end{cases}$$

Fórmula 3-19 Descripción de la saturación en el modelo HSI

## 3.3 Definición de procesamiento de imagen digital

No hay un consenso general de dónde empieza y dónde acaba la ciencia del procesamiento de imagen. En (R. C. González, 2002) se recurre al siguiente paradigma para tratar de definir el procesamiento de imagen, además de clasificar los procesos pertenecientes al procesamiento de imagen en tres tipos. El primer tipo son los procesos de bajo nivel, en los que normalmente tanto la entrada como la salida al proceso son imágenes (por ejemplo, el preprocesamiento de la imagen para reducir ruido). El segundo tipo son los procesos de nivel medio, en los que generalmente la entrada es una imagen, pero en los que la salida es un conjunto de atributos extraído de la imagen (por ejemplo extracción de características de objetos segmentados). El

último tipo son los procesos de alto nivel, donde se toma como entrada atributos extraídos de la imagen, y se les da significado en las funciones cognitivas asociadas a la visión.

Un proceso sobre una imagen digital bidimensional, se puede entender como un conjunto de funciones que actúan sobre los píxeles de dicha imagen, haciendo una transformación.

$$g_1: {}^1P \rightarrow D'_1, \dots, g_n: {}^1P \rightarrow D'_n$$

$$\text{Proceso} = \{g_1, \dots, g_n\}$$

**Fórmula 3-20 Proceso sobre una imagen como conjunto de funciones**

Cuando el proceso en concreto, transforma la imagen de entrada en otra imagen, se puede describir dicho proceso y la imagen resultante como se muestra en las fórmulas siguientes.

$$g_x: {}^1P \rightarrow X', g_y: {}^1P \rightarrow Y', g_1: {}^1P \rightarrow D'_1, \dots, g_n: {}^1P \rightarrow D'_n$$

$$\text{Proceso} = \{g_x, g_y, g_1, \dots, g_n\}$$

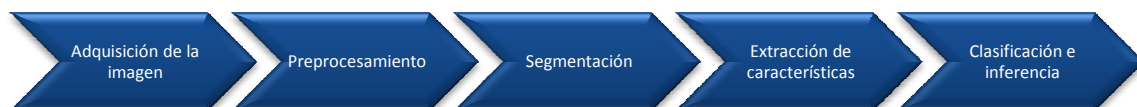
$$f'_1: X' \times Y' \rightarrow D'_1, \dots, f'_n: X' \times Y' \rightarrow D'_n$$

$$I' = \{f'_1, \dots, f'_n\}$$

**Fórmula 3-21 Transformación de una imagen en otra como conjunto de funciones**

## 3.4 Etapas del procesamiento de imagen digital

Un sistema de procesamiento de imagen, que tiene como entrada imágenes que el sistema mismo se encargaría de capturar y que tiene como salida un conocimiento extraído de las mismas, se puede dividir en un conjunto de etapas secuenciales, según se representa en la siguiente figura.



**Ilustración 3-9 Etapas secuenciales del procesamiento de imagen**

A continuación se explican secuencialmente en qué consisten estas etapas, además de que en la explicación de cada una de ellas se incorpora la descripción de técnicas conocidas pertenecientes a la etapa pertinente.

## 3.5 Adquisición de la imagen

En esta etapa, la entrada, se tiene que capturar, ofreciendo como salida una imagen digital, o varias. La entrada puede ser cualquier fuente perceptible, no hace falta que sea luz (por ejemplo

puede ser sonido). Sea cual sea la fuente perceptible, habrá uno o varios sensores (donde para la mayoría de tipos de percepción cada uno de los cuales incorporará un transductor). El cometido de los sensores es transformar la fuente perceptible entrante en una diferencia de potencial eléctrico. La cuantía de la diferencia de potencial eléctrico dependerá de la observación realizada por los sensores a la fuente perceptible. Tanto la diferencia de potencial eléctrico que indica la amplitud de la señal de la fuente perceptible, como la ubicación espacial de dicha captura en la imagen que se está representando con los resultados de la captura, pertenecen cada uno a un dominio continuo. Por tanto, ambos han de discretizarse para obtener una imagen digital. La digitalización de las coordenadas espaciales se llama muestreo, y la digitalización de la señal percibida se llama cuantificación. En este punto hay que tomar las decisiones de cómo se realizará el muestreo y cómo se realizará la cuantificación. Para el muestreo hay que definir una función de un dominio continuo en un dominio discreto y finito. También para la cuantificación hay que definir una función de un dominio continuo en un dominio discreto y finito. Lo convencional en el muestreo es, definir una unidad básica indivisible, y hacer que dados los valores de cada una de las componentes de los elementos del conjunto de destino ordenados, se separen en intervalos de igual tamaño a dicha unidad. Lo mismo se suele hacer con la cuantificación.

### 3.5.1 El ruido

Un problema que se suele presentar en la adquisición de imágenes es el del ruido. El ruido en un píxel puede definirse como la perturbación que conlleva el hecho de que el valor del píxel no corresponda con la realidad que pretende representar. Esta perturbación viene producida por el sensor receptor de la magnitud física medible a la que reacciona, o por el método físico del tratamiento de los datos posterior a la percepción mediante el sensor.

## 3.6 Preprocesamiento

La finalidad de esta etapa es la de mejorar la imagen resultante de la etapa de adquisición de imagen. La idea es hacer la imagen más apropiada para un proceso específico posterior, por lo que las técnicas que se emplean en el preprocesamiento no son universales para todos los problemas, pues en función del problema, algunas tendrán mejor respuesta que otras. Las técnicas empleadas en el preprocesamiento se pueden dividir en dos grupos. El primer grupo de técnicas, se identifican por operar en el dominio espacial. Es decir, dichas técnicas trabajan sobre la imagen digital entrante directamente, manipulando los valores de los píxeles directamente. El segundo grupo de técnicas, se reconocen por operar con una transformada de la imagen. Es decir, a la imagen se le aplica una función que la convierte en otra, con la cual se opera, para luego aplicar la función inversa de la función que la ha transformado en primera instancia, para retornar así la imagen al dominio espacial. Existen muchos tipos de transformadas de imagen, si bien algunas son más conocidas que otras. Aquí, aparte de las técnicas de preprocesamiento en el dominio espacial, se hablará de las técnicas que operan en el dominio de la frecuencia, donde estas últimas trabajan con una transformada de la imagen entrante a la etapa de preprocesamiento. En éstas técnicas, a la imagen digital entrante, se le aplica la transformada de Fourier, quedando una imagen en el dominio de la frecuencia. Una vez aplicada la transformada, la técnica perteneciente a este dominio opera sobre la imagen en el dominio de frecuencia, modificando la imagen transformada, para después finalmente aplicar la transformada inversa.

Antes se ha hablado de que algunas técnicas pueden funcionar mejor que otras para un proceso específico posterior perteneciente a otra etapa del procesamiento de imagen. Cabe plantearse la cuestión, que no es trivial, de cómo se define “mejor”. En función de la clase de problema, se puede tener una labor a la hora de definir “mejor” más fácil o más difícil. En el caso fácil, se puede probar el sistema de procesamiento de imágenes completo, con todas sus etapas, y sin sesgos, estimar cuantitativamente el acierto del sistema. En éste caso, las técnicas de preprocesamiento que ayuden al sistema a llegar a un nivel de acierto más alto, serán consideradas mejores. En el caso difícil, lo anterior no es posible, y la condición de “mejor” de una técnica de preprocesamiento queda supeditada al juicio directo de los humanos. Lo usual es que cuando se deba juzgar si una técnica de preprocesado es mejor o peor, no se recurre a cualquier juicio humano sobre ello, si no que se procura recurrir a la opinión de gente experta en el modelo de negocio con el que el sistema pretende lidiar.

### **3.6.1 Técnicas de preprocesamiento en el dominio espacial**

#### ***3.6.1.1 Transformaciones de niveles de gris***

En este tipo de técnicas, cada píxel de la imagen original se transforma en un píxel en la imagen de destino, preservándose la posición del píxel de la imagen original a la transformada. Además el valor de un píxel transformado depende en exclusiva del valor del píxel original con el que comparte posición. La idea es medir el nivel de gris de cada píxel de la imagen, a través del valor del píxel, transformar dicho nivel de gris, y con dicho nivel de gris transformado transformar el valor del píxel original. A continuación se muestra una gráfica con transformaciones de niveles de gris habituales (donde  $m$  y  $M$  representan el valor de nivel de gris mínimo y máximo respectivamente).

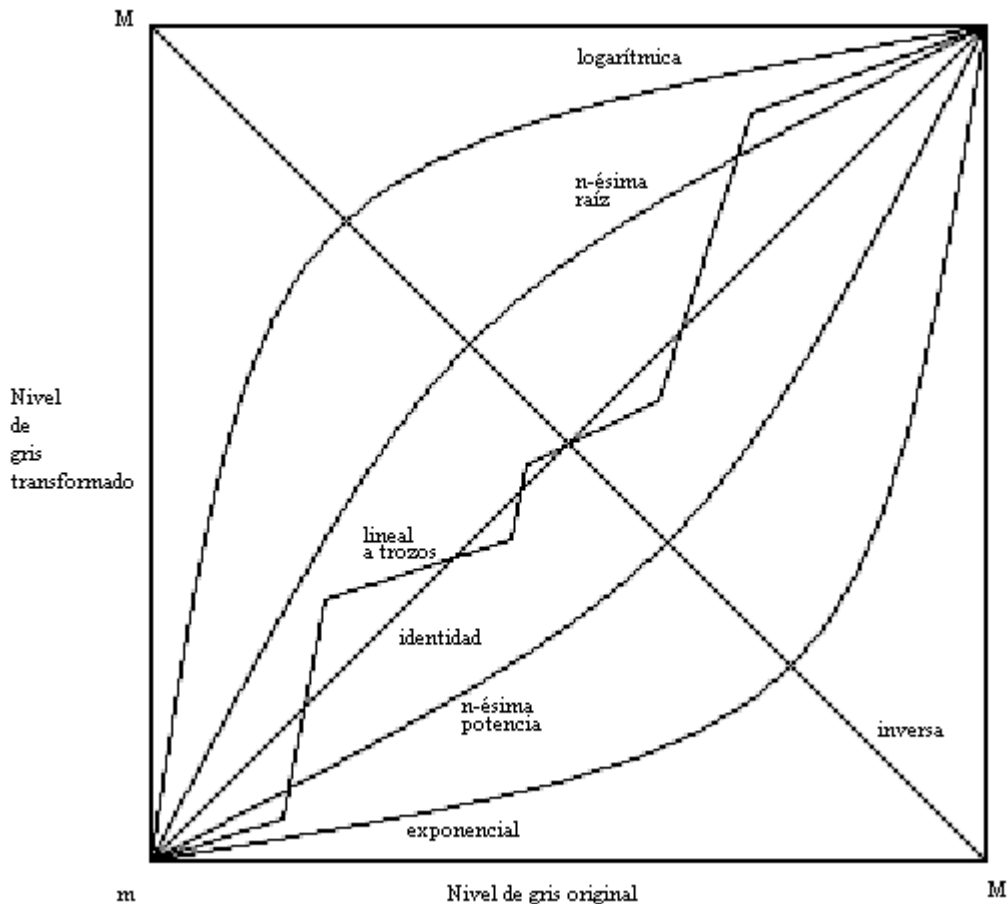


Ilustración 3-10 Transformaciones de niveles de gris habituales

Dependiendo de la aplicación posterior que se le pretenda dar a la imagen transformada, convendrá escoger una función de transformación de niveles de gris u otra. La idea de las transformaciones de niveles de gris, es la de ensalzar los píxeles deseados imprimiendo más luz en ellos y rebajar los píxeles no deseados restando luz en ellos.

### 3.6.1.2 Transformaciones del histograma

Un histograma de una imagen se puede describir como, un conjunto donde cada elemento es un par de números. Hay tantos elementos en el conjunto como elementos hay entre el nivel mínimo de gris y el nivel máximo de gris (ambos incluidos) posibles a ser encontrados en la imagen. Además el primer término de cada elemento del conjunto, alude a un nivel de gris específico de entre los posibles a ser encontrados en la imagen, y no se repite la alusión a un mismo nivel de gris en distintos elementos. El segundo término de los elementos alude, a la proporción de píxeles de la imagen que presentan un nivel de gris igual al que alude el primer término de los elementos. Puede verse como un histograma, en el cual se suele representar en el eje de las abscisas los niveles de gris y en el eje de las ordenadas la proporción de píxeles de la imagen. Cabe indicar que la proporción de píxeles en la imagen en los segundos términos de los elementos del conjunto del histograma, puede venir en términos absolutos (hay un número  $n$  determinado de píxeles que



tienen el nivel de gris  $m$ ) o en términos relativos (hay  $n/N$  píxeles con el nivel de gris específico, donde  $N$  es el número total de píxeles de la imagen).

Entre las técnicas de transformaciones del histograma, aquí se destaca la técnica conocida como ecualización del histograma. En esta técnica se trata de emplear la transformación del histograma que deje como resultado un histograma transformado que cumple con la propiedad de ser uniforme (aunque en el caso discreto, puede ocurrir que no se consiga ecualizar el histograma, si bien se aproxima bastante bien, y el histograma transformado resultante de la ecualización de un histograma en el caso discreto, se extiende por los niveles de gris posibles). Al ecualizar una imagen, se incrementa la calidad visual de la misma, y pues se aumenta el contraste existente y la imagen lucirá un amplio espectro de valores de luminosidad. Esta técnica suele resultar de utilidad cuando tanto lo que se pretende resaltar (generalmente objetos) como aquello que se pretende descartar (generalmente fondo) son ambas cosas muy oscuras o muy claras. Una de las ventajas que ofrece es que la técnica en sí se puede aplicar sobre una imagen sin necesidad de información adicional. Otra de ellas es que dada la función de ecualización, se puede encontrar la función inversa para recuperar la imagen original.

En la siguiente imagen se muestran un par de ejemplos de ecualización del histograma. La imagen siguiente es fruto de la composición de cuatro imágenes, donde el par de imágenes de la izquierda resultan ser las originales y el par de imágenes de la derecha son las imágenes precedentes ecualizadas.



Ilustración 3-11 Muestras de la ecualización del histograma

### **3.6.1.3 Promediado de imágenes**

Cuando una imagen presenta ruido con ciertas propiedades, y la captura (la adquisición de la imagen) de dicha imagen se puede repetir, se puede resolver de la forma que se indica en este apartado. Las propiedades que debe presentar el ruido de la imagen para que pueda solventarse con el método aquí descrito, es que el ruido de la imagen debe ser gaussiano, además debe tener media cero (para cada componente la media del ruido es cero) y que el ruido no esté correlacionado con las posiciones de los píxeles de la imagen.

La técnica consiste en repetir la captura de la imagen un número dado de veces, y después usar todas esas imágenes para operar con ellas y generar como producto una única imagen resultado de este proceso, por el cual, se trata de mejorar la calidad de la imagen reduciendo el ruido presente en la imagen. En la imagen final, el valor de cada píxel existente en ella es resultado del promedio estadístico de los valores de los píxeles del conjunto de imágenes que se tiene, donde dichos píxeles deben cumplir con la condición de que coinciden en posición con el píxel de la imagen final.

Conforme el número de imágenes se incrementa, la desviación típica se reducirá y uniendo esto al hecho de que la media del ruido en las imágenes es cero, sucederá que conforme el número de imágenes se incrementa, el ruido en la imagen resultante se atenuará.

### **3.6.1.4 Filtrado espacial**

Bajo filtrado espacial se agrupan un conjunto de técnicas que transforman una imagen operando píxel por píxel con las siguientes particularidades. El valor de cada píxel transformado, depende de los coeficientes asociados al filtro y de la máscara definida para el píxel. El valor final de cada píxel transformado, vendrá dado por una relación predefinida de aquellas variables de las que depende. La máscara es un subconjunto de píxeles de la imagen original, en la cual la magnitud de sus dimensiones es igual al del vecindario de los píxeles. Es decir, por ejemplo, si se emplea un criterio de 8-vecindad, la máscara será una matriz de 3x3, resultando ser un conjunto de 9 píxeles donde el píxel central es igual al píxel original que se desea transformar, y cada uno de los píxeles vecinos en la matriz de la máscara se corresponde con uno de los vecinos del píxel original. Los coeficientes asociados al filtro, suelen presentarse repartidos dentro de una matriz del mismo tamaño que la máscara, pues es usual que cada coeficiente esté asociado a un píxel de la máscara. Dentro del filtrado espacial, se conoce como filtrado lineal espacial a los filtros que hacen que el píxel transformado sea un valor que depende de las combinaciones lineales de los valores de la máscara.

En el filtrado espacial caben los filtros de suavizado espaciales. Estos filtros se usan para eliminar ruido de la imagen y para desenfocarla hasta cierto punto, tratando de lograr la eliminación de pequeños detalles en objetos grandes y de pequeñas discontinuidades en líneas. Por otra parte, el desenfocado provoca la atenuación de las discontinuidades de color en la imagen. Este hecho supone una desventaja ya que los bordes aportan información a la hora de detectar objetos.

A continuación se explica como ejemplo de las técnicas de filtrado de suavizado espacial, un filtrado de suavizado espacial lineal, en donde la transformación se basa en los niveles de gris. Se emplea un criterio de 8-vecindad y se define una máscara de 3x3 centrada en el píxel a

transformar. Se define un conjunto de 9 coeficientes, donde cada coeficiente está asociado a un píxel de la máscara distinto. Entonces para cada píxel de la máscara, se calcula el valor de la característica del nivel de gris. Una vez se tiene el valor de dicha característica para cada píxel de la máscara, se realiza una combinación lineal de dichos valores, multiplicando cada valor de nivel de gris por el coeficiente asociado al píxel del cual se ha extraído el respectivo valor de nivel de gris. Después se suman todos estos productos, obteniendo el valor de nivel de gris transformado para el píxel a transformar. Finalmente, operando con el nivel de gris transformado y con el valor del píxel a transformar, se obtiene el valor del píxel transformado. En la siguiente fórmula, cada subíndice representa una dimensión en la máscara,  $g$  representa un valor de nivel de gris y  $c$  representa un coeficiente.

$$g'_{22} = \frac{\sum_{i=1}^3 \sum_{j=1}^3 g_{ij} \cdot c_{ij}}{\sum_{i=1}^3 \sum_{j=1}^3 c_{ij}}$$

**Fórmula 3-22 Ejemplo: Nivel de gris transformado**

Es importante comentar que el hecho de que en la ecuación, en el término de la derecha figure en el denominador del cociente la suma de todos los coeficientes. Ese término figura para que el valor de nivel de gris transformado sea una interpolación ponderada de los valores de nivel de gris de los píxeles de la máscara, en lugar de ser una adición de los mismos.

Dentro del filtrado espacial, se encuentran los filtros de afilado. Estos filtros es la de resaltan los detalles y las discontinuidades en la imagen, tales como los bordes de objetos. Como ejemplo se expone un subconjunto de ellos, que se basan en las derivadas de la imagen en sus dos dimensiones espaciales. Concretando el ejemplo, se define la derivada de primer orden mediante la siguiente fórmula.

$$df = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}$$

$$df = (f(x + 1, y) - f(x, y)) + (f(x, y + 1) - f(x, y))$$

**Fórmula 3-23 Derivada de primer orden**

La derivada de segundo orden se define con la siguiente fórmula.

$$d^2f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$d^2f = (f(x + 1, y) + f(x - 1, y) - 2f(x, y)) + (f(x, y + 1) + f(x, y - 1) - 2f(x, y))$$

$$d^2f = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)$$

**Fórmula 3-24 Derivada de segundo orden**

Las derivadas anteriores pueden verse como coeficientes de una máscara de filtrado. Finalmente se define el valor del píxel transformado como el valor del original, más la combinación lineal de los valores de los píxeles de la máscara. La combinación lineal específica se define como la suma

de productos, donde cada producto es el producto entre el valor de un píxel y su coeficiente asociado. Como la suma de coeficientes en ambos casos es cero, y la combinación lineal se le suma al valor del píxel original, no se dan ni ganancias ni pérdidas indeseadas.

### 3.6.2 Técnicas de preprocesamiento en el dominio de frecuencia

En este punto resulta adecuado indicar que se dispone en el apartado de anexos un anexo que trata la transformada de Fourier.

#### 3.6.2.1 Filtros de suavizado

En una imagen digital, las discontinuidades de color, los bordes y el ruido contribuyen bastante en las frecuencias altas de la transformada de dicha imagen al dominio de la frecuencia. Por este hecho, para suavizar o desenfocar una imagen digital, se le puede aplicar la transformada de Fourier, después aplicarle un filtro de suavizado (también llamado filtro paso-baja) de imágenes en el dominio de la frecuencia que de alguna forma atenuará las altas frecuencias de la imagen y finalmente aplicar la transformada inversa para obtener la imagen suavizada en el dominio espacial. Una forma convencional de llevar a cabo este filtrado, es la de multiplicar para cada píxel de la imagen transformada, el valor de él, por un escalar real acotado en el rango  $[0,1]$ . Esta operación se muestra en la siguiente ecuación, donde  $F$  es la imagen transformada al dominio de la frecuencia,  $ff_s$  es la función que aplica el filtro de suavizado, donde todos los valores resultado de emplear  $ff_s$  sobre elementos del dominio de definición de la función, están, como se ha mencionado, acotados en el intervalo  $[0,1]$ . Finalmente  $F'$  es la imagen resultado de filtrar la imagen transformada al dominio de la frecuencia.

$$F'(u, v) = ff_s(u, v) \cdot F(u, v)$$

#### Fórmula 3-25 Ecuación convencional de los filtros de suavizado en el dominio de la frecuencia

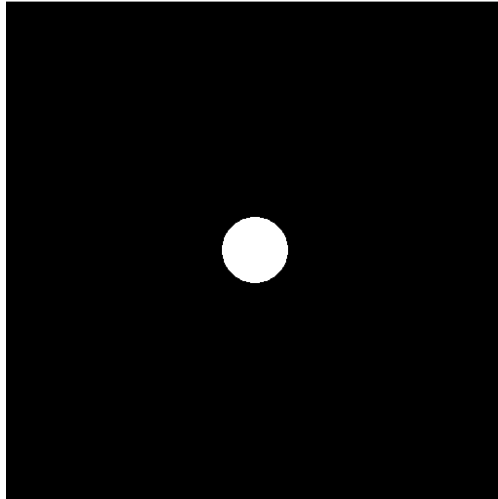
Así pues, lo convencional es que la función que aplica el filtro de suavizado, ofrezca como resultado de dicha función valores más próximos al cero cuanto más lejos se está del centro de la imagen transformada (que es donde se acumulan las altas frecuencias). Existen gran variedad de filtros de suavizado en el dominio de la frecuencia, y aquí se mencionan los filtros de tipo ideal, y los filtros de tipo gaussiano.

El filtro ideal es una función que dado un píxel de la imagen transformada, mide su distancia al centro de la imagen, siendo el resultado de la función del filtro cero cuando dicha distancia es mayor a cierto valor dado, y siendo uno cuando es menor o igual a dicho valor umbral. Si bien se pueden emplear distintas funciones de distancia en la función del filtro de suavizado, lo convencional es emplear la distancia euclídea. En la siguiente fórmula se muestra la función del filtro de suavizado de tipo ideal, donde  $d$  es una función de distancia,  $(u_c, v_c)$  es la posición central de la imagen transformada y  $d_u$  es el umbral de distancia.

$$ff_s(u, v) \begin{cases} 1 & \Leftrightarrow d((u, v), (u_c, v_c)) \leq d_u \\ 0 & \Leftrightarrow d((u, v), (u_c, v_c)) > d_u \end{cases}$$

#### Fórmula 3-26 Función de filtro de suavizado de tipo ideal

Es usual presentar los filtros visualmente como una imagen digital en escala de grises. Para un píxel dado del filtro, cuanto mayor es su valor, más próximo es su color al blanco, mientras que cuanto menor es su valor, más próximo es al negro. Así pues, se presenta en la siguiente ilustración un ejemplo de representación visual de un filtro de suavizado de tipo ideal.



**Ilustración 3-12 Filtro ideal de suavizado**

Tras la aplicación de un filtro de suavizado como el de la ilustración anterior sucede que los valores de los píxeles de la imagen transformada a filtrar que caen dentro de la región blanca de la imagen anterior, se preservan íntegros, mientras que todos los demás tomarán valor igual a cero tras la aplicación del filtro. En la siguiente ilustración se muestra el resultado de la aplicación de un filtro de suavizado. En la ilustración, la primera imagen es la imagen digital a procesar, mientras que la segunda imagen es el resultado de transformar la primera mediante la transformada de Fourier, para después aplicarle un filtro ideal y finalmente retornarla al dominio espacial con la transformada inversa de Fourier.



**Ilustración 3-13** Ejemplo de uso de un filtro de suavizado ideal

El filtro gaussiano para una imagen bidimensional, se basa en una distribución gaussiana bidimensional centrada en el centro de la imagen, en donde la distribución gaussiana toma como valor máximo el uno. En la siguiente fórmula se muestra la función del filtro de suavizado de tipo gaussiano, donde  $d$  es una función de distancia,  $(u_c, v_c)$  es la posición central de la imagen transformada y  $\sigma$  es la desviación típica de la gaussiana.

$$ffs(u, v) = e^{-d^2((u,v),(u_c,v_c))/2\sigma^2}$$

**Fórmula 3-27** Función de filtro de suavizado de tipo gaussiano

### 3.6.2.2 Filtros de afilado

Se ha mencionado previamente que en los bordes, las discontinuidades de color, el ruido y los detalles, realizan un aporte importante en las altas frecuencias de la imagen. Como el cometido de un filtro de afilado es resaltar detalles, bordes y discontinuidades de color, en esta ocasión se tenderá a atenuar las bajas frecuencias y a preservar las altas frecuencias. Se puede describir un filtro de afilado igual que un filtro de suavizado según la ecuación siguiente, en donde  $ffa$  es la función del filtro de afilado.

$$F'(u, v) = ffa(u, v) \cdot F(u, v)$$

**Fórmula 3-28 Ecuación convencional de los filtros de afilado en el dominio de la frecuencia**

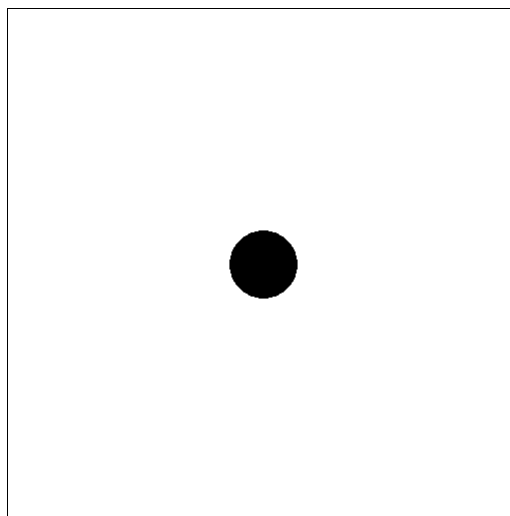
Así pues, lo convencional es que la función que aplica el filtro de afilado, ofrezca como resultado de dicha función valores más próximos al uno cuanto más lejos se está del centro de la imagen transformada (que es donde se acumulan las altas frecuencias). Cabe mencionar, que una vez conocidos los filtros de suavizado, es fácil definir el filtro de afilado asociado a un filtro de suavizado dado, donde dicho filtro de afilado tomará como valor en cada píxel de la imagen del mismo el valor resultante de restarle al número uno, el valor que ofrece el filtro de suavizado al cual está asociado, para dicho píxel. Existen gran variedad de filtros de afilado en el dominio de la frecuencia, y aquí se mencionan los filtros de tipo ideal, y los filtros de tipo gaussiano.

El filtro ideal de afilado, indica que todos los valores de píxeles cuyas posiciones son lo suficientemente próximas al centro de la imagen según una función de distancia, se modifican para que tengan valor igual a cero, mientras que en otro caso preservarán su valor. Si bien se pueden emplear distintas funciones de distancia en la función del filtro de afilado, lo convencional es emplear la distancia euclídea. En la siguiente fórmula se muestra la función del filtro de afilado de tipo ideal, donde  $d$  es una función de distancia,  $(u_c, v_c)$  es la posición central de la imagen transformada y  $d_u$  es el umbral de distancia.

$$ffs(u, v) \begin{cases} 0 & \Leftrightarrow d((u, v), (u_c, v_c)) \leq d_u \\ 1 & \Leftrightarrow d((u, v), (u_c, v_c)) > d_u \end{cases}$$

**Fórmula 3-29 Función de filtro de afilado de tipo ideal**

Es usual presentar los filtros visualmente como una imagen digital en escala de grises. Para un píxel dado del filtro, cuanto mayor es su valor, más próximo es su color al blanco, mientras que cuanto menor es su valor, más próximo es al negro. A continuación, se realiza una presentación visual en la siguiente ilustración de un ejemplo de filtro de afilado de tipo ideal.



**Ilustración 3-14 Filtro ideal de afilado**

Tras la aplicación de un filtro de afilado como el de la ilustración anterior sucede que los valores de los píxeles de la imagen transformada a filtrar que caen dentro de la región blanca de la imagen anterior, se preservan integros, mientras que todos los demás tomarán valor igual a cero tras la aplicación del filtro. En la siguiente ilustración se muestra el resultado de la aplicación de un filtro de afilado. En la ilustración, la primera imagen es la imagen digital a procesar, la segunda imagen es la anterior transformada al dominio de la frecuencia, la tercera imagen es el filtro aplicado, la cuarta imagen es el resultado de aplicar a la segunda imagen el filtro de la tercera imagen, mientras que la última imagen es el resultado de retornar la cuarta imagen del dominio de la frecuencia al dominio espacial (aunque para la presentación visual en la siguiente imagen, se ha realizado un binarizado de la misma. Los píxeles que en un espacio RGB tengan un valor superior a treinta en todas sus componentes, quedan blancos, y los demás, quedan negros).



Ilustración 3-15 Ejemplo de uso de un filtro ideal de afilado

El filtro gaussiano para una imagen bidimensional, se basa en la resta hecha a un filtro que fronterapreserva el valor de cada píxel integro, de una distribución gaussiana bidimensional centrada en el centro de la imagen, en donde la distribución gaussiana toma como valor máximo uno. En la siguiente fórmula se muestra la función del filtro de afilado de tipo gaussiano, donde  $d$  es una función de distancia,  $(u_c, v_c)$  es la posición central de la imagen transformada y  $\sigma$  es la desviación típica de la gaussiana.

$$ffa(u, v) = 1 - e^{-d^2((u,v),(u_c,v_c))/2\sigma^2}$$

Fórmula 3-30 Función de filtro de afilado de tipo gaussiano

### 3.7 Segmentación

En esta etapa, la entrada, resulta ser una imagen digital, mientras que el resultado de esta etapa, suele presentarse como un conjunto de imágenes digitales. La finalidad de esta etapa es la de dividir la imagen entrante en regiones u objetos que figuran en la misma. Dependiendo del problema específico planteado, los objetos de interés presentarán unas características u otras, siendo la etapa de segmentación adaptada al problema.

La gran mayoría de algoritmos de segmentación se agrupan bajo alguna de las dos tipos de técnicas más extendidas. El primer tipo de técnicas, se centran en segmentar la imagen basándose en los bordes y discontinuidades presentes en la misma, como por ejemplo las discontinuidades



de la intensidad de la luz a lo largo de la imagen. La idea es emplear estos bordes y discontinuidades como fronteras que separen regiones de la imagen, donde cada región finalmente será un objeto de la imagen. El segundo tipo de técnicas, se centra en dividir la imagen en un conjunto de regiones, de forma que las regiones de dicho conjunto presenten un conjunto de propiedades especificado previamente a la segmentación. A continuación se habla de la segmentación basada en detección de bordes y posteriormente se habla de la segmentación por identificación de regiones.

### 3.7.1 Detección de discontinuidades

Existen una gran cantidad de técnicas de detección de discontinuidades en una imagen digital y aquí se van a comentar algunas de las más extendidas.

Trabajando en escala de grises, se pueden detectar píxeles aislados que son significativamente diferentes de sus vecinos, mediante el empleo de la siguiente técnica. Para examinar si un píxel en una imagen es significativamente distinto de sus vecinos, se puede definir una máscara que incorpore al píxel a examinar y a sus vecinos. A continuación se presenta un ejemplo de este procedimiento. En los coeficientes asociados a cada píxel, se definen los de los vecinos con valor menos uno y el del píxel a examinar con un valor igual al número de vecinos que tiene. Entonces se calcula la suma de productos, donde cada producto es la multiplicación de un píxel de la máscara por su coeficiente. Entonces si el valor absoluto del resultado de la suma de productos es mayor que un cierto umbral, el píxel se considera significativamente diferente de sus vecinos.

Para detectar líneas, trabajando en escala de grises, se puede emplear una técnica muy similar a la anterior. Una línea en una imagen digital, puede verse como un conjunto de píxeles similares alineados, que son significativamente diferentes de los píxeles adyacentes que no forman parte de la línea en la que están alineados. Se examinará, para cada píxel, si éste pertenece a una línea o no, mediante el empleo de una máscara igual que en la técnica anterior. De nuevo dicha máscara cubre al píxel a evaluar y a sus vecinos. La primera diferencia es en los coeficientes asociados a cada término de la máscara. En esta ocasión los coeficientes positivos posicionados sobre los píxeles a los cuales están asociados respectivamente, deberán describir una línea física, o al menos aproximarla dentro mediante la ayuda de la geometría discreta. Los coeficientes negativos, deberán estar fuera de esa línea. Como en la técnica anterior, la suma de coeficientes deberá ser igual a cero. Lo que se está midiendo es la diferencia ponderada entre los píxeles que recaen sobre la línea hipotética descrita por los coeficientes positivos plasmados sobre sus píxeles asociados, y los píxeles que se encuentran fuera de dicha línea. Así pues, el resultado de la suma de productos, cuanto más alto es, más plausible vuelve la hipótesis de que dicho píxel está realmente en la imagen dentro de una línea. Cabe tener en cuenta, que el resultado ofrecido para líneas paralelas presentes en la imagen de distintos grosores, fácilmente variará al aplicar esta técnica.

Se han explicado la detección de puntos y líneas discrepantes con su entorno y a continuación se explica la detección de bordes. Para las técnicas aquí planteadas, se rescatan los conceptos de primera derivada y segunda derivada de la imagen digital, expuestos en el apartado de filtrado espacial, en la parte en la cual se comentan los filtros de afilado. De nuevo en las técnicas aquí presentadas se trabaja en escala de grises. Los bordes pueden verse como discontinuidades en la

imagen. Al aplicar la primera derivada a la imagen y después generar una gráfica para una columna o fila dadas, se puede observar que las planicies de la gráfica se corresponden con valores iguales a lo largo de la columna o fila, mientras que en las variaciones de valor entre píxeles se pueden observar pendientes ascendentes o descendentes en la gráfica. Al aplicar la segunda derivada a la imagen, y después generar la gráfica para una columna o fila dadas, se puede observar que en los lugares en los que para la gráfica generada de la primera derivada para la columna o fila dada se pasa de tener una planicie a una pendiente y viceversa, la segunda derivada toma valores distintos de cero. Por tanto tomando el primer lugar en el que se pase de tener una planicie a una pendiente o viceversa y tomando el siguiente lugar que presente ésta propiedad, se puede localizar el centro del borde mediante la interpolación de los valores de sus posiciones. Cabe mencionar lo sensibles que son la primera derivada y la segunda derivada a la presencia de ruido en la imagen.

Una forma de decidir si un píxel forma parte de un borde, consiste en calcular su derivada bidimensional de primer orden, para después ver si su valor es superior a un cierto umbral, que caso de verificarse la superación de dicho umbral, dicho punto será considerado como un píxel de borde. Finalmente un borde quedará compuesto por un conjunto de píxeles de borde que son conexos entre sí dada una función de conexión. Igual que en el filtrado espacial, sobre un píxel dado se aplica una máscara que lo contiene a él y a sus vecinos.

Otra forma de decidir si un píxel forma parte de un borde, consiste en el cálculo del gradiente basado en máscaras. La ecuación del gradiente se muestra a continuación.

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{pmatrix}$$

### Fórmula 3-31 Gradiente en el caso bidimensional

Ejemplos de este método son el empleo de las máscaras de Prewitt y de Sobel, ambas de uso extendido, para el cálculo del gradiente. Los coeficientes de las máscaras de Prewitt y de Sobel se presentan en la siguiente ilustración.

-1	0	1
-1	0	1
-1	0	1

**PREWITT**

-1	0	1
-2	0	2
-1	0	1

**SOBEL**

### Ilustración 3-16 Coeficientes de las máscaras de Prewitt y Sobel

Mencionar que las máscaras anteriores son para el cálculo de una de las componentes del gradiente (horizontal), mientras que para la otra componente se empleará la traspuesta de la matriz de coeficientes correspondiente. Tras el cálculo del gradiente, se puede calcular el módulo del mismo, según la siguiente ecuación.

$$|\nabla f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

**Fórmula 3-32 Módulo del gradiente en el caso bidimensional**

Aunque éste cálculo suele ser caro computacionalmente y el valor resultante del módulo se aproxima mediante la suma de las dos componentes del gradiente. Finalmente, dado el valor del módulo (sea aproximado o no), se puede testear frente a un valor umbral, que caso de superarlo indicará que el píxel para el cual se ha calculado el gradiente corresponde a un píxel de borde.

### 3.7.2 Segmentación por identificación de regiones

Existen muchas técnicas de segmentación que tratan de identificar regiones directamente y aquí se realiza un esbozo de la problemática y se comentan algunas técnicas someramente.

Las regiones a identificar en una imagen digital, pueden verse como conjuntos que todos ellos resultan ser una partición de la imagen digital. Además, en cada conjunto para que éste pueda considerarse una región, debe suceder que todos los píxeles están conectados según una función de conectividad dada. Una forma de definir la plausibilidad de las regiones aportadas por una partición de la imagen digital, consiste en definir una expresión lógica evaluable sobre una región o una unión de regiones, que se verifica exclusivamente cuando se le pasa una única región.

Una forma de abordar el problema de identificación de regiones, consiste en identificar un conjunto de píxeles que actuarán de semilla, para posteriormente expandir las regiones que comienzan en cada píxel semilla. La decisión de qué píxeles actuarán de semilla es dependiente del negocio que se tenga entre manos, para el cual se está empleando el procesamiento de imagen. Así pues se declara una región por cada píxel semilla, que contendrá únicamente al píxel semilla respectivo. Tras esto, se van expandiendo las regiones. Para dicha expansión, los píxeles que no son semilla se van incorporando a las regiones existentes hasta que todo píxel pertenece a una y sólo una región. La incorporación de un píxel a una región u otra depende de un criterio, el cual dependerá del problema planteado, si bien es convencional, incorporar en dicho criterio, entre otras restricciones, que un píxel candidato a ser incorporado en una región, debe ser al menos, vecino de algún píxel de dicha región, según un criterio de vecindad dado.

