



Universidad  
Carlos III de Madrid

Departamento de Teoría de la Señal y Comunicaciones.

## PROYECTO FIN DE CARRERA Ingeniería de Telecomunicación

# DETECCIÓN Y ESTIMACIÓN EN REDES DE SENSORES INALÁMBRICAS CON CENTRO DE FUSIÓN

Autor: José Luis Barbosa Moreno

Tutor: David Luengo García

Leganes, 27 de Enero de 2011



Título:  
Autor:  
Director:

EL TRIBUNAL

Presidente: Ángel María Bravo Santos

Vocal: Elisa María Molanes López

Secretario: Marcelino Lázaro Teja

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 27 de Enero de 2011 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

SECRETARIO

PRESIDENTE

VOCAL



# Agradecimientos

Quiero dar mi más sincero agradecimiento al profesor David Luengo García por haberme dirigido este proyecto final de carrera, por todo el tiempo que me ha dedicado, por su cercanía y amabilidad.

Este proyecto pone punto y final a mi carrera universitaria, por eso aprovecho para mencionar en estos agradecimientos a quienes han estado a mi lado durante todo este tiempo. No pueden faltar ni Miriam, ni mi familia, apoyo inagotable e incalculable en la distancia. Mención especial para mi abuelo Curro, al que estoy seguro hubiese hecho realmente orgulloso.

Han sido unos años en los que he disfrutado y sufrido con esta carrera, y con el ambiente de esta universidad y de su residencia. Un tiempo que da paso a nuevas experiencias pero que deja grandes compañeros de viaje y mejores amigos, como Antonio y Kiko, que forman parte importante de mi vida.

No puedo olvidar a Blanca, que me ayudó a entender en el momento adecuado las SVMs, ni a la oficina técnica del Departamento de Electrónica, en especial a Jesús. Gracias por vuestra comprensión y por todas las facilidades que me habéis dado.



# Resumen

El presente proyecto trata de analizar las redes de sensores inalámbricas y la posible detección y estimación descentralizada a partir de la información obtenida por los sensores de dichas redes. Para ello se ha considerado un escenario en el cual los sensores están distribuidos de modo uniforme por toda el área monitorizada, en la que se pretende detectar la presencia de un blanco móvil.

Concretamente, se ha implementado en Matlab, mediante el uso de una máquina de vectores soporte (SVM), un detector global de presencia que coincide con el centro de fusión de datos de la red. La regla de decisión global de este centro se genera en base a las decisiones locales tomadas por cada nodo-sensor.

La necesidad de tomar las decisiones con carácter local, y de este modo obligar a la inferencia descentralizada, tiene como origen las limitaciones que se imponen a la capacidad del canal de comunicaciones entre los sensores y el centro de fusión, tratando de reducir al mínimo (hasta 1 bit) la información que cada nodo puede enviar de forma paralela al mencionado centro.

Por ello, cada sensor dispone de una regla de decisión local establecida bajo el criterio de Neyman-Pearson, que nos garantiza unos niveles programables de probabilidad de error a partir de los cuales se obtienen umbrales de potencia con los que comparar la potencia recibida en un entorno con ruido Gaussiano.

Posteriormente se modifica el escenario para dar la oportunidad a cada sensor de abstenerse de enviar información si considera que su información es confusa, aumentando indirectamente la información que recibe el centro de fusión.

Una vez obtenidos y evaluados los detectores globales, el proyecto se completa con una primera aproximación a la estimación, tratando de comprobar en qué medida es posible estimar la posición y recrear la trayectoria una vez detectado el blanco.

**Palabras clave:** Redes de sensores inalámbricas (WSN), detección distribuida con centro de fusión, criterio de Neyman-Pearson, máquinas de vectores soporte (SVM).



# Abstract

The main goal of this project is analyzing Wireless Sensor Networks and how decentralized detection and estimation can be performed in them thanks to the sensed information. In order to achieve this goal, an environment where sensors are uniformly distributed inside the supervised area has been considered. Detecting a potential mobile target crossing mentioned area is the ultimate purpose in this scenario.

In particular a global detector which corresponds to the data fusion center has been implemented in Matlab using Support Vectors Machines (SVMs). In this way a global decision rule is generated based on local decisions made by each sensor-node.

Due to the established limits on the communications channel capacity between the sensors and the fusion center, local decisions are required, as well as decentralized inference. The scheme is designed to try to reduce it as much as possible (up to 1 bit) information sent in a parallel way from each node to the mentioned fusion center.