Otra forma de abordar el problema, parte de concebir una única región en la imagen digital que contiene todos sus píxeles. Se debe definir un criterio que permita discernir si una región es consistente o no, para lo cual se define una expresión lógica evaluable sobre una región dada, que se verifica cuando la región entrante es consistente. Es convencional que en dicho criterio, tenga lugar entre otras restricciones, que todos los píxeles de la región entrante deban estar conectados para que la región pueda ser considerada consistente. Tras esto se comienza un proceso iterativo de partición de las regiones reconocidas en la imagen digital hasta el momento. En este proceso iterativo se evalúa el criterio de consistencia de una región para cada región reconocida en la imagen hasta el momento. Caso de no verificarse para alguna región, dicha región se particiona en cuatro regiones (para el caso bidimensional), seccionándola por la mitad en cada eje que describe su posición, para finalmente volver a comenzar un nuevo ciclo en el que se evaluará la consistencia de todas las regiones. Después del proceso iterativo mencionado, comienza otro

proceso iterativo de unificación de las regiones reconocidas en la imagen digital hasta el momento. En este proceso iterativo, se consideran todas las posibles unificaciones de pares de regiones que son adyacentes. Una unificación de un par de regiones se lleva a cabo si la unión de ambas resulta ser una región consistente. Si en un ciclo del proceso iterativo no se puede llevar a cabo ninguna unión de regiones, el proceso iterativo concluye.

### **3.8 Extracción de características**

En esta etapa, la entrada, convencionalmente resulta ser un conjunto de imágenes digitales, resultado dicho conjunto de una segmentación previa de una imagen digital, donde en cada imagen del mencionado conjunto figura una única región (en cada imagen una región distinta) de aquellas que fueron identificadas en la etapa de segmentación. La finalidad de esta etapa es la de describir las regiones de las imágenes entrantes mediante características. Dichas características pueden ser de distinta naturaleza, como por ejemplo pueden ser discretas o continuas, finitas o infinitas, acotadas o no acotadas y ordenables o no ordenables. Las características extraíbles de una región, sea pueden agrupar en dos grandes conjuntos según la naturaleza de la característica que describen. Por un lado, están las características que describen la forma de la región y por otro lado, están las características que describen la distribución del color en la región. Suele ser habitual que para las etapas posteriores del procesamiento de imagen, sea deseable que las características extraídas para una región sean independientes del tamaño de la misma, independientes de la orientación de la misma e independientes de la posición de la misma, por eso deben definirse características extraíbles de la región, que traten en la medida de lo posible de tener esto en cuenta.

Antes de exponer algunas características extraíbles tanto de la forma de una región como de la distribución del color de una región, se habla de técnicas que permiten condensar la información de una región, favoreciendo los cálculos de procedimientos posteriores.

#### **3.8.1 Condensación de regiones**

Los códigos de cadenas, consisten en representar la forma de la región entrante mediante un código que describe la orientación promedio de la frontera a intervalos. Convencionalmente un código de cadenas es una secuencia finita de números enteros y además, suelen presentarse en base cuatro o en base ocho, convencionalmente dependiendo este último punto del criterio de conectividad que se emplee para la descripción del código de cadenas. Un elemento de un código de cadenas dado para una región, sabido el muestreo realizado a la región, indica la orientación promedio de la frontera en un intervalo acotado de la frontera. Es habitual que en lugar de mostrar con cada elemento del código la orientación del segmento de frontera que va de un píxel de la frontera al siguiente píxel de la frontera, cada elemento del código muestre la orientación promedio de un segmento de frontera más grande. Es decir, que se realiza un muestreo. Esto se realiza tanto como para atenuar el ruido a pequeña escala que pueda alterar la forma de la región, como para evitar que la descripción de la región sea demasiado grande. Cabe tener en cuenta que el código de cadenas para una misma región, dependerá de en qué punto de la frontera se comience a describir la región. Esto es una propiedad no deseable. Existen diferentes formas para resolver este problema: Una de ellas consiste en “girar la cadena” en cualquiera de los dos sentidos hasta que se encuentra la cadena de menor valor, ya que la cadena realmente es

circular al describir una región cuya frontera es una curva cerrada. Para tener códigos de cadenas del mismo tamaño para objetos de diferente tamaño, se pueden aplicar muestreos de distinta granularidad.

Otra forma de condensar la forma de una región, es mediante una aproximación poligonal de la misma. Una forma de llevar a cabo esta aproximación, en imágenes bidimensionales, consiste en poner la región sobre una malla y marcar todas aquellas celdas que contienen algún píxel de la frontera de la región. Entonces se encuentra el polígono de menor perímetro que verifica que todos los puntos que lo describen están contenidos dentro de alguna celda marcada. Otro método de aproximación poligonal, consiste en lo siguiente. Lo primero, se traza una recta que une los dos píxeles de la frontera de la región más distantes entre sí. Una vez hecho esto, hasta que se satisfaga cierta condición, se irán añadiendo vértices que junto a la línea describirán un polígono. Los vértices se añaden en lugares que forman parte de la frontera de la región. Cada vez que se añade un nuevo vértice, se rompe la recta que une los dos vértices preexistentes más próximos a él, y se generan dos rectas, ambas del vértice nuevo, pero cada una a un vértice preexistente vecino distinto. La única excepción es cuando se añade un vértice nuevo cuando sólo hay dos preexistentes. En dicho caso, la recta que conecta los preexistentes se preserva. Como ejemplo de la condición a satisfacer para detener la adición de vértices que añaden más lados al polígono que aproxima la región, se podría definir un umbral y un test a realizar sobre cada píxel de la frontera, que caso de superarse para todos, implicaría la cesión de adiciones de vértices al polígono de aproximación. Como ejemplo de test, en un píxel de la frontera dado y dado cierto umbral, se puede decir que dicho píxel no pasa el test si la distancia a todo punto del perímetro del polígono de aproximación es mayor que dicho umbral.

Otro método de condensar la forma de una región, consiste en describirla en coordenadas polares tomando como origen de coordenadas el centro de la región. Así pues según se incrementa el ángulo, la distancia del centro a la frontera irá variando. Así pues se hace un muestreo a intervalos regulares del ángulo, obteniendo así para todas las regiones un condensado de igual tamaño. Para que esta condensación sea independiente de la orientación del objeto se puede establecer como ángulo cero en las coordenadas aquel para el cual la distancia entre el centro y la frontera es mayor. Una forma de tratar que la condensación sea independiente del tamaño de la región, es mediante la normalización para una región dada, de la distancia entre el centro y la frontera en el condensado.

La idea de esqueleto de una región, alude al conjunto de píxeles equidistantes a las fronteras de la misma. Es decir, para un píxel dado, se mide su distancia a los píxeles de la frontera y si el valor de distancia mínimo se da para dos o más píxeles de la frontera, el píxel dado es entonces considerado como parte del esqueleto de la región.

### **3.8.2 Características para la descripción de la forma**

Una característica relacionada con la forma y fácil de extraer de una región es el perímetro de la región. Dicho perímetro se puede aproximar como el número de píxeles que componen la frontera de la región. El valor exacto puede ser calculado mediante el empleo de un código de cadenas en el que se emplee 8-conectividad y en el que la granularidad del muestreo permita describir la inclinación de un píxel al siguiente, en lugar de un conjunto de píxeles a otro.

Concretamente se calcula como la suma ponderada del número de inclinaciones horizontales, inclinaciones verticales e inclinaciones diagonales. El peso de las inclinaciones horizontales y verticales será la unidad, mientras que en diagonales será raíz de dos. Otra característica sencilla, es el área de la región, que se define como el número de píxeles que componen la región. Otra métrica interesante es la compacidad de la región, que se define en función del perímetro y el área de la región, mediante la siguiente ecuación.

$$\text{compacidad} = \frac{\text{perímetro}^2}{\text{área}}$$

### Fórmula 3-33 Compacidad de la región

Otra característica relacionada con la forma, es el diámetro de la región, que dada una función de distancia, se define como el valor ofrecido por dicha distancia que resulta ser máximo entre todas las posibles combinaciones de pares de posiciones tomadas de los píxeles que forman la frontera de la región.

Otra característica es la orientación del eje mayor, donde eje mayor se llama a la línea que une los dos puntos que definen el diámetro de la región para el cual se está calculando la orientación del eje mayor. El eje menor es una recta perpendicular al eje mayor, que une dos puntos. Dichos dos puntos están cada uno de ellos situado a la distancia del centro necesaria y suficiente como para que se cumpla la siguiente condición. Primero se trazan dos rectas perpendiculares al eje mayor cada una en un extremo del mismo. Después se trazan dos rectas perpendiculares al eje menor cada una en un extremo del mismo. Entonces debe cumplirse que el rectángulo descrito por las cuatro rectas trazadas y que toma como vértices los puntos de intersección entre ellas, contiene a toda la región. Al mencionado rectángulo, se le llama rectángulo básico. Una característica útil de la forma de la región es la excentricidad de la frontera, que es la razón entre el eje mayor y el eje menor. Otra característica interesante es la curvatura, que se define como la tasa de variación de la pendiente en la frontera. Convencionalmente para calcular la curvatura se emplea la variación entre los segmentos de frontera, es decir, se observa en los vértices del polígono que describe la frontera, la variación de la pendiente del segmento precedente al siguiente. Si se realiza el cálculo en el sentido de las agujas del reloj, si en un vértice la variación de la pendiente es negativa indicará que es un punto convexo y si en él la variación de la pendiente es positiva, indicará que es un punto cóncavo. En caso de realizar el cálculo en sentido inverso al de las agujas del reloj, la variación positiva de la pendiente pasará a indicar un punto cóncavo y la variación negativa un punto convexo.

Otro método para describir la forma, se basa en la transformada de Fourier. Lo que se hace, es describir cada posición del píxel de la frontera de la región, como un valor complejo, donde la parte real se corresponde con la componente de las abscisas y donde la parte imaginaria se corresponde con la componente de las ordenadas. Descritas así las posiciones de los píxeles de la frontera, se aplica la transformada de Fourier discreta, para cada una de las posiciones. A los valores transformados se les llama descriptores de Fourier. Si bien empleando todos a la hora de realizar la transformada inversa se consigue recuperar los valores originales, lo interesante es que mediante el empleo de un número muy pequeño de descriptores de Fourier, se puede preservar la esencia general de la forma de la región. Esto se logra preservando los descriptores de Fourier

de baja frecuencia, mientras que los de alta frecuencia se eliminan, removiendo así los detalles. Aunque los descriptores de Fourier no son independientes directamente a la escala, el punto de comienzo, la rotación y la traslación, se pueden aplicar funciones propias del cálculo de transformadas de Fourier para resolver el problema.

Otro método para describir la forma, se basa en las propiedades topológicas de lo que se presenta en la región. La topología se centra en el estudio de las propiedades de los objetos que permanecen invariantes ante cualquier deformación, que no rasgue el objeto ni lo una con otros. Una característica topológica sencilla es el número de agujeros dentro de la región (usualmente las regiones contienen objetos sobre un fondo, por lo que se consideran los agujeros de los objetos). Otra característica topológica es el número de componentes conexas en la región (de igual modo, se consideran las componentes conexas de los objetos). Si al número de componentes conexas en una región se le resta el número de agujeros en dicha región, se obtiene un valor denominado número de Euler, el cual también es una propiedad topológica.

### 3.8.3 Características para la descripción de la distribución del color

Un concepto importante es la textura. Si bien no existe una definición exacta para la textura en el entorno en que aquí se emplea, se puede decir que la textura trata de describir propiedades como la regularidad, la suavidad o la aspereza. Una aproximación para la descripción de la textura, consiste en extraer estadísticos (normalizados) de los niveles de gris de la región, como pueden ser los momentos como la media y la varianza, la mediana, el valor máximo y el mínimo del nivel de gris, o el número de píxeles por debajo de la media o el número de los que están por encima. Basándose en la varianza del nivel de gris de la región, se define un descriptor relativo a la suavidad de la región, dado por la ecuación siguiente.

$$R = 1 - \frac{1}{1 + \sigma^2}$$

**Fórmula 3-34** Descriptor R de la suavidad

El momento estadístico de tercer orden, la asimetría, permite medir las desigualdades de la distribución del gris por la región. El momento estadístico de cuarto orden, la curtosis, mide el grado de apilamiento de la distribución. Una curtosis alta sugiere que la varianza es el resultado de la gran influencia de casos atípicos.

Teniendo la proporción normalizada de los distintos niveles de gris, es decir, teniendo una distribución de los distintos niveles de gris, se puede calcular la uniformidad como la suma de los cuadrados de las distintas probabilidades. También partiendo de esta distribución, se puede calcular la entropía promedio, como la suma de productos, donde cada producto es la probabilidad  $i$ -ésima multiplicada por el logaritmo de la probabilidad  $i$ -ésima.

## 3.9 Clasificación e inferencia

En esta etapa, la entrada, convencionalmente suele ser una matriz bidimensional, donde cada fila representa una región u objeto y cada columna representa una característica del mismo. El resultado de esta etapa dependerá del negocio que se tenga entre manos, aunque existen un

conjunto de anhelos que se presentan constantemente. Uno de ellos es la clasificación de los objetos, agrupando los objetos de la matriz en subconjuntos, según el valor de una característica. En concreto, la clasificación supervisada trata de deducir el valor de los objetos entrantes según la característica, por la cual se les pretende agrupar en subconjuntos a los objetos, mediante el aprendizaje que se puede realizar de un conjunto de objetos vistos previamente cuyo valor de la característica por la cual se pretende agrupar los objetos, se conoce. La asignación del valor de la mencionada característica (convencionalmente denominada clase) para ese conjunto de objetos previos, puede hacerse manualmente, ya sea mediante pruebas que permitan verificar el valor de la característica clase o mediante juicio experto. La inferencia de los siguientes casos entrantes se realizará mediante técnicas de clasificación supervisada, tema que se trata en el siguiente apartado.





## 4 Clasificación supervisada

El área de la clasificación supervisada comprende los métodos que a partir de un conjunto de casos de clase conocida, tratan de clasificar los nuevos casos que se vayan presentando cuya clase no sea conocida. Estos métodos o algoritmos, se conocen como clasificadores. El problema a resolver, que es el de la clasificación nuevos casos, se suele plantear visualmente como se muestra en la siguiente tabla.

$X$	$X_1$	...	$X_{n-1}$	$C = X_n$
$X^1 = (x_1^1, \dots, x_n^1)$	$x_1^1$	...	$x_{n-1}^1$	$x_n^1$
$X^2 = (x_1^2, \dots, x_n^2)$	$x_1^2$	...	$x_{n-1}^2$	$x_n^2$
...	...	...	...	...
$X^m = (x_1^m, \dots, x_n^m)$	$x_1^m$	...	$x_{n-1}^m$	$x_n^m$
$X^{m+1} = (x_1^{m+1}, \dots, x_{n-1}^{m+1}, \text{¿¿¿??})$	$x_1^{m+1}$	...	$x_{n-1}^{m+1}$	¿¿¿???

Tabla 4-1 Planteamiento del problema de la clasificación supervisada

En la anterior tabla la primera fila presenta los índices y el resto de filas presentan cada una un caso. La primera columna alude a todas las características de los casos, mientras que el resto de columnas, alude cada columna a una característica. La última columna alude a la característica llamada clase y es por eso que en el último caso que figura en la tabla éste no tiene valor. Los  $m$  primeros casos son los casos de clase conocida. Si ahora se emplea sobre la anterior tabla uno de los algoritmos de clasificación supervisada, dicho algoritmo tratará de determinar el valor de la característica clase para el último caso. Cabe indicar que el valor de la característica clase que ofrezca para ese último caso, será una especulación basada en la información que figura en el resto de la tabla.

Las variables que no son la variable clase (llamadas variables predictoras) pueden ser de distinta naturaleza, a saber, pueden ser variables discretas o continuas, finitas o infinitas, ordenables o no ordenables. Sin embargo, la variable clase debe cumplir con la propiedad de ser una variable discreta y finita. También se debe cumplir que el número de casos debe ser finito, y que el número de características por caso debe ser finito.

Los distintos métodos del área del conocimiento que aquí se aborda, cada uno de ellos pueden ofrecer resultados mejores o peores. Ahora cabe decir que implica “mejor” o “peor”. Dado un clasificador, y dado un conjunto de casos de clase conocida, se puede probar el comportamiento del clasificador pasándole el conjunto de casos, pero vetándole el conocimiento del valor de la característica clase para un subconjunto de casos del conjunto de casos. Es decir, del conjunto de

casos que se tiene al principio, se le pasa un subconjunto del conjunto de casos al clasificador como casos de clase conocida, y se le pasa otro subconjunto del conjunto de casos al clasificador como casos de clase desconocida. Cuando el clasificador termine de operar y se conozca las predicciones que realiza para los casos que se le pasaron como de clase desconocida, se podrán contrastar los resultados que otorgue con los valores de la variable clase que dichos casos realmente tenían.

El contraste entre valores de la variable clase predichos por el clasificador y los valores reales de la variable clase se puede plantear como un problema de minimización de costes. Así pues cuando un clasificador realiza predicciones de valores de la variable clase para un conjunto de casos, se contrasta con los valores reales de la variable clase de dicho conjunto de casos, mediante el cálculo del coste que implican las predicciones realizadas. El coste se define como la suma de los productos elemento a elemento de la matriz de confusión y de la matriz de coste. Para entender esta última oración, lo primero es indicar qué son la matriz de confusión y la matriz de coste. La matriz de confusión es una matriz cuadrada de tamaño igual al número de distintos posibles valores que puede tomar la variable clase. Cada número del uno al número de distintos posibles valores que puede tomar la variable clase, se vincula a un valor específico de los posibles de la variable clase (cada uno se vincula a uno distinto). Una de las dimensiones de la matriz aludirá a los valores reales de los casos, mientras que la otra dimensión de la matriz aludirá a los valores predichos por el clasificador para los casos. Así pues cuando un clasificador predice valores de la variable clase para un conjunto de casos de valor de clase conocida, pero cuyo conocimiento se ha vetado al clasificador, se puede ubicar cada caso en la matriz de confusión, en función del valor real de la variable clase y del valor predicho para dicho caso por el clasificador. Por tanto, un elemento de la matriz de confusión, representa el número de casos que se han etiquetado con un valor del tipo que representa el número de la columna (fila) en la que está el elemento de la matriz pero que realmente tenían un valor del tipo que representa el número de la fila (columna) en la que está el elemento de la matriz. Los valores de los elementos de la matriz de costes las define ad hoc el usuario. Es decir, el que pretende emplear un clasificador para predecir los valores de la variable clase de futuros casos, debe decidir cómo de costoso es etiquetar un caso con un valor concreto de la variable clase cuando realmente ese caso tiene un valor de la variable clase específico (y según el caso ambos valores concretos coincidirán o no). A continuación se realiza una presentación visual de la matriz de confusión, de la matriz de costes y del coste de etiquetar n casos.

$$\text{matriz de confusión} = \begin{pmatrix} nc_{11} & \cdots & nc_{n1} \\ \vdots & \ddots & \vdots \\ nc_{1n} & \cdots & nc_{nn} \end{pmatrix}$$

**Fórmula 4-1 Matriz de confusión**

$$\text{matriz de costes} = \begin{pmatrix} c_{11} & \cdots & c_{n1} \\ \vdots & \ddots & \vdots \\ c_{1n} & \cdots & c_{nn} \end{pmatrix}$$

**Fórmula 4-2 Matriz de costes**

$$coste = \sum_{i=1}^n \sum_{j=1}^n nc_{ij} \cdot c_{ij}$$

#### Fórmula 4-3 Coste

Como muestra, el siguiente ejemplo, en el que en la matriz de confusión las filas representan la clase real y las columnas representan la clase predicha por el clasificador (por ejemplo, si se predice que un caso es del tipo uno y luego es realmente del tipo tres, dicho caso se reflejará).

$$matriz\ de\ confusi3n = \begin{pmatrix} 15 & 8 \\ 3 & 24 \end{pmatrix} \quad matriz\ de\ costes = \begin{pmatrix} 0 & 2 \\ 5 & 0 \end{pmatrix}$$

$$coste = (15 \cdot 0) + (8 \cdot 2) + (3 \cdot 5) + (24 \cdot 0) = 31$$

#### F3rmula 4-4 Ejemplo de coste

Denotar que, si bien se puede definir cualquier matriz de coste, no todas se usan con la misma frecuencia. As3 pues aqu3 se presenta la m3s usual, en la que se representa que cada fallo en la clasificaci3n es igual de costoso.

$$matriz\ de\ confusi3n = \begin{pmatrix} nc_{11} & \cdots & nc_{n1} \\ \vdots & \ddots & \vdots \\ nc_{1n} & \cdots & nc_{nn} \end{pmatrix}$$

$$(nc_{ij} = 0 \Leftrightarrow i = j \quad nc_{ij} = 1 \Leftrightarrow i \neq j) \forall i, j \in [1, n]$$

#### F3rmula 4-5 Matriz de confusi3n usual

Cuando se trabaja con la anterior matriz de costes, normalmente la calidad de un clasificador no se suele presentar como su coste, si no como su tasa de acierto, donde cada caso que se etiqüete con el valor de clase que realmente tiene se considera una acierto, y se etiqueta con un valor distinto al que realmente tiene, se toma como un fallo. As3 pues se suele presentar su tasa de acierto como un porcentaje que indica la proporci3n de casos que se han etiquetado correctamente.

Cabe mencionar en este punto que se disponen como apoyo los ap3ndices acerca de las t3cnicas de validaci3n y la teor3a de la informaci3n estadística, pues en adelante se aludir3 a algunos de los conceptos de las dos materias mencionadas.

## 4.1 El problema del sobreentrenamiento

Lo que se desea de un clasificador cuando se le otorga un conjunto de casos cuyos valores de clase son conocidos, es que aprenda de esos hechos y genere una funci3n que permita determinar el valor de la variable clase dependiendo de los valores de las variables predictoras, para el problema planteado. Por tanto se desea que el clasificador posea una capacidad generalizadora de lo que aprende del conjunto de casos que se le ha entregado, para que pueda predecir el valor de la variable clase para casos que con los que no hab3a entrenado (un clasificador se entrena con los casos de los que aprende). A la hora de realizar el aprendizaje, se puede presentar el problema del sobreentrenamiento. Este problema se presenta cuando el

clasificador se ajusta demasiado a los casos con los que entrena, y pierde su capacidad generalizadora. Si paralelamente se observase la tasa de acierto sobre el conjunto de casos con los que entrena de un clasificador según va aprendiendo, y la tasa de acierto sobre un conjunto de casos con los que no entrena de ese mismo clasificador según va aprendiendo, el momento en el que la tasa de acierto sobre los casos con los que no ha entrenado comienza a ser decreciente indica la aparición del sobreentrenamiento. En la siguiente ilustración se muestra un ejemplo de este acontecimiento en una gráfica.

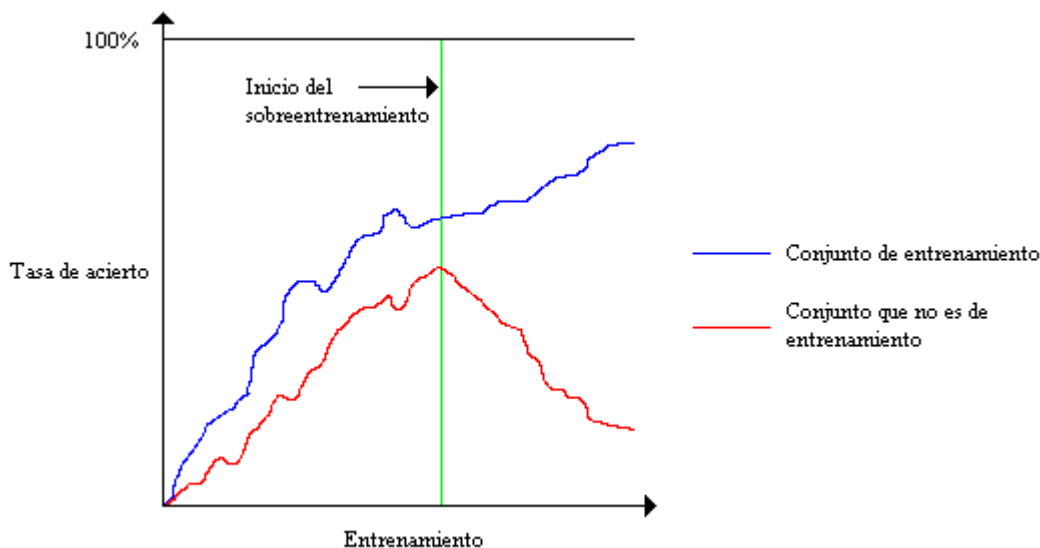


Ilustración 4-1 Ejemplo de sobreentrenamiento

## 4.2 Paradigmas de clasificadores

Los métodos de clasificación supervisada, no son todos de la misma naturaleza, pues se pueden diferenciar en grupos según el concepto en el que se basen. Aquí se exponen las ideas en las que se basan algunos de esos grupos, exponiéndose algunas de las más conocidas.

### 4.2.1 Clasificadores K-NN

Los clasificadores K-NN (K-Nearest Neighbour) se basan en la idea de asignar como valor de la variable clase a un caso entrante aquel valor de clase que más veces se repita entre los  $k$  vecinos ( $k$  entero mayor que cero) más cercanos al caso a clasificar (habrá que definir un sistema de desempate para los casos en que distintos valores de clase empaten en ser el valor de clase que más veces se repite). Por tanto habrá que definir una función de distancia a aplicar entre pares de casos, basada en las variables predictoras. Dado un conjunto de casos de valor de clase conocida, dada una función de distancia entre pares de casos, dado un valor de  $y$  y dado un nuevo caso de valor de clase desconocida, el algoritmo K-NN operaría de la siguiente forma. Primero, calcularía la distancia del nuevo caso a cada caso del conjunto de casos dado. Después, ordenaría las distancias resultantes de menor a mayor. Las  $k$  primeras distancias aluden cada una a la distancia del nuevo caso a un caso concreto del conjunto de casos. Entonces de esos  $k$  casos concretos aludidos cada uno por una de las  $k$  primeras distancias, se observan sus valores de la variable

clase y se cuentan para ver qué valor es el más repetido. El algoritmo K-NN predecirá entonces como valor de la variable clase del nuevo caso, aquel valor que se repita más entre los  $k$  casos mencionados.

Los clasificadores K-NN encajan dentro de los clasificadores denominados “vagos”. Esta categorización se debe al hecho de que los clasificadores K-NN no generan un modelo explícito de clasificación como otros paradigmas. En su lugar el modelo de clasificación de los clasificadores K-NN está implícito en el conjunto de casos de clase conocida que emplea como conjunto de entrenamiento.

Anteriormente se ha mencionado que a la hora de clasificar un caso nuevo, el algoritmo K-NN se puede encontrar con el problema de que dos o más valores de la variable clase empatan en ser los valores de la variable clase con mayor presencia entre los  $k$  casos más próximos al nuevo caso. También se ha dicho que habrá que definir un sistema de desempate. Algunos ejemplos de sistemas de desempate son elegir el valor de variable clase que tenga el caso más cercano al caso nuevo, o reintentar la clasificación reduciendo la  $K$  en un entero sucesivamente hasta que no se den empates. El algoritmo K-NN antes explicado es una de las variantes más sencillas de los clasificadores K-NN.

Una variación que se puede introducir a los algoritmos K-NN es la de introducir un valor de clase de rechazo. Así, todos aquellos nuevos casos para los que el clasificador no prediga una etiqueta lo suficientemente plausible, se etiquetarán con el valor rechazo que se ha incorporado a los valores de la variable clase. Se puede medir la plausibilidad de una etiqueta para un caso nuevo en función de la proporción del valor de la variable clase más repetido en los  $k$  más cercanos. Así, se puede indicar que si dicha proporción no supera un determinado umbral, el nuevo caso deberá ser etiquetado con el valor de rechazo.

Otra variación es la de asignar al nuevo caso el valor de la variable clase igual al valor de la variable clase del centroide más cercano al nuevo caso. Habrá un centroide por cada valor de la variable clase diferente presente entre los  $k$  casos más próximos al nuevo caso, y cada centroide estará ligado a uno de esos valores de la variable clase. Así pues la posición del centroide resulta ser la media de todos los casos de entre los  $k$  casos cuyo valor de la variable clase es el mismo al que el centroide está ligado.

Otro añadido, es que variar la influencia de cada caso de los  $k$  casos en a la hora de ver qué valor de la variable clase es el más repetido. Es decir, en lugar de contar cada caso de los  $k$  casos una vez (o con un peso igual a uno), se puede ponderar su contribución. Un ponderado habitual es el de valorar la contribución de cada caso de los  $k$  casos de forma inversamente proporcional a la distancia al caso nuevo. Así, los casos más cercanos tendrán mayor importancia o peso a la hora de decidir qué valor de la variable clase se predice para el nuevo caso.

### **4.2.2 Reglas de clasificación**

El paradigma de las reglas de clasificación se basa en generar un modelo que resulta ser un conjunto de reglas. Cada regla está compuesta por una condición y una consecuencia. La condición es una expresión lógica que puede ser cierta o falsa y que depende de las variables predictoras. La consecuencia es un valor de la variable clase. Así pues cuando se evalúa una regla

para un caso entrante, si la condición de ésta se satisface con dicho caso, se pronostica que el valor de la variable clase para dicho caso es la consecuencia asociada a la condición que verificó el caso. Existen varios algoritmos para construir modelos de clasificación basados en reglas, y aquí se exponen algunos de los más conocidos. En algunos de ellos el orden de evaluación de las reglas tiene relevancia, mientras que en otros es indiferente.

### **4.2.2.1 El algoritmo AQ**

El sistema AQR de inducción de reglas emplea el algoritmo AQ básico para generar un conjunto de reglas que sirvan de modelo de clasificación. El algoritmo AQ fue presentado en (Michalski, 1969), y AQR se muestra en (P. Clark, 1988) y (Sierra, 2006). En AQR, el orden de evaluación de las reglas no tiene relevancia. Antes de mostrar el pseudocódigo del AQR, cabe definir describir algunos términos mencionados originados en (Michalski, 1969). Un cover es la condición de una regla. Un selector es un test sobre una variable predictora, basado en una ecuación o en una inecuación, donde en uno de los lados de la ecuación o inecuación, figura dicha variable predictora. Un complejo es una expresión lógica formada por una de conjunción de selectores. Un cover, es una expresión lógica formada por una disyunción de complejos. Así pues un cover es una expresión lógica en forma normal disyuntiva. Cuando un caso verifica un cover, se dice que dicho cover cubre al caso.

La mecánica del algoritmo consiste en ir generando covers para cada valor de la variable clase. La generación de un cover para un valor de la variable clase específico, al que se liga, comienza por dividir el conjunto de casos en dos particiones, pos y neg. En pos estarán todos los casos cuyo valor de la variable clase sea igual que el valor para el cual se está generando un cover. En neg estarán todos los demás casos. Entonces se inicializa un cover que no cubre ningún caso. Iterativamente se van seleccionando casos de pos no cubiertos por el cover. Cuando todos los casos de pos sean cubiertos por el cover, el proceso iterativo finalizará, y el cover será el resultado. En cada una de esas iteraciones se construye un conjunto de complejos que se inicializa con un único complejo que cubre todos los casos. La construcción de dicho conjunto de complejos es un proceso iterativo anidado. Dicho proceso iterativo anidado, en el cual en cada iteración el primer paso es elegir un caso de neg cubierto por algún complejo del conjunto de complejos descrito. El segundo paso es especializar los complejos del conjunto de complejos mencionado para que no cubran el caso de neg elegido. El proceso iterativo anidado concluye cuando no existe un caso en neg que sea cubierto por algún complejo en el conjunto de complejos. Tras la construcción del conjunto de complejos, se escoge de entre ellos el más simple y se añade como disyunción al cover.

A continuación se muestra el pseudocódigo de AQR, y después se muestra el procedimiento STAR del cual AQR hace uso. El proceso STAR se corresponde con el proceso iterativo anidado antes mencionado.

```
AQR(pos,neg){
  cover ← {falso}
  mientras cover no cubra todos los elementos de pos{
    seed ← un elemento cualquiera de pos que cumpla con la condición de no
    ser cubierto por cover
    star ← Star(seed,neg)
    best ← complejo perteneciente a star más simple
    cover ← cover best
  }
  devolver cover
}
```

Ilustración 4-2 El algoritmo AQR

```
STAR(seed, neg){
  star ← {verdadero}
  mientras algún complejo en Star cobra algún elemento de neg{
    n ← un elemento de cualquiera de neg que cumpla con la condición de ser
    cubierto por algún complejo que pertenezca a star
    extension ← conjunto de selectores que cubren seed y no n
    star ← {xy|: x ∈ star y ∈ extension}
    eliminar los complejos en star contenidos en otros complejos en star
    (contenido en el sentido de que toda evaluación cierta por parte del
    complejo contenido también es cierta en el complejo contenedor)
  }
}
```

Ilustración 4-3 El procedimiento STAR

Así pues se genera un cover para cada valor de la variable clase, donde cada cover cubre los todos los casos cuyo valor de la variable clase es igual al valor de la variable clase al que está ligado al cover. Además cada cover no cubre ningún caso que cuyo valor de la variable clase sea diferente al valor de la variable clase que está ligado al cover.

### 4.2.2.2 El algoritmo CN2

A continuación se va a exponer el algoritmo CN2. En él, el orden de evaluación de las reglas sí tiene relevancia, por lo que a la hora de predecir el valor de la variable clase para un caso entrante con un conjunto de reglas inferidas con CN2, lo que se hace es buscar en orden cual es la primera regla que se cumple. Así, se predecirá como valor de la variable clase aquel que está ligado a dicha regla. El algoritmo que fue presentado en (P. Clark, 1988), también viene descrito en (Sierra, 2006).

Al algoritmo se le ofrece un conjunto de casos el cual empleará para inferir reglas. Durante la ejecución del CN2, dicho conjunto se irá actualizando, suprimiendo casos del conjunto según



vayan siendo cubiertos por los complejos que se van generando. CN2 comienza por declarar una lista de reglas vacía, tras lo cual comienza el proceso iterativo de generación de reglas. El primer paso en dicho proceso iterativo es calcular el mejor complejo para el conjunto de casos. El segundo paso consiste en reducir el conjunto de casos sustrayendo de él los casos del conjunto de casos que son cubiertos por el mejor complejo calculado en esa iteración. El último paso del proceso iterativo es el de la adición de una regla justo después de la última regla en la lista de reglas que el CN2 devolverá como resultado. La regla que se añade tiene como condición el mejor complejo calculado en esa iteración, y tiene como consecuencia el valor de la variable clase que más veces se presenta en los casos que se han sustraído del conjunto de casos en esa iteración.

La elección del mejor complejo para un conjunto de casos se realiza de la forma siguiente. Primero se parte de un mejor complejo que se verifica para cualquier caso, y se inicializa un conjunto que sólo contiene un complejo que se verifica para cualquier caso. Entonces comienza un proceso iterativo en el que se especializa el conjunto de complejos antes mencionado, además de que también se va actualizando el mejor complejo.

Dados dos complejos y un conjunto de casos, La forma de discriminar cuál es mejor, es definida por el usuario. Una forma posible de llevar a cabo ésta discriminación es la siguiente. Para cada uno de los dos complejos se calcula el conjunto de casos cubiertos sobre el conjunto de casos dado. Entonces se calcula para esos dos conjuntos de casos cubiertos, la entropía de la variable clase. Aquella entropía más alta, indica mayor aporte de información, e indica que el complejo que genera esa entropía en la variable clase es mejor que el otro. Otra forma es realizar unas extracciones aleatorias de los casos del conjunto de casos dado para formar un conjunto del cual se mide la distribución de la variable clase, para medir la divergencia Kullback-Leibler. Las extracciones aleatorias se realizan empleando la distribución de probabilidad de la variable clase del conjunto dado. Así pues, se calcula la divergencia Kullback-Leibler en dos ocasiones. En una se mide entre la distribución de la variable clase de uno de los conjuntos de casos cubiertos y la distribución de la variable clase en el conjunto compuesto de extracciones aleatorias. En la otra se mide entre la distribución de la variable clase del otro de los conjuntos de casos cubiertos y la distribución de la variable clase en el conjunto compuesto de extracciones aleatorias.

A continuación se muestra el pseudocódigo de CN2, y después se muestra el procedimiento MejorComplejo del cual hace uso, y que se encarga de encontrar en cada iteración del algoritmo CN2 de encontrar el MejorComplejo.

```

CN2(cc){
  ListaReglas ← {∅}
  MejorComplejo ← {verdadero}
  mientras ¬(MejorComplejo = ∅) ¬(cc = ∅) {
    MejorComplejo ← MejorComplejo(cc)
    si MejorComplejo ≠ ∅ {
      cc' ← conjunto de casos de cc cubiertos por MejorComplejo
      cc ← cc - cc'
      vccc' ← valor de la variable clase más presente en cc'
      ListaReglas ← {ListaReglas, ((MejorComplejo) → valor de la variable
        clase = vccc')}
    }
  }
  devolver ListaReglas
}
    
```

Ilustración 4-4 Algoritmo CN2

```

MejorComplejo(cc){
  Star ← {verdadero}
  MejorComplejo ← {verdadero}
  Selectores ← conjunto de todos los posibles selectores
  mientras Star ≠ ∅ {
    NuevoStar ← {xy | x ∈ Star y y ∈ Selectores}
    Eliminar de NuevoStar los complejos que pertenecen a Star y los complejos
    para los cuales ningún caso posible puede verificar el complejo
    Para cada complejo de Nuevo Star{
      si el complejo actual es estadísticamente significativo y mejor que
      MejorComplejo{
        MejorComplejo ← complejo actual
      }
    }
    Eliminar los peores complejos de NuevoStar hasta que este tenga un
    tamaño aceptable.
    Star ← NuevoStar
  }
  devolver MejorComplejo
}
    
```

Ilustración 4-5 Procedimiento MejorComplejo

#### 4.2.2.3 El algoritmo IREP

El algoritmo IREP (Incremental Reduced Error Pruning) genera reglas que se disponen en forma normal disyuntiva. Este algoritmo se presenta en (J. Fürnkranz, 1994). Ahora se va a emplear de

nuevo la terminología expuesta en el apartado del algoritmo AQ, en referencia a lo que es un cover, un complejo y un selector. Así pues en IREP los covers se disponen en forma normal disyuntiva.

Al algoritmo se le pasan tres parámetros. Los dos primeros son el conjunto de casos positivos y el conjunto de casos negativos en los que se particiona el conjunto de casos para el cual se va a inferir un cover. El último parámetro indica la proporción entre casos de crecimiento y casos de poda en los que se dividirá un conjunto dado dentro del algoritmo. Lo primero que ocurre es que el algoritmo se inicializa declarando un cover vacío. Después se da paso a un proceso iterativo de varias etapas. La primera etapa es dividir los conjuntos de casos positivos y negativos, cada uno, en dos conjuntos, según los casos se vayan a emplear para el crecimiento o la poda de un complejo. La siguiente etapa del proceso iterativo, es declarar un complejo vacío, para a continuación hacerlo crecer mediante el empleo del conjunto de casos positivos para crecimiento y del conjunto de casos negativos para crecimiento, y después podarlo con el uso del conjunto de casos positivos para poda y del conjunto de casos negativos para poda. Una vez generado el complejo de esa iteración, se compara si éste es mejor que un complejo vacío o no. Si resulta peor o igual, se ha terminado de construir el cover, se finaliza el proceso iterativo y se devuelve el cover como resultado. Si resulta mejor, se remueven del conjunto de casos positivos los casos de dicho conjunto cubiertos por el complejo, se remueven del conjunto de casos negativos los casos de dicho conjunto cubiertos por el complejo y se añade el complejo al cover como disyunción. El proceso iterativo, también podrá finalizar cuando el conjunto de casos positivos se quede vacío.

La construcción de un complejo, es un proceso de adición de selectores voraz. Se añade siempre el selector que muestre el mayor valor en la siguiente ecuación.

$$valor(c, c', pos - grow, neg - grow) = \frac{pos'}{pos} \cdot \left[ -\log_2 \left( \frac{pos}{pos + neg} \right) + \log_2 \left( \frac{pos'}{pos' + neg'} \right) \right]$$

**Fórmula 4-6 Criterio de mejora en el crecimiento de un complejo en IREP**

En donde  $c$  es el complejo antes de la adición del selector actual,  $c'$  es el complejo con el selector añadido como conjunción,  $pos - grow$  es el conjunto de casos positivos para el crecimiento del complejo,  $neg - grow$  es el conjunto de casos negativos para el crecimiento del complejo,  $pos$  es el número de casos cubiertos por  $c$  en el conjunto  $pos - grow$ ,  $pos'$  es el número de casos cubiertos por  $c'$  en el conjunto de  $pos - grow$ ,  $neg$  es el número de casos cubiertos por  $c$  en el conjunto de  $neg - grow$  y  $neg'$  es el número de casos cubiertos por  $c'$  en el conjunto de  $neg - grow$ . El proceso de crecimiento de un complejo concluye cuando no se puede añadir ningún selector que mejore el valor de la ecuación.

La poda de un complejo, es un proceso de eliminación de selectores secuencial, en orden inverso al que se han añadido al complejo. Así pues se eliminan selectores mientras se incrementa el valor de la siguiente ecuación.

$$valor = (c, pos - prune, neg - prune) = \frac{pos + (N(neg - prune) - neg)}{N(pos - prune) + N(neg - prune)}$$

**Fórmula 4-7 Criterio de mejora en la poda de un complejo en IREP**

En donde  $c$  es el complejo antes de la adición del selector actual,  $pos - prune$  es el conjunto de casos positivos para la poda del complejo,  $neg - prune$  es el conjunto de casos negativos para la poda del complejo,  $pos$  es el número de casos cubiertos por  $c$  en el conjunto  $pos - prune$ ,  $neg$  es el número de casos cubiertos por  $c$  en el conjunto de  $neg - prune$  y  $neg$  es el número de casos cubiertos por  $c$  en el conjunto de  $N$  es el operador que calcula la cardinalidad de un conjunto. El proceso de poda de un complejo concluye cuando no se puede remover ningún selector que mejore el valor de la ecuación.

A continuación se muestra el pseudocódigo de IREP.

```
IREP (pos, neg, proporcion) {
  cover ← ∅
  mientras pos ≠ ∅ {
    dividir (proporcion, pos, pos-grow, pos-prune)
    dividir (proporcion, neg, neg-grow, neg-prune)
    complejo ← ∅
    complejo ← GrowComplejo (clausula, pos-grow, neg-grow)
    complejo ← PruneComplejo (clausula, pos-prune, neg-prune)
    if precision (complejo) ≤ precision(∅){
      devolver (cover)
    }
    else {
      pos ← pos - casos de pos cubiertos por el complejo
      neg ← neg - casos de neg cubiertos por el complejo
      cover ← cover complejo
    }
  }
  devolver (regla)
}
```

Ilustración 4-6 Algoritmo IREP

#### 4.2.2.4 El algoritmo RIPPER

El algoritmo RIPPER (Repeated Incremental Pruning Produce Error Reduction), que fue descrito en (Cohen, 1995), es una mejora del algoritmo IREP. Aquí también se empleará la terminología de cover, complejo y selector expuesta anteriormente. Igual que en IREP, en RIPPER cada cover se dispone en forma normal disyuntiva.

Los cambios con respecto a IREP, se pueden dividir en tres partes. La primera parte de cambios consiste en reemplazar el criterio de mejora de la poda de un complejo, reemplazándolo por el siguiente.

$$valor = (c, pos - prune, neg - prune) = \frac{pos - neg}{pos + neg}$$

Fórmula 4-8 Criterio de mejora en la poda de un complejo en RIPPER

La siguiente parte de cambios consiste en modificar la condición de parada en la adición de complejos al cover. Se añade a lo existente en IREP, la condición de parada por la cual se deberá dejar de añadir complejos al cover si se verifica que la longitud del cover es  $x$  bits más larga que el cover más pequeño posible. En (Cohen, 1995) se prueba con una diferencia de longitud de 64.

La última parte de cambios consiste en un proceso de optimización de los covers. En dicho proceso se van examinando los complejos del cover secuencialmente. Para cada complejo, se plantean dos complejos alternativos, y se deberá escoger uno de los tres para que finalmente ocupe el lugar del complejo original. El primer complejo alternativo es un complejo que se hace crecer y se poda teniendo como criterio de mejora en la poda el deseo de minimizar el error de todo el cover sobre el conjunto  $pos - prune \cup neg - prune$ . El segundo complejo alternativo, se obtiene partiendo del complejo original, se le hace crecer y luego se poda, empleando como criterio de mejora en la poda el deseo de minimizar el error de todo el cover sobre el conjunto  $pos - prune \cup neg - prune$ .

### 4.2.3 Árboles de clasificación

El paradigma de los árboles de clasificación se basa en generar un modelo que resulta ser una estructura arbórea. La estructura arbórea es un grafo compuesto por arcos y nodos. Los arcos son direccionales y cada arco sale de un nodo considerado el padre y entra en otro nodo considerado el hijo, en la relación establecida entre ambos nodos por dicho arco. En la estructura arbórea se cumple que hay un único nodo que no tiene padres, el cual se llama nodo raíz. Además, también se verifica que todo nodo tiene como máximo un padre. A los nodos que no tienen hijos, se les suele llamar nodos hoja. En dicha estructura arbórea, los nodos hoja contienen valores de la variable clase. Para clasificar un nuevo caso entrante, se recorrerá el árbol como si fuera un camino desde el nodo raíz hasta un nodo hoja. A la hora de recorrer el camino, en cada punto al que se llegue y en el que suceda que el camino se divide en varios, se debe elegir aquel de ellos, que se ajuste al caso. Lo que sucede es que todo nodo del árbol que no es hoja, con cada una de sus ramas conforma una expresión lógica evaluable para los casos entrantes. El caso más sencillo de este hecho es que un nodo aluda a una variable de los casos que no es la variable clase, y cada arista que sale de él, contenga una expresión que unida a la variable aludida por el nodo que toma el papel de padre en la arista conforme un selector (término explicado en el apartado del algoritmo AQ de reglas de clasificación). Se verifica que las distintas ramas que salen de un nodo, particionan el conjunto de casos que pueden llegar hasta ese nodo, generando un subconjunto por rama. Antes de exponer algunos algoritmos concretos, se expone el algoritmo TDIDT (Top Down Induction Decision Trees) en el cual se basan la mayoría de algoritmos de generación de modelos de clasificación basados en árboles. A continuación se muestra el algoritmo TDIDT.

```

TDIDT (cc) {
  if (todos los casos de cc tienen el mismo valor, c, en la variable clase){
    a ← nodo hoja con valor de variable clase c
  }
  else {
    v ← se escoge una variable predictora según un criterio dado
    se particiona cc en  $c^1, \dots, c^p$  separado los casos de distinto valor en la
    variable v ( $v^1, \dots, v^p$ ) en distintos conjuntos, y manteniendo los casos con
    igual valor en la variable v en el mismo conjunto. p es el número de
    distintos valores de la variable v en cc.
    i ← 1
    mientras  $i \leq p$  {
       $a^i \leftarrow TDIDT(c^i)$ 
       $i \leftarrow i + 1$ 
    }
    a ← construir un árbol con un nodo ligado a la variable v como raíz. El árbol
    tendrá p ramas, donde la i-ésima rama tiene como selector  $v = v^i$ , y como
    nodo hijo el nodo raíz del árbol  $a^i$ .
  }
  devolver a
}

```

Ilustración 4-7 Algoritmo TDIDT

### 4.2.3.1 El algoritmo ID3

El algoritmo ID3 fue desarrollado por John Ross Quinlan y se comenta en (Quinlan, 1986). Aquí se comenta su versión determinista, organizada en un pseudocódigo propio. Al algoritmo se le da un conjunto de casos. El árbol que devolverá se va construyendo iterativamente. Antes de entrar a dicho proceso iterativo, se define el tamaño de una ventana, que será en el fondo un conjunto de casos, por lo que su tamaño indica la cardinalidad de dicho conjunto de casos. Tras la declaración del tamaño de la ventana, se construye la ventana realizando extracciones sin reemplazamiento del conjunto de casos dados al algoritmo hasta alcanzar el tamaño de ventana especificado. Entonces se declara un árbol vacío. A continuación, se manda expandir el árbol vacío con la ventana, mediante el procedimiento ID3\_ea (ver más adelante), el cual modificará el árbol parámetro de dicho procedimiento para que consiga etiquetar correctamente los casos del conjunto de casos parámetro de dicho procedimiento. Tras esto, se comprueba si el árbol generado clasifica correctamente todos los casos del conjunto de casos dado al algoritmo que no están en la ventana. Si clasifica todos esos casos correctamente no se llega a entrar en el proceso iterativo y se devuelve el árbol. En otro caso, se comienza un proceso iterativo, en el que se añaden los casos que no están en la ventana y que fueron clasificados incorrectamente por el último árbol generado a la ventana, después se expande el último árbol generado con la nueva ventana en empleando el procedimiento ID3\_jea, para volver a comprobar si clasifica correctamente todos los casos del conjunto de casos dado al algoritmo que no están en la ventana. Tras dicha comprobación, el proceso iterativo terminará o comenzará una nueva

iteración, expandiendo la ventana, luego el árbol y finalmente comprobando de nuevo. A continuación se muestra el pseudocódigo del algoritmo ID3.

```
ID3 (cc) {
    c ← variable clase
    tv ← entero a definir por el usuario que representa el tamaño inicial de la ventana
    w ← subconjunto de casos de cc de tamaño tv donde cada caso se ha escogido
    aleatoriamente sin reemplazamiento
    a ← ∅
    a ← ID3_ea (w, a)
    while (a clasifique algún caso de cc – w erroneamente) {
        w ← w ∪ casos de cc – w mal clasificados por a
        a ← ID3_ea (w, a)
    }
    devolver a
}
```

### Ilustración 4-8 Algoritmo ID3

El procedimiento ID3\_ea como se ha comentado sirve para expandir un árbol. Si el árbol es vacío o está compuesto exclusivamente por un nodo hoja, se construye un árbol con el procedimiento ID3\_ca (ver más adelante), que será el árbol devuelto. En otro caso se manda expandir cada árbol cuyo nodo raíz es hijo del nodo raíz del árbol que se está expandiendo. A cada una de esas expansiones, se le pasará un subconjunto de casos del conjunto de casos dado para el árbol que se está expandiendo. Dicho subconjunto estará compuesto por todos los elementos del conjunto dado para el árbol que se está expandiendo que verifican la expresión lógica compuesta por la variable aludida por el nodo raíz del árbol que se está expandiendo y por la expresión ligada a la rama que va desde el nodo raíz del árbol que se está expandiendo al hijo pertinente de la expansión que se pretende lanzar. A continuación se muestra el pseudocódigo del procedimiento ID3\_ea (ea de expansión de árbol).

```

ID3_ea (w, a) {
  If (a se compone de un único nodo hoja o a es vacío) {
    a ← ID3_ca (w)
  }
  else {
    Para cada rama ri que sale del nodo raíz de a {
      ai ← árbol que tiene como nodo raíz el nodo al que toma el papel
      de hijo en la rama ri
      wi ← subconjunto del conjunto w en el cual todo elemento verifica
      la expresión lógica compuesta por la variable aludida por el nodo
      raíz de a y la expresión ligada a la rama ri, y que además en el
      conjunto w – wi dicha condición no se cumple para ningún
      elemento
      ai ← ID3_ea (wi, ai)
    }
  }
}

```

**Ilustración 4-9 Procedimiento de ID3 para la expansión de un árbol**

El procedimiento ID3\_ca como se ha comentado sirve para construir un árbol. Si el conjunto parámetro es vacío devuelve un árbol vacío. Si no ocurre nada de lo anterior y si no hay variables predictoras en los elementos del conjunto parámetro devuelve un árbol que se compone de un nodo hoja en el que figura el valor de la variable clase más presente en el conjunto de casos. Si no ocurre nada de lo anterior y si todos los casos del conjunto de casos dado tienen el mismo valor en la variable clase, se devuelve un árbol compuesto por un nodo hoja en el que figura dicho valor de la variable clase. Si no ocurre nada de lo anterior, se calcula la información mutua de cada variable predictora con la variable clase, se escoge la variable que otorgue la información mutua más alta de ella con la variable clase, y se le hace un test estadístico de independencia. Si se acepta la hipótesis de independencia, se devuelve un árbol que se compone de un nodo hoja en el que figura el valor de la variable clase más presente en el conjunto de casos. Si se rechaza la hipótesis de independencia, se procede a una generación del árbol recursiva. En dicha construcción, la primera etapa consiste en declarar un nodo raíz en el que figura la variable predictora elegida que realiza un aporte significativo. Entonces se llama a ejecutar al procedimiento ID3\_ca varias veces, cada vez sobre un subconjunto distinto. Los conjuntos sobre los que se ejecuta el ID3\_ca en esas llamadas, se generan de la siguiente forma. Cada uno de esos conjuntos se genera a partir del conjunto de casos dado. Para la generación de cada uno de esos conjuntos, se filtra el conjunto de casos dado eliminando todos los casos que no tengan cierto valor en la variable predictora elegida, valor que será diferente para cada conjunto a generar. Dicho cierto valor, es distinto cada vez, quedando así un conjunto de subconjuntos del conjunto de casos dado, donde la colección de subconjuntos resulta ser un particionamiento del conjunto de casos dado. A continuación, en cada elemento de la citada colección se elimina la variable predictora elegida (se borra para cada caso perteneciente al elemento). El conjunto de subconjuntos resultante de esta última operación es la colección de elementos sobre la que se realizarán las llamadas del ID3\_ca, realizando una llamada cada vez sobre un elemento distinto.



Tras cada llamada al `ID3_ca`, si el árbol que retorna es no vacío, se añade una rama del nodo raíz al árbol devuelto por la llamada, ligando a la rama la expresión " $= v$ " donde  $v$  en este caso será el valor de la variable predictora con el cual se ha filtrado previamente el elemento con el cual se ha realizado dicha llamada a `ID3_ca`. Para ver todo este proceso más claro, a continuación se presenta el pseudocódigo del algoritmo `ID3_ca`, y justo después se presentan las funciones auxiliares de las que hace uso.

```

ID3_ca (cc) {
    c ← variable clase
    if (N(cc) = 0) {
        a ← ∅
        devolver a
    }
    a ← nuevo nodo
    if (no hay variables predictoras en cc) {
        etiquetar a con el valor v de la variable clase que más se presente en cc
    }
    else {
        if (todos los casos de cc tienen el mismo valor v en la variable clase) {
            etiquetar a con el valor v de la variable clase
        }
        else {
            para cada variable predictora vari en cc {
                calcular InformaciónMutua(vari, c)
            }
            var ← variable predictora para la cual su InformaciónMutua (vari, c) calculada es la máxima
            If (se rechaza la hipótesis de independencia en un test estadístico de independencia entre var y c) {
                etiquetar a con var
                para cada valor v de var {
                    ai ← ID3_ca (Subtabla(cc, var, v))
                    if (a ≠ ∅) {
                        construir una rama de a a ai a la que se le asocia la expresión " = v"
                    }
                }
            }
            else {
                etiquetar a con el valor v de la variable clase que más se presente en en cc
            }
        }
    }
    devolver a
}

```

**Ilustración 4-10 Procedimiento de ID3 para la construcción de un árbol**

A continuación se explican los procedimientos auxiliares. El procedimiento Subtabla de dos parámetros, consiste en remover del conjunto de casos parámetro la variable parámetro. EL procedimiento Subtabla de tres parámetros, remueve del conjunto de casos parámetro los casos

que no tengan en la variable parámetro el valor parámetro, y a continuación procede a remover del conjunto de casos parámetro la variable parámetro. Seguidamente se muestra su pseudocódigo.

```
Subtabla (cc, var) {  
    scc ← eliminar de cada caso de scc la variable var  
    devolver scc  
}  
  
Subtabla (cc, var, v) {  
    scc ← subconjunto de casos de cc, donde cada caso c del subconjunto cumple la  
    condición siguiente,  $c_{var} = v$   
    scc ← eliminar de cada caso de scc la variable var  
    devolver scc  
}
```

### Ilustración 4-11 Procedimientos auxiliares de ID3

Comentar finalmente, que está matemáticamente demostrado que el empleo de la información mutua como herramienta de elección de variables predictoras para la generación del árbol, es un método sesgado que favorece la selección de variables predictoras en las cuales la cardinalidad del conjunto de valores que pueden tomar es más alta.

#### 4.2.3.2 El algoritmo C4.5

El algoritmo C4.5 es una propuesta realizada en (Quinlan, 1993). C4.5 es una mejora del algoritmo ID3. En él se reemplaza el criterio de selección de variables predictoras para la generación del árbol de la información mutua por el criterio del cociente compuesto por el numerador, que en esta ocasión es la información mutua, y el denominador, que en este caso es la entropía de Shannon de la variable predictora dada para el cálculo del criterio. A este criterio se le llama gain ratio, y a continuación se muestra su ecuación.

$$\text{Gain Ratio } (X_i, C) = \frac{I(X_i, C)}{H(X_i)}$$

#### Fórmula 4-9 Gain Ratio

Otra mejora con respecto a ID3, que C4.5 incorpora, es un sistema de poda que se aplica tras haber inferido el árbol. La idea de éste sistema de poda es la de indicar si un subárbol dado no trivial (que no es vacío ni es un árbol que está compuesto por un único nodo hoja) del árbol inferido, tiene razón de ser. Es decir, se plantea que quizá sería mejor reemplazar dicho subárbol por un único nodo hijo que tendrá el valor de la variable clase que más se presente en los casos que lleguen hasta dicho nodo. En la siguiente fórmula, conjunto a la exposición del criterio en sí que permite discernir cuando se realiza la poda y cuando no, viene expuesto previamente una función de la que depende el criterio. La inequación que figura al final de la fórmula, caso de cumplirse implicará la conveniencia de la poda, y en cualquier otro caso indicará la conveniencia de no realizar dicha poda.

$$ne'(a_i) = ne(a_i) + \frac{nh(a_i)}{2}$$
$$ne'(a) + \sqrt{\frac{ne'(a) \cdot [nc(a') - ne'(a)]}{nc(a')}} \geq ne'(a')$$

### Fórmula 4-10 Criterio de poda en C4.5 tras la inducción del árbol

En la anterior fórmula,  $a_i$  es un árbol entrante a la función auxiliar,  $a$  es el subárbol que se está planteando reemplazar,  $a'$  es el subárbol en el que se trasformaría  $a$  caso de verificarse la inecuación anterior. La función  $ne$  devuelve el número de errores que produce el árbol parámetro de dicha función, al clasificar un conjunto de casos. Dicho conjunto de casos es un subconjunto de casos con el cual se ha inferido el árbol global. Dicho subconjunto está por compuesto por los casos que según se intentan clasificar, llegan al nodo raíz del árbol parámetro de la función  $ne$ . La función  $nh$  devuelve el número de nodos del árbol parámetro que son nodos hoja. La función  $nc$  devuelve la cardinalidad de un conjunto de casos. Dicho conjunto de casos es un subconjunto de casos con el cual se ha inferido el árbol global. Dicho subconjunto está por compuesto por los casos que según se intentan clasificar, llegan al nodo raíz del árbol parámetro de la función  $nc$ .

### 4.2.4 Redes bayesianas

El paradigma de las redes bayesianas se basa en la teoría de la probabilidad, que modela los fenómenos no deterministas. Como clasificador se genera un tipo de modelo gráfico probabilístico que se compone de un grafo acíclico dirigido, al que se llama red bayesiana. En él los nodos representan variables y los arcos representan dependencias condicionales. La idea es obtener la red bayesiana de los datos, y a través de la red bayesiana, realizar inferencias para obtener conocimiento. Cuando un arco direccional sale de una variable y entra en otra, se dice que aquella de la que sale es el padre y aquella en la que entra es el hijo, en la relación entre ambas variables establecida por dicho arco dirigido. Además del modelo gráfico, para cada variable se tienen las probabilidades condicionadas a los valores de sus variables padres, y su distribución de probabilidad a priori.

#### 4.2.4.1 Criterio de D-separación

Tomando un modelo gráfico de red bayesiana, se pueden deducir independencias condicionales, en base al criterio conocido como D-separación. Un primer conjunto de variables es independiente de un segundo conjunto dado un tercero, si ocurre que para el modelo gráfico transformado a no dirigido (cambiando todos los arcos dirigidos por no dirigidos), sucede que para todos los caminos entre cualquier nodo perteneciente al primer conjunto y cualquier nodo perteneciente al segundo conjunto, hay un nodo intermedio que verifica al menos una de las dos siguientes premisas. O bien ese nodo pertenece al tercer conjunto y no es de aristas convergentes (tomando las dos aristas que forman el camino que pasa por el nodo, se observa si las aristas originales (dirigidas) que fueron sustituidas por esas, no son convergentes, es decir, al menos una de las dos no apunta hacia el nodo), o bien el nodo es de aristas convergentes y ni él ni sus descendientes están en el tercer conjunto. Otra técnica para deducir las independencias condicionales es el siguiente. Si el tercer conjunto separa al primer y al segundo conjunto en el

grafo moral del menor subconjunto ancestral que contenga a los tres conjuntos de nodos, sucede que el primer y segundo conjuntos son condicionalmente independientes dado el tercer conjunto. El menor subconjunto ancestral para un conjunto de nodos es aquel que los incluye a ellos y a todos sus antecesores. El grafo moral de un grafo dirigido, es aquel que se obtiene tras reemplazar los arcos dirigidos por no dirigidos y tras unir por un arco a los padres que tengan hijos (tomando como referencia el grafo original) comunes que aún no estén conectados. Una factorización de una distribución de probabilidad conjunta de un conjunto de variables alude a la igualdad de este término con el producto de probabilidades condicionadas de cada variable de ese conjunto a sus respectivas variables padres.

### **4.2.4.2 Inferencia**

Ahora se trata la parte de obtener conocimiento dada la red bayesiana. La probabilidad de una variable se puede calcular aplicando el teorema de la probabilidad total descomponiendo su probabilidad a una suma de productos entre probabilidades condicionales a sus antecesores y las probabilidades de la conjunción de dichos antecesores. Teniendo en cuenta el modelo gráfico de la red bayesiana y los criterios de D-separación, la expresión anterior se puede simplificar reduciendo el número de cálculos necesarios (también la agrupación de sumas mediante la marginalización de variables puede ayudar). Esta descomposición también se puede aplicar al cálculo de probabilidades condicionadas de una variable a otra.

La idea es poder dar respuesta a preguntas como cuál es la probabilidad de una variable, o cuál es la probabilidad de una variable condicionada a otra (p.e. cuál es la probabilidad de extraer una bola blanca a sabiendas de que urna se ha extraído). Estos interrogantes son generalizables a conjuntos de variables, es decir, plantearse cuál es la probabilidad de que un conjunto de variables tome unos resultados concretos, o cual es la probabilidad del hecho anterior condicionado a que otro conjunto de variables tome unos resultados concretos. Dentro de las preguntas planteables a la red bayesiana, se puede discernir entre preguntas predictivas (en las que se calcula la probabilidad de una variable condicionada normalmente a sus antecesores), preguntas de diagnóstico (en las que la variable condicionada suele ser un ancestro de las variables que la condicionan), o preguntas que son un mixto de las dos anteriores, en las que entre las variables condicionantes suele haber tanto ancestros como descendientes de la variable condicionada. Estas discriminaciones de los tipos de preguntas son generalizables a casos en los que haya tanto conjuntos de variables condicionadas como conjuntos de variables condicionantes. Otro tipo de pregunta planteable es la de buscar el conjunto de valores que toma un conjunto de variables que hace maximiza la plausibilidad de que otro conjunto de variables tome otro conjunto de valores (p.e. si se tiene un saco con muchos dados de diferente número de caras, y se extraen tres al azar y se lanza cada uno una vez, se puede calcular que conjunto de tres dados hace más plausible obtener el resultado que se ha obtenido).

Hasta ahora se ha hablado de los objetivos (los distintos tipos de preguntas a los que se desean respuesta), ahora se habla del viaje, los métodos, la forma de llegar a esos objetivos. Un método puede distinguirse entre exacto y aproximado, sucediendo que en los métodos exactos se obtiene la respuesta cierta ante una cuestión sobre una probabilidad, mientras que en los métodos no exactos se obtiene una respuesta que no es cierta, pero que es lo suficientemente próxima al valor real. El primer método exacto es el cálculo por fuerza bruta. Por ejemplo se trata de

averiguar que probabilidad tiene un resultado concreto de una variable que se va descomponiendo mediante el teorema de la probabilidad total hasta depender de todos sus ancestros. El problema es que este cálculo es intratable para la mayoría de problemas, pero la esperanza reside en que este cálculo es muy ineficiente, y se puede mejorar, reduciendo el tiempo necesario para tratarlo. La primera mejora a este tratamiento está basada en el reordenamiento de las sumas, ya que hay términos que se repiten muchas veces, y mediante un buen reordenamiento, se puede tratar de agrupar dichos términos, evitando tener que calcularlos varias veces. El problema es que encontrar la solución óptima (aquella que implicara menor número de cálculos) al problema del reordenamiento de sumas consume un volumen de tiempo que crece de forma exponencial con respecto al número de variables (de ahí que se apliquen heurísticos de búsqueda para resolver el problema).

El algoritmo de eliminación de variables es una respuesta al problema del reordenamiento de sumas, en el caso de que se le plantee una pregunta en la que se tienen una serie de evidencias (un conjunto de variables toma una serie de valores concretos) y en el que se desea conocer la distribución de probabilidad de una variable. Así pues lo primero que hace es agregar las evidencias a una lista en la que figuran todas las funciones de probabilidad. Después se define un orden que contenga a todas las variables salvo a la variable objetivo. Entonces se entra en un proceso iterativo que se repite un número de veces igual a la longitud del orden definido anteriormente. Entonces en cada iteración, se coge la variable del orden de igual posición al número de la iteración. Tras esto se suprimen de la lista en la que se integraron las evidencias todas aquellas funciones que contengan a la variable elegida. Entonces se crea una nueva función de probabilidad que es igual a la suma para todos los valores de la variable elegida, de los productos de todas las funciones suprimidas en esta iteración de la lista en la que se integraron las evidencias. Después se agrega esta función de probabilidad a la lista donde se integraron las evidencias, tras lo que puede comenzar la próxima iteración. Finalmente se coge la lista en la que se integraron las evidencias, y tras todo el proceso iterativo, se combinan todas las funciones contenidas en una única función. Hay que indicar, que la elección del orden es crítico, pues la complejidad del algoritmo dependerá del orden elegido.

Otra respuesta al problema de la agrupación de sumas es el algoritmo de propagación, que es un proceso basado en el paso de mensajes, mediante los cuales se propaga evidencia por el grafo. En el proceso, se elige un nodo. Entonces este nodo solicita al resto información. Una vez la tiene, la propaga por la red, y una vez se ha propagado, en cada nodo se marginaliza hasta que sólo quedan las dimensiones del propio nodo y de sus padres.

El algoritmo de eliminación de variables y el algoritmo de propagación no sirven para grafos en los que existan ciclos, aunque estos ciclos solo existan en el grafo no dirigido asociado al original dirigido, y en este no existan, sigue sin servir. Para este problema se ofrecen como solución los métodos de agrupamiento, que tratan de agrupar nodos de forma que se eliminen los ciclos generando estructuras auxiliares, y tras aplicar un método de agrupamiento, quedara una red bayesiana acíclica en la que ya se pueden aplicar las técnicas anteriormente descritas. Por ejemplo, uno de los métodos de agrupamiento, consiste en la siguiente secuencia de pasos. Primero, moralizar el grafo (unir los padres que tengan hijos comunes y tomar su grafo no dirigido asociado). Después, se triangula el grafo resultante, haciendo que cada ciclo de longitud cuatro o superior, se parta. Esto se hace, eliminando los nodos del grafo en cualquier secuencia, haciendo

que cada vez que se elimine uno, y si ese pertenece a un ciclo o más, para cada ciclo del que formaba parte, se unan cada una de las dos variables vecinas con las que formaba un segmento del ciclo mediante un arco. Cada vez que se elimina un nodo, si se debe corregir en un ciclo que conformaba junto a otras variables mediante la unión de sus dos vecinas, se dirá que ese nodo y esas dos vecinas conformaban un clique (un clique es un subgrafo en el que todo par de nodos está unido por un arco), que aunque no lo conformaban realmente en el grafo original, sirve como expresión para describir el proceso siguiente. Ahora se concibe un nuevo grafo, en el que solo hay nodos, y en donde cada nodo contiene lo que se acaba de identificar como "clique". Es decir, cada nodo es una conjunción de las tres variables que antes conformaban el clique. Ahora usando el orden en el que se han eliminado los nodos del grafo anterior y en el que se han obtenido los cliques, se une el clique al que le toca el turno con aquel que tenga más variables (de las que agrupa dentro cada nodo) en común, siempre que no formen un ciclo en este nuevo grafo. Ahora se asignan potenciales a cada nodo, que serán para cada uno el producto de los potenciales de las variables que engloba (para la determinación de estos términos se considera el grafo original. Es decir, se observa la estructura original entre las tres variables, y a partir de ahí, se determinan los términos, por ejemplo, si una es el padre de las otras dos, el producto de los potenciales que engloba el nodo que contiene a estas tres variables, será el producto de la probabilidad del padre, por la probabilidad condicionada de un hijo al padre, por la probabilidad condicionada del otro hijo al padre). Si queda algún nodo sin potencial asociado, este se definirá con valor igual a uno. Tras haber concluido todas estas etapas, se aplica nuevamente un algoritmo de paso de mensajes sobre el grafo que se ha obtenido en última instancia.

### **4.2.4.3 Construcción de la red bayesiana**

Ahora se habla de cómo obtener la red bayesiana de los datos. Para la construcción de una red bayesiana, primero, se debe elegir que variables tomarán parte en ella y definir que rangos tomarán los valores de cada variable. Después se debe decidir que dependencias se dan entre variables (relaciones padre-hijo). Para cada variable se deben calcular sus parámetros, que son su distribución de probabilidad a priori y las distribuciones de probabilidad condicionadas a los valores de sus padres. Así, en la red se puede distinguir entre determinar los parámetros (los valores anteriores) y determinar la estructura de la red (entre que pares de variables hay relaciones padre-hijo).

Se comienza hablando del cálculo de parámetros. Ahora procede explicar el concepto de estimación de máxima verosimilitud. Imaginar que se dan dos distribuciones de probabilidad sobre el lanzamiento de un dado de seis caras, en el que una distribución es equilibrada (todos los posibles resultados son equiprobables), y en la otra distribución los tres valores más altos tienen cada uno el doble de probabilidades que cada uno de los más bajos. Teniendo una serie de casos, donde cada uno es el resultado de un lanzamiento, se puede ver cuál de las dos distribuciones es más verosímil en cuanto a esos casos. Es decir, cuál de las dos es más probable que genere esos resultados concretos. Para el cálculo se emplea la función de verosimilitud, en donde para una serie de casos y una distribución de probabilidad se multiplica la serie de probabilidades donde cada probabilidad es la que correspondiente a la probabilidad de obtener el resultado de un caso de la serie de casos. Así, cuanto más alto sea el resultado de la función de verosimilitud, más plausible es que dicha función de probabilidad genere la salida obtenida (la sucesión de casos). Interesa encontrar los parámetros para la red bayesiana que sean máximo verosímiles respecto al

conjunto de datos desde el que se extrae dicha red. El interés es calcular las probabilidades a priori de las variables y las probabilidades condicionadas de cada variable a sus padres que son máximo verosímiles. Primero se asume un conocimiento a priori, modelado como una distribución de probabilidad por cada variable. Si este conjunto de distribuciones, cumplen con la propiedad de ser de Dirichlet, el problema tiene una resolución analítica, permitiendo encontrar respuesta a la búsqueda de probabilidades antes planteada. La respuesta es un cociente que contiene cuatro términos, dos sumando en el numerador y dos sumando en el denominador. Así pues, cuando se quiera calcular la probabilidad de que una variable tome un valor concreto o la probabilidad condicionada de que una variable tome un valor concreto condicionada a valores concretos de sus padres se recurrirá a esta fórmula. En ella, uno de los términos del numerador es igual al número de casos de la base de datos que se tiene, que cumplen con ciertas restricciones. Las restricciones son que tomen en las variables que describen la probabilidad que se está calculando, los valores específicos que toman en la descripción de dicha probabilidad. Uno de los términos del denominador es igual al número de casos de la base de datos que se tiene, que cumplen con ciertas restricciones. Las restricciones son que tomen en las variables que condicionan en la probabilidad que se está calculando, los valores específicos que toman en la descripción de dicha probabilidad. Los dos términos restantes, el del numerador representa lo mismo que el primer término del numerador, y el del denominador representa lo mismo que el segundo término del numerador, pero en un entorno virtual. Es decir, es como si ya se hubiera visto otra serie de muestras con otra serie de resultados. Estos dos términos pueden ayudar a eliminar problemas que se presentan cuando la base de datos que se tiene es pequeña.

En una red bayesiana, dado un conjunto de casos, se pueden extraer los parámetros máximo verosímiles de probabilidades y probabilidades condicionadas. En el caso de calcular la probabilidad máximo verosímil de una variable para un valor concreto, será máximo verosímil el cociente entre el número de casos que presentan ese valor concreto en esa variable concreta sobre el total de casos de la base de datos. En el caso de calcular la probabilidad máximo verosímil de una variable condicionada a sus padres, será máximo verosímil el cociente entre el numerador, que será el número de casos que presentan el valor concreto en esa variable concreta y que además presentan en las variables que son condicionantes en la probabilidad los valores concretos que toman en dicha probabilidad, y el denominador que será el número de casos que presentan en las variables que son condicionantes en la probabilidad los valores concretos que toman en dicha probabilidad.

Ahora se habla de la determinación de la estructura de la red. Los métodos que determinan una estructura de red se pueden agrupar en tres tipos distintos. El primer grupo se basa en la detección de independencias condicionales entre variables, el segundo grupo se basa en la búsqueda por resultados (se busca la estructura que otorgue mejores resultados, buscando en el espacio de posibles redes mediante heurísticos o métricas de evaluación), y el último grupo recogería las técnicas que hibridan los métodos empleados en los dos grupos anteriores.

En las técnicas del primer grupo se puede requerir información adicional, como ordenación entre las variables, o el tipo de estructura deseada. De entre todos los detalles a tener en cuenta, cabe destacar que al testear independencias condicionales, la complejidad del test crece exponencialmente con respecto al orden de las independencias condicionales, donde el orden es el número de variables condicionantes dentro del test de independencia. La función del test de



independencia está basada en la función de información mutua de dos variables condicionadas a un conjunto de variables. Conviene explicar cuando dos DAG (directed acyclic graph) son equivalentes, ya que al emplear la independencia condicional para calcular la estructura de la red lo más probable es que se obtenga un DAG parcial, donde no todos los arcos son dirigidos, y de él se pueden extraer DAGs que presentan diferencias visuales pero que son equivalentes (son representantes de la misma clase). Dos DAG son equivalentes si la independencia condicional de dos conjuntos de variables condicionados a un tercer conjunto se presenta en uno de los DAGs, implica que se presenta en el otro. Sin embargo a la hora de observar los DAGs dibujados, probablemente sea más sencillo atender a la siguiente definición de equivalencia. Dos DAGs son equivalentes si y solo si tienen los mismos arcos (sin dirección) y presentan los mismos patrones cabeza-cabeza (el patrón cabeza-cabeza es que en una terna de variables, dos de ellas no tienen arco entre sí y ambas son padres de la que queda). El algoritmo PC, es una solución para generar un DAG parcial, en la cual se parte de un grafo completo no dirigido, se itera desde cero hasta un valor arbitrario, de uno en uno. Dentro del ciclo se itera hasta que todos los pares de variables ordenados (el test es conmutativo sobre el par de variables elegidas, por lo que no tiene sentido hacerlo dos veces) que presenten una conexión hayan sido testeados. En el interior de este bucle se escogen un número de variables vecinas a las dos variables anteriores igual al valor del contador del bucle externo (de ser posible). Entonces si las dos variables que afrontan el test son condicionalmente independientes dado ese vecindario, se borra el arco entre ambas.

En las técnicas de búsqueda por resultados (score+search) existen diferentes tipos de métricas de la bondad de la red. Una métrica es el logaritmo de la verosimilitud penalizado (penalized log-likelihood). El valor de esta métrica sin penalización será igual a la suma de para todas las variables, y dentro de cada variable para cada posible estado, y dentro de cada estado para cada posible estado de los padres de dicha variable (un sumatorio triple), el logaritmo de la probabilidad máximo verosímil de que la variable que especifica el índice de la primera suma esté en el estado concreto que especifica el índice de la segunda suma dado que sus padres están en la configuración que especifica el índice de la tercera suma, a la potencia del número de casos de la base de datos que cumplen con la restricción de presentar en la variable especificada por el índice de la primera suma el valor especificado por el índice de la segunda suma dado que las variables padre de dicha variable presentan la configuración especificada por el índice de la tercera suma. A continuación se muestra la fórmula que define el logaritmo de la verosimilitud sin penalización.

$$\log P(cc : dag, \theta) = \sum_{i=1}^n \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \log \left( (\theta_{ijk})^{N_{ijk}} \right)$$

**Fórmula 4-11 Logaritmo de la verosimilitud sin penalización**

En la anterior fórmula  $cc$  representa el conjunto de casos dado,  $dag$  representa la red bayesiana para la cual se está calculando el logaritmo de la verosimilitud sin penalización,  $\theta$  es el vector de probabilidades de la red,  $\theta_{ijk}$  es la probabilidad específica de que la variable  $i$  se encuentre en el estado  $j$  dado que sus padres se encuentran en el estado  $k$ , y  $N_{ijk}$  es el número de casos de  $cc$  en los cuales la variable  $i$  se encuentre en el estado  $j$  dado que sus padres se encuentran en el estado  $k$ . El problema de esta métrica es que es monótona creciente respecto a la complejidad de la red, lo que puede inducir a un sobreentrenamiento que aleje a la red de la capacidad

generalizadora que se espera de ella. Por ello se introduce una penalización, que es una resta al término anterior. Dicha resta es el producto de la dimensión del modelo por una función de penalización no negativa. La dimensión del modelo se comprende como la suma de para todas las variables el producto de los distintos valores menos uno que puede tomar la variable especificada por el índice de la suma por el número de distintas posibles configuraciones que pueden tomar sus padres. Se han propuesto varias funciones para la función de penalización no negativa antes mencionada, de las que se destacan la del criterio de Akaike, donde siempre toma valor uno, y la del criterio de información bayesiana, donde toma valor igual a la mitad del logaritmo del número de casos existentes en la red. Otra métrica es la selección del modelo bayesiano (Bayesian model selection) en la cual se busca la estructura de red que es máximo verosímil dados los datos (es decir, la red que tiene mayor probabilidad condicionada dados los datos, máxima verosimilitud marginal). Este máximo es proporcional al producto entre la probabilidad de la red (probabilidad a priori sobre el conjunto de estructuras) por la probabilidad condicionada de los datos a la red (verosimilitud marginal de los datos). Si la probabilidad de la red es uniforme (el máximo que alcanza la probabilidad condicionada de la red a los datos es igual al máximo que alcanza la probabilidad condicionada de los datos a la red), se puede intentar encontrar la red con máxima verosimilitud marginal. K2 es un algoritmo que plantea una solución a la búsqueda de la red que es máximo verosímil dados los datos. Para la búsqueda de una resolución analítica que deje una formula cerrada para la probabilidad condicionada de los datos a la red, existe la métrica K2, la cual parte de la premisa de que si la probabilidad condicionada de los parámetros de la red a la red es uniforme, es posible encontrar una solución analítica. También existe la métrica BDe (Bayesian-Dirichlet equivalence) la cual parte de la premisa de que si la probabilidad condicionada de los parámetros de la red a la red sigue una distribución de Dirichlet, es posible hallar una resolución analítica.