Hence, each single node performs its decision according to a local detection rule obtained under Neyman-Pearson criterion. This criterion guarantees fixed probability of error levels from which power thresholds are derived and used to be compared with the received power in a Gaussian noise environment.

Later on, this scenario is modified to provide sensors the opportunity of refraining from transmitting information if they consider that their local information is confusing. This means that amount of information received by the fusion center increases indirectly.

Once global detectors have been obtained and evaluated, the project is completed with a first approach to estimation. The purpose now is taking advantage of previous work to check whether it's possible or not to estimate the target's position and reconstruct the followed path when the target has already been detected.

**Keywords:** Wireless Sensor Networks (WSNs), distributed detection with fusion center, Neyman-Pearson criterion, Support Vector Machine (SVM).



# Índice general

<b>1. INTRODUCCIÓN Y DESCRIPCIÓN DEL PROBLEMA .....</b>	<b>1</b>
1.1 Introducción.....	1
1.2 Descripción del problema.....	2
1.3 Fases del desarrollo.....	2
1.4 Medios empleados .....	3
1.5 Estructura de la memoria.....	3
<b>2. REDES DE SENSORES .....</b>	<b>5</b>
2.1 Introducción.....	5
2.2 Evolución histórica .....	5
2.3 Componentes del nodo .....	8
2.3.1 <i>Sensores y actuadores</i> .....	9
2.3.2 <i>Transceptor</i> .....	9
2.3.3 <i>Microcontrolador y memoria</i> .....	10
2.3.4 <i>Batería y administrador de la energía</i> .....	10
2.4 Arquitectura de un nodo sensor .....	10
2.5 Topologías .....	12
2.6 Requisitos de una WSN y posibles retos.....	13
2.6.1 <i>Escalabilidad</i> .....	14
2.6.2 <i>Consumo de energía</i> .....	14
2.6.3 <i>Enrutado</i> .....	15
2.6.4 <i>Restricciones hardware de memoria</i> .....	15
2.6.5 <i>Agregación y disseminación automática de datos</i> .....	16
2.6.6 <i>Topología. Auto organización de la red</i> .....	16
2.6.7 <i>Peticiones de información</i> .....	16
2.6.8 <i>Entorno y seguridad</i> .....	16
2.7 Redes Ad-hoc frente a redes con centro de fusión.....	17
2.7.1 <i>Redes con centro de fusión</i> .....	17
2.7.2 <i>Redes Ad-hoc con procesado interno</i> .....	18
2.8 Aplicaciones .....	19
2.8.1 <i>Infraestructuras de seguridad</i> .....	19
2.8.2 <i>Monitorización del entorno y el hábitat</i> .....	20
2.8.3 <i>Aplicaciones a la agricultura</i> .....	20
2.8.4 <i>Sensores en el entorno industrial</i> .....	21
2.8.5 <i>Sensores en el control de tráfico</i> .....	21
2.8.6 <i>Apoyo logístico</i> .....	21
2.8.7 <i>Aplicaciones centradas en los humanos</i> .....	22
2.8.8 <i>Aplicaciones en robótica</i> .....	22
2.9 Discusión .....	23
<b>3. DETECCIÓN, CLASIFICACIÓN Y ESTIMACIÓN EN WSN.....</b>	<b>25</b>
3.1 Introducción.....	25
3.2 Detección/Clasificación .....	27
3.2.1 <i>Modo de trabajo general</i> .....	27
3.2.2 <i>Un problema simple: comunicación binaria</i> .....	28

3.2.3	<i>El criterio de Neyman-Pearson</i> .....	33
3.2.4	<i>El criterio de Bayes</i> .....	36
3.2.5	<i>Criterio Minimax</i> .....	38
3.2.6	<i>Estadístico suficiente</i> .....	40
3.2.6	<i>Test de hipótesis compuestas</i> .....	41
3.2.7	<i>Clasificación</i> .....	43
3.3	<i>Estimación</i> .....	46
3.3.1	<i>Tipos de problemas de estimación</i> .....	47
3.3.2	<i>Estimación Bayesiana</i> .....	47
3.3.3	<i>Estimador de Máxima Verosimilitud (ML)</i> .....	51
3.4	<i>Discusión</i> .....	52
<b>4.</b>	<b>DETECCIÓN/ESTIMACIÓN DISTRIBUIDA EN WSNs CON CENTRO DE FUSIÓN</b> .....	<b>55</b>
4.1	<i>Introducción</i> .....	55
4.2	<i>Aprendizaje e inferencia distribuido</i> .....	56
4.2.1	<i>Un modelo general</i> .....	56
4.2.2	<i>Modelos paramétricos y no paramétricos</i> .....	57
4.2.3	<i>Un ejemplo concreto</i> .....	62
4.3	<i>Aprendizaje distribuido en WSNs ad-hoc con procesado en la propia red</i> .....	63
4.4	<i>Aprendizaje distribuido en WSNs con un centro de fusión</i> .....	66
4.5	<i>Limites estadísticos del aprendizaje distribuido</i> .....	67
4.6	<i>Discusión</i> .....	71
<b>5.</b>	<b>DESCRIPCIÓN DE LOS MÉTODOS USADOS EN DETECCIÓN/ESTIMACIÓN</b> .....	<b>73</b>
5.1	<i>Introducción</i> .....	73
5.2	<i>El escenario</i> .....	74
5.2.1	<i>Los sensores</i> .....	74
5.2.2	<i>Las trayectorias</i> .....	79
5.3	<i>Detección</i> .....	80
5.3.1	<i>Trabajo con los sensores</i> .....	80
5.3.2	<i>Detección en el centro de fusión</i> .....	88
5.4	<i>Estimación</i> .....	103
5.4.1	<i>Método escogido para la estimación de la posición</i> .....	105
5.4.2	<i>Modificaciones al método</i> .....	107
5.4.3	<i>Error, esperanza y sesgo</i> .....	108
5.5	<i>Discusión</i> .....	109
<b>6.</b>	<b>RESULTADOS</b> .....	<b>111</b>
6.1	<i>Introducción</i> .....	111
6.2	<i>Aspectos comunes a los dos escenarios</i> .....	112
6.2.1	<i>El área y los sensores</i> .....	112
6.2.2	<i>Distribuciones de las señales</i> .....	112
6.2.3	<i>El Kernel</i> .....	117
6.3	<i>Despliegue del primer escenario: clasificador binario</i> .....	117
6.4	<i>Despliegue del segundo escenario: considerando abstención</i> .....	118
6.5	<i>Generando los datos</i> .....	122
6.5.1	<i>Trayectorias tipo</i> .....	122
6.5.2	<i>Comparativa de proporciones de datos</i> .....	123
6.6	<i>Precisión, sensibilidad y especificidad</i> .....	124
6.6.1	<i>Comparativa de los dos escenarios para <math>P_{fa}=P_{nd}= 0,01</math></i> .....	125
6.6.2	<i>Comparativa de los dos escenarios para <math>P_{fa}=P_{nd}= 0,1</math></i> .....	126
6.7	<i>El Kernel Gaussiano como alternativa</i> .....	127
6.8	<i>La estimación</i> .....	131
6.8.1	<i>Estimación sin abstención</i> .....	131
6.8.2	<i>Estimación con abstención</i> .....	137
6.8.3	<i>Estimación sin limitación en el número de bit transmitidos</i> .....	141

6.8.4 Comparación de escenarios.....	142
6.9 Discusión.....	147
<b>7. CONCLUSIONES Y LÍNEAS FUTURAS .....</b>	<b>149</b>
7.1 Introducción.....	149
7.2 Conclusiones.....	149
7.3 Posibles mejoras y líneas futuras .....	151
7.3.1 Ampliación del tipo de trayectorias .....	151
7.3.2 El canal binario simétrico.....	152
7.3.3 Mejoras del método de estimación.....	153
7.3.4 Consideración de otros escenarios.....	153
<b>8. PRESUPUESTO DEL PROYECTO.....</b>	<b>155</b>
A.1 Coste de los medios materiales.....	155
A.2 Coste del personal .....	156
A.3 Coste de la dirección .....	156
A.4 Coste total del proyecto.....	156
<b>9. REFERENCIAS.....</b>	<b>157</b>