Ahora se hablará de soluciones concretas en la construcción de la red. Aunque previamente cabe recordar que una de las respuestas que se pretende otorgar, es cuál es la configuración de valores para un conjunto de variables (variables objetivo) que hace máximo verosímil la probabilidad condicionada de que dicho conjunto de variables tome dicha configuración de variables condicionada por otro conjunto de variables que toma una configuración de valores específica. Es decir, en base a una serie de evidencias en una serie de características de un problema, se quiere saber en otra serie de características de valor desconocido cuál es su configuración de valores más probable dadas las evidencias anteriores. El valor de esta probabilidad condicionada es proporcional al producto entre la probabilidad a priori de que las variables objetivo tomen dicha configuración por la probabilidad condicionada de que las variables evidencia tomen la configuración que han tomado condicionado a que las variables objetivo toman la configuración antes mencionada. El problema es que a poco que se incremente el número de variables objetivo y de variables evidencia, el número de parámetros crece inadmisiblemente (pues el crecimiento es exponencial con respecto a los dos números antes mencionados). Incluso si el número de variables objetivo se reduce a uno, el número de parámetros sigue siendo prohibitivo cuando se tiene un número de variables evidencia aun razonable. Así pues no parece factible tener una red completa y hay que simplificar.

### **4.2.4.4 Naïve-Bayes**

Una primera solución es Naive-Bayes, donde se asume que la probabilidad condicionada de que la variable objetivo tome un valor sujeto a la condición de que las variables evidencia tomen una configuración de valores concreto, es igual al producto de la probabilidad de que la variable objetivo tome dicho valor concreto multiplicado por una serie de probabilidades condicionadas. En esta serie de probabilidades condicionadas cada una de ellas estaría descrita por la probabilidad condicionada de que una variable evidencia tome el valor concreto que le corresponde condicionado a que la variable clase tome el valor específico de antes (tomándose cada vez en la serie de probabilidades condicionadas anteriormente mencionada, en cada probabilidad condicionada distinta, una variable evidencia distinta). Así Naive-Bayes clasificara un caso que le entra con una configuración de valores concreta en sus variables evidencia, con el valor en la variable objetivo que maximiza el producto anteriormente descrito. Cabe denotar, que con el procedimiento empleado en el clasificador Naive-Bayes, se está asumiendo que todas las variables evidencia son independientes entre sí, cosa que puede realmente no ser cierta. Este hecho en problemas en donde las variables evidencia tienen una dependencia entre sí suficiente, puede llevar a la red construida con Naive-Bayes a otorgar resultados pobres. En Naive-Bayes, la estructura de la red tiene un arco de la variable objetivo a cada una de las variables evidencia, donde la variable objetivo siempre es el padre.

### **4.2.4.5 Seminaïve-Bayes**

En el clasificador Seminaive-Bayes donde existe una alternativa a considerar todas las variables evidencia como condicionalmente independientes. A la hora de generar la red, Seminaive-Bayes parte de un grafo en el que sólo está el nodo de la variable objetivo (variable clase o clase), y clasifica todos los casos con el valor moda de la variable clase. Entonces mientras la tasa de aciertos en clasificación mejora, se sigue un proceso iterativo. En él, en cada iteración se elige la opción que más incrementa la tasa de aciertos de entre las dos siguientes. En la primera se considera cada variable que no está presente en el grafo, como candidata para ser añadida como condicionalmente independiente de todas las variables evidencia presentes en el grafo, y como hija de la variable objetivo. En la segunda se considera cada variable que no está presente en el grafo, como candidata para formar una variable compuesta con cualquier variable evidencia que sea simple o compuesta, y que la nueva variable compuesta preserve las conexiones en el grafo de la variable evidencia que formaba parte del grafo que ha desaparecido para dejarle lugar a esta nueva variable compuesta. La generación de una variable compuesta es la generación de un nuevo conjunto que se define como el producto cartesiano de los conjuntos de las variables que la componen.

### **4.2.4.6 TAN**

TAN (Tree augmented Naive-Bayes) es un clasificador bayesiano que basa la generación de la estructura de la red en la función de información mutua entre dos variables condicionadas a un conjunto de variables, procedente de la teoría de la información. En TAN lo primero que se hace es calcular la información mutua entre cada par de variables evidencia condicionadas a la variable clase. Entonces se construye un grafo no dirigido que contenga a todas las variables evidencia (o variables predictoras), que aún no contiene arcos. Ahora resta añadir un número de arcos igual al

número de variables predictoras menos uno. Entonces se itera hasta que se añade dicho número de arcos. En cada iteración, se añade el arco entre el par de variables que mayor información mutua condicionada a la variable clase ofrezcan (que ya se calculó antes), siempre que dicho arco no exista ya, porque tal vez se añadió en iteraciones anteriores, y siempre que dicho arco no genere un ciclo en el grafo (en cualquiera de esos dos casos, se obvia ese arco como candidato y se busca el siguiente que tenga mayor valor). Cuando se termina el proceso iterativo, se transforma el grafo en dirigido, escogiendo alguna de las variables predictoras como raíz, y haciendo que todos los arcos conduzcan a ella. Finalmente se añade un nodo que representa la variable clase al grafo, y se añade un arco de ella a cada una de las variables predictoras, donde la variable clase juega el papel de padre.

#### **4.2.4.7 *k-DB***

*k-DB* es un clasificador bayesiano en el que se requiere elegir un valor entero positivo  $k$  (no es necesario escogerlo antes de comenzar el proceso de generación de la red, pues podría ser interesante hacer variantes del algoritmo *k-DB* donde el valor  $k$  se determine en tiempo de ejecución). En *k-DB* primero se calcula la información mutua entre cada variable predictora y la variable clase, y la información mutua entre cada par de variables predictoras condicionadas a la variable clase. Se comienza con un grafo que solo tiene el nodo de la variable clase. En cada iteración se añade la variable que mayor información mutua tiene con la variable clase (y que aún no está en el grafo) y se añade un arco de ella a la variable clase, donde la variable clase es el padre. Además, también se añaden  $k$  arcos entre la variable que se acaba de añadir (siempre como hija) y las  $k$  variables de mayor valor en la información mutua entre ellas y la variable que se acaba de añadir condicionadas a la variable clase (esto se ha calculado al principio).



## 5 Imágenes de sonar

### 5.1 El sonido

El sonido es una onda longitudinal (la perturbación es paralela a la dirección de la propagación) mecánica (P. A. Tipler, 2005). Dichas ondas se propagan por las moléculas de un gas, líquido o sólido, que oscilan según la dirección de la propagación. La velocidad de la propagación del sonido depende de las características de la materia por la que se propaga, como por ejemplo el estado de agregación de la misma, su compresibilidad y su densidad. Una propiedad importante del sonido es su intensidad, que se define como la potencia acústica por unidad de área a la dirección de propagación

Denotar la diferencia entre el sonido en sí y el sonido audible por los humanos, pues éste último, es sonido que cumple un conjunto de condiciones necesarias para que un humano lo pueda oír, como por ejemplo que los sonidos audibles por seres humanos deben estar en un intervalo acotado específico de frecuencias o que deben de tener un mínimo de intensidad.

### 5.2 El sonar

El sonar (SOund NAVigation and Ranging), es un aparato que recoge o propaga información mediante el empleo de ondas de acústicas en un medio elástico (su uso principal es en el agua). Se puede distinguir entre sonar pasivo y sonar activo. El sonar pasivo únicamente capta ondas acústicas del medio. El sonar activo además de percibir ondas acústicas, puede emitir sonido. Dicha emisión puede tener el propósito de propagar información a otros sonares o puede tener como objetivo captar el eco de la señal acústica emitida. Teniendo ésta última motivación en mente, la captación del eco de una señal acústica emitida, puede emplearse para tratar de detectar entidades físicas en el espacio y para reconocer características de las mismas como pueden ser su ubicación espacial o su velocidad. En 1915 Paul Langévin inventó el primer sonar para la detección de submarinos, aunque no fue hasta 1918 cuando se desarrollaron los sonares activos. Como se ha mencionado antes, el medio en el que más se emplea el sonar es en el agua. Este hecho es en parte consecuencia de que no se pueda emplear el radar en el agua, pues dada la alta conductividad del agua, la señal se dispersa rápidamente. Sin embargo, el sonar sí ofrece un buen comportamiento en el agua.

El funcionamiento del sonar activo, comienza por la emisión de una señal eléctrica a la cual se le realiza una transducción para obtener la señal acústica que el sonar emite. Entonces la señal acústica se propaga por el medio, hasta que encuentra algún obstáculo y rebota. Entonces el eco que llega al sonar, sufre una transducción, quedando en una señal eléctrica que se puede representar en un monitor. Al ser la velocidad del sonido en el agua constante, se puede medir la diferencia temporal desde que se envía una señal acústica desde el sonar hasta que se recibe un rebote, para posteriormente medir la distancia al objeto que ha provocado el rebote. Lo convencional en sonares activos es emplear señales acústicas inaudibles tanto para humanos como para los seres vivos de la vida marina.

Se pueden destacar cuatro características deseables en un sonar. Una de ellas es tener un transmisor de señales acústicas de gran potencia, incrementando así la probabilidad de recibir el

eco de la señal en aguas profundas y en condiciones de agua adversas. Otra característica es tener un buen transductor, en el sentido de que, aparte de que debe poder soportar la posible alta potencia del transmisor, debe convertir la señal eléctrica en señal acústica con la menor pérdida posible de intensidad, además de que tiene que poder detectar los ecos de baja potencia. Otra propiedad deseable, es que el receptor, al tener que lidiar con un amplio rango de señales, debe reducir las señales entrantes de muy alta potencia y debe amplificar las señales entrantes de potencia muy reducida. Adicionalmente debe distinguir objetos cercanos en impulsos distintos a la hora de representarlo en pantalla. La última característica deseable, es que la pantalla del sonar debería ser de alta resolución y de buen contraste, para que no se pierda detalle.

El transductor de un sonar propaga la señal formando un cono con la base abombada que según se propaga crece por la base. El cono abombado es de tal forma que se incrementa la distancia del vértice del cono al centro de la base con respecto a un cono de base no abombada. El vértice del cono se sitúa en el propio transductor, mientras que la orientación del cono abombado en el espacio depende de la orientación del traductor al emitir la señal. La recta que rige la orientación del transductor atraviesa el vértice del cono y el centro de su base. La intensidad de la señal es más fuerte en la recta que pasa por el vértice y por el centro de la base del cono y se va atenuando según se aleja de esta recta.

Se ha mencionado anteriormente las condiciones de agua adversas, que son un conjunto de condiciones que pueden presentarse en el agua y que afectan a los resultados del sonar, degradándolos. Una de ellas, es que en agua salada, las entidades suspendidas en el agua, afectan a la señal. Como ejemplo de estas entidades, se encuentran los microorganismos como el plancton o las algas, aunque también se pueden encontrar minerales y sales. Además en aguas saladas (casi siempre mares abiertos) el entorno es bastante dinámico por las corrientes de aire y agua y el oleaje. Estos eventos provocan que el aire se mezcle con el agua en la superficie y se puedan encontrar burbujas de aire en el agua cerca de la superficie. Estas burbujas de aire también suponen obstáculos que afectan al rendimiento del sonar. Estos problemas también se dan en aguas dulces, pero su impacto en el rendimiento del sonar es menor.

### **5.2.1 El sonar empleado**

Aquí se comentan algunas particularidades inherentes a la conjunción del modelo de sonar específico con el cual se han extraído las imágenes de sonar con las que se ha trabajado en esta tesina de fin de máster y su configuración en el barco.

El sonar empleado, cumple con la funcionalidad de realizar detecciones a largo alcance, haciendo principalmente referencia a una gran distancia en el plano horizontal. Así pues no es de extrañar que el sonar usualmente esté ligeramente inclinado con respecto al plano horizontal, cabeceando hacia abajo, es decir, apunta ligeramente hacia abajo. El sonar emite en 360º con respecto al eje vertical, por lo que más que ofrecer un cono da un volumen generado por la revolución de un cono generado por el sonar para ángulo cualquiera.

Cabe tener en cuenta la disposición física del sonar en el barco pues aporta información sobre posibles fenómenos que aparecen en las imágenes de sonar generadas. Una consecuencia es, que su disposición física unida al hecho de que el sonar hará barridos con muy poca diferencia de

ángulo con respecto al plano horizontal, puede desembocar ante un fuerte oleaje que incline la embarcación (y por tanto el sonar), que una sección del barrido del sonar llegue a la superficie y rebote en ella, produciendo ruido en la imagen del sonar. Otro evento a tener en cuenta, es que el registro que haga el sonar, también se verá afectado por las intervenciones que realice la embarcación, aparte de su inclinación, como es su propio desplazamiento sobre las aguas. El sistema de propulsión bate el agua, generando por una parte burbujas en el agua que queda detrás de la embarcación, mientras que por otra parte la embarcación deja a su paso una influencia en el oleaje de la superficie, que se puede apreciar muy bien en aguas calmas, donde el único oleaje perceptible es el generado por la propia embarcación.

## 5.3 Las imágenes de sonar

En este apartado se habla de características inherentes al tipo de imágenes de sonar con las cuales se ha trabajado en esta tesis de fin de máster. Seguidamente se muestra una imagen recogida del sonar empleado.

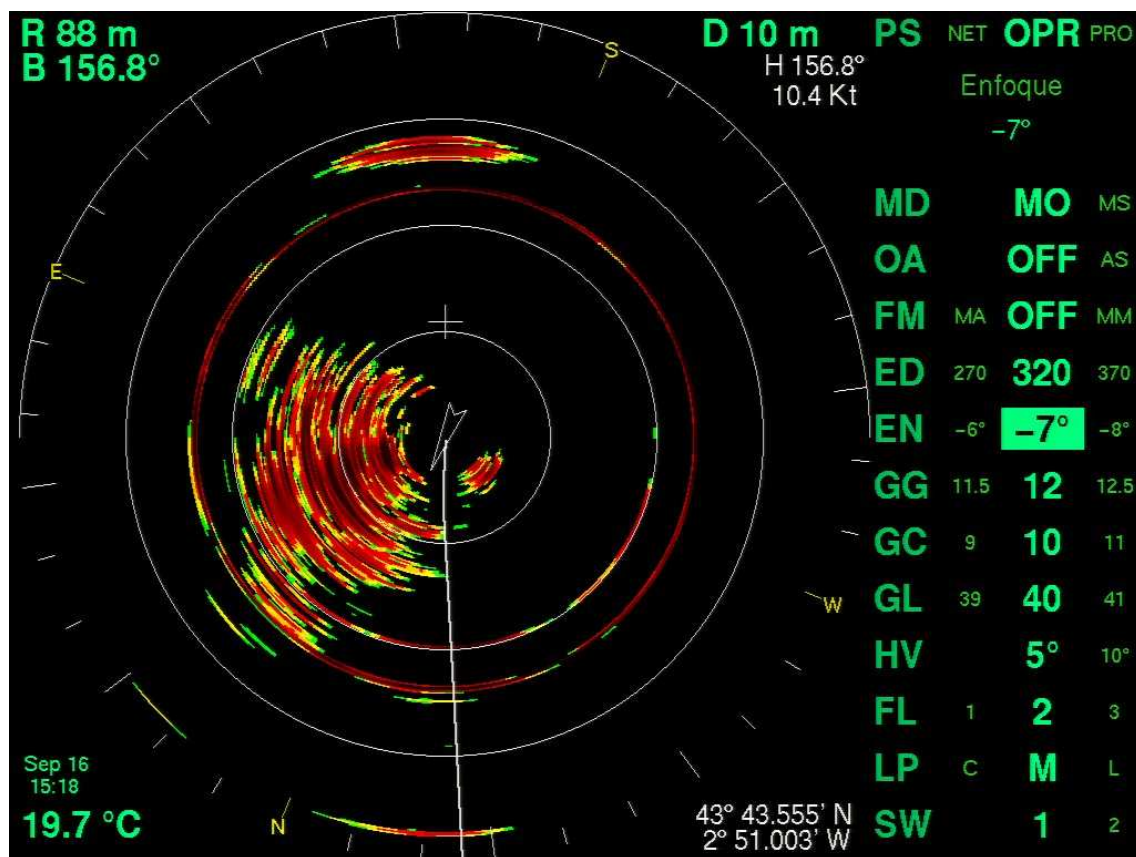


Ilustración 5-1 Ejemplo de imagen generada por el sonar empleado

En la anterior imagen se pueden distinguir dos partes. Por una parte está el círculo que contiene el eco recibido por el sonar en un instante del tiempo específico y un conjunto de agregados gráficos, mientras que por otra parte estaría todo lo no contenido en ese círculo. Lo no contenido en dicho círculo, son mayoritariamente parámetros relacionados con la navegación. De lo contenido en el círculo, indicar cuáles son los agregados gráficos exactamente. Las iniciales de los puntos cardinales ubicadas en los extremos del círculo son agregados gráficos. Toda recta blanca



dentro del círculo que extendiéndose pase por el centro del mismo también es un agregado gráfico. Existe una recta por cada punto cardinal, donde cada una se sitúa en el entorno de una inicial distinta de los puntos cardinales, además de que la prolongación de cada una de esas rectas pasa por el centro del círculo. Dichas cuatro rectas también son un agregado gráfico. Las circunferencias y semicircunferencias de color blanco también son agregados gráficos. La cruz blanca situada ligeramente por encima de la parte alta de la circunferencia blanca de radio más pequeño, también es un agregado gráfico. La línea blanca, que comienza en algún punto de la circunferencia de menor radio que circunscribe al círculo que contiene el eco del sonar, y que termina en el centro de dicho círculo, también es un agregado gráfico. Por último el polígono blanco de cuatro lados situado en el centro, también es un agregado gráfico.

Cuando se ha mostrado la imagen anterior, se ha presentado como “una imagen recogida del sonar empleado”. Es decir, la imagen anterior, tal vez no sea exactamente lo que muestra el sonar, pues al pasar la imagen mostrada por la pantalla del sonar a soporte digital, se lleva a cabo un proceso de captación de imagen. En el desarrollo de esta tesis de fin de máster no se ha investigado sobre dicho proceso, debido a los inconvenientes prácticos y al hecho de centrar los esfuerzos en otras áreas, pero es un hecho a tener en cuenta en posteriores investigaciones.

Ahora se comenta lo que representa lo contenido dentro del círculo que muestra el eco captado por el sonar, que no es agregado gráfico. Es decir, se habla de las manchas de múltiples colores y de formas variadas presentes en dicho círculo. Primero denotar que los píxeles se corresponden con regiones del espacio físico. Los colores guardan una relación con la intensidad del eco percibido, en el sentido de que, existe una función que dado un valor de intensidad del eco, devuelve un color. Así para un píxel dado contenido dentro del círculo para el cual se desea mostrar su color según la intensidad del eco percibido por el sonar en la región del espacio correspondiente a dicho píxel, se aplica la mencionada función sobre la intensidad registrada y se obtiene el color. Sin embargo cabe denotar un hecho. El hecho es que para el desarrollo de esta tesis de fin de máster no se ha investigado sobre dicha función. Los valores de los píxeles de la imagen a color de la captura de pantalla de sonar mostrada anteriormente, suelen estar expresados en un espacio de color tridimensional, como puede ser el espacio RGB. La intensidad del rebote, es una magnitud física que resulta ser un escalar. Así pues la función de intensidad en color desconocida se puede describir parcialmente de la siguiente forma.

$$f: (\{0, \mathbb{R}^+\}) \rightarrow (\mathbb{Z} \cap [0,255], \mathbb{Z} \cap [0,255], \mathbb{Z} \cap [0,255])$$

### **Fórmula 5-1 Función de intensidad en color desconocida**

Donde resulta que los elementos del conjunto de origen tienen una única componente, y los elementos del conjunto de destino tienen tres componentes. De la anterior descripción de la función se deduce que no es una función inyectiva. Se conoce que el intervalo sin discontinuidades más pequeño que contiene a todos los valores de intensidades que tienen como imagen un mismo color, no contiene valor alguno de intensidad que tenga como imagen un color distinto. A pesar de la carencia de la propiedad inyectiva de la función, teniendo en cuenta lo enunciado en la última frase se podrían reordenar los colores del conjunto de destino de forma que los intervalos de intensidades que generan los distintos colores estuvieran ordenados de mayor a menor, o viceversa. En este contexto se entiende que un intervalo es mayor que otro, si

todos los elementos de dicho conjunto son al menos no inferiores al valor más alto del segundo conjunto. Análogamente, se entiende que un intervalo es menor que otro si todos los elementos de dicho conjunto son al menos no mayores que el valor más bajo del segundo conjunto. Para llevar a cabo la reordenación antes mencionada, se debería conocer la función antes mencionada. Llevar a cabo dicha reordenación es interesante pues permitiría decidir qué color es mayor que otro, en el sentido que las intensidades que representa el primero, son al menos no inferiores a las que representa el segundo. Por ello este es un tema a plantear en investigaciones posteriores. Sin embargo, aunque no se tiene un conocimiento completo de qué color es “mayor” que qué otro color, se tiene una descripción intuitiva de la ordenación de los colores. Los valores de intensidad más bajos, vienen representados por el color negro. Aumentando la intensidad, los siguientes colores que se encuentran son aquellos situados en el entorno del amarillo, si se sigue incrementando entonces se encuentran los entornos de verdes, si se incrementa más entornos de rojos y si aún se incrementa más se encuentran entornos de rojos oscuros.



### 6 Diseño de experimentos

En este apartado se habla de las pruebas realizadas, es decir, de qué se ha hecho. Se ha buscado bibliografía para la problemática específica empleada, pero no se ha logrado encontrar información accesible con respecto al área del procesamiento de señales de atún mediante sonar de largo alcance. Por tanto lo aquí elaborado parte de cero en ese aspecto.

Indicar que los ensayos se realizan con las ayudas de dos productos software. Uno es la aplicación Sonar presentada como parte de la aportación de esta tesis de fin de máster, de la cual se incorpora un manual en el CD asociado a la memoria. Otra es la aplicación Weka, de libre distribución bajo licencia GNU. Se puede localizar en la dirección url localizada en la referencia bibliográfica (WekaPrincipal), se puede descargar directamente de la url localizada en la referencia bibliográfica (WekaDescarga) y se puede encontrar documentación relativa a la aplicación Weka (como tutoriales, APIs o FAQs) en la dirección url localizada en la referencia bibliográfica (WekaDocumentación).

Comentar que si bien en las imágenes originales vienen dadas en formato con extensión “.jpg”, en todos los procesos en los que se manipulan imágenes se trabaja con formato con extensión “.bmp”, salvo en las imágenes de entrada a los preprocesamientos, que serán de extensión “.jpg”. Esto se debe a la naturaleza de las segmentaciones empleadas, pues se estima que su rendimiento caería bastante de emplearse formatos de imagen con pérdida.

Primero de todo, comentar que se han empleado tres secuencias de imágenes de sonar. Se bautizan como “secuencia con atún”, “secuencia vacía” y “secuencia con ruido de superficie”. En la “secuencia con atún” predominan las imágenes en las cuales se presenta un banco de atún y figura un ruido moderado. Con el empleo de juicio experto, en dicha secuencia de 31 imágenes, se identifican 28 en las que hay presencia de un banco de atún, mientras que en las otras dos secuencias no se presenta banco de atún alguno. A continuación se muestra una imagen típica de esta secuencia, en la que la mancha colorada que figura más arriba es el banco de atún y el resto de manchas se corresponden con ruido.

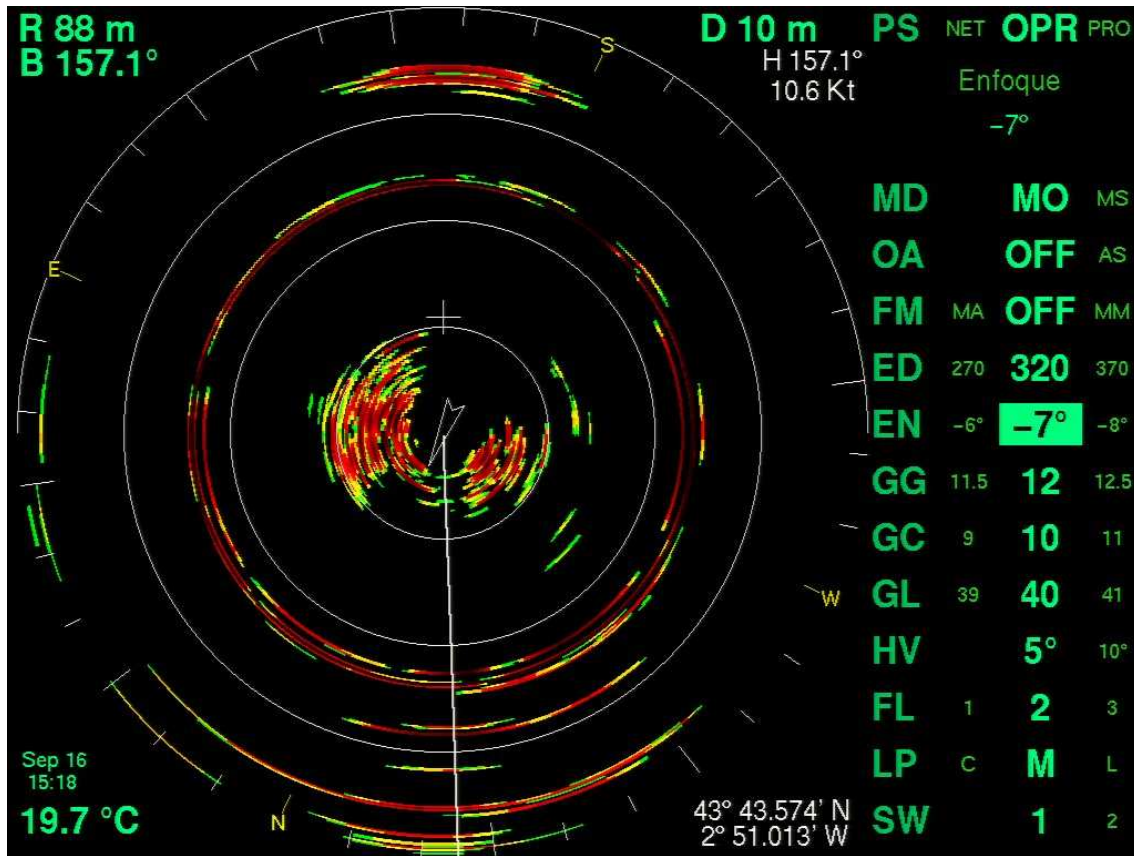


Ilustración 6-1 Imagen típica de la “secuencia con atún”

En la “secuencia vacía” predominan las imágenes en las cuales no hay presencia de bancos de atún y apenas hay ruido. Es una secuencia de 41 imágenes. A continuación se muestra una imagen típica de esta secuencia.

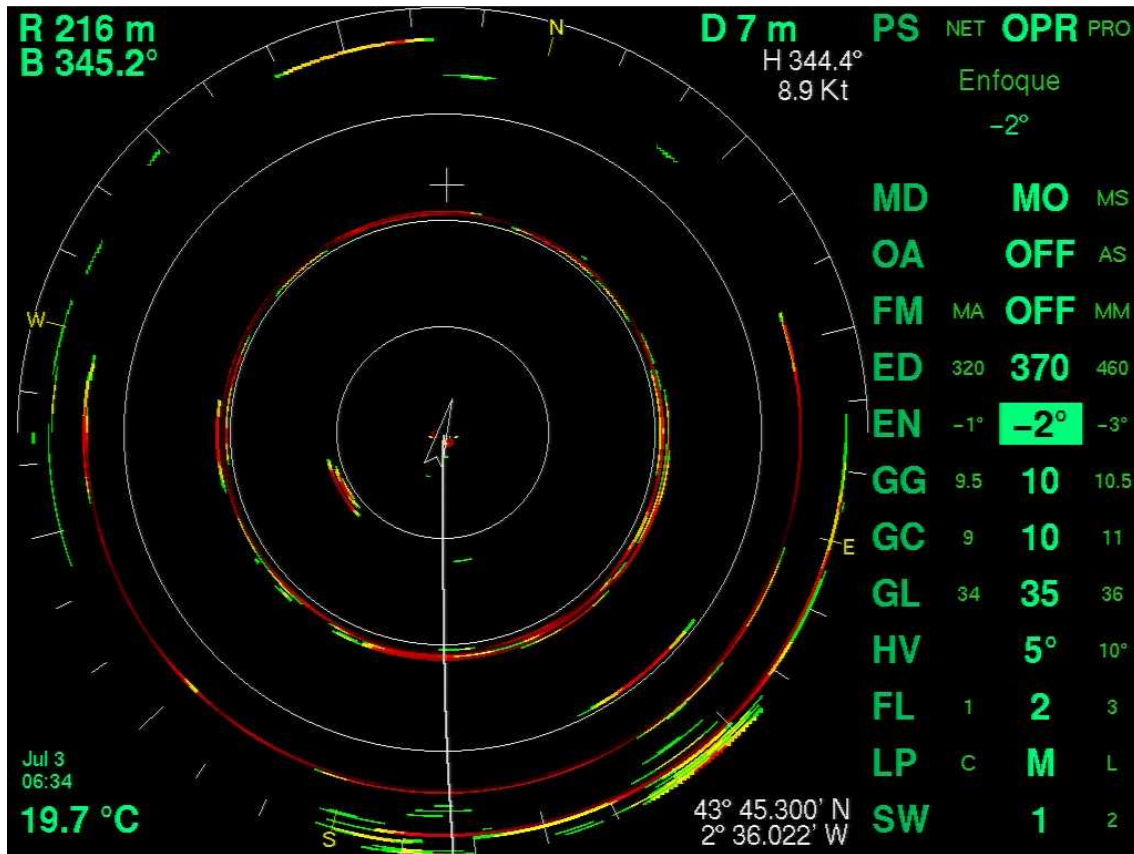


Ilustración 6-2 Imagen típica de la “secuencia vacía”

En la “secuencia con ruido de superficie” predominan las imágenes en las cuales no hay presencia de bancos de atún y existe un ruido de la superficie considerable. Es una secuencia de 67 imágenes. A continuación se muestra una imagen típica de esta secuencia.

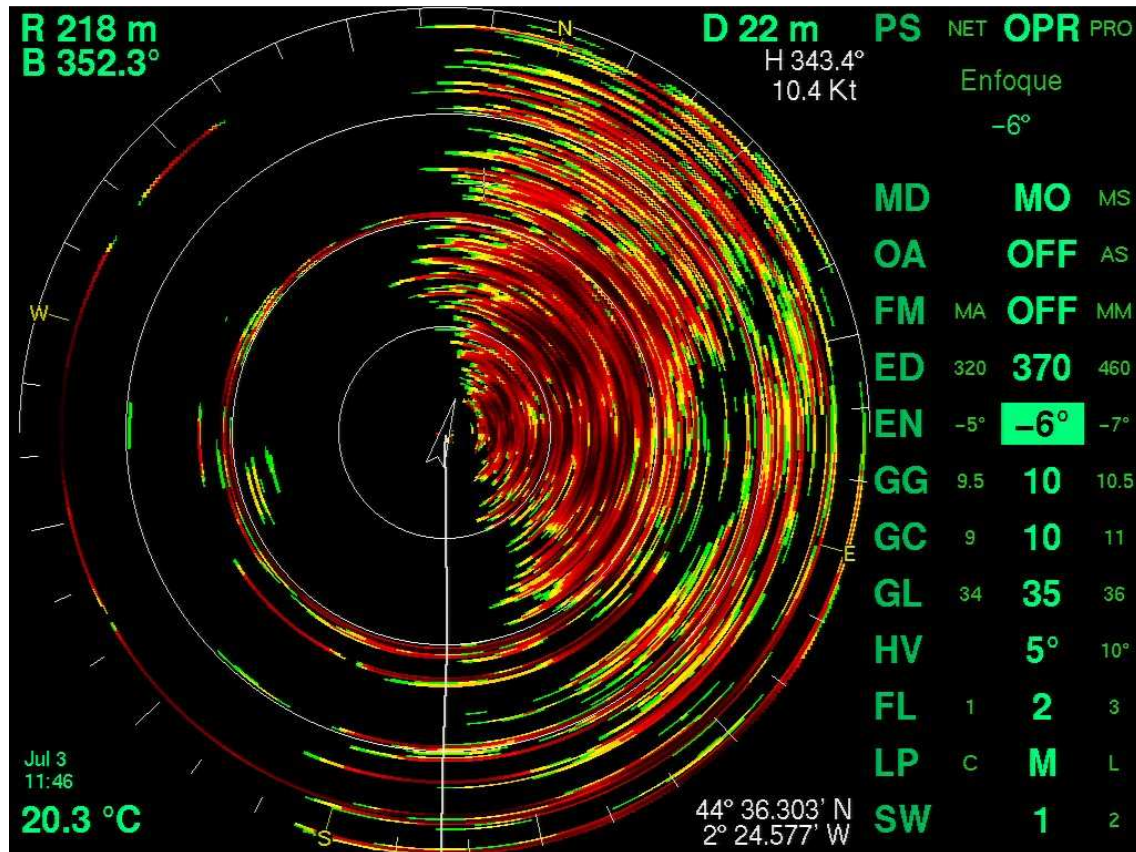


Ilustración 6-3 Imagen típica de la “secuencia con ruido de superficie”

Comentar que se ha trabajado con la mitad superior del círculo del sonar. Esto se debe a la idea de ordenar la investigación tratando de realizar primero las tareas más sencillas para luego ir aumentando la complejidad de los métodos empleados. Realmente resulta más sencillo trabajar con la mitad superior del círculo del sonar que con todo el círculo del sonar, la gran mayoría del ruido usualmente se encuentra en la mitad inferior del círculo del sonar. Además lo más frecuente es que el atún se localice en el sonar según aparece por el frente (en el círculo el frente se puede describir por el vector director que va desde el centro del círculo del sonar al punto más alto de la circunferencia mínima que lo circunscribe) o en entornos de la dirección frontal, debido a la diferencia de velocidad habitual existente entre la embarcación y el atún.

Cabe comentar que se considera un proceso completo aquel que incorpora el preprocesamiento, la segmentación, la extracción de características, el etiquetado de instancias y el algoritmo de aprendizaje. Como el etiquetado de instancias (mediante juicio experto) es el mismo para todos los procesos completos empleados, la mención de éste se omite en la descripción de un proceso completo particular en este documento. Así pues, un proceso completo específico, viene definido por el preprocesamiento, la segmentación, la extracción de características y el algoritmo de aprendizaje específicos empleados. Tanto los preprocesamientos, como las segmentaciones, como las extracciones de características aquí empleadas carecen de nombre propio y en lugar de nombre tienen un identificador numérico. Por ello dentro de proceso completo específico se puede aludir a su preprocesamiento, su segmentación y su extracción de características de forma bastante resumida. Así pues, en lugar de decir que el proceso completo específico *PC* emplea el preprocesamiento 1, la segmentación 2, la extracción de características 3 y el algoritmo de

aprendizaje  $X$ , se puede abreviar mediante la expresión “el proceso completo 1 2 3  $X$ ”. Por ejemplo, si un proceso completo específico emplea el preprocesamiento 1, la segmentación 2, la extracción de características 3 y el algoritmo de aprendizaje JRip, se le puede aludir como “proceso completo 1 2 3 JRip”. De forma análoga, para procesos parciales, en los que tan sólo se especifica el preprocesamiento, la segmentación y la extracción de características, se puede realizar una alusión resumida igual que la anterior, salvo que el nombre del algoritmo de aprendizaje no figurará al final.

Recordar que para realizar los procesos completos empleados en esta tesis de fin de máster, se han empleado las dos aplicaciones software antes mencionadas Sonar y Weka. Mencionar que Weka es una herramienta software que aborda la problemática de la minería de datos y el aprendizaje automático, que permite examinar problemas de clasificación supervisada. Para la realización de los preprocesamientos, segmentaciones y extracciones de características, se emplea el software Sonar. Para el etiquetado de instancias, primero se distinguen los distintos tipos de etiquetas y se decide qué etiqueta corresponde a cada instancia mediante ayuda de juicio experto. Para el desarrollo del etiquetado de instancias, en la aplicación Sonar se dispone de la funcionalidad de etiquetado asistido que facilita esta tarea. Para el empleo de algoritmos de aprendizaje, se hace uso de la aplicación Weka. Realmente tras la etapa de extracción de características y la etiquetación de instancias, el producto entrante para la etapa de aplicación de algoritmos de aprendizaje es una base de datos que resulta ser un fichero con extensión “.arff” que Weka es capaz de manejar, para realizar experimentos. Mencionar que la sintaxis de los ficheros “.arff” se puede encontrar en la dirección url localizada en la referencia bibliográfica (Arff). Para la experimentación con Weka, se ha empleado la interfaz “experimenter” de dicha aplicación, programando en ella los experimentos pertinentes con cada base de datos entrante.