# Índice de figuras

Figura 1. Componentes de un nodo de una WSN [9].....	8
Figura 2. Pila de protocolos típica de un sensor [13]. .....	11
Figura 3. Topologías típicas de una WSN[14].....	13
Figura 4. Factores que intervienen en el diseño de una WSN. ....	14
Figura 5. Modelo de ejemplo para el ahorro del consumo [1]. .....	15
Figura 6. Topología de una WSNs con centro de fusión [21].....	18
Figura 7. Topología de una WSNs sin centro de fusión [21].....	19
Figura 8. Imagen del prototipo de guante con sensores y movimientos detectados [34].	23
Figura 9. Disposición matricial de los datos. ....	26
Figura 10. Etapas en el diseño de un clasificador. ....	28
Figura 11. Regiones de probabilidad de error para un test de hipótesis binario [37].....	32
Figura 12. Funciones de densidad condicionadas y regiones de clasificación [37].....	46
Figura 13. Función de coste cuadrática para el estimador MMSE [37]. ....	49
Figura 14. Función de coste uniforme para el estimador MAP [37]. ....	50
Figura 15. Producto coste uniforme por probabilidad condicionada. ....	50
Figura 16. (a) Red paralela con un centro de fusión. (b) Red “Ad-hoc” con procesado interno [21]. ....	56
Figura 17. Comparativa de las funciones de Kernel: Lineal, “Naive” y Gaussiana para una dimensión.....	62
Figura 18. Entrenamiento distribuido basando en subgradiente incremental [21].....	64
Figura 19. Entrenamiento distribuido basado en proyecciones alternadas [21].....	65
Figura 20. Ejemplo de distribución aleatoria uniforme de 300 sensores sobre un área sobre un área de 200m x 200m. ....	74
Figura 21. Curvas de nivel de SNR (dB) obtenidas supuesto un blanco en el centro del área normalizada. Se representan desde el centro hacia fuera 30, 20, 10, 5, 0, -5 y -10 decibelios. ....	78
Figura 22. Trayectorias generadas de modo aleatorio sin restricciones. ....	79
Figura 23. Trayectorias contempladas y trayectorias que se excluyen (en rojo) por comenzar y terminar en la misma frontera (a). Trayectorias contempladas tras la restricción (b).....	79
Figura 24. Trayectorias contempladas tras la primera restricción y trayectorias que se excluyen (en rojo) por no cruzar el área de modo significativo (a). 6 Trayectorias supervivientes y que se contemplan en el escenario desplegado (b). ....	80
Figura 25. Escenario con umbral único dependiente de la probabilidad de falsa alarma.	81
Figura 26. Dos configuraciones posibles del escenario formado a partir de dos umbrales. ....	81
Figura 27. Comparativa de radios de detección y no detección para los dos escenarios posibles.....	88
Figura 28. Esquema de trabajo de una SVM: transformación de los datos a otra dimensión donde se trabaja y crea la frontera lineal y transformación de vuelta al espacio de entrada [51].....	89

## Índice de figuras

Figura 29. Comparativa de los márgenes obtenidos por distintas fronteras (SVM) [53].	91
Figura 30. Transformación a un espacio de mayor dimensión para facilitar la separación[51].	91
Figura 31. Los conjuntos de datos de entrenamiento no separables acarrear un coste [51].	92
Figura 32. Ejemplos de sobreajuste y subajuste del modelo [51].	93
Figura 33. Fases del trabajo con la SVM.	94
Figura 34. Agrupamiento de los datos disponibles para entrenamiento y test.	96
Figura 35. Proceso de validación cruzada mediante la técnica “Leave-one-out”.	99
Figura 36. Generación del modelo	100
Figura 37. Uso del modelo para generar predicciones a partir de los datos de test.	101
Figura 38. Proceso de validación de datos en el escenario completo.	102
Figura 39. Proceso de regresión en función de la detección del blanco y la situación real.	104
Figura 40. Proceso interno del K-NN. Estimación de una posición a partir de dos bloques de sensores y un radio de referencia.	106
Figura 41. Caso de no estimación a pesar de detección correcta.	107
Figura 42. Distribución aleatoria de 10, 50, 100, 300, 500 y 1000 sensores sobre el área.	113
Figura 43. Funciones de densidad de probabilidad (izquierda) y de distribución de probabilidad (derecha) de la potencia de ruido. Teórica (rojo) frente a empírica (azul).	114
Figura 44. Distribución de los 300 sensores en el área (azul) y los puntos aleatorios para el cálculo empírico de las distribuciones (rojo).	114
Figura 45. Función de densidad de probabilidad de la distancia.	115
Figura 46. Funciones de densidad de probabilidad (izquierda) y de distribución de probabilidad (derecha) de la potencia recibida en función de la distancia para ausencia (arriba) y presencia (abajo) de ruido.	116
Figura 47. Representación de la función $J$ para $Pnd = 0,01$ .	119
Figura 48. Representación de la función $J$ para $Pnd = 0,1$ .	119
Figura 49. Comparativa umbrales frente a probabilidades de no detección y falsa alarma.	120
Figura 50. Umbrales para los clasificadores con abstención.	121
Figura 51. Trayectorias tipo (6) catalogadas en el proyecto.	122
Figura 52. Comparativa: predicción y presencia real para $N=10, 100, 500$ y $1000$ . $Pfa = 0,01$	132
Figura 53. Comparativa trayectoria real y posiciones estimadas para $N=10, 50$ y $100$ sensores. $Pfa = 0,01$ . 2 trayectorias tipo.	133
Figura 54. Comparativa trayectoria real y posiciones estimadas para $N=300, 500$ y $1000$ sensores. $Pfa = 0,01$ . 2 trayectorias tipo.	134
Figura 55. Diferencia coordenada a coordenada de la posición real con la posición estimada. Para 10(a), 50(b), 100(c), 300(d), 500(e) y 1000(f) sensores. Para $Pfa = 0,01$ .	135
Figura 56. Comparativa estimaciones con pesos fijos para distintas condiciones de parada y para 300 sensores, en un escenario con abstención $Pfa = 0,01$ y $Pnd = 0,01$ .	138
Figura 57. Representación de la diferencia entre posiciones de la trayectoria y estimaciones cuando se produce detección del blanco. Pesos fijos para distintas condiciones de parada: original (arriba) y modificada (abajo). Para $Pfa = 0,01$ y $Pnd = 0,01$ .	139