En esta tesis de fin de máster se realizan tres experimentos. En el primer experimento se aplican los procesos completos 1 1 1 ZeroR, 1 1 1 IBk, 1 1 1 JRip, 1 1 1 J48, 1 1 1 NaiveBayes, 1 1 1 BayesNet, 2 1 1 ZeroR, 2 1 1 IBk, 2 1 1 JRip, 2 1 1 J48, 2 1 1 NaiveBayes, 2 1 1 BayesNet, 3 1 1 ZeroR, 3 1 1 IBk, 3 1 1 JRip, 3 1 1 J48, 3 1 1 NaiveBayes, 3 1 1 BayesNet, 4 1 1 ZeroR, 4 1 1 IBk, 4 1 1 JRip, 4 1 1 J48, 4 1 1 NaiveBayes, 4 1 1 BayesNet, 5 1 1 ZeroR, 5 1 1 IBk, 5 1 1 JRip, 5 1 1 J48, 5 1 1 NaiveBayes y 5 1 1 BayesNet sobre la secuencia de imágenes de sonar llamada “secuencia con atún”. Es decir, se aplican cinco procesos parciales, 1 1 1, 2 1 1, 3 1 1, 4 1 1 y 5 1 1 sobre la secuencia de imágenes llamada “secuencia con atún” que junto a las etiquetaciones de instancias dejan como producto cinco bases de datos en ficheros de extensión “.arff”. Después con ellas se experimenta con la interfaz “experimenter” de Weka con los algoritmos de aprendizaje ZeroR, IBk, JRip, J48, NaiveBayes y BayesNet. Tras ejecutar todos esos procesos completos sobre ese conjunto de imágenes, se contrastan los resultados obtenidos entre los distintos procesos completos empleados en este experimento. En el segundo experimento se aplican los procesos completos completos 5 1 1 ZeroR, 5 1 1 IBk, 5 1 1 JRip, 5 1 1 J48, 5 1 1 NaiveBayes, 5 1 1 BayesNet, 5 2 1 ZeroR, 5 2 1 IBk, 5 2 1 JRip, 5 2 1 J48, 5 2 1 NaiveBayes, 5 2 1 BayesNet, 5 3 1 ZeroR, 5 3 1 IBk, 5 3 1 JRip, 5 3 1 J48, 5 3 1 NaiveBayes y 5 3 1 BayesNet sobre el conjunto de imágenes compuesto por las tres secuencias, a saber, “secuencia con atún”, “secuencia vacía” y “secuencia con ruido de superficie”. En esta ocasión son los procesos parciales 5 1 1, 5 2 1 y 5 3 1 empleados sobre el conjunto de imágenes que contiene las tres secuencias los que dejan junto con el etiquetado de instancias tres bases de datos en ficheros “.arff”. Como en el experimento 1, con las bases de



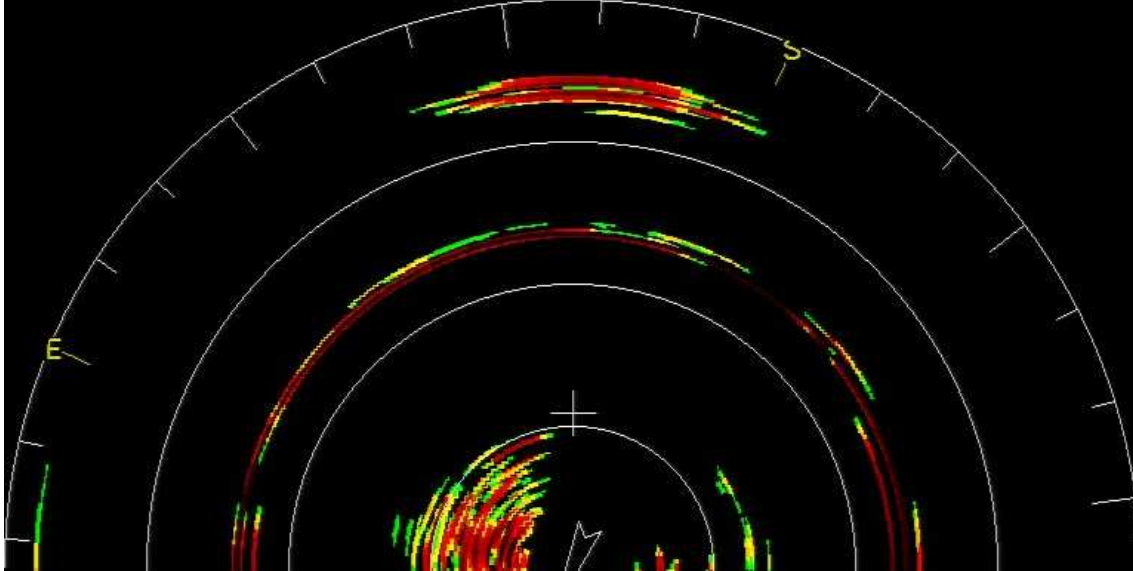
datos en ficheros “.arff” se experimenta con la interfaz “experimenter” de Weka con los algoritmos de aprendizaje ZeroR, IBk, JRip, J48, NaiveBayes y BayesNet. Igual que en el experimento anterior, tras realizar la experimentación se comparan los resultados obtenidos para los distintos procesos empleados. En el tercer experimento se aplican los procesos completos 5 3 1 ZeroR, 5 3 1 IBk, 5 3 1 JRip, 5 3 1 J48, 5 3 1 NaiveBayes, 5 3 1 BayesNet, 5 3 2 ZeroR, 5 2 1 IBk, 5 3 2 JRip, 5 3 2 J48, 5 3 2 NaiveBayes, 5 3 2 BayesNet, 5 3 3 ZeroR, 5 3 3 IBk, 5 3 3 JRip, 5 3 3 J48, 5 3 3 NaiveBayes y 5 3 3 BayesNet sobre el conjunto de imágenes compuesto por las tres secuencias, a saber, “secuencia con atún”, “secuencia vacía” y “secuencia con ruido de superficie”. Aquí el empleo de los procesos parciales 5 3 1, 5 3 2 y 5 3 3 sobre el conjunto de imágenes que contiene a las tres secuencias junto con el etiquetado de instancias, dejan como producto tres bases de datos en ficheros “.arff” (aunque cabe tener en cuenta que la base de datos generada por 5 3 1, ya fue generada en el experimento anterior, por lo que se puede reutilizar). Aquí de nuevo, igual que en anteriores experimentos se experimenta con la interfaz “experimenter” de Weka con los algoritmos de aprendizaje ZeroR, IBk, JRip, J48, NaiveBayes y BayesNet sobre las tres bases de datos entrantes. Nuevamente al concluir la experimentación se contrastan los resultados de los procesos completos empleados en este experimento.

Cabe plantearse por qué no se comparan todos los procesos completos mencionados anteriormente en un único experimento. No se lleva a cabo la experimentación de esta forma debido a dos motivos. Uno de ellos es que puede suceder que en un proceso completo dado, la primera etapa, el preprocesamiento, deje como producto un conjunto de imágenes que resulte inviable tratar en las etapas posteriores. Es decir, que puede suceder que un preprocesamiento dado, genere como producto un conjunto de imágenes que provocarán problemas en las etapas dadas posteriores. Por ejemplo puede que un preprocesamiento dado sea inadecuado y favorezca en demasía que la segmentación identifique demasiados segmentos y que la extracción de características tenga que trabajar para cada uno de ellos, además de generar una base de datos para Weka de tamaño considerable, lo que puede suponer un problema en términos de tiempo. El otro motivo, es que una segmentación inadecuada puede desembocar en un problema con las mismas consecuencias que el anterior. Por ello, en el primer experimento se trata de encontrar el preprocesamiento más adecuado operando sobre un conjunto de imágenes relativamente pequeño (sobre una sólo de las tres secuencias). Por la misma razón, en el segundo experimento se trata de encontrar la segmentación más adecuada, sin variar el preprocesamiento ni la extracción de características empleada. Finalmente en el tercer experimento se trata de encontrar la extracción de características más adecuada, para por último determinar dados el preprocesamiento, la segmentación y la extracción de características más adecuada cual es el clasificador que mejores prestaciones ofrece.

Lo siguiente es, comentar en qué consiste cada etapa. Así pues, primero se explican los preprocesamientos empleados, después las segmentaciones y tras esto las extracciones de características. Cuando se termine de comentar todo lo anterior, se comentará el etiquetado realizado con ayuda de juicio experto. Finalmente se habla de los algoritmos de aprendizaje empleados

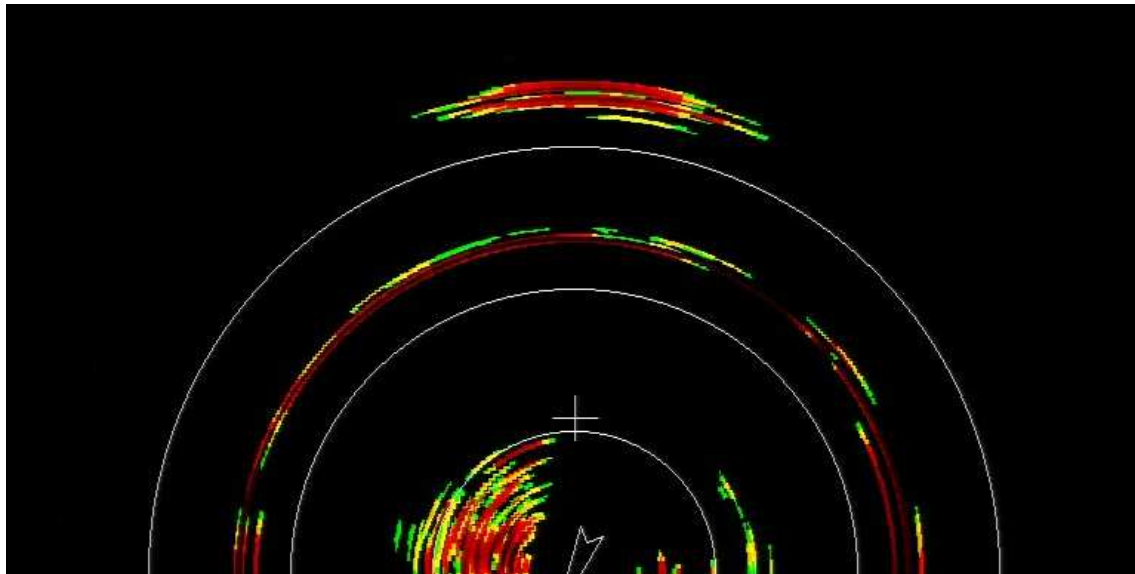
El preprocesamiento 1, realiza las siguientes tareas. Para una imagen entrante al preprocesamiento 1, éste primero recorta la imagen quedándose con el cuadrado mínimo que contiene la mitad superior del círculo del sonar. Después cambia el valor de todos los píxeles que

estén a una distancia (distancia euclídea) del centro del círculo del sonar mayor a cierta distancia dada, de forma que el nuevo valor representará el color negro (en un espacio RGB las tres componentes valdrán cero). La cierta distancia dada, es la distancia existente del centro del círculo del sonar a cualquier punto de la circunferencia mínima que lo circunscribe. A continuación se muestra una imagen resultante de este preprocesamiento como ejemplo.



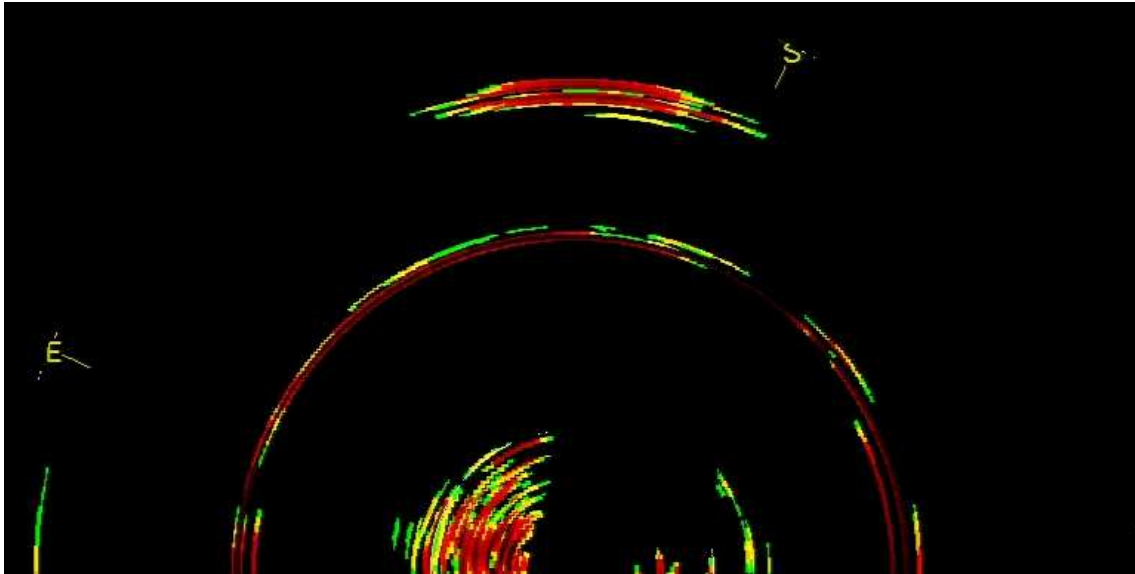
**Ilustración 6-4 Ejemplo de preprocesamiento 1**

El preprocesamiento 2, es muy similar al preprocesamiento 1, pues la única diferencia estriba en la cierta distancia dada, que rige qué píxeles preservarán su valor y qué píxeles pasarán a representar con su valor el negro. En esta ocasión, la cierta distancia dada, es menor que en la del preprocesamiento 1. Concretamente, es la mayor distancia que permite eliminar los píxeles (cambiar su valor a negro) pertenecientes al gran semicírculo, las rectas de color blanco que intersectan con él, la representación visual de las iniciales de los puntos cardinales y la recta asociada a cada uno de ellos. Esto implica que los registros del sonar que estén tan alejados del centro como los agregados gráficos que se pretenden eliminar en este preprocesamiento, se perderán. A continuación se muestra una imagen resultante de este preprocesamiento como ejemplo.



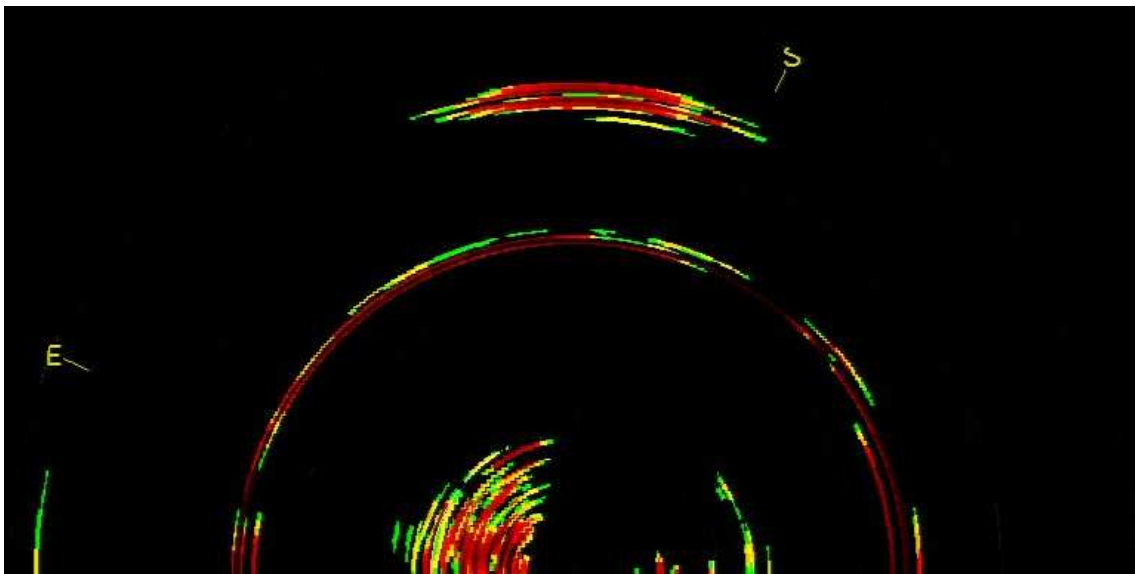
**Ilustración 6-5** Ejemplo de preprocesamiento 2

El preprocesamiento 3, realiza las operaciones que realiza el preprocesamiento 1, además de realizar una operación adicional. Dicha operación adicional, consiste en eliminar todos los píxeles cuyo valor se encuentre en entornos de la escala de gris. Es decir, del negro al blanco en el espacio RGB se examinan todos los niveles de gris, y se ve para un píxel dado, si su valor está en el entorno de alguno de ellos. Caso de estar en el entorno de alguno de ellos, el píxel pasará a tener valor negro. La discriminación de si un píxel está en algún entorno de gris, se realiza mediante un valor que actuará de umbral, y una función de distancia. El valor que actúa de umbral en el preprocesamiento 3 es un valor igual a 6. La función de distancia empleada, toma dos valores de píxeles y mide su distancia. La función de distancia empleada en el preprocesamiento 3, es una distancia Manhattan que se aplica en el espacio RGB de los valores de los píxeles. Así, si según esta distancia y el umbral mencionado, un píxel está lo suficientemente próximo a algún nivel de gris, el píxel perderá su valor y pasará a ser negro. A continuación se muestra una imagen resultante de este preprocesamiento como ejemplo. Se ha empleado esta función de distancia en concreto, tanto por la facilidad de su implementación, como por su sencillez en los cálculos, que se traduce en menor costo computacional.



**Ilustración 6-6** Ejemplo de preprocesamiento 3

El preprocesamiento 4, realiza las operaciones que realiza el preprocesamiento 1, además de realizar una operación adicional. Dicha operación adicional, consiste en eliminar todos los píxeles cuyo valor de saturación en el espacio HSV no supere determinado umbral. En el preprocesamiento 4 el umbral a superar es igual al valor 66/255. A continuación se muestra una imagen resultante de este preprocesamiento como ejemplo.



**Ilustración 6-7** Ejemplo de preprocesamiento 4

El preprocesamiento 5, realiza las operaciones que realiza el preprocesamiento 2. Además realiza posteriormente la operación adicional que realiza el preprocesamiento 3 con respecto al preprocesamiento 1. Tras esto realiza la operación adicional que realiza el preprocesamiento 4 con respecto al preprocesamiento 1. A continuación se muestra una imagen resultante de este preprocesamiento como ejemplo.

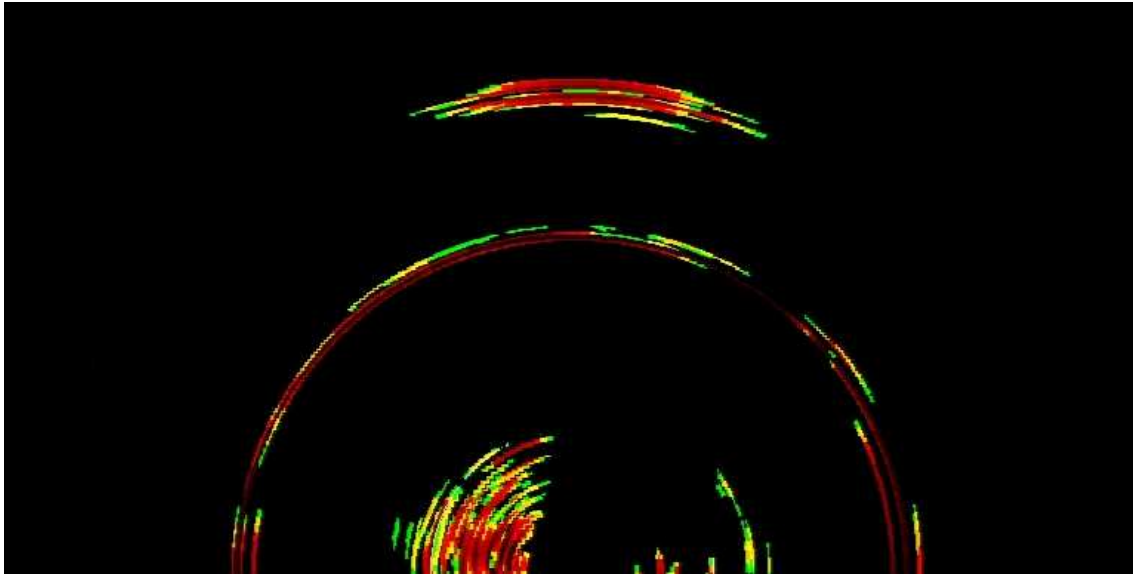


Ilustración 6-8 Ejemplo de preprocesamiento 5

Para ver la complejidad relativa de los preprocesamientos de forma clara, se ofrece la siguiente ilustración. En ella se puede observar un diagrama que contiene un grafo dirigido. Cada nodo del grafo alude a un preprocesamiento. Un arco dirigido entre dos nodos, denota que el nodo del que sale, es un nodo más sencillo que aquel nodo al que el arco dirigido entra. En este grafo en concreto, se indica qué preprocesamiento es más sencillo que qué otro. En concreto, para que un preprocesamiento sea más sencillo que un segundo preprocesamiento, el conjunto de operaciones del preprocesamiento candidato a ser más sencillo debe ser un subconjunto del conjunto de operaciones que realiza el mencionado segundo preprocesamiento.

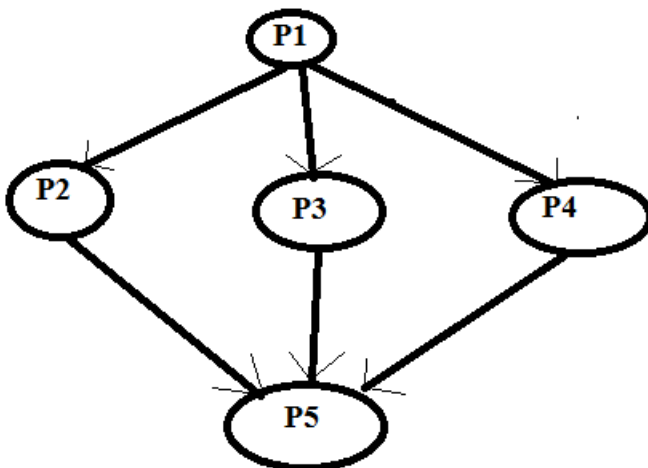


Ilustración 6-9 Diagrama de la complejidad relativa de preprocesamientos

Para ver qué operaciones que realiza cada preprocesamiento de una forma sencilla, se muestra la siguiente ilustración. Ella contiene un diagrama consistente en un grafo no dirigido y en una tabla informativa. La tabla informativa indica qué color está asociado a qué preprocesamiento en el grafo no dirigido. En el grafo no dirigido, se presentan dos nodos especiales, llamados “Inicio” y “Fin”, además de otros nodos. Para conocer las operaciones que realiza un preprocesamiento específico, se deben seguir los arcos de su color asociado en el grafo, partiendo desde el nodo “Inicio” hasta llegar al nodo “Fin”, sin repetir nunca el paso por ningún arco. Todos aquellos nodos por los que pase el recorrido creado, obviando los nodos “Inicio” y “Fin”, contienen todas las operaciones que realizará el preprocesamiento para el cual se ha examinado su recorrido. Además, según se recorre el camino, las operaciones se presentan en el mismo orden en que se ejecutan en la aplicación Sonar.

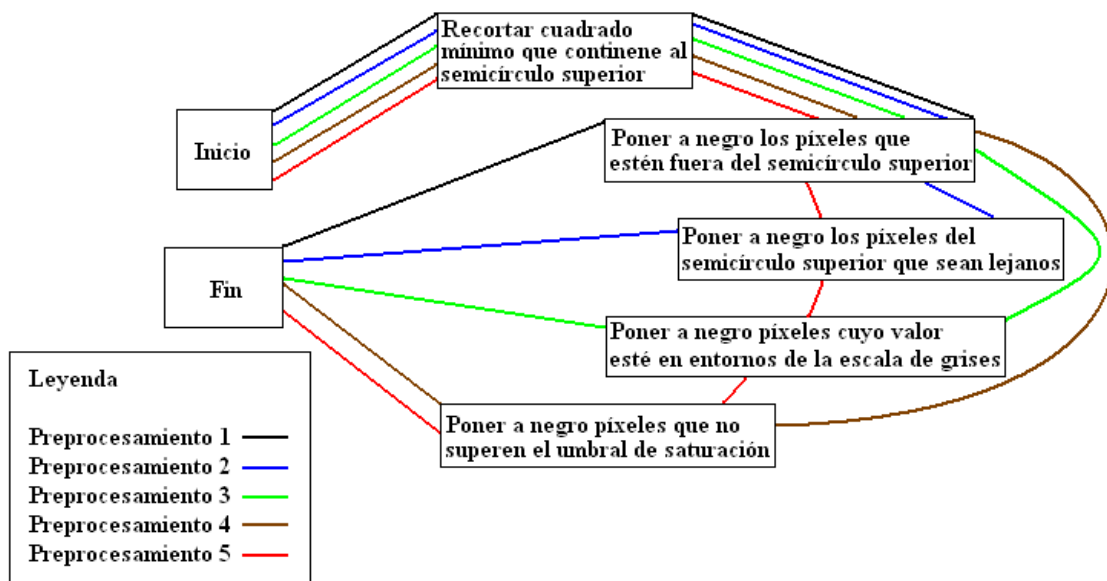


Ilustración 6-10 Diagrama de las operaciones de los preprocesamientos

La segmentación 1, es un proceso de segmentación muy simple, que toma los píxeles de color negro como píxeles de fondo y que toma los píxeles que no son de color negro como píxeles pertenecientes a un objeto. Entonces se determina si dos píxeles que pertenecen a un objeto pertenecen al mismo objeto si existe un camino que los une según una 8-adyacencia. De los objetos identificados, no todos serán elegidos para ser presentados como resultado a la segmentación. Esto es porque aquellos objetos que no tengan un mínimo de área no serán presentados como resultado. Dicho mínimo exigido implica que para que un objeto sea válido debe tener un área superior a 15 píxeles. Así pues para una imagen a segmentar se genera como producto resultante un conjunto de imágenes. En cada imagen de dicho conjunto de imágenes aparecerá un objeto diferente extraído de la imagen entrante, que ha superado el umbral de área. Dicho objeto aparecerá en la posición en la que aparecía en la imagen entrante al proceso de segmentación y todos aquellos píxeles que no pertenezcan al objeto serán negros. La eliminación de objetos de área pequeña responde a la siguiente motivación. La idea es evitar que se genere un número de imágenes de objetos abusivamente grande, sin eliminar posibles bancos de atún, al

ser los objetos eliminados demasiado pequeños como para que se correspondan con bancos de atún.

La segmentación 2, es muy similar a la segmentación 1. De hecho sólo varía el valor del parámetro que indica el mínimo de área requerido a un objeto para que sea considerado. En este caso el mínimo se establece en 50, por lo que el área de un objeto dado deberá ser superior a dicho valor, para que el objeto sea considerado.

Con la segmentación 3, ocurre lo mismo que con la segmentación 2. Es pues que la segmentación 3 es realmente similar a la segmentación 1 y a la segmentación 2. De nuevo, varía únicamente el valor del parámetro que indica el mínimo de área requerido a un objeto para que sea considerado. En esta ocasión dicho mínimo se establece en 100.

La extracción de características 1, toma como entrada un conjunto de imágenes, y genera como producto una base de datos incompleta que resulta ser un fichero “.arff” de Weka. Dicha base de datos incompleta, se diferencia de una base de datos estándar de Weka en lo siguiente. Una de las diferencias en la cabecera de la base de datos es que aparece un atributo de los casos que resulta ser el nombre de los casos y es de tipo cadena de caracteres. La otra diferencia en la cabecera, es que se define la variable clase como nominal, pero que no toma ningún valor (está vacía). Una de las diferencias en el cuerpo de la base de datos es que el primer campo es el nombre de la imagen desde la cual se generó el caso. La otra diferencia en el cuerpo de la base de datos es que el último campo, que es la variable clase, figura con valor desconocido “?”. Para una imagen con un objeto dado a la extracción de características 1, ésta calcula las siguientes propiedades del objeto. Calcula su área, descrita por el atributo “Area”. Calcula su centroide, es decir, promedia las posiciones de los píxeles que componen el objeto, expresando después su centroide según sus dos componentes espaciales, quedando descrito por los atributos “X” e “Y”. Calcula su perímetro, descrito por el atributo “Perimeter”. Calcula el rectángulo de menor tamaño que contiene al objeto, quedando éste descrito por su anchura, por su altura y por la posición de su esquina superior izquierda, descrita por sus dos componentes espaciales. Esta descripción del rectángulo mínimo se recoge en las variables “Width”, “Height”, “BX” y “BY”. Se calcula la elipse que mejor se ajusta al objeto. Esta elipse queda descrita por la longitud del eje de mayor elongación de la elipse, por la longitud del eje de menor elongación de la elipse y por el ángulo que forma el eje de mayor elongación de la elipse con el eje de las abscisas. Estos tres descriptores de la elipse se recogen en los atributos “Major”, “Minor” y “Angle” respectivamente. Se calcula el cociente entre la elongación del eje mayor (numerador) y el eje menor (denominador) recogido en el atributo “AR” aspect ratio. Se calcula la circularidad, dada por la fórmula siguiente.

$$circularidad = \frac{4\pi \cdot area}{perimetro^2}$$

### Fórmula 6-1 Circularidad

Cuanto más próximo esté el valor de la circularidad a uno, más próxima a un círculo será su forma, mientras que valores próximos a cero sugieren una forma alargada. Se calcula la redondez, propiedad definida por la siguiente fórmula.

$$\text{redondez} = \frac{4 \cdot \text{area}}{\pi \cdot \text{eje mayor}^2}$$

## Fórmula 6-2 Redondez

Se calcula la solidez, que es el cociente entre el área del objeto (numerador) y el área convexa del objeto (denominador). En este caso, área convexa del objeto, hace referencia a lo siguiente. Se coge el objeto original, y se rellenan sus concavidades hasta hacer desaparecerlas todas, sin rellenar más a partir de ahí. Entonces el área de ese objeto relleno es el área convexa del objeto. La solidez se recoge en el atributo "Solidity". Se calcula la mayor distancia existente entre cualquier par de píxeles del objeto, que se recoge en el atributo "Feret". En el atributo "Feret Angle" se recoge el ángulo de la línea que describe la distancia de "Feret". En "FeretX" y "FeretY" se recoge la ubicación de uno de los dos extremos de la línea que describe la distancia "Feret". Situándose sobre un punto intermedio de dicha línea, y observando la orientación descrita por el vector director que toma como ángulo sobre el eje de las abscisas el valor "Feret Angle", resulta que el último punto perteneciente a la línea siguiendo esa dirección es el punto extremo descrito por "FeretX" y "FeretY". Se calcula el valor "Min Feret" que es la longitud de una línea perpendicular y que interseca por el centro a la línea que describe la distancia de "Feret". La longitud de dicha línea debe ser la mínima que permita cumplir la siguiente condición. Primero tanto en los extremos de la línea que describe "Feret" como en los extremos de la que describe "Min Feret", se traza en cada uno una línea que resulta ser perpendicular a la línea a la cual pertenece el punto extremo desde el cual se está trazando la nueva línea. Para estas nuevas cuatro líneas, se observa el polígono descrito por las intersecciones entre las mismas. Se debe cumplir que dicho polígono debe contener al objeto.

La extracción de características 2, extrae las características que extraía la extracción de características 1 y adicionalmente extrae algunas más. Las características adicionales extraídas de un objeto son, la media, el valor mínimo, el valor máximo, la mediana, la desviación típica, el coeficiente de asimetría y el coeficiente de kurtosis del objeto en escala de grises. Es decir, en el caso de la media, se calcula para cada píxel del objeto su nivel de gris en base a su valor, para tras haberlos calculado todos, calcular la media de los niveles de gris. La asimetría queda descrita por la siguiente fórmula.

$$\text{asimetría} = \frac{1}{n} \cdot \sum_{i=1}^n \left( \frac{x_i - \mu}{\sigma} \right)^3$$

## Fórmula 6-3 Coeficiente de asimetría

El coeficiente de kurtosis queda descrito por la siguiente fórmula.

$$\text{kurtosis} = \frac{1}{n} \cdot \sum_{i=1}^n \left( \frac{x_i - \mu}{\sigma} \right)^4$$

## Fórmula 6-4 Coeficiente de kurtosis

El nivel de gris para un píxel dado en el espacio RGB, se calcula según la siguiente fórmula.

$$\text{nivel de gris} = 0,299 \cdot \text{rojo} + 0,587 \cdot \text{verde} + 0,114 \cdot \text{azul}$$



### Fórmula 6-5 Cálculo del nivel de gris

En la extracción de características 3, se extrae las características que se extraen en la extracción de características 2 y adicionalmente se extraen algunas más. Las características adicionales extraídas de un objeto son, la media, el valor mínimo, el valor máximo, la mediana, la desviación típica, el coeficiente de asimetría y el coeficiente de curtosis del objeto para su color rojo, su color verde y su color azul.

Cabe mencionar el empleo del Imagej en el cálculo de las características de objetos. En concreto, para características de objetos relativas a la forma de los mismos, se ha accedido a funciones de Imagej para realizar su cálculo. Por otra parte, la extracción de las características relativas al color de los objetos, no se realiza mediante el acceso a funciones de Imagej, si no que estos cálculos son realizados por la programación de la aplicación Sonar.

El etiquetado de instancias realizado ha sido el mismo para todas las bases de datos generadas. Primero, se ha distinguido entre dos tipos de etiquetas. Una de ellas representa los objetos que son atún con apenas interferencias. La otra etiqueta representa todo aquello que no tiene cabida en la anterior etiqueta. Como al etiquetado de instancias se llega tras realizar la segmentación y la extracción de características, se pueden separar las imágenes segmentadas entre los valores que toman sus etiquetas, asignados mediante juicio experto. Es decir, se observa cada objeto extraído de la segmentación y se decide si lo que se observa es atún con apenas interferencias, o si es cualquier otra cosa. Tras realizar esta decisión para cada objeto extraído de la segmentación, se toma la base de datos incompleta que deja la extracción de características realizada mediante la aplicación Sonar, y mediante el uso de la funcionalidad de etiquetado asistido de la aplicación Sonar, se logra construir una base de datos completa, que es apta para ser usada mediante la aplicación Weka.

Sobre los algoritmos de aprendizaje, cabe comentar someramente la naturaleza de cada uno de ellos. El clasificador "ZeroR" consiste en etiquetar todos los casos con el valor de la variable clase que más se presente en la base de datos. El clasificador "IBk" es un clasificador de la familia de los clasificadores K-NN y se deja configurado con sus opciones por defecto en las experimentaciones que hacen uso de él en esta tesis de máster. "JRip" es un clasificador de la familia de los clasificadores basados en reglas, "J48" es un clasificador de la familia de los clasificadores basados en árboles y "NaiveBayes" y "BayesNet" son clasificadores de la familia de los clasificadores basados en redes bayesianas. Todos ellos, "JRip", "J48", "NaiveBayes" y "BayesNet", al igual que "IBk", se dejan configurados con sus opciones por defecto en las experimentaciones que hacen uso de ellos en esta tesis de máster.



## 7 Aplicación sonar

En este apartado se realiza una somera descripción de la aplicación Sonar a nivel técnico, mediante el esbozo del diagrama de casos de uso, entre otros. Primero de todo declarar que la aplicación Sonar se ha desarrollado en el lenguaje de programación Java. Internamente la aplicación Sonar hace uso de la librería ij, correspondiente a la aplicación Imagej. Imagej es una aplicación de dominio público escrita en java desarrollada por el instituto nacional de la salud de los Estados Unidos. Está diseñado de tal forma que se pueden integrar plugins dentro del mismo. Imagej es una aplicación cuya funcionalidad principal es el procesamiento de imágenes. Se puede localizar en la dirección url localizada en la referencia bibliográfica (ImagejPrincipal) y se puede descargar directamente de la dirección url localizada en la referencia bibliográfica (ImagejDescarga). Se puede encontrar documentación relativa tales como tutoriales o APIs en la dirección url localizada en la referencia bibliográfica (ImagejDocumentación).

La aplicación Sonar, en lugar de integrarse como un plugin de Imagej, se ha desarrollado como una aplicación propia que accede a las funcionalidades de Imagej a través de su librería. La aparente obligación de Imagej de desarrollar un plugin en una sola clase (y los problemas derivados de esta restricción) fue la motivación que desembocó en tal decisión.

### 7.1 Diagrama de casos de uso

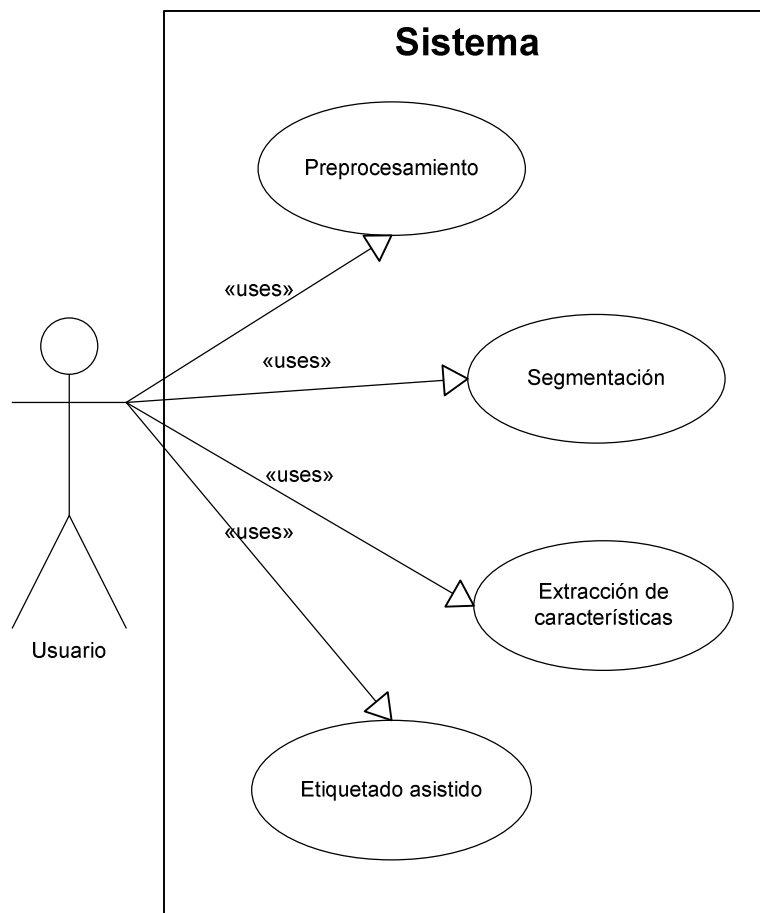


Ilustración 7-1 Diagrama de casos de uso

## 7.2 Casos de uso expandidos

Caso de uso	CU00: Preprocesamiento
Actores	Usuario
Propósito	El propósito es generar un conjunto de imágenes preprocesadas a partir de un conjunto de imágenes originales de sonar sin preprocesar.
Resumen	El usuario escogerá el directorio donde residen las imágenes originales y después escogerá el directorio donde residirán las imágenes preprocesadas.
Precondiciones	El directorio donde se guarden las imágenes preprocesadas no debería contener imágenes.
Postcondiciones	En el directorio donde se guardan las imágenes preprocesadas, se guardarán las imágenes preprocesadas en el formato de archivo especificado. Además sólo se preprocesarán del directorio de imágenes originales aquellas que sean del formato de archivo especificado.
Flujo principal	
Usuario	Sistema
1. Accede a la funcionalidad preprocesamiento. 3. El usuario elige las opciones generales de preprocesamiento. 5. El usuario elige el directorio donde residen las imágenes originales a preprocesar. 7. El usuario elige el directorio donde residirán las imágenes preprocesadas.	2. Muestra las opciones generales de preprocesamiento, como son elegir el tipo de preprocesamiento, el tipo de sonar ligado a las imágenes originales, el formato de las imágenes originales y el formato de las imágenes preprocesadas. 4. Muestra un selector de ficheros para elegir el directorio donde residen las imágenes originales a preprocesar. 6. Muestra un selector de ficheros para elegir el directorio donde residirán las imágenes preprocesadas. 8. El sistema preprocesa las imágenes originales del formato indicado, según el tipo de preprocesamiento indicado para el tipo de imágenes de sonar indicado, dejando en el directorio indicado las imágenes preprocesadas en el formato indicado.
Subflujos	
	Alternativa1: El usuario en 3, puede cancelar la funcionalidad en curso. Alternativa2: El usuario en 5, puede cancelar la funcionalidad en curso. Alternativa3: El usuario en 7, puede cancelar la funcionalidad en curso.

Tabla 7-1 CU00: Preprocesamiento

## Tesis de máster

Caso de uso	CU01: Segmentación
Actores	Usuario
Propósito	El propósito es generar un conjunto de imágenes segmentadas a partir de un conjunto de imágenes de sonar preprocesadas.
Resumen	El usuario escogerá el directorio donde residen las imágenes preprocesadas y después escogerá el directorio donde residirán las imágenes segmentadas.
Precondiciones	El directorio donde se guarden las imágenes segmentadas no debería contener imágenes. Las imágenes preprocesadas deben haber sido preprocesadas por la aplicación Sonar.
Postcondiciones	En el directorio donde se guardan las imágenes segmentadas, se guardarán las imágenes segmentadas en el formato de archivo especificado. Además sólo se segmentarán del directorio de imágenes preprocesadas aquellas que sean del formato de archivo especificado.
Flujo principal	
Usuario	Sistema
<p>1. Accede a la funcionalidad segmentación.</p> <p>3. El usuario elige las opciones generales de segmentación.</p> <p>5. El usuario elige el directorio donde residen las imágenes preprocesadas a segmentar.</p> <p>7. El usuario elige el directorio donde residirán las imágenes segmentadas.</p>	<p>2. Muestra las opciones generales de segmentación, como son elegir el tipo de segmentación, el formato de las imágenes preprocesadas y el formato de las imágenes segmentadas.</p> <p>4. Muestra un selector de ficheros para elegir el directorio donde residen las imágenes preprocesadas a segmentar.</p> <p>6. Muestra un selector de ficheros para elegir el directorio donde residirán las imágenes segmentadas.</p> <p>8. El sistema segmenta las imágenes preprocesadas del formato indicado, según el tipo de segmentación indicada, dejando en el directorio indicado las imágenes segmentadas en el formato indicado.</p>
Subflujos	
	<p>Alternativa1: El usuario en 3, puede cancelar la funcionalidad en curso.</p> <p>Alternativa2: El usuario en 5, puede cancelar la funcionalidad en curso.</p> <p>Alternativa3: El usuario en 7, puede cancelar la funcionalidad en curso.</p>

**Tabla 7-2 CU01: Segmentación**

## Tesis de máster

Caso de uso	CU02: Extracción de características
Actores	Usuario
Propósito	El propósito es generar un fichero “.arff” incompleto a partir de un conjunto de imágenes de sonar segmentadas.
Resumen	El usuario escogerá el directorio donde residen las imágenes segmentadas y después escogerá el lugar donde residirá el fichero “.arff” incompleto.
Precondiciones	Las imágenes deben haber sido segmentadas con la aplicación Sonar
Postcondiciones	En el directorio destino especificado se guardará el fichero “.arff” incompleto. Además sólo para la construcción del mismo sólo se tendrán en cuenta las imágenes del directorio de imágenes segmentadas que estén en el formato de archivo especificado.
Flujo principal	
Usuario	Sistema
1. Accede a la funcionalidad extracción de características. 3. El usuario elige las opciones generales de extracción de características. 5. El usuario elige el directorio donde residen las imágenes segmentadas. 7. El usuario elige el lugar donde residirá el fichero “.arff” incompleto.	2. Muestra las opciones generales de extracción de características, como son elegir el tipo de extracción de características y el formato de las imágenes segmentadas. 4. Muestra un selector de ficheros para elegir el directorio donde residen las imágenes segmentadas. 6. Muestra un selector de ficheros para elegir el lugar donde residirá el fichero “.arff” incompleto. 8. El sistema extrae característica de las imágenes segmentadas del formato indicado, dejando en el lugar indicado el archivo “.arff” incompleto con las descripciones de las imágenes segmentadas según sus características extraídas.
Subflujos	
	Alternativa1: El usuario en 3, puede cancelar la funcionalidad en curso. Alternativa2: El usuario en 5, puede cancelar la funcionalidad en curso. Alternativa3: El usuario en 7, puede cancelar la funcionalidad en curso.

**Tabla 7-3 CU02: Extracción de características**

## Tesis de máster

Caso de uso	CU03: Etiquetado asistido
Actores	Usuario
Propósito	El propósito es generar un fichero “.arff” completo a partir de un fichero “.arff” incompleto y un conjunto de imágenes segmentadas divididas en n directorios que están recogidos en un único directorio base.
Resumen	El usuario escogerá el lugar donde reside el fichero “.arff” incompleto, después elegirá el directorio base donde residen los n directorios, en los cuales se ha repartido el conjunto de imágenes segmentadas. Por último el usuario escogerá el lugar donde residirá el fichero “.arff” completo.
Precondiciones	Las imágenes segmentadas tienen que haber sido segmentadas con la aplicación Sonar. El fichero “.arff” incompleto tiene que haber sido generado con la aplicación Sonar.
Postcondiciones	En el fichero “.arff” completo será una modificación del fichero “.arff” incompleto. En concreto respecto a este, se elimina de la cabecera la variable nombre y de los datos el campo correspondiente. Además en la cabecera se especificará un número de valores de la variable clase igual al número de los n directorios, y cada valor distinto tomará como valor el nombre de uno de los n directorios (cada uno, uno distinto). Además, en el cuerpo de los datos se reemplazarán los valores desconocidos de la variable clase por el valor que corresponde a la instancia en curso. En concreto, para una instancia dada, antes de eliminar su nombre, se busca en cuál de los n directorios reside su imagen segmentada asociada, y el nombre del directorio en el que resida será su valor en la variable clase.
Flujo principal	
Usuario	Sistema
1. Accede a la funcionalidad de etiquetado asistido. 3. Elige el lugar donde reside el fichero “.arff”. 5. El usuario elige el directorio donde residen los n directorios que en conjunto contienen las imágenes segmentadas. 7. El usuario elige el lugar donde residirá el fichero “.arff” completo.	2. Muestra un selector de ficheros para elegir el lugar donde reside el fichero “.arff” incompleto. 4. Muestra un selector de fichero para elegir el directorio donde residen los n directorios que en total contienen las imágenes segmentadas. 6. Muestra un selector de ficheros para elegir el lugar donde residirá el fichero “.arff” completo. 8. El sistema lleva a cabo el etiquetado asistido, dando a cada instancia del fichero “.arff” el valor de la variable clase que le corresponde, registrando los distintos valores de la variable clase en la cabecera y eliminando de la cabecera y del cuerpo de los datos la variable nombre.
Subflujos	
	Alternativa1: El usuario en 3, puede cancelar la funcionalidad en curso. Alternativa2: El usuario en 5, puede cancelar la funcionalidad en curso. Alternativa3: El usuario en 7, puede cancelar la funcionalidad en curso.

Tabla 7-4 CU03: Etiquetado asistido

### 7.3 Dependencias entre paquetes

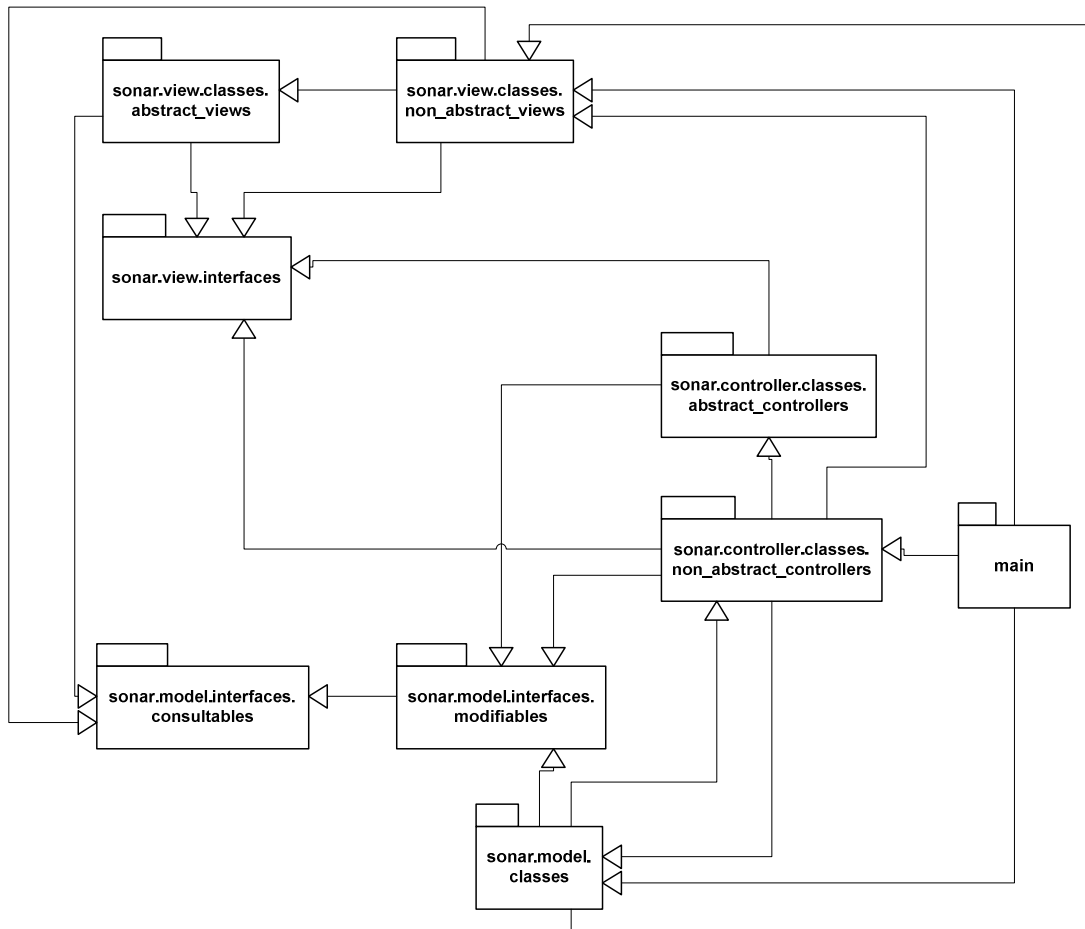
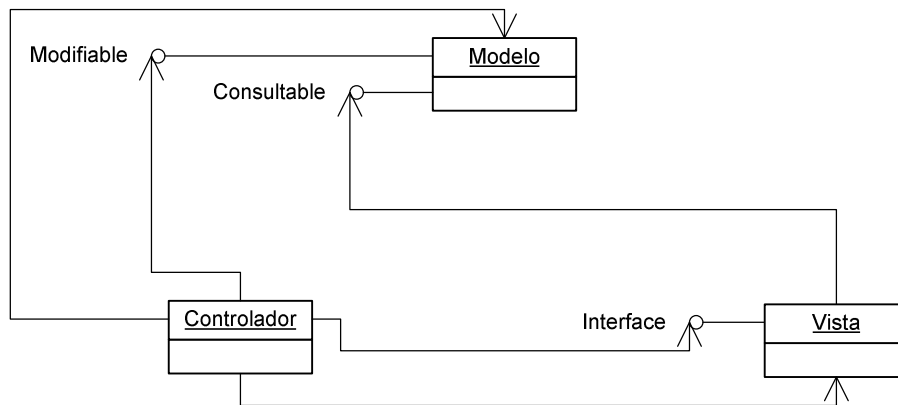


Ilustración 7-2 Dependencias entre paquetes

### 7.4 Patrón modelo-vista-controlador

El diagrama de la ilustración siguiente representa la tónica general en la arquitectura de objetos dentro de la aplicación Sonar. Los objetos de la vista sólo acceden al modelo como consulta, nunca con idea de modificar. El modelo, recoge la estructura del negocio en curso sin mayores implicaciones. Es decir, el modelo describe el procesamiento de imágenes de sonar, pero no ve objetos de la vista ni se preocupa de controlarlos. Son los objetos controladores, los que se encargan de coordinar vista y modelo. Ellos se encargan de lanzar la vista pertinente, de atender a los eventos que esta genere y de traducirlos en acciones en el modelo. También cabe resaltar, que los controladores sólo lidian con objetos del modelo y de la vista directamente, en lugar de con sus interfaces, cuando se disponen a lanzar una nueva vista. Mientras se trate de mantener el control de una vista en curso, los controladores tan sólo trabajarán con interfaces del modelo e interfaces de la vista.





**Ilustración 7-3 Patrón Modelo-Vista-Controlador específico implementado**

Sin embargo existen unas pocas excepciones muy puntuales a esta implementación específica del patrón modelo-vista-controlador. En algunas ocasiones el modelo trabaja con objetos controladores y objetos de vista. Este es un percance a reparar en el trabajo futuro.



## 8 Resultados

En este apartado se habla de los resultados obtenidos en las pruebas realizadas. Así pues se comentan los resultados obtenidos experimento por experimento. En general, salvo que se indique lo contrario, se supondrá que a la hora de extraer estimaciones directas acerca de la bondad del comportamiento de un algoritmo de aprendizaje sobre una base de datos, estas estimaciones se realizan sobre una 10-cross validación repetida 10 veces (la k-cross validación se explica en el anexo de técnicas de validación). Se emplea una 10-cross validación repetida 10 veces porque es el método que por defecto ofrece la interfaz “experimenter” de Weka.

Realizar un inciso, para denotar que algunos cálculos matemáticos para la obtención de los resultados se reflejan en el anexo llamado “Cálculos matemáticos de resultados”. Se considera que lo recomendable es leer primeramente este apartado de resultados y luego consultar el anexo en el que figuran los fundamentos matemáticos.

En este momento procede comentar algunos aspectos del método de comparación empleado. En función del preprocesamiento específico y la segmentación específica empleados, a la base de datos correspondiente llegarán distinto número de instancias en las que además la proporción entre los distintos valores de clase normalmente será diferente. Además hay que tener en cuenta, que por otra parte el interés real es triple. Uno de los deseos, es maximizar la detección de objetos que son atún. Otro de los deseos es el de minimizar el error de identificar como atún objetos que realmente no lo son. Por último existe el deseo de minimizar el error de no identificar como atún objetos que realmente son atún. Todos estos hechos, llevan a hacer que se deba reflexionar sobre qué método de comparación de procesos completos puede resultar adecuado. Por todo lo anterior, se descarta la idea de comparar directamente el porcentaje de bien clasificados entre procesos completos. Concretamente como el tándem preprocesamiento-segmentación puede generar mayor o menor desproporción en la presencia en distintos valores de clase según el caso, puede hacer que el porcentaje de bien clasificados se incremente simplemente con una mayor proporción de objetos que no son atún en la base de datos. Es decir, la comparación de porcentaje de bien clasificados, sirve para medir la bondad de los procesos completos con respecto a un objetivo que no es el que aquí se busca. Por los motivos aquí mencionados, también se descarta la idea de contrastar los procesos completos mediante la comparación del área bajo la curva ROC, ya que esta medida depende de dos parámetros, que son la especificidad y la sensibilidad, cuyas ecuaciones se muestran a continuación (en la referencia (Fawcett) se puede encontrar información acerca de la curva ROC y conceptos como la sensibilidad, la especificidad, la precisión y el F1 score).

$$\text{sensibilidad} = \frac{TP}{TP + FN}$$

**Fórmula 8-1 Sensibilidad**

$$\text{especificidad} = \frac{TN}{TN + FP}$$

**Fórmula 8-2 Especificidad**

En las anteriores ecuaciones,  $TP$  (verdaderos positivos) representa el conjunto de instancias que son realmente atún y se han reconocido como tal,  $TN$  (verdaderos negativos) representa el conjunto de instancias que no son realmente atún y se han reconocido como objetos que no son atún,  $FN$  (falsos negativos) representa el conjunto de instancias que son realmente atún, pero que se han reconocido como objetos que no son atún y finalmente  $FP$  (falsos positivos) representa el conjunto de instancias que son objetos que no son atún pero que se ha reconocido como objetos que son atún. Resulta pues que el área bajo la curva ROC depende de estos dos parámetros, y si bien la sensibilidad puede resultar de interés, la especificidad es un parámetro que no se corresponde con los objetivos aquí marcados. Esto se debe a que el interés real es maximizar la detección de bancos de atún, minimizar el número de bancos de atún que quedan sin detectar y minimizar el número de errores consistentes en identificar como atún un objeto que realmente no es atún. Es decir el interés real es maximizar  $TP$ , minimizar  $FN$  y minimizar  $FP$ . Teniendo lo anterior en cuenta, el número de objetos que realmente no son atún y se han identificado como tal, no resulta de aporte para la problemática que aquí se está abordando. Es decir, el interés se centra en los valores  $TP$ ,  $FP$  y  $FN$ .

Así pues se idea la siguiente forma de comparar procesos completos (o conjuntos de procesos completos). Primero se extraen los valores  $TP$ ,  $FP$  y  $FN$  para cada prueba. En este punto cabe comentar, que el valor  $TP$  y  $FP$  es el valor directamente extraído con Weka. Sin embargo  $FN$  no es el valor calculado mediante Weka. Esto se debe a que se están comparando procesos completos y no clasificadores. Resulta que en todos los ensayos, el número de objetos que son atún es igual a 28. Teniendo este dato en mente, restando a dicho valor, 28, el valor  $TP$  se obtiene el valor  $FN$ . De este hecho se deduce que el denominador de la sensibilidad en los experimentos realizados, será siempre igual al valor 28, pues el denominador es la suma de  $TP$  y de  $FN$ .

Así pues se extraen con la interfaz experimenter de Weka, el número de verdaderos positivos y falsos positivos para cada ejecución de una 10-cross validación. Como se ha aplicado una 10-cross validación, el número de verdaderos positivos y falsos positivos es proporcional al que sería para toda la base de datos, sólo que es diez veces inferior, pues el tamaño de cada rodaja es diez veces inferior. Por tanto ambos valores se multiplican por diez. El tamaño de las muestra de una ejecución de una 10-cross validación sobre un proceso completo es 10 pues se realizan 10 ejecuciones de 10-cross validaciones en cada proceso completo. Tras esto, se calcula la sensibilidad y la precisión. La precisión viene dada por la siguiente ecuación.

$$precisión = \frac{TP}{TP + FP}$$

### Fórmula 8-3 Precisión

En este punto, cada par de procesos completos (o cada par de conjuntos de procesos completos) se contrasta mediante un test t emparejado de independencia sobre la sensibilidad, con un nivel de confianza del 95%. Igualmente se contrasta cada par de procesos completos (o cada par de conjuntos de procesos completos) mediante un test t emparejado de independencia sobre la precisión, con un nivel de confianza del 95%. Después se calcula el F1 score, que viene determinado por la siguiente ecuación.

$$F1 = \frac{\textit{precisión} \cdot \textit{sensitividad}}{\textit{precisión} + \textit{sensitividad}}$$

### Fórmula 8-4 F1 score

Tras su cálculo, se realiza un contraste entre cada par de procesos completos (o cada par de conjuntos de procesos completos) mediante un test t emparejado de independencia, con un nivel de confianza del 95%. Tras tener los contrastes para la sensibilidad, la precisión y el F1 score, se define qué proceso completo (o conjunto de procesos completos) es completamente superior a qué otro. Se dice que un proceso completo (o conjunto de procesos completos) es completamente superior a otro, si es al menos igual de bueno tanto en precisión, como en sensibilidad y en F1 score, y al menos es mejor en alguna de dichas tres métricas. En todas las tablas de contrastes, las de sensibilidad, las de precisión y las de F1 score, además de en las tablas de superioridad completa, se emplea la siguiente nomenclatura. Para una celda dada, se presenta una “M” si el proceso completo fila es significativamente mejor que el proceso completo columna, se presenta una “P” si el proceso completo fila es significativamente peor que el proceso completo columna, se presenta una “I” si no se puede asegurar que el proceso completo fila es significativamente superior ni significativamente inferior al proceso completo columna y finalmente, se presenta una “X” cuando el proceso completo fila es el mismo proceso completo que el proceso completo columna. Concretamente, superior en precisión, implica una que se posee una precisión más alta, superior en sensibilidad, implica que se posee una sensibilidad más alta y superior en F1 score, implica que se posee un F1 score más alto.

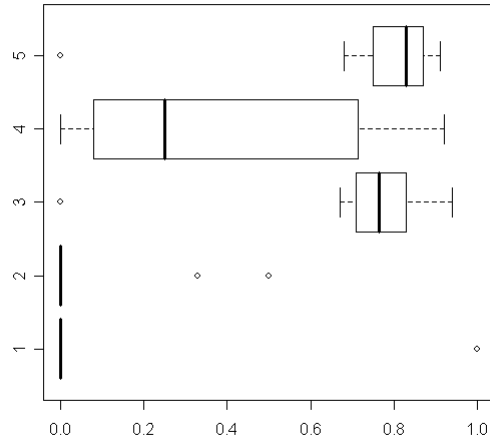
Indicar que, debido a la envergadura de las tablas generadas, éstas se dejan reflejadas en el anexo llamado tablas de resultados. Aquí según se vayan exponiendo los resultados se irán referenciando las tablas pertinentes que figuran en el anexo mencionado.

En concreto, en el experimento uno se ha operado de la siguiente forma. Tras ejecutar el experimento y obtener las distribuciones de *TP*, *FP* (obtenidas de Weka) para cada proceso completo, se extraen las distribuciones de *FN*, la precisión, la sensibilidad y el F1 score para cada proceso completo. Después se han extraído las medias y las desviaciones típicas de *TP*, *FP* y *FN* para cada proceso completo ejecutado. Esto se recoge en la tabla llamada “Experimento 1: Medias y desviaciones típicas de TP, FP y FN” en el anexo de tablas de resultados. Después se han extraído las medias y las desviaciones típicas de la precisión, la sensibilidad y el F1 score. Esto se recoge en la tabla llamada “Experimento 1: Medias y desviaciones típicas de precisión, sensibilidad y F1 score” en el anexo de tablas de resultados.

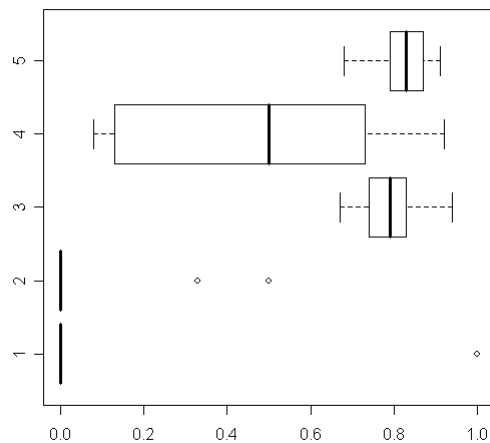
Después se realizan dos agrupaciones. En la primera se agrupan los procesos completos ejecutados en el experimento, según el preprocesamiento que empleen. En la segunda se agrupan los procesos completos ejecutados en el experimento, según el preprocesamiento que empleen, pero dejando fuera de todos los grupos de esta agrupación a aquellos que empleen el clasificador ZeroR. Cada grupo de cada agrupación representará el preprocesamiento que comparten todos los miembros del grupo. Entonces en ambas agrupaciones se crea una distribución para cada preprocesamiento representado por cada propiedad de la terna precisión, sensibilidad y F1 score. La distribución para un preprocesamiento representado específico para una propiedad específica de la terna en una agrupación específica, es aquella que se forma a

partir de un conjunto de valores. Dicho conjunto de valores, contiene a todos los valores con que se formaron cada una de las distribuciones de la misma propiedad para cada elemento del grupo (que está contenido en la agrupación mencionada) que representa el preprocesamiento para el cual se está generando la distribución de la mencionada propiedad. Es decir, de forma simplificada, además de tener distribuciones de precisión, sensibilidad y F1 score para cada proceso completo empleado en el experimento, se calcula la distribución de precisión, sensibilidad y F1 score para preprocesamiento empleado, agrupando los valores según el preprocesamiento para generar las distribuciones mencionadas. Sin embargo, se genera para cada preprocesamiento, dos distribuciones por cada propiedad de la terna precisión, sensibilidad y F1 score. En una de ellas se tienen en cuenta todos los procesos completos, mientras que en la otra quedan al margen los procesos que emplean ZeroR. Entonces quedan seis distribuciones según el preprocesamiento empleado, dos de precisión, dos de sensibilidad y dos de F1 score. Para tres de ellas se ha tenido en cuenta todos los procesos a la hora de generarlas, mientras que para las otras tres tan sólo se ha tenido en cuenta los clasificadores “inteligentes”. Es sobre estas seis distribuciones, entre cada par de preprocesamientos, sobre las que se realiza un test t emparejado de independencia, con un nivel de confianza del 95%. Todos estos contrastes se recogen en tablas en el anexo de tablas de resultados. El contraste entre pares de preprocesamientos según sus distribuciones de la precisión teniendo en cuenta todos los clasificadores se refleja en la tabla de nombre “Experimento 1: Contraste de preprocesamientos según precisión teniendo en cuenta todos los clasificadores”, que sigue la nomenclatura especificada previamente. El contraste entre pares de preprocesamientos según sus distribuciones de la precisión teniendo en cuenta los clasificadores “inteligentes” se refleja en la tabla de nombre “Experimento 1: Contraste de preprocesamientos según precisión teniendo en cuenta los clasificadores “inteligentes””, que sigue la nomenclatura especificada previamente. El contraste entre pares de preprocesamientos según sus distribuciones de la sensibilidad teniendo en cuenta todos los clasificadores se refleja en la tabla de nombre “Experimento 1: Contraste de preprocesamientos según sensibilidad teniendo en cuenta todos los clasificadores”, que sigue la nomenclatura especificada previamente. El contraste entre pares de preprocesamientos según sus distribuciones de la sensibilidad teniendo en cuenta los clasificadores “inteligentes” se refleja en la tabla de nombre “Contraste de preprocesamientos según sensibilidad teniendo en cuenta los clasificadores “inteligentes””, que sigue la nomenclatura especificada previamente. El contraste entre pares de preprocesamientos según sus distribuciones del F1 score teniendo en cuenta todos los clasificadores se refleja en la tabla de nombre “Experimento 1: Contraste de preprocesamientos según F1 score teniendo en cuenta todos los clasificadores”, que sigue la nomenclatura especificada previamente. El contraste entre pares de preprocesamientos según sus distribuciones del F1 score teniendo en cuenta los clasificadores “inteligentes” se refleja en la tabla de nombre “Experimento 1: Contraste de preprocesamientos según F1 score teniendo en cuenta los clasificadores “inteligentes””, que sigue la nomenclatura especificada previamente. Dados los tres contrastes que tienen en cuenta todos los clasificadores, se genera una tabla de superioridad completa, que se recoge en el anexo de tablas de resultados con el nombre de “Experimento 1: Superioridad completa entre preprocesamientos teniendo en cuenta todos los clasificadores”. Dados los otros tres contrastes que tienen en cuenta solo los clasificadores “inteligentes”, se genera una tabla de superioridad completa, que se recoge en el anexo de tablas de resultados con el nombre de “Experimento 1: Superioridad completa entre preprocesamientos teniendo en cuenta los clasificadores “inteligentes””. Aparte de los resultados de los contrastes

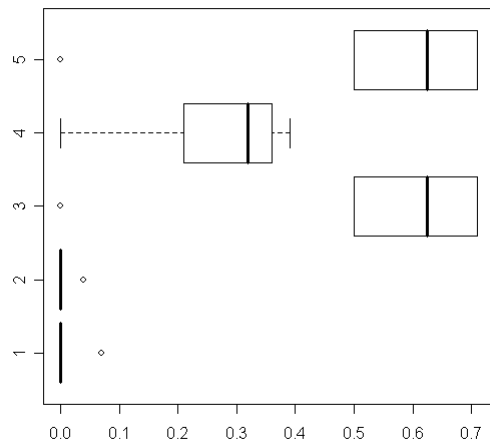
para cada una de las seis distribuciones, se muestra para cada una de ellas, un diagrama boxplot. En cada diagrama, se muestra una caja por cada preprocesamiento. En todos estos diagramas, el eje de las abscisas representa el valor que toma la propiedad, mientras que el eje de las ordenadas representa el preprocesamiento empleado. Todos estos diagramas se muestran a continuación.



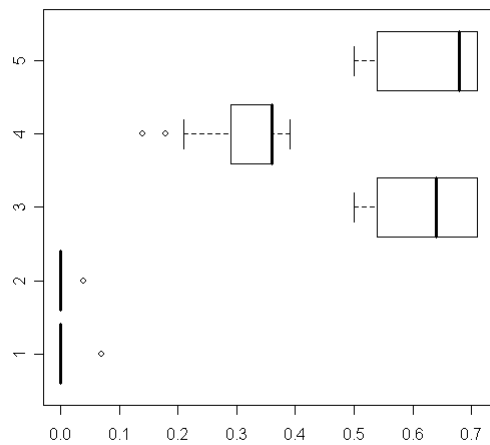
**Ilustración 8-1 Experimento 1: Boxplot de preprocesamientos para la precisión tenidos en cuenta todos los clasificadores**



**Ilustración 8-2 Experimento 1: Boxplot de preprocesamientos para la precisión tenidos en cuenta los clasificadores "inteligentes"**

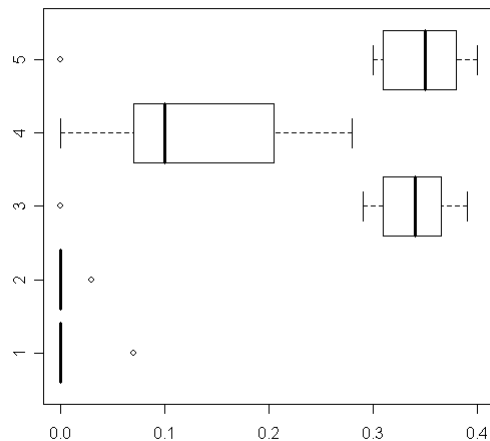


**Ilustración 8-3 Experimento 1: Boxplot de preprocesamientos para la segmentación tenidos en cuenta todos los clasificadores**

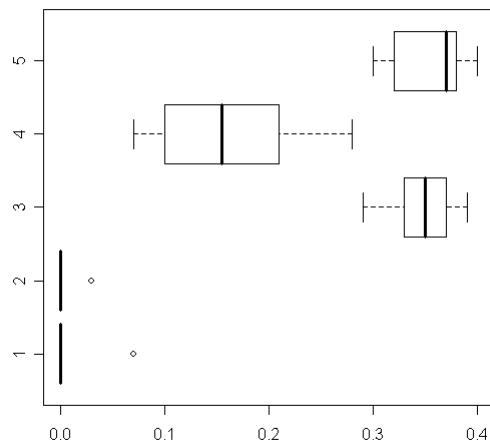


**Ilustración 8-4 Experimento 1: Boxplot de preprocesamientos para la segmentación tenidos en cuenta los clasificadores "inteligentes"**





**Ilustración 8-5 Experimento 1: Boxplot de preprocesamientos para el F1 score tenidos en cuenta todos los clasificadores**

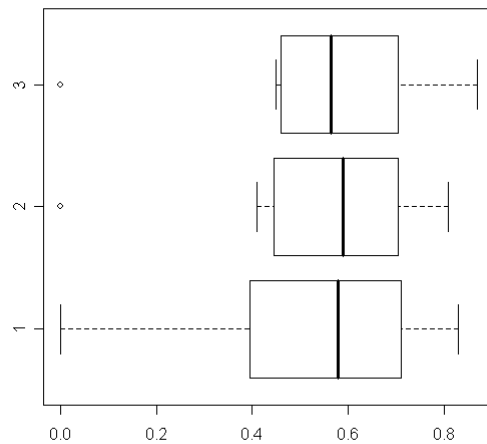


**Ilustración 8-6 Experimento 1: Boxplot de preprocesamientos para el F1 score tenidos en cuenta los clasificadores "inteligentes"**

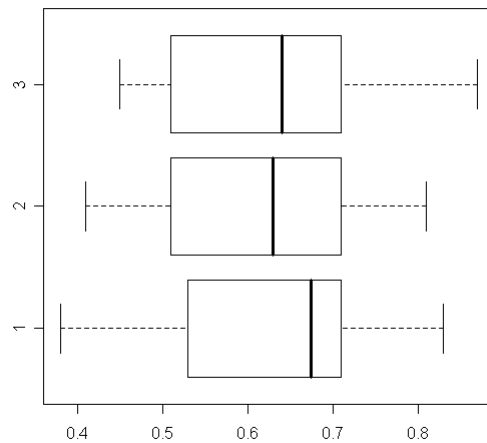
Recurriendo a las dos tablas de superioridad sobre preprocesamientos, tan sólo quedan los preprocesamientos 3 y 5 que no son superados por ningún otro en ambas tablas. Si bien los resultados del preprocesamiento 5 son ligeramente superiores a los del preprocesamiento 3, no se puede concluir que sean significativamente superiores (para ello quizá se podrían realizar muestras de mayor tamaño). Sin embargo, el preprocesamiento 5 ofrece una ventaja con respecto al preprocesamiento 3 y es que se generan un menor número de objetos con la segmentación. Esto implica que Weka tendrá que trabajar con menos instancias, con lo que el volumen de cómputo resultará menor. Por el contrario, el preprocesamiento 5 tiene como desventaja frente al preprocesamiento 3 que requiere mayor cómputo. Un observación directa de ambas diferencias en vivo, lleva a observar que la ventaja del preprocesamiento 5 compensa la desventaja del mismo con respecto al preprocesamiento 3. Teniendo en cuenta todo lo anterior, se considera que el preprocesamiento 5 es el preprocesamiento más adecuado.

Cabe comentar el número de instancias que llegan a la base de datos y que resultan ser atún, según el proceso parcial empleado. Así pues, con los procesos parciales 1 1 1 y 2 1 1, sólo llegan a la base de datos 2 casos que se identifican como atún. Con el proceso parcial 4 1 1 llegan 12 y con los procesos parciales 3 1 1 y 5 1 1 llegan 22.

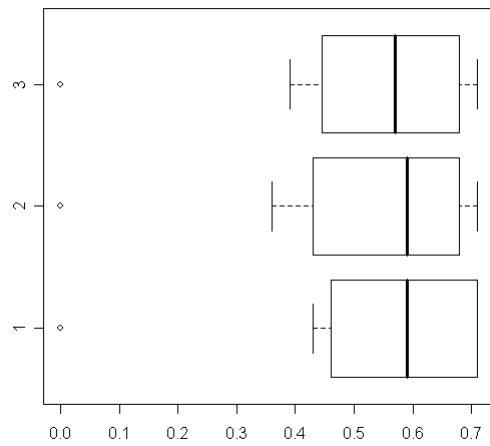
En el segundo experimento, se opera de forma idéntica, salvo que las agrupaciones en lugar de ser realizadas por preprocesamientos, se realizan por segmentaciones. Los resultados análogos quedan reflejados en las tablas de nombres análogos. A continuación se muestran los boxplots análogos, recordando que en esta ocasión, el eje de las ordenadas representa la segmentación específica y no el preprocesamiento específico.



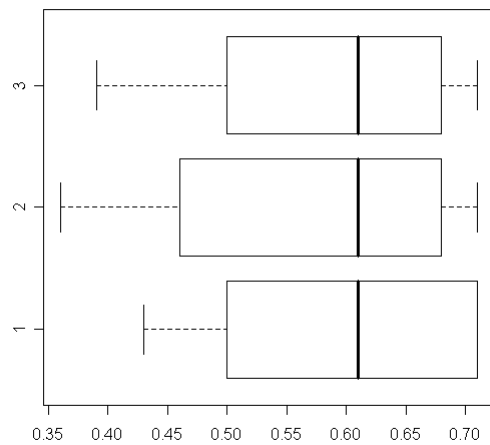
**Ilustración 8-7 Experimento 2: Boxplot de segmentaciones para la precisión tenidos en cuenta todos los clasificadores**



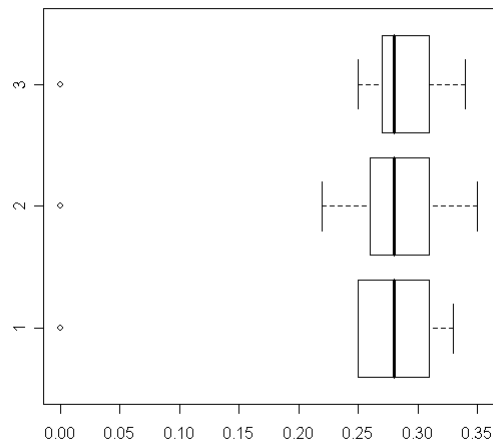
**Ilustración 8-8 Experimento 2: Boxplot de segmentaciones para la precisión tenidos en cuenta los clasificadores "inteligentes"**



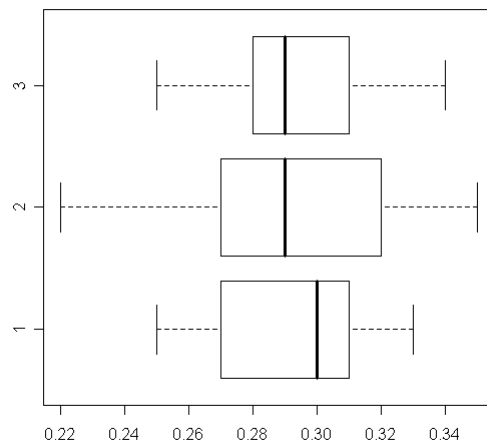
**Ilustración 8-9 Experimento 2: Boxplot de segmentaciones para la segmentación tenidos en cuenta todos los clasificadores**



**Ilustración 8-10 Experimento 2: Boxplot de segmentaciones para la segmentación tenidos en cuenta los clasificadores "inteligentes"**



**Ilustración 8-11 Experimento 2: Boxplot de segmentaciones para el F1 score tenidos en cuenta todos los clasificadores**

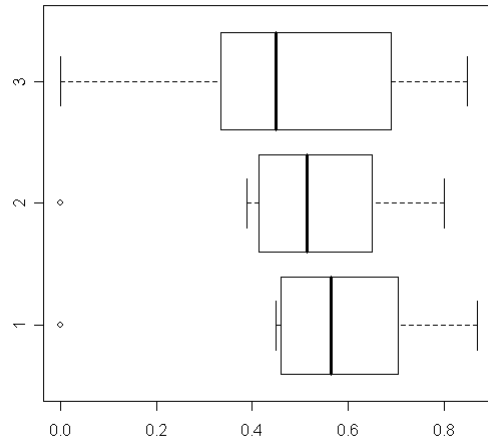


**Ilustración 8-12 Experimento 2: Boxplot de segmentaciones para el F1 score tenidos en cuenta los clasificadores "inteligentes"**

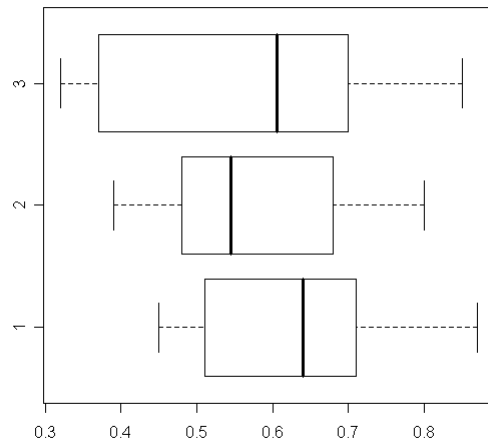
Observando las dos tablas de superioridad sobre segmentaciones, se percibe que ninguna es superior a ninguna otra en ninguna de las dos tablas. Sin embargo, sucede que cada una filtra mayor o menor número de objetos para procesos posteriores. Como la segmentación 3 es la segmentación que más objetos filtra y ofrece un rendimiento que no se puede asegurar que sea ni mejor ni peor que el de las otras dos, se considera que es la más adecuada. El hecho de dejar menos objetos para procesos posteriores reduce el volumen de cómputo.

En el segundo experimento, se opera de forma idéntica que en las anteriores ocasiones, salvo que las agrupaciones en lugar de ser realizadas por preprocesamientos o segmentaciones, se realizan por extracciones de características. Los resultados análogos quedan reflejados en las tablas de nombres análogos. A continuación se muestran los boxplots análogos, recordando que en esta

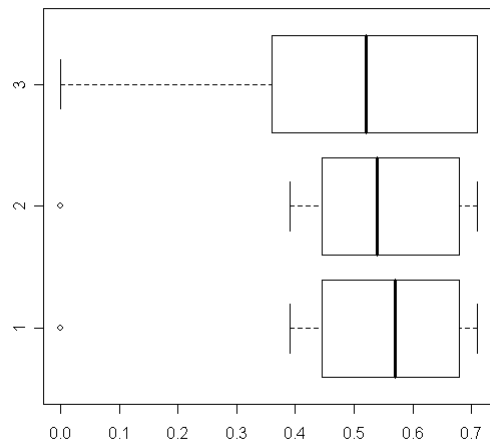
ocasión, el eje de las ordenadas representa la extracción de características específica y no el preprocesamiento específico ni la segmentación específica.



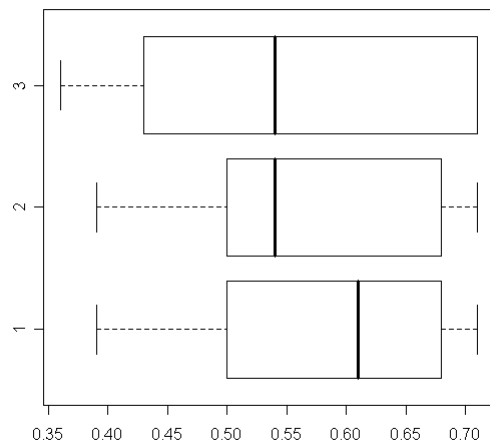
**Ilustración 8-13 Experimento 3: Boxplot de extracciones de características para la precisión tenidos en cuenta todos los clasificadores**



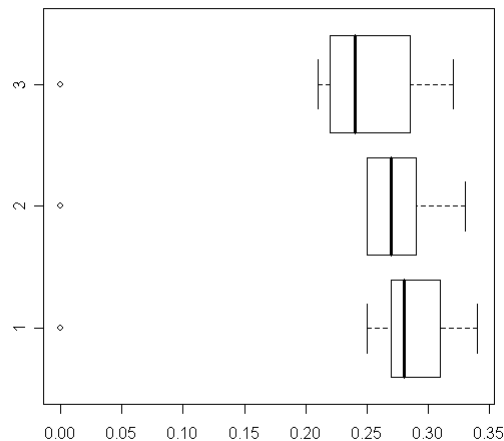
**Ilustración 8-14 Experimento 3: Boxplot de extracciones de características para la precisión tenidos en cuenta los clasificadores "inteligentes"**



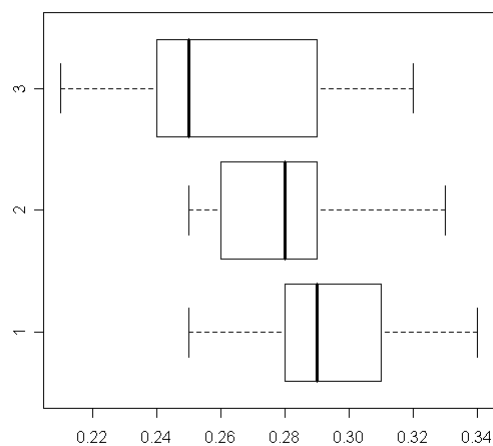
**Ilustración 8-15 Experimento 3: Boxplot de extracciones de características para la segmentación tenidos en cuenta todos los clasificadores**



**Ilustración 8-16 Experimento 3: Boxplot de extracciones de características para la segmentación tenidos en cuenta los clasificadores “inteligentes”**



**Ilustración 8-17 Experimento 3: Boxplot de extracciones de características para el F1 score tenidos en cuenta todos los clasificadores**



**Ilustración 8-18 Experimento 3: Boxplot de extracciones de características para el F1 score tenidos en cuenta los clasificadores “inteligentes”**

Recurriendo a las dos tablas de superioridad sobre extracciones de características, que tan sólo la 1 es superior a la 3 cuando se consideran exclusivamente los clasificadores inteligentes. Además la extracción de características 1 requiere menor volumen de cómputo cuando se ejecuta y genera menor volumen de cómputo para los procesos posteriores. Requiere un menor volumen de cómputo cuando se ejecuta ya que las características que extrae son subconjunto de las características que extraen cualquiera de las otras dos extracciones de características. Genera menor volumen de cómputo para los procesos posteriores ya que genera menor número de características que las otras dos extracciones de características. Por todo lo anterior, se considera que la extracción de características 1 es la más adecuada.

Dado que se ha deliberado qué preprocesamiento, qué segmentación y qué extracción de características resultan más adecuados, cabe observar, qué clasificador, muestra mejor

comportamiento. Así pues, se realizan tres contrastes de los procesos completos que tienen el preprocesamiento, la segmentación y la extracción de características considerados más adecuados. El primer contraste es sobre la precisión, el segundo sobre la sensibilidad y el último sobre el F1 score. Estos tres contrastes se recogen en las tablas con nombres “Experimento 3: Contraste de clasificadores según precisión”, “Experimento 3: Contraste de clasificadores según sensibilidad” y “Experimento 3: Contraste de clasificadores según F1 score” en el anexo de tablas de resultados. Finalmente se elabora una tabla de superioridad completa dados estos tres contrastes, dejando dicha tabla en el anexo de tablas de resultados con el nombre “Experimento 3: Superioridad completa entre clasificadores”.

Observando esta tabla de contrastes, se ve que sólo los clasificadores IBk y NaiveBayes son los que no son superados completamente por ningún otro. IBk aporta mejores resultados en precisión que NaiveBayes, pero NaiveBayes muestra un mejor comportamiento en sensibilidad que IBk. En F1 score IBk se comporta mejor que NaiveBayes. En definitiva IBk no es completamente superior a NaiveBayes, pero lo supera en dos de las tres propiedades clave. Por ello si los deseos de obtener buenos resultados tanto en precisión como en sensibilidad son igual de intensos, parece congruente decantarse por IBk como mejor opción. Por último, de forma análoga a anteriores ocasiones, se muestran a continuación, los diagramas boxplot. Cabe indicar que en esta ocasión, al no haber agrupaciones, tan sólo se genera un boxplot por propiedad clave (precisión, sensibilidad y F1 score). En esta ocasión se comparan los clasificadores, que se reflejan en el eje de las ordenadas. En ese eje, el valor 1 representa al clasificador ZeroR, el valor 2 representa al clasificador IBk, el valor 3 representa al clasificador JRip, el valor 4 representa al clasificador J48, el valor 5 representa al clasificador NaiveBayes y el valor 6 representa al clasificador BayesNet.

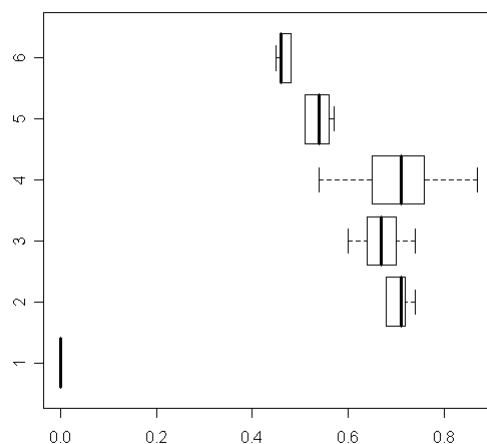


Ilustración 8-19 Experimento 3: Boxplot de clasificadores para la precisión



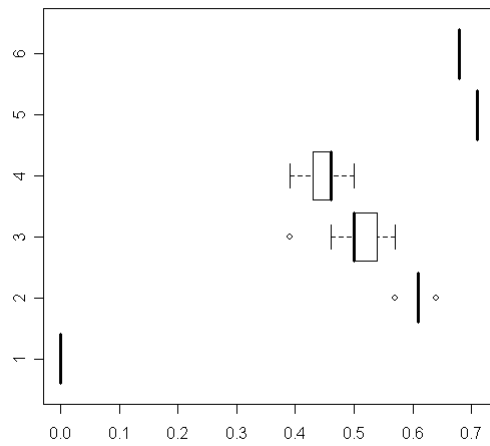


Ilustración 8-20 Experimento 3: Boxplot de clasificadores para la sensibilidad

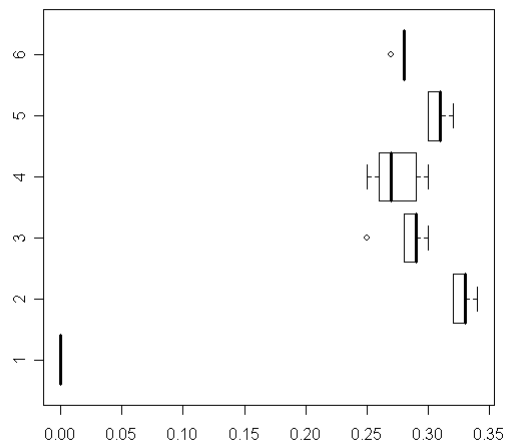


Ilustración 8-21 Experimento 3: Boxplot de clasificadores para el F1 score



## 9 Conclusiones

En este apartado se exponen las conclusiones extraídas de los ensayos realizados y los resultados obtenidos. Primero de todo indicar que se puede discernir entre las conclusiones extraídas de la generación y experimentación con las cinco primeras bases de datos y las conclusiones extraídas de la generación y experimentación con las tres bases de datos posteriores.

Del primer experimento en el que se generan las cinco primeras bases de datos, se extrae lo siguiente. Vistos los resultados, respecto a lo que interesa, una alta sensibilidad, una alta precisión y un alto F1 score, se puede concluir que los preprocesamientos 3 y 5 parecen otorgar mejor respuesta que los demás. Recordar que el preprocesamiento 1 era el preprocesamiento más sencillo, pues no hacía nada salvo quedarse con la mitad superior del círculo del sonar. Igualmente cabe recordar que los preprocesamientos 2, 3 y 4 son en esencia porciones del preprocesamiento 5, que resulta de la composición ellos. Con esto en mente y los resultados en la memoria, se puede aventurar la conclusión de que la operación adicional que realiza el preprocesamiento 3 es la que tiene mayor aporte tiene en el ensalce de la imagen. Dicha operación adicional se basaba en la eliminación de valores de píxel (corrimento al negro) que estuvieran en entornos (mediante el empleo de la distancia Manhattan) de cualquier color perteneciente a la escala de grises. Es consonante con los resultados deducir que es esta operación la que realiza mayor aporte al ensalce de las imágenes para los procesos posteriores.

Por otra parte, se puede concluir que la operación adicional del preprocesamiento 4 resulta de aporte a la problemática, mientras que en el caso de la operación adicional del preprocesamiento 2 no se puede aventurar tal conclusión. De hecho la operación adicional del preprocesamiento 2 parece no tener un aporte al ensalce, por lo que resulta natural plantearse eliminar el preprocesamiento 2 y eliminar la operación adicional correspondiente al preprocesamiento 2 residente en el preprocesamiento 5 en futuros experimentos. Es decir, quizá no tenga porqué eliminarse los objetos más lejanos que están dentro del círculo del sonar. Lo que se traduce en, que quizá el efecto que tienen los agregados gráficos del semicírculo blanco exterior, las rectas blancas con las que intersecta, las iniciales de los puntos cardinales y sus cuatro rectas asociadas, sea demasiado tenue como para influir en los resultados de un proceso completo. De ser así se podría mejorar el proceso removiendo la mencionada operación adicional. Las ventajas serían el ahorro de cómputo y la posibilidad de detectar los bancos de atún más lejanos. La desventaja sería que podría desembocar en mayor número de instancias tras la segmentación, lo que implicaría mayor volumen de cómputo.

Otro aspecto a considerar, es el de la conjunción de las operaciones adicionales de los preprocesamientos 3 y 4 en el preprocesamiento 5. En la experimentación realizada no se ha podido concluir que el preprocesamiento 5 resulte de mayor aporte de cara a los resultados que el preprocesamiento 3, aunque sus números son algo mejores. Por ello, cabe plantearse si recogiendo un mayor número de muestras, se podría distinguir si el preprocesamiento 5 es de mayor aporte que el preprocesamiento 3 o no. Esto permitiría distinguir si la operación adicional del preprocesamiento 4 es prescindible o no de cara a los resultados, pues caso de ser de igual aporte el preprocesamiento 5 y el preprocesamiento 3, se estaría indicando que las operaciones adicionales de los preprocesamientos 2 y 4 no realizan ningún aporte que la operación adicional del preprocesamiento 3 no realice.

Sobre la segmentación cabe decir lo siguiente. Primero cabe recalcar su sencillez, pues ésta hace que objetos que se superponen espacialmente no puedan ser distinguidos como objetos separados. El preprocesamiento 5 parece favorecer la segmentación, aunque esta es una impresión visual extraída a la hora de contrastar la segmentación hecha para distintos preprocesamientos previos. No obstante esto es algo difícil de cuantizar, problema de medición el cual se podría plantear resolver en el futuro.

Del segundo experimento en el que se generan las tres bases de datos posteriores, se extraen las siguientes conclusiones. En general, los resultados decaen en la precisión, con respecto a la experimentación anterior con las primeras cinco bases de datos. Este decaimiento puede venir provocado por uno o ambos de los siguientes motivos. Por un lado en término medio, ha crecido la desproporción existente entre clases, pudiendo incrementarse en medio orden de magnitud fácilmente. Por otro lado, el empleo de las tres secuencias de imágenes para la generación de las bases de datos hace que se introduzcan objetos de nuevas formas y distribuciones de colores en la base de datos. Esto puede complicar el problema con respecto al procesamiento de una sola secuencia, pues puede que sea más difícil discernir en las nuevas formas aportadas por las dos secuencias de imágenes adicionales agregadas.

Sobre los resultados generales en el empleo de las distintas tres segmentaciones, se concluye lo siguiente. Primero resaltar la discrepancia entre resultados y cálculos. Lo esperado era que la segmentación 2 favoreciera a los resultados más que la segmentación 1 y que la segmentación 3 favoreciera a los resultados más que la segmentación 1 y la segmentación 2. Sin embargo tal conclusión no parece factible a la vista de los resultados. La esperanza anterior se basaba en el supuesto de que quizá la creciente desproporción entre clases estaba afectando a los resultados. Sin embargo ante estos resultados dicha idea pierde peso. Por tanto la idea de que el decaimiento de resultados (en precisión) está provocado por una mayor dificultad de discernir la naturaleza de los objetos del conjunto de objetos generado por las tres secuencias de imágenes gana peso. Sin embargo, la segmentación 3 parece ofrecer en general al menos resultados igual de buenos que la segmentación 1 y la segmentación 2. La cuestión es que la segmentación 3 tiene una ventaja adicional sobre las otras dos y es que la segmentación 3 tiene un umbral más alto para el filtrado de objetos por su área. Esto hace que los objetos que distinga la segmentación 3, sean un subconjunto de los distinguidos por la segmentación 2 y también sean un subconjunto de los distinguidos por la segmentación 1. Esto implica que la segmentación 3 en general implicará un menor volumen de cómputo en etapas posteriores. Por ello, si el rendimiento en clasificación de la segmentación 3 es igual al de las segmentaciones 1 y 2, es preferible usar la segmentación 3 por el beneficio en tiempo que ofrece.

Del tercer experimento en el que se generan las tres últimas bases de datos, se extraen las siguientes conclusiones. Al parecer, la incorporación de características extraídas de los niveles de gris, de los niveles de rojo, de los niveles de verde y de los niveles de azul, no se traduce en un incremento de las prestaciones. Esto puede deberse, a que las nuevas variables incorporadas, son redundantes entre sí, o a que son irrelevantes para la predicción de la clase.

Comprendiendo ahora los resultados en general, se puede decir que las técnicas aquí desarrolladas aportan netamente más que las técnicas sin tratamiento real. Como ejemplo de esto, indicar que el preprocesamiento 5 ha mostrado un mejor comportamiento que el

preprocesamiento 1. Teniendo en cuenta todo lo anterior se considera que el proceso parcial 5 3 1 es el más adecuado. Tras esto la búsqueda del clasificador más adecuado para conjuntar con el proceso parcial mencionado, lleva a observar que IBk y NaiveBayes son los mejores candidatos, si bien presentan diferencias entre ambos. IBk obtiene mejores resultados en precisión y NaiveBayes obtiene mejores resultados en sensibilidad. Por ello se considera que los procesos completos 5 3 1 IBk y 5 3 1 NaiveBayes, son los procesos completos más adecuados y cabrá decidir, en función de qué se desee más, si se desee más precisión o más sensibilidad, cual de los dos emplear.



### 10 Líneas futuras

En este apartado se recogen un conjunto de futuras posibilidades de desarrollo del trabajo iniciado en esta tesis de fin de máster. Así pues se realiza una descripción somera de las técnicas aquí propuestas.

Un posible avance sería la extracción de características de los objetos hasta ahora no extraídas, pudiendo ser dichas características basadas en el color, o en la forma, o en una combinación de ambas.

Otro posible avance sería la selección de variables en el aprendizaje automático, ya que no todas contribuyen en la misma cuantía y quizá la contribución de algunas sea demasiado escasa como para ser significativa.

Otro posible avance sería probar segmentaciones más sofisticadas, como las basadas en detección de bordes y discontinuidades o las que se centran en detectar las regiones directamente.

Otro posible avance sería, conocer la función que hace corresponder intervalos de señal acústica con colores digitales. Dada dicha función, las imágenes se podrían pasar a escala de grises donde el nivel de gris es un reflejo directo de la intensidad de la señal acústica. Al deshacer la transformación intermedia desconocida, se sabe con seguridad qué valor de píxel representa qué valor de intensidad de señal acústica.

Otro posible avance sería, lo que aquí se llama segmentación supervisada. No se ha podido comprobar si el concepto tras la denominación de segmentación supervisada ya existe en el área del conocimiento del procesamiento de imagen. La idea subyacente es, coger un conjunto de imágenes de sonar preprocesadas pero no segmentadas y realizar una segmentación manual con ayuda de juicio experto en la materia. Así se tiene una correspondencia entre imágenes sin segmentar y de imágenes segmentadas fiel a lo que se desea. Con dicha base de datos, la idea es realizar un aprendizaje automático de cómo se deben segmentar las imágenes.

Análogo al caso anterior, sería el avance del preprocesamiento supervisado.

Otro posible avance sería, el no procesar las imágenes de sonar individualmente. Es decir, tener en cuenta la relación temporal entre los instantes en los que se capturaron las imágenes y tener en cuenta la relación espacial existente entre los objetos de una imagen con los objetos pertenecientes a imágenes contenidas dentro de un entorno temporal de la primera imagen mencionada.

Otro posible avance sería, mediante una aplicación de reconocimiento de caracteres, recoger los datos de navegación y de otras naturalezas que figuran fuera del círculo del sonar, que podrían ayudar a la clasificación.

Otro posible avance sería el siguiente. Se parte de que se dispone ya de un conjunto de imágenes segmentadas y etiquetadas. Entonces se diseña un proceso evolutivo que genera conjuntos de características extraíbles de cada objeto de las imágenes segmentadas. Sería un proceso iterativo basado en el tándem formado por el proceso evolutivo y la valoración de las soluciones ofrecidas

por el mismo mediante el empleo de la aprendizaje supervisado sobre la base de datos resultante de las etiquetas, los objetos, y las características generadas por el proceso evolutivo para el ciclo en el cual se está realizando la valoración de dichas características.

Otra idea que puede suponer un avance, es aplicar la programación evolutiva a la segmentación, dados un conjunto de imágenes preprocesadas y el conjunto resultante de segmentar el primer conjunto de imágenes. La diferencia entre la segmentación ofrecida por la programación evolutiva y los resultados esperados definidos por la correspondencia entre el par de conjuntos de imágenes antes mencionado, serviría como métrica de la bondad de la solución propuesta por la programación evolutiva.

Otra idea análoga a la anterior, es la de aplicar la programación evolutiva al preprocesamiento.

Otra posible idea, es la de definir un número de imágenes, para a continuación hacer lo siguiente. Dado ese número, de una secuencia de imágenes se escoge un entorno temporal de forma que dentro de él queden ese número de imágenes y se trata de decir en conjunto si en esa secuencia de imágenes hay bancos de atún, no a nivel de imagen individual.

Otra mejora, en este caso de cara a su empleo cotidiano, sería el de integrar la aplicación Sonar dentro de Imagej.





## 11 Anexos

### 11.1 Conceptos básicos de probabilidad

La probabilidad se encarga de modelar los fenómenos no deterministas. En este apéndice se rememoran algunos conceptos básicos de la teoría de la probabilidad. Primero de todo, recordar que una variable aleatoria discreta  $X$  queda definida por el conjunto de posibles valores que puede tomar,  $\{x_1, \dots, x_n\}$ , y el conjunto formado por las probabilidades asociadas a cada posible valor que puede tomar  $\{p(x_1), \dots, p(x_n)\}$ . Recordar que la probabilidad de un suceso  $x_i$ ,  $P(X = x_i)$ , indica la proporción de las veces que se presenta sobre el total de ensayos, cuando el número total de ensayos tiende al infinito. Este valor puede ser como mínimo cero, indicando que el suceso nunca ocurre, o uno, indicando que ocurre en todos los casos sobre el total de la proporción.

Una distribución de probabilidad de una variable aleatoria discreta puede definirse como un vector que agrupa las probabilidades de todos los distintos resultados posibles de un ensayo. Por ejemplo al lanzar un dado de seis caras, hay seis posibles resultados, al conjunto de seis valores que recoge las probabilidades de cada resultado, se le llama distribución de probabilidad (la suma de todos sus valores es igual a la unidad).

La probabilidad conjunta de dos sucesos específicos  $x_i, y_j$  pertenecientes a las variables aleatorias  $X, Y$  respectivamente, se denota por  $P(X = x_i \cap Y = y_j)$ , e indica la probabilidad de que se cumpla que la variable aleatoria  $X$  tome el valor  $x_i$  y de que también se cumpla que la variable aleatoria  $Y$  tome el valor  $y_j$ .

Una distribución de probabilidad conjunta de dos variables recoge, las probabilidades de todas las distintas combinaciones posibles de pares de resultados de las dos variables (este concepto es generalizable al número de variables que se deseen). Es decir, se recoge en un conjunto las probabilidades asociadas a cada uno de los elementos del conjunto formado por el producto cartesiano entre el conjunto de posibles sucesos de una primera variable aleatoria y el conjunto de posibles sucesos de una segunda variable aleatoria.

La probabilidad condicionada indica como es de probable un suceso de una variable aleatoria dado que otro suceso de otra variable aleatoria ha acontecido. Así, la probabilidad condicionada del suceso  $x_i$  de la variable aleatoria  $X$  al suceso  $y_j$  de la variable aleatoria  $Y$  se denota por  $P(X = x_i | Y = y_j)$ . Por ejemplo, teniendo dos urnas, una con una bola blanca y otra con una bola negra, y se realiza una extracción de una urna al azar, se observa que la probabilidad de extraer una bola blanca sin saber de qué urna se realizó la extracción no es igual a la probabilidad de extraer una bola blanca a sabiendas de qué urna se realizó la extracción. En concreto, la probabilidad de un suceso condicionado a otro, es igual a la proporción de probabilidad conjunta de ambos sucesos dividida por la probabilidad del segundo suceso, igualdad que se muestra en la siguiente fórmula.

$$P(\mathbf{X} = x_i | \mathbf{Y} = y_j) = \frac{P(\mathbf{X} = x_i \cap \mathbf{Y} = y_j)}{P(\mathbf{Y} = y_j)}$$

### Fórmula 11-1 Probabilidad condicionada

El teorema de la probabilidad total indica que la probabilidad de un primer suceso es igual a la suma de probabilidades condicionadas de ese suceso a un resultado concreto de un segundo suceso, multiplicadas cada una por la respectiva probabilidad de que suceda ese resultado concreto en ese segundo suceso, siendo en cada término de la suma, el resultado concreto en ese segundo suceso, distinto cada vez. A continuación se muestra la ecuación que resume el teorema de la probabilidad total.

$$P(\mathbf{X} = x_i) = \sum_{j \in I} P(\mathbf{X} = x_i \cap \mathbf{Y} = y_j) \cdot P(\mathbf{Y} = y_j)$$

### Fórmula 11-2 Teorema de la probabilidad total

La fórmula del teorema de Bayes, se saca del teorema de la probabilidad total y de la definición de probabilidad condicionada. Partiendo de esta última definición, se reemplaza en el numerador mediante la propia definición de probabilidad condicionada, salvo que se reemplaza haciendo aparecer la probabilidad condicionada del segundo suceso al primero, y no al revés (cosa que puede hacerse ya que las variables son mudas). Tras este reemplazamiento, se sustituye el denominador mediante el teorema de la probabilidad total, llegando así, a la fórmula de Bayes. Seguidamente se muestra la ecuación que resume el teorema de Bayes.

$$P(\mathbf{X} = x_i | \mathbf{Y} = y_j) = \frac{P(\mathbf{Y} = y_j | \mathbf{X} = x_i) \cdot P(\mathbf{X} = x_i)}{\sum_{i \in I} P(\mathbf{Y} = y_j | \mathbf{X} = x_i) \cdot P(\mathbf{X} = x_i)}$$

### Fórmula 11-3 Teorema de Bayes

La definición de distribución de probabilidad condicionada es análoga a las definiciones de distribuciones anteriores, solo que ahora se compondrá de un vector de probabilidades condicionadas (siempre todas ellas condicionadas a la misma conjunción de resultados concretos, claro).

Dos variables aleatorias se dicen independientes si se da el hecho de que el conocimiento del resultado concreto de una, no aporta nada de conocimiento con respecto a la otra (la probabilidad del primer suceso a priori es igual a la probabilidad del primer suceso sabiendo el resultado específico del segundo suceso). En la siguiente fórmula se muestran las implicaciones directas de la independencia entre dos variables.

$$\mathbf{X} \text{ e } \mathbf{Y} \text{ son i.} \Leftrightarrow P(\mathbf{Y} = y | \mathbf{X} = x) = P(\mathbf{Y} = y)$$

$$\mathbf{X} \text{ e } \mathbf{Y} \text{ son i.} \Leftrightarrow P(\mathbf{X} = x | \mathbf{Y} = y) = P(\mathbf{X} = x)$$

$$\mathbf{X} \text{ e } \mathbf{Y} \text{ son i.} \Leftrightarrow P(\mathbf{X} = x \cap \mathbf{Y} = y) = P(\mathbf{X} = x) \cdot P(\mathbf{Y} = y)$$

### Fórmula 11-4 Independencia entre dos variables

Dos variables aleatorias se dicen condicionalmente independientes dada una tercera variable aleatoria, si se da el hecho de que conocido el resultado de la tercera, las otras dos son independientes entre sí. En la siguiente fórmula se muestran las implicaciones directas de la independencia condicional entre dos variables dada una tercera.

$$X \text{ e } Y \text{ son c. i. dada } Z \Leftrightarrow P(Y = y | X = x \cap Z = z) = P(X = x | Z = z)$$

$$X \text{ e } Y \text{ son c. i. dada } Z \Leftrightarrow P(X = x | Y = y \cap Z = z) = P(Y = y | Z = z)$$

$$X \text{ e } Y \text{ son c. i. dada } Z \Leftrightarrow P(X = x \cap Y = y | Z = z) = P(X = x | Z = z) \cdot P(Y = y | Z = z)$$

### Fórmula 11-5 Independencia condicional entre dos variables dada una tercera

La independencia condicional de entre dos variables aleatorias dada una tercera, se puede generalizar. Así, la independencia condicional entre dos variables puede verse restringida no sólo a una tercera variable, una única variable, si no a  $n$  variables.

Los potenciales son parecidos a las probabilidades, en el sentido que cuanto más alto es el valor del potencial, más probable es que suceda el suceso al que está asociado, y que los potenciales nunca son menores que cero. El potencial es un valor no acotado por arriba, y además proporcional a la probabilidad.

Dada una distribución de probabilidad conjunta de dos variables, extraer la distribución de probabilidad de una de ellas se denomina marginalización sobre dicha variable, y se realiza sumando sobre la dimensión de la otra variable.

## 11.2 La transformada de Fourier

Antes de hablar de técnicas de preprocesamiento en el dominio de frecuencia, cabe explicar qué es el dominio de la frecuencia, cómo llevar una imagen digital dada en el dominio espacial al dominio de la frecuencia, y cómo devolver la imagen que se ha transformado para que resida en el dominio de la frecuencia, al dominio espacial.

La transformada de Fourier es una función matemática. Lo que se hace en la transformada de Fourier es conseguir describir la función entrante a la transformada, como una superposición de un conjunto de ondas senoidales. Así, en el dominio de la frecuencia se muestra cómo oscilan las componentes en las que se ha descrito la función entrante tras la aplicación de la transformada de Fourier.

Para que exista la transformada de Fourier de una función  $f(x)$ , ésta, debe cumplir con un conjunto de propiedades, conocidas como condiciones de Dirichlet. La primera propiedad es que  $f(x)$  debe ser absolutamente integrable, es decir, se verifica la inecuación siguiente.

$$\int_{-\infty}^{\infty} |f(x)| dx < \infty$$

### Fórmula 11-6 Condición de integrabilidad absoluta

## Tesis de máster

---

La segunda propiedad que  $f(x)$  debe cumplir, es que de tener discontinuidades, el número de éstas debe ser finito y deben estar presentes en cualquier intervalo finito.

Una vez establecidas las condiciones que una función debe cumplir para que su transformada de Fourier exista, se presenta la transformada de Fourier, en la siguiente ecuación.

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-2\pi iux} dx$$

### Fórmula 11-7 Transformada de Fourier

En la anterior ecuación  $f(x)$  es la función a transformar y  $F(u)$  es la función transformada.

Si la transformada de Fourier es la función que lleva a cabo la obtención  $F(u)$ , la función transformada, a partir de  $f(x)$ , la función original, la antitransformada de Fourier o la transformada inversa de Fourier, es la función que obtiene  $f(x)$  a partir de  $F(u)$ . Seguidamente se muestra la transformada inversa de Fourier definida por la siguiente ecuación.

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{2\pi iux} du$$

### Fórmula 11-8 Transformada inversa de Fourier

La transformada de Fourier se puede generalizar para el caso de que se desee transformar una función de varias variables. A continuación se muestra la transformada de Fourier y la transformada inversa de Fourier para el caso de una función de dos variables.

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y)e^{-2\pi i(ux+vy)} dx dy$$

### Fórmula 11-9 Transformada de Fourier para el caso bidimensional

$$f(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u)e^{2\pi i(ux+vy)} dudv$$

### Fórmula 11-10 Transformada inversa de Fourier para el caso bidimensional

Planteando la transformada de Fourier para el caso bidimensional, la proximidad al caso de interés es mayor, pues es, en imágenes digitales bidimensionales donde se desea aplicar la transformada de Fourier. Por ello se plantea la transformada de Fourier discreta, pues se está hablando de imágenes digitales. En las siguientes fórmulas se muestran la transformada de Fourier discreta y la transformada inversa de Fourier discreta tanto para el caso unidimensional como para el caso bidimensional.

$$F(u) = \sum_{x=0}^{M-1} f(x) \cdot e^{-i2\pi ux/M} \quad \text{para } u = 0, \dots, M-1$$

### Fórmula 11-11 Transformada de Fourier discreta

$$f(x) = \sum_{u=0}^{M-1} F(u) \cdot e^{i2\pi ux/M} \quad \text{para } x=0, \dots, M-1$$

**Fórmula 11-12 Transformada inversa de Fourier discreta**

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot e^{-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad \text{para } u=0, \dots, M-1 \quad v=0, \dots, N-1$$

**Fórmula 11-13 Transformada de Fourier discreta para el caso bidimensional**

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) \cdot e^{i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} \quad \text{para } x=0, \dots, M-1 \quad y=0, \dots, N-1$$

**Fórmula 11-14 Transformada inversa de Fourier discreta para el caso bidimensional**

Cabe mencionar, que la transformada de Fourier discreta y la transformada inversa de Fourier discreta, siempre existen, cuando la función a transformar toma un número de valores finito.

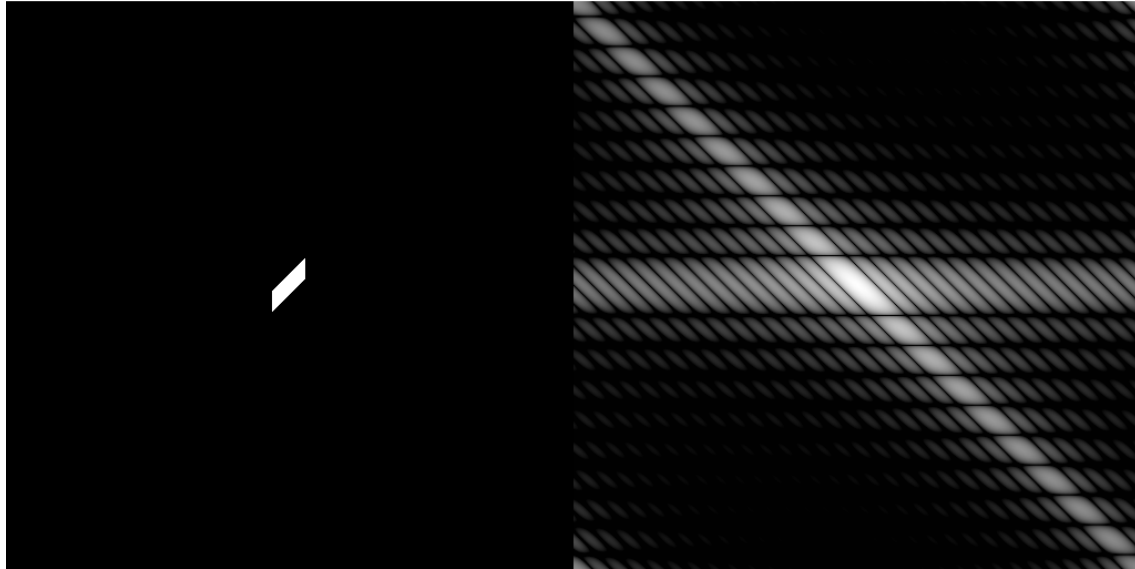
La transformada de Fourier deja como producto una función compleja, la cual en ocasiones resulta ventajoso presentarla en forma polar.

$$|F(u, v)| = \sqrt{R(F(u, v))^2 + I(F(u, v))^2}$$

$$\theta(u, v) = \tan^{-1} \left( \frac{I(F(u, v))}{R(F(u, v))} \right)$$

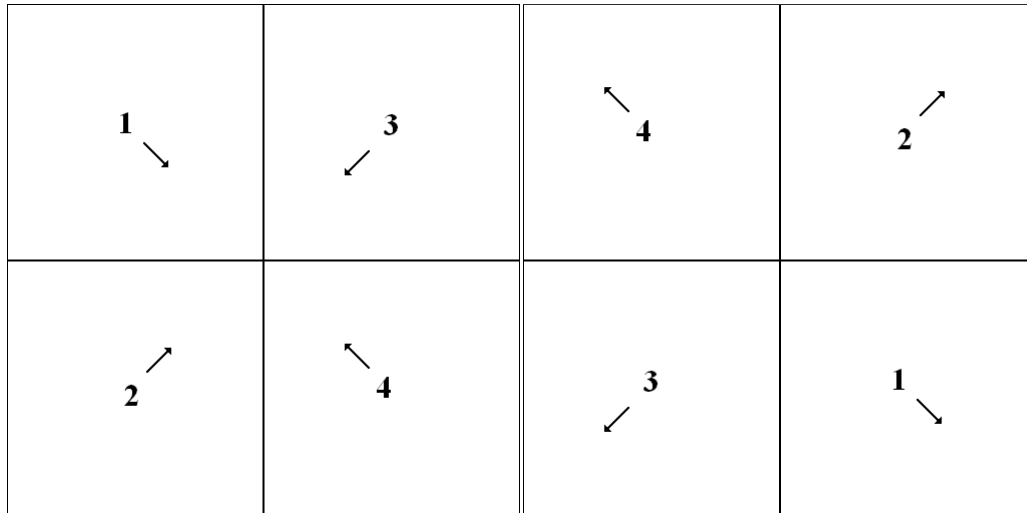
**Fórmula 11-15 Resultado de la transformada en coordenadas polares**

Donde en la fórmula anterior, la primera ecuación representa el módulo, la segunda ecuación representa el ángulo,  $R()$  es una función que devuelve la parte real de un número dado e  $I()$  es una función que devuelve la parte imaginaria de un número dado. A continuación se muestra como ejemplo, una imagen digital, y la presentación visual de la imagen digital tras ser transformada mediante la transformada de Fourier. Los valores de los píxeles de la imagen transformada, valores los cuales son de una única componente, están basados en el valor  $|F(u, v)|$ . Al ser de una única componente dichos valores, la imagen presentada de la transformada de Fourier se ofrece en escala de grises.



**Ilustración 11-1 Ejemplo: Imagen e imagen transformada**

Así pues una imagen digital bidimensional descrita en el dominio espacial, se transforma y se obtiene una imagen digital bidimensional descrita en el dominio de la frecuencia. Las imágenes del dominio de frecuencia en ocasiones se suelen representar visualmente, pero, a la hora de realizar su presentación visual, las imágenes transformadas al dominio de la frecuencia se transforman para poder realizar su presentación visual. Así pues en el caso anterior, la imagen en el dominio de la frecuencia, se ha obtenido tras aplicar dos transformaciones, la primera consiste en transformar la imagen original mediante la transformada de Fourier, y la segunda consiste en realizar una transformación para enriquecer la presentación visual. Esta nueva transformación tiene como objetivo facilitar la interpretación visual. La transformación convencional para la presentación visual de una imagen en el dominio de la frecuencia se compone de dos partes. La primera parte consiste en realizar una operación por cada dimensión espacial de la imagen. En cada una de esas operaciones, se escoge una dimensión distinta de las que se escogen en el resto de esas operaciones. Entonces, para una operación con una dimensión espacial escogida, se ordenan los píxeles de la imagen según el valor que tengan en dicha componente de menor a mayor. Se realiza una partición de los píxeles de la imagen en dos conjuntos, donde el primer conjunto tomará los primeros elementos y el segundo conjunto los últimos. La cardinalidad de los dos conjuntos de la partición deberá ser lo más similar posible. Ahora los elementos de ambos conjuntos resultantes de la partición se ordenan según el mismo criterio. Entonces se crea un nuevo conjunto unión de los dos anteriores, preservando el ordenamiento de los píxeles de cada uno de los conjuntos, pero poniendo primero los píxeles del segundo conjunto y poniendo después los del primero. Entonces en dicho conjunto, se cogen los valores de los píxeles en la componente espacial escogida, y se reordenan dichos valores reasignándose a distintos píxeles en el conjunto. Se reordenan de tal forma que los valores queden ordenados de mayor a menor. Ahora este último conjunto es el producto de realizar la operación en la dimensión espacial escogida. Dicho a “grosso modo”, esta parte de la transformación consiste en dada una imagen digital, dividirla por una mitad en una dimensión e intercambiar sus píxeles, y realizar este intercambio una vez por cada dimensión espacial de la imagen. A continuación, como ejemplo, en la siguiente ilustración se muestra la aplicación de esta etapa de la transformación en una imagen digital.



**Ilustración 11-2 Favoreciendo la presentación visual mediante el intercambio de mitades**

La segunda etapa de esta transformación que favorece la interpretación visual, consiste en realizar un escalado de los valores del módulo  $|F(u, v)|$ , ya que el rango de los valores es mucho mayor que en la imagen original. En una presentación visual convencional de una imagen digital a la que se le ha aplicado la transformada de Fourier, los valores de los píxeles de la imagen presentada, son una función dependiente de  $|F(u, v)|$ . Dicha función se presenta a continuación, en la cual  $c$  representa una variable que se determina en función de la escala específica a aplicar.

$$D(u, v) = c \cdot \log(1 + |F(u, v)|)$$

**Fórmula 11-16 Escalado de la imagen descrita en el dominio de la frecuencia**

Cabe mencionar, que en una imagen a la que se le ha aplicado la transformada de Fourier y después la transformada para favorecer la presentación visual, la frecuencia descrita por cada píxel es más baja cuanto más próxima al centro de la imagen está.

### 11.3 Técnicas de validación

Un problema a plantear es que, a la hora de medir como de bien se comporta un clasificador, normalmente, en principio se tiene un conjunto de casos de clase conocida, y posteriormente llegan los casos de clase desconocida. Por lo tanto, para estimar como de bien se comporta, hay que basarse en el contraste de valores reales de la variable clase para los casos de valor de variable clase conocido, con las predicciones que realiza el clasificador para dichos casos. Pero esta estimación no es un problema trivial. Que un clasificador aprenda de un conjunto de casos de clase conocida y luego se mida la calidad de sus resultados en función de cómo clasifica cada uno de los casos de dicho conjunto de casos, resulta poco honesto. Esto se debe a que resulta más fácil clasificar casos que ya se han visto y con los que se ha entrenado que clasificar casos no jamás vistos y que pueden ser completamente diferentes. A lo anterior se une el hecho de que se espera que al clasificador a emplear, le vayan llegando casos de todo tipo, los cuales tendrá que clasificar. Por lo tanto una estimación del comportamiento del clasificador empleando el conjunto de casos con el que se ha entrenado, será una estimación que tenderá a valorar el clasificador como más bueno de lo que realmente es. Así que hay que buscar otra forma de estimación para



evadir este sesgo. Las técnicas de validación plantean soluciones al problema de estimar como de bueno (bondad) es un clasificador, y a continuación se exponen algunas de las más conocidas. Todas las técnicas de validación emplean al menos dos conjuntos de casos, a los que se les llaman conjunto de casos de entrenamiento y conjunto de casos de testeo. El conjunto de casos de entrenamiento el clasificador lo empleará para aprender, y el conjunto de casos de testeo se empleará para medir la calidad del clasificador.

### 11.3.1 Estimación de la bondad con el conjunto de entrenamiento

Esta es la técnica cuyos problemas se ha planteado anteriormente. Es la técnica mediante la cual se estima la bondad de un clasificador, empleando como conjunto de casos de entrenamiento y como conjunto de casos de testeo el mismo conjunto de casos.

### 11.3.2 Holdout o método H

Esta es una de las técnicas más sencillas. Teniendo un conjunto de  $m$  casos de valor de clase conocida, en este método se dejan  $r$  ( $r$  menor que  $m$ ) casos como casos de testeo y  $s$  ( $s$  igual a  $m$  menos  $r$ ) casos como casos de entrenamiento. Es decir, se hace una partición del conjunto de casos en dos particiones. Se aprende a partir de los  $s$  casos y se calcula su bondad clasificando los  $r$  casos. Esta técnica se puede generalizar ejecutando sobre el conjunto de casos varias veces el método H cada una de ellas empleando sus propias particiones del conjunto de casos en un conjunto de casos de entrenamiento y un conjunto de casos de test, resultando la bondad del clasificador la media de las bondades de las distintas ejecuciones del método H. Esta técnica y otras, se pueden refinar realizando un estratificado, acción que implica preservar la proporción de los distintos valores de la variable clase existentes en el conjunto de casos, en cada una de las distintas particiones que se vayan realizando de la base de casos (en la medida en la que sea posible).

### 11.3.3 Validación cruzada o k-fold cross validation

Esta es una técnica muy empleada. En ella el conjunto de casos se particiona en  $k$  subconjuntos del mismo tamaño (en la medida en la que sea posible), y se emplean  $k - 1$  conjuntos para entrenar, y el sobrante como conjunto de testeo. Este proceso se realiza para todas las posibles combinaciones de  $k - 1$  conjuntos que se pueden realizar extrayendo (extracción sin reemplazamiento) cada uno de ellos de los  $k$  subconjuntos en los que se ha partido el conjunto de casos (es decir, dejando cada vez un conjunto distinto como conjunto de entrenamiento). Esta técnica también se puede refinar mediante estratificado.

### 11.3.4 Bootstrapping

En ésta técnica se usa como conjunto de entrenamiento un conjunto del mismo tamaño que el conjunto de casos. Este conjunto de entrenamiento se construye realizando extracciones aleatorias con reemplazamiento del conjunto de casos (por tanto, se permite que se repitan casos en el conjunto de entrenamiento). Los casos que no sean escogidos para el conjunto de entrenamiento formaran parte del conjunto de test. La probabilidad de que un caso no sea escogido en una extracción es igual a  $1 - 1/m$  y de que no sea escogido en  $m$  extracciones es el producto de que no sea escogido en cada una de las  $m$  extracciones, por lo que  $(1 - 1/m) \times \dots \times$

$(1 - 1/m) = (1 - 1/m)^m$  y resulta que  $(1 - 1/m)^m = e^{-1} \approx 0,368$ . Por tanto la probabilidad de que un caso del conjunto de casos no este en el conjunto de entrenamiento es de 0,368, por lo que la probabilidad de que esté es de  $1 - 0,368 = 0,632$ . En esta técnica, se estima la bondad del clasificador de una forma mixta, ya que combina la bondad calculada al clasificar los casos del conjunto de entrenamiento con la bondad calculada al clasificar los casos del conjunto de test, según la siguiente fórmula, concediendo más importancia a la clasificación de casos de test que a la clasificación de los casos de entrenamiento.

$$bondad = 0,368 \cdot bondad_{entrenamiento} + 0,632 \cdot bondad_{test}$$

**Fórmula 11-17 Bondad estimada por Bootstrapping**

Repitiendo  $q$  veces este proceso de calcular un conjunto de entrenamiento, su conjunto asociado de test, y la bondad del clasificador generado por dicho conjunto de entrenamiento empleando la fórmula anterior, se realiza un promediado, quedando finalmente como estimación de la bondad del clasificador la estimación descrita por la siguiente fórmula.

$$bondad = 1/q \times \sum_{i=1}^q bondad_i$$

**Fórmula 11-18 Bondad estimada por Bootstrapping reiterado**

### 11.4 Teoría de la información estadística

La teoría de la información es una rama de las matemáticas aplicadas relacionada con la cuantificación de la información. Aquí se exponen algunos conceptos importantes relacionados con la cuantificación de la información. Para una variable aleatoria dada, se puede medir la cantidad de información aportada por cada uno de los distintos valores que puede tomar. Cuanto más probable es que al variable aleatoria tome un valor específico, menos información aporta dicho valor específico. Así pues cuando dicho valor específico es un suceso seguro, la cantidad de información aportada cuando dicho suceso sucede es cero, puesto que ya se sabía anteriormente que la probabilidad de dicho suceso implicaba que era un suceso seguro. En la siguiente fórmula se muestra la ecuación para el cálculo de la cantidad de información aportado por un suceso específico, en una variable aleatoria discreta  $X$  que puede tomar los valores  $x_1, \dots, x_n$  donde cada uno tiene asociada la probabilidad correspondiente de entre las probabilidades  $p(x_1), \dots, p(x_n)$ .

$$I(x_i) = -\log_2 p(x_i)$$

**Fórmula 11-19 Cantidad de información de un suceso**

De formar similar se puede definir la variable aleatoria que describe la cantidad de información asociada a una variable aleatoria discreta  $X$ , como aquella con posibles valores  $I(x_1), \dots, I(x_n)$  y donde cada valor tiene asociada la probabilidad correspondiente de entre las probabilidades  $p(x_1), \dots, p(x_n)$ .

Se puede medir la cantidad de incertidumbre de una variable aleatoria discreta, mediante la entropía de Shannon,  $H(x)$  que se define como la esperanza matemática de la variable aleatoria  $I(\mathbf{X})$ . En la siguiente fórmula puede observarse la ecuación que define la entropía de Shannon.

$$H(\mathbf{X}) = E(I(\mathbf{X})) = - \sum_{i=1}^n p(x_i) \cdot \log_2 p(x_i)$$

#### Fórmula 11-20 Entropía de Shannon

Con la excepción de que cuando para cada suceso  $p(x_i)$  de la variable aleatoria cuya probabilidad sea cero se reemplaza la indeterminación resultante en  $p(x_i) \log_2 p(x_i)$  por un cero. A continuación se muestran la entropía de Shannon para una variable aleatoria bidimensional conjunta (siendo ésta discreta en ambas componentes).

$$H(\mathbf{X}, \mathbf{Y}) = - \sum_{i=1}^n \sum_{j=1}^m p(x_i, y_j) \cdot \log_2 p(x_i, y_j)$$

#### Fórmula 11-21 Entropía de la variable aleatoria bidimensional conjunta

En la siguiente fórmula se muestra la entropía de Shannon para una variable aleatoria discreta condicionada a un valor específico de otra variable aleatoria discreta.

$$H(\mathbf{X}|\mathbf{Y} = y_j) = - \sum_{i=1}^n p(x_i|y_j) \cdot \log_2 p(x_i|y_j)$$

#### Fórmula 11-22 Entropía de una variable aleatoria condicionada a un valor específico de otra

También cabe mostrar, la entropía de Shannon para una variable aleatoria discreta condicionada a otra variable aleatoria discreta.

$$H(\mathbf{X}|\mathbf{Y}) = \sum_{j=1}^m p(y_j) \cdot H(\mathbf{X}|\mathbf{Y} = y_j)$$

#### Fórmula 11-23 Entropía de una variable aleatoria condicionada a otra

Cabe indicar que se verifica la equidad mostrada en la siguiente fórmula, que se conoce como ley de entropías totales.

$$H(\mathbf{X}, \mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y}|\mathbf{X})$$

#### Fórmula 11-24 Ley de entropías totales

Si  $\mathbf{X}$  e  $\mathbf{Y}$  son variables aleatorias independientes, se verifica lo siguiente.

$$H(\mathbf{X}, \mathbf{Y}) = H(\mathbf{X}) + H(\mathbf{Y}) = H(\mathbf{X}|\mathbf{Y}) + H(\mathbf{Y}) = H(\mathbf{X}|\mathbf{Y}) + H(\mathbf{Y}|\mathbf{X})$$

#### Fórmula 11-25 Propiedad relativa a la entropía de variables aleatorias independientes

Otra cuantificación de interés, es la de la distancia entre dos distribuciones de probabilidad, definidas sobre una misma variable aleatoria. La divergencia de Kullback-Leibler ofrece una solución a este problema, en la que una de las dos distribuciones actúa como referencia. A continuación en la siguiente fórmula se muestra la ecuación de la divergencia Kullback-Leibler.

$$D_{K-L}(p_1, p_2) = \sum_{i=1}^n p_1(x_i) \cdot \log \frac{p_1(x_i)}{p_2(x_i)}$$

**Fórmula 11-26 Divergencia Kullback-Leibler**

Se puede plantear un problema cuando se desea cuantificar la reducción de incertidumbre de una variable al conocer el valor de una segunda variable. La cantidad de información mutua ofrece una solución a este problema, indicando que la reducción de incertidumbre en una variable aleatoria al conocer otra variable aleatoria es igual a la entropía de Shannon de la primera variable aleatoria menos la entropía de Shannon de la primera variable aleatoria condicionada a la segunda. Dicha equidad se muestra en la siguiente fórmula.

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y, X)$$

**Fórmula 11-27 Cantidad de información mutua**

### 11.5 Cálculos matemáticos de resultados

En este anexo se precisan los cálculos matemáticos empleados para obtener los resultados. El primer cálculo a realizar es, el de resolver la indeterminación que se da en el cálculo de la precisión y la sensibilidad, cuando todas las variables de las que dependen toman valor igual a cero. Procede pues, emplear el cálculo infinitesimal para tratar de encontrar una solución

Intentando encontrar el límite doble para la sensibilidad y la precisión mediante el cálculo de los límites reiterados se observa lo siguiente (en las ecuaciones siguientes se emplean variables mudas).

$$\lim_{x \rightarrow 0} \left( \lim_{y \rightarrow 0} \frac{x}{x + y} \right) = \lim_{x \rightarrow 0} \left( \frac{x}{x + 0} \right) = \lim_{x \rightarrow 0} \left( \frac{x}{x} \right) = \lim_{x \rightarrow 0} (1)$$

$$\lim_{y \rightarrow 0} \left( \lim_{x \rightarrow 0} \frac{x}{x + y} \right) = \lim_{y \rightarrow 0} \left( \frac{0}{0 + y} \right) = \lim_{y \rightarrow 0} \left( \frac{0}{y} \right) = \lim_{y \rightarrow 0} (0)$$

**Fórmula 11-28 Límites reiterados de la sensibilidad y la precisión para el punto (0,0)**

El límite doble para la sensibilidad y la precisión no existe, al ser sus límites reiterados respectivos, de distinto valor. Por convenio se establece que el valor a tomar sea igual a cero.

El siguiente cálculo a plantear, es el de la indeterminación que se da en el cálculo del F1 score cuando todas las variables de las que depende toman valor igual a cero. Mediante cálculo infinitesimal se trata de encontrar el límite doble para el F1 score, mediante el cálculo de los límites reiterados. En esta ocasión, se observa lo siguiente (en las ecuaciones siguientes se emplean variables mudas).

$$\lim_{x \rightarrow 0} \left( \lim_{y \rightarrow 0} \frac{x \cdot y}{x + y} \right) = \lim_{x \rightarrow 0} \left( \frac{x \cdot 0}{(x + 0)^2} \right) = \lim_{x \rightarrow 0} \left( \frac{0}{x^2} \right) = \lim_{x \rightarrow 0} (0)$$

$$\lim_{y \rightarrow 0} \left( \lim_{x \rightarrow 0} \frac{x \cdot y}{x + y} \right) = \lim_{y \rightarrow 0} \left( \frac{0 \cdot y}{(0 + y)^2} \right) = \lim_{y \rightarrow 0} \left( \frac{0}{y^2} \right) = \lim_{y \rightarrow 0} (0)$$

**Fórmula 11-29 Límites reiterados del F1 score para el punto (0,0)**

En el caso del cálculo del F1 score, los límites reiterados son iguales tomando valor cero, por lo que el límite doble en caso de existir tomará valor cero. Así pues se establece por convenio que en este caso se tome un valor igual a cero como resultado.

### 11.6 Tablas de resultados

En este anexo se recogen las tablas que reflejan los resultados de la experimentación, que no se han mostrado en el capítulo de resultados.

## Tesis de máster

Preprocesamiento	Segmentación	Extracción de características	Clasificador	Media de TP	Media de FP	Media de FN	Desviación típica de TP y FN	Desviación típica de FP
1	1	1	ZeroR	0	0	28	0	0
1	1	1	IBk	2	0	26	0	0
1	1	1	JRip	0	0	28	0	0
1	1	1	J48	0	0	28	0	0
1	1	1	NaiveBayes	0	0	28	0	0
1	1	1	BayesNet	0	0	28	0	0
2	1	1	ZeroR	0	0	28	0	0
2	1	1	IBk	1	1,1	27	0	0,32
2	1	1	JRip	0	1,9	28	0	0,32
2	1	1	J48	0	0	28	0	0
2	1	1	NaiveBayes	0	1	28	0	0
2	1	1	BayesNet	0	1,8	28	0	0,42
3	1	1	ZeroR	0	0	28	0	0
3	1	1	IBk	18,2	5,9	9,8	0,42	0,74
3	1	1	JRip	14,7	4,2	13,3	0,95	1,87
3	1	1	J48	14,7	2,5	13,3	0,67	1,35
3	1	1	NaiveBayes	20	3,9	8	0	0,57
3	1	1	BayesNet	19,9	7	8,1	0,32	1,41
4	1	1	ZeroR	0	0	28	0	0
4	1	1	IBk	10,1	2,1	17,9	0,57	0,57
4	1	1	JRip	6,2	5,9	21,8	1,48	1,6
4	1	1	J48	7,6	3,6	20,4	1,07	1,35
4	1	1	NaiveBayes	10	65,3	18	0	1,25
4	1	1	BayesNet	10	110,2	18	0	1,75
5	1	1	ZeroR	0	0	28	0	0
5	1	1	IBk	19,2	3,8	8,8	0,42	0,92
5	1	1	JRip	15,2	4,	12,8	0,42	1,43
5	1	1	J48	14,4	2,8	13,6	0,52	1,14
5	1	1	NaiveBayes	19,9	3,3	8,1	0,32	0,48
5	1	1	BayesNet	19,9	4	8,1	0,32	0,47

Tabla 11-1 Experimento 1: Medias y desviaciones típicas de TP, FP y FN

## Tesis de máster

Preprocesamiento	Segmentación	Extracción de características	Clasificador	Media de precisión	Media de sensibilidad	Media de F1 score	Desviación típica de precisión	Desviación típica de sensibilidad	Desviación típica de F1 score
1	1	1	ZeroR	0	0	0	0	0	0
1	1	1	IBk	1	0,07	0,07	0	0	0
1	1	1	JRip	0	0	0	0	0	0
1	1	1	J48	0	0	0	0	0	0
1	1	1	NaiveBayes	0	0	0	0	0	0
1	1	1	BayesNet	0	0	0	0	0	0
2	1	1	ZeroR	0	0	0	0	0	0
2	1	1	IBk	0,48	0,04	0,03	0,05	0	0
2	1	1	JRip	0	0	0	0	0	0
2	1	1	J48	0	0	0	0	0	0
2	1	1	NaiveBayes	0	0	0	0	0	0
2	1	1	BayesNet	0	0	0	0	0	0
3	1	1	ZeroR	0	0	0	0	0	0
3	1	1	IBk	0,76	0,65	0,35	0,03	0,02	0,01
3	1	1	JRip	0,78	0,53	0,31	0,08	0,03	0,02
3	1	1	J48	0,86	0,53	0,33	0,06	0,02	0,01
3	1	1	NaiveBayes	0,84	0,71	0,39	0,02	0	0
3	1	1	BayesNet	0,74	0,71	0,36	0,04	0,01	0,01
4	1	1	ZeroR	0	0	0	0	0	0
4	1	1	IBk	0,83	0,36	0,25	0,04	0,02	0,01
4	1	1	JRip	0,51	0,22	0,15	0,09	0,05	0,03
4	1	1	J48	0,68	0,27	0,19	0,07	0,04	0,02
4	1	1	NaiveBayes	0,13	0,36	0,1	0	0	0
4	1	1	BayesNet	0,08	0,36	0,07	0	0	0
5	1	1	ZeroR	0	0	0	0	0	0
5	1	1	IBk	0,84	0,69	0,38	0,04	0,02	0,01
5	1	1	JRip	0,78	0,54	0,32	0,06	0,02	0,01
5	1	1	J48	0,84	0,51	0,32	0,05	0,02	0,01
5	1	1	NaiveBayes	0,86	0,71	0,39	0,02	0,01	0
5	1	1	BayesNet	0,83	0,71	0,38	0,02	0,01	0,01

Tabla 11-2 Experimento 1: Medias y desviaciones típicas de precisión, sensibilidad y F1 score

## Tesis de máster

---

Preprocesamiento	1	2	3	4	5
1	X	I	P	I	P
2	I	X	P	P	P
3	M	M	X	M	I
4	I	M	P	X	P
5	M	M	I	M	X

**Tabla 11-3 Experimento 1: Contraste de preprocesamientos según precisión teniendo en cuenta todos los clasificadores**

Preprocesamiento	1	2	3	4	5
1	X	I	P	I	P
2	I	X	P	P	P
3	M	M	X	M	I
4	I	M	P	X	P
5	M	M	I	M	X

**Tabla 11-4 Experimento 1: Contraste de preprocesamientos según precisión teniendo en cuenta los clasificadores “inteligentes”**

Preprocesamiento	1	2	3	4	5
1	X	I	P	P	P
2	I	X	P	P	P
3	M	M	X	M	I
4	M	M	P	X	P
5	M	M	I	M	X

**Tabla 11-5 Experimento 1: Contraste de preprocesamientos según sensibilidad teniendo en cuenta todos los clasificadores**

Preprocesamiento	1	2	3	4	5
1	X	I	P	P	P
2	I	X	P	P	P
3	M	M	X	M	I
4	M	M	P	X	P
5	M	M	I	M	X

**Tabla 11-6 Experimento 1: Contraste de preprocesamientos según sensibilidad teniendo en cuenta los clasificadores “inteligentes”**

Preprocesamiento	1	2	3	4	5
1	X	I	P	P	P
2	I	X	P	P	P
3	M	M	X	M	I
4	M	M	P	X	P
5	M	M	I	M	X

**Tabla 11-7 Experimento 1: Contraste de preprocesamientos según F1 score teniendo en cuenta todos los clasificadores**



Preprocesamiento	1	2	3	4	5
1	X	I	P	P	P
2	I	X	P	P	P
3	M	M	X	M	I
4	M	M	P	X	P
5	M	M	I	M	X

**Tabla 11-8 Experimento 1: Contraste de preprocesamientos según F1 score teniendo en cuenta los clasificadores “inteligentes”**

Preprocesamiento	1	2	3	4	5
1	X	I	P	P	P
2	I	X	P	P	P
3	M	M	X	M	I
4	M	M	P	X	P
5	M	M	I	M	X

**Tabla 11-9 Experimento 1: Superioridad completa entre preprocesamientos teniendo en cuenta todos los clasificadores**

Preprocesamiento	1	2	3	4	5
1	X	I	P	P	P
2	I	X	P	P	P
3	M	M	X	M	I
4	M	M	P	X	P
5	M	M	I	M	X

**Tabla 11-10 Experimento 1: Superioridad completa entre preprocesamientos teniendo en cuenta los clasificadores “inteligentes”**

## Tesis de máster

Preprocesamiento	Segmentación	Extracción de características	Clasificador	Media de <i>TP</i>	Media de <i>FP</i>	Media de <i>FN</i>	Desviación típica de <i>TP</i> y <i>FN</i>	Desviación típica de <i>FP</i>
5	1	1	ZeroR	0	0	28	0	0
5	1	1	IBk	17	7,6	11	0	1,35
5	1	1	JRip	14	6,2	14	1,05	1,99
5	1	1	J48	13	5	15	1,05	1,05
5	1	1	NaiveBayes	20	16,8	8	0	1,4
5	1	1	BayesNet	20	30,6	8	0	1,71
5	2	1	ZeroR	0	0	28	0	0
5	2	1	IBk	17,2	6,7	10,8	0,42	0,95
5	2	1	JRip	13,9	6,8	14,1	1,45	2,1
5	2	1	J48	12,2	5,9	15,8	1,03	1,37
5	2	1	NaiveBayes	20	16,8	8	0	1,99
5	2	1	BayesNet	19	23,9	9	0	1,6
5	3	1	ZeroR	0	0	28	0	0
5	3	1	IBk	17	7,1	11	0,47	0,74
5	3	1	JRip	14	7,1	14	1,33	1,85
5	3	1	J48	12,6	5,6	15,4	0,84	2,5
5	3	1	NaiveBayes	20	17,1	8	0	1,45
5	3	1	BayesNet	19	21,9	9	0	0,74

Tabla 11-11 Experimento 2: Medias y desviaciones típicas de TP, FP y FN

## Tesis de máster

Preprocesamiento	Segmentación	Extracción de características	Clasificador	Media de precisión	Media de sensibilidad	Media de F1 score	Desviación típica de precisión	Desviación típica de sensibilidad	Desviación típica de F1 score
5	1	1	ZeroR	0	0	0	0	0	0
5	1	1	IBk	0,69	0,61	0,32	0,04	0	0,01
5	1	1	JRip	0,7	0,5	0,29	0,07	0,04	0,02
5	1	1	J48	0,72	0,46	0,28	0,04	0,04	0,02
5	1	1	NaiveBayes	0,54	0,71	0,31	0,02	0	0,01
5	1	1	BayesNet	0,4	0,71	0,25	0,01	0	0,01
5	2	1	ZeroR	0	0	0	0	0	0
5	2	1	IBk	0,72	0,61	0,33	0,03	0,02	0,01
5	2	1	JRip	0,68	0,5	0,29	0,07	0,05	0,02
5	2	1	J48	0,68	0,44	0,26	0,06	0,04	0,02
5	2	1	NaiveBayes	0,54	0,71	0,31	0,03	0	0,01
5	2	1	BayesNet	0,44	0,68	0,27	0,02	0	0,01
5	3	1	ZeroR	0	0	0	0	0	0
5	3	1	IBk	0,71	0,61	0,33	0,02	0,02	0,01
5	3	1	JRip	0,67	0,5	0,28	0,04	0,05	0,01
5	3	1	J48	0,7	0,45	0,27	0,09	0,03	0,02
5	3	1	NaiveBayes	0,54	0,71	0,31	0,02	0	0,01
5	3	1	BayesNet	0,46	0,68	0,28	0,01	0	0

Tabla 11-12 Experimento 2: Medias y desviaciones típicas de precisión, sensibilidad y F1 score

# Tesis de máster

---

Segmentación	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-13 Experimento 2: Contraste de segmentaciones según precisión teniendo en cuenta todos los clasificadores

Segmentación	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-14 Experimento 2: Contraste de segmentaciones según precisión teniendo en cuenta los clasificadores "inteligentes"

Segmentación	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-15 Experimento 2: Contraste de segmentaciones según sensibilidad teniendo en cuenta todos los clasificadores

Segmentación	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-16 Experimento 2: Contraste de segmentaciones según sensibilidad teniendo en cuenta los clasificadores "inteligentes"

Segmentación	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-17 Experimento 2: Contraste de segmentaciones según F1 score teniendo en cuenta todos los clasificadores

Segmentación	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-18 Experimento 2: Contraste de segmentaciones según F1 score teniendo en cuenta los clasificadores "inteligentes"

## Tesis de máster

---

Segmentación	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

**Tabla 11-19 Experimento 2: Superioridad completa entre segmentaciones teniendo en cuenta todos los clasificadores**

Segmentación	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

**Tabla 11-20 Experimento 2: Superioridad completa entre segmentaciones teniendo en cuenta los clasificadores "inteligentes"**

## Tesis de máster

Preprocesamiento	Segmentación	Extracción de características	Clasificador	Media de <i>TP</i>	Media de <i>FP</i>	Media de <i>FN</i>	Desviación típica de <i>TP</i> y <i>FN</i>	Desviación típica de <i>FP</i>
5	3	1	ZeroR	0	0	28	0	0
5	3	1	IBk	17	7,1	11	0,47	0,74
5	3	1	JRip	14	7,1	14	1,33	1,85
5	3	1	J48	12,6	5,6	15,4	0,84	2,5
5	3	1	NaiveBayes	20	17,1	8	0	1,45
5	3	1	BayesNet	19	21,9	9	0	0,74
5	3	2	ZeroR	0	0	28	0	0
5	3	2	IBk	15,1	12,8	12,9	0,88	1,03
5	3	2	JRip	14	6,3	14	1,41	1,83
5	3	2	J48	12,4	5,9	15,6	1,07	1,52
5	3	2	NaiveBayes	20	21,1	8	0	1,45
5	3	2	BayesNet	19,2	27	8,8	0,42	1,94
5	3	3	ZeroR	0	0	28	0	0
5	3	3	IBk	15,3	6,9	12,7	0,48	0,74
5	3	3	JRip	12,7	5,2	15,3	1,25	1,99
5	3	3	J48	10,5	5,9	17,5	0,71	3,03
5	3	3	NaiveBayes	20	38,9	8	0	2,77
5	3	3	BayesNet	20	33,6	8	0	1,84

Tabla 11-21 Experimento 3: Medias y desviaciones típicas de TP, FP y FN

## Tesis de máster

Preprocesamiento	Segmentación	Extracción de características	Clasificador	Media de precisión	Media de sensibilidad	Media de F1 score	Desviación típica de precisión	Desviación típica de sensibilidad	Desviación típica de F1 score
5	3	1	ZeroR	0	0	0	0	0	0
5	3	1	IBk	0,71	0,61	0,33	0,02	0,02	0,01
5	3	1	JRip	0,67	0,5	0,28	0,04	0,05	0,01
5	3	1	J48	0,7	0,45	0,27	0,09	0,03	0,02
5	3	1	NaiveBayes	0,54	0,71	0,31	0,02	0	0,01
5	3	1	BayesNet	0,46	0,68	0,28	0,01	0	0
5	3	2	ZeroR	0	0	0	0	0	0
5	3	2	IBk	0,54	0,54	0,27	0,02	0,03	0,01
5	3	2	JRip	0,69	0,5	0,29	0,06	0,05	0,02
5	3	2	J48	0,68	0,44	0,27	0,05	0,04	0,01
5	3	2	NaiveBayes	0,49	0,71	0,29	0,02	0	0,01
5	3	2	BayesNet	0,42	0,69	0,26	0,02	0,02	0,01
5	3	3	ZeroR	0	0	0	0	0	0
5	3	3	IBk	0,69	0,55	0,3	0,03	0,02	0,01
5	3	3	JRip	0,71	0,45	0,28	0,09	0,04	0,03
5	3	3	J48	0,66	0,38	0,24	0,13	0,03	0,02
5	3	3	NaiveBayes	0,34	0,71	0,23	0,02	0	0,01
5	3	3	BayesNet	0,37	0,71	0,25	0,01	0	0,01

Tabla 11-22 Experimento 3: Medias y desviaciones típicas de precisión, sensibilidad y F1 score

# Tesis de máster

---

Extracción de características	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-23 Experimento 3: Contraste de extracciones de características según precisión teniendo en cuenta todos los clasificadores

Extracción de características	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-24 Experimento 3: Contraste de extracciones de características según precisión teniendo en cuenta los clasificadores “inteligentes”

Extracción de características	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-25 Experimento 3: Contraste de extracciones de características según sensibilidad teniendo en cuenta todos los clasificadores

Extracción de características	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-26 Experimento 3: Contraste de extracciones de características según sensibilidad teniendo en cuenta los clasificadores “inteligentes”

Extracción de características	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

Tabla 11-27 Experimento 3: Contraste de extracciones de características según F1 score teniendo en cuenta todos los clasificadores

Extracción de características	1	2	3
1	X	I	M
2	I	X	I
3	P	I	X

Tabla 11-28 Experimento 3: Contraste de extracciones de características según F1 score teniendo en cuenta los clasificadores “inteligentes”



## Tesis de máster

Extracción de características	1	2	3
1	X	I	I
2	I	X	I
3	I	I	X

**Tabla 11-29 Experimento 3: Superioridad completa entre extracciones de características teniendo en cuenta todos los clasificadores**

Extracción de características	1	2	3
1	X	I	M
2	I	X	I
3	P	I	X

**Tabla 11-30 Experimento 3: Superioridad completa entre extracciones de características teniendo en cuenta los clasificadores “inteligentes”**

Clasificador	ZeroR	IBk	JRip	J48	NaiveBayes	BayesNet
ZeroR	X	P	P	P	P	P
IBk	M	X	M	I	M	M
JRip	M	P	X	I	M	M
J48	M	I	I	X	M	M
NaiveBayes	M	P	P	P	X	M
BayesNet	M	P	P	P	P	X

**Tabla 11-31 Experimento 3: Contraste de clasificadores según precisión**

Clasificador	ZeroR	IBk	JRip	J48	NaiveBayes	BayesNet
ZeroR	X	P	P	M	P	P
IBk	M	X	M	M	P	P
JRip	M	P	X	M	P	P
J48	M	P	P	X	P	P
NaiveBayes	M	M	M	M	X	M
BayesNet	M	M	M	M	P	X

**Tabla 11-32 Experimento 3: Contraste de clasificadores según sensibilidad**

Clasificador	ZeroR	IBk	JRip	J48	NaiveBayes	BayesNet
ZeroR	X	P	P	P	P	P
IBk	M	X	M	M	M	M
JRip	M	P	X	M	P	M
J48	M	P	P	X	P	I
NaiveBayes	M	P	M	M	X	M
BayesNet	M	P	P	I	P	X

**Tabla 11-33 Experimento 3: Contraste de clasificadores según F1 score**

Clasificador	ZeroR	IBk	JRip	J48	NaiveBayes	BayesNet
ZeroR	X	P	P	P	P	P
IBk	M	X	M	M	I	I
JRip	M	P	X	M	I	I
J48	M	P	P	X	I	I
NaiveBayes	M	I	I	I	X	M
BayesNet	M	I	I	I	P	X

Tabla 11-34 Experimento 3: Superioridad completa entre clasificadores



## 12 Bibliografía

Arff [En línea]. - <http://www.cs.waikato.ac.nz/~ml/weka/arff.html>.

**Cohen W. W.** Fast effective rule induction [Conferencia] // Proceedings of the Twelfth International Conference on Machine Learning. - 1995.

**Fawcett T.** Cours [En línea]. - [https://cours.etsmtl.ca/sys828/REFS/A1/Fawcett\\_PRL2006.pdf](https://cours.etsmtl.ca/sys828/REFS/A1/Fawcett_PRL2006.pdf).

**Froufe A.** Java 2 Manual de usuario y tutorial [Libro]. - [s.l.] : Ra-Ma, 2005. - 4ª. - 84-7897-679-5.

**Hernández E.** Álgebra y geometría [Libro]. - Madrid : Addison-Wesley/Universidad Autónoma de Madrid, 1994. - 2ª Edición. - 978-84-7829-024-6.

**Hernando B.** Problemas sobre espacios métricos, normados y de Hilbert [Libro]. - Madrid : UNED, 1999. - 1ª Edición. - 84-362-3949-0.

Imagej-Descarga [En línea]. - <http://rsbweb.nih.gov/ij/download.html>.

Imagej-Documentación [En línea]. - <http://rsbweb.nih.gov/ij/docs/index.html>.

Imagej-Principal [En línea]. - <http://rsbweb.nih.gov/ij/>.

**J. F. Velez A. B. Moreno, A. Sánchez, J. L. Esteban** Visión por computador [Libro]. - 2007. - 2ª Edición.

**J. Fürnkranz G. Widner** Incremental Reduced Error Pruning [Conferencia] // Proceedings of the Eleventh International Conference on Machine Learning. - 1994.

**J. Sánchez G. Huecas, B. Fernández, P. Moreno** Java 2 [Libro]. - [s.l.] : McGraw-Hill, 2005. - 2ª. - 84-481-9816-6.

**J.M. Pérez G. Madariaga, J. Etxebarria, J. Ortega, R.B. Pérez** Fundamentos de técnicas experimentales. - Bilbao : [s.n.], 2007. - Apuntes del departamento de física de la materia condensada de la UPV/EHU.

**Larman C.** UML y Patrones [Libro]. - [s.l.] : Pearson Prentice Hall, 2003. - 2ª. - 978-84-205-3438-1.

**M. T. González A. P. De Vargas** Estadística Aplicada Una visión instrumental [Libro]. - [s.l.] : Diaz de Santos, 2004. - 978-84-7978-913-8.

**Michalski R.S.** On the quasi-minimal solution of the general covering problem [Conferencia] // In Proceedings of the First International Symposium of Information Processing. - Bled, Yugoslavia : [s.n.], 1969. - págs. 125-128.

**Novoa J. F.** Análisis Matemático 1 [Libro]. - Madrid : UNED, 1991. - 4ª Edición : Vol. II : II. - 84-362-1667-9.

## Tesis de máster

---

**Novoa J.F.** Análisis Matemático 1 [Libro]. - Madrid : UNED, 1991. - 4ª Edición : Vol. I : II. - 84-362-1667-9.

**P. A. Tipler G. Mosca** Física para la ciencia y la tecnología [Libro]. - Barcelona : Reverté, 2005. - 5ª Edición : Vol. I : II. - 84-291-4411-0.

**P. A. Tipler G. Mosca** Física para la ciencia y la tecnología [Libro]. - Barcelona : Reverté, 2005. - 5ª Edición : Vol. II : II. - 84-291-4412-9.

**P. Clark T. Nibblet** The CN2 Induction Algorithm [Publicación periódica] // Machine Learning Journal 3. - 1988. - págs. 261-281.

**Quinlan J. R.** C4.5: Programs for Machine Learning [Libro]. - [s.l.] : Morgan Kaufmann, 1993.

**Quinlan J. R.** Induction of decision trees [Publicación periódica] // Machine Learning 1. - 1986. - págs. 81-106.

**R. C. González R. E. Woods** Digital Image Processing [Libro]. - Upper Saddle River : Pearson Prentice-Hall, 2002. - 2ª Edición.

**R. Jain R. Kasturi, B. G. Schunck** Machine Vision [Libro]. - Nueva York : McGraw-Hill, 1995. - 0-07-032018-7.

Real Academia Española [En línea]. - <http://www.rae.es/rae.html>.

**S. Russell P. Norvig** Inteligencia Artificial Un Enfoque Moderno [Libro]. - Madrid : Pearson Prentice-Hall, 2004. - 2ª Edición. - 0-13-790395-2.

**Sierra B.** Aprendizaje Automático: conceptos básicos y avanzados [Libro]. - Madrid : Pearson Prentice-Hall, 2006. - 978-84-8322-318-5.

Weka - Descarga [En línea]. - [http://www.cs.waikato.ac.nz/ml/weka/index\\_downloading.html](http://www.cs.waikato.ac.nz/ml/weka/index_downloading.html).

Weka - Documentación [En línea]. - [http://www.cs.waikato.ac.nz/ml/weka/index\\_documentation.html](http://www.cs.waikato.ac.nz/ml/weka/index_documentation.html).

Weka - Principal [En línea]. - <http://www.cs.waikato.ac.nz/ml/weka/>.

Wikipedia [En línea]. - <http://www.wikipedia.org/>.