Figura 58. Comparativa trayectoria real y posiciones estimadas para $N=100, 300$ y $1000$ , sensores, sin limitación en el número de bits transmitidos. $Pfa = Pnd = 0,01$ . 2 trayectorias tipo. ....	141
Figura 59. Comparativa de 4 estimaciones de trayectorias distintas para 500 sensores. Sin abstención y $Pfa = 0,01$ , (Arriba izq.). Sin abstención y $Pfa = 0,1$ , (Abajo izq.). Con abstención y $Pfa = Pnd = 0,01$ , (Arriba der.). Con abstención y $Pfa = Pnd = 0,1$ . (Abajo der.).....	144
Figura 60. Comparativa de 4 estimaciones de trayectorias distintas para 500 sensores. Sin abstención y $Pfa = 0,01$ , (Arriba izq.). Sin abstención y $Pfa = 0,1$ , (Abajo izq.). Con abstención y $Pfa = Pnd = 0,01$ , (Arriba der.). Con abstención y $Pfa = Pnd = 0,1$ . (Abajo der.).....	144
Figura 61. Comparativa de 3 estimaciones de trayectorias distintas para redes de 300 sensores en función de 3 distancias distintas (12,28m, 24,56m y radio infinito) para limitar la influencia de los sensores en la estimación. 2 trayectorias tipo, con abstención, $Pfa = Pnd = 0,01$ . ....	146
Figura 62. Trayectorias actuales (izquierda) y propuestas para líneas futuras (derecha). ....	152
Figura 63. Canal binario simétrico. ....	152
Figura 64. Canal binario simétrico con borrado.....	153

## Índice de figuras

# Índice de tablas

Tabla 1. Evolución histórica de las redes de sensores [2].	8
Tabla 2. Tabla de costes correspondientes a cada clasificación según hipótesis (Minimax).	40
Tabla 3. Tabla de costes correspondientes a cada clasificación según hipótesis (Bayessiano).	40
Tabla 4. Variación del radio de detección con respecto a la probabilidad de falsa alarma tolerada.	87
Tabla 5. Umbrales de potencia y radios de detección en función de la probabilidad de falsa alarma $Pfa$ .	117
Tabla 6. Mínimos de la función J para distintos valores de probabilidad de no detección $Pnd$ .	119
Tabla 7. Umbrales de potencia y distancias equivalentes.	120
Tabla 8. Distribución real de los datos por clases.	123
Tabla 9. Datos recogidos de precisión (P), sensibilidad (S) y especificidad (E) para entrenamiento, test y validación. $Pfa = 0,01$ .	125
Tabla 10. Datos de precisión (P), sensibilidad (S) y especificidad (E) para entrenamiento, test y validación. $Pfa = 0,01$ y $Pnd = 0,01$ .	126
Tabla 11. Datos recogidos de precisión (P), sensibilidad (S) y especificidad (E) para entrenamiento, test y validación. $Pfa = 0,1$ .	127
Tabla 12. Datos de precisión (P), sensibilidad (S) y especificidad (E) para entrenamiento, test y validación. $Pfa = 0,1$ y $Pnd = 0,1$ .	127
Tabla 13. Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo). $Pfa = 0,01$ .	128
Tabla 14. Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo). $Pfa = 0,1$ .	128
Tabla 15. Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo). $Pfa = 0,01$ y $Pnd = 0,01$ .	129
Tabla 16. Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo). $Pfa = 0,1$ y $Pnd = 0,1$ .	129
Tabla 17. Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo). Red de 500 sensores.	131
Tabla 18. Sesgo, varianza y error cuadrático para 10, 50, 100, 300, 500 y 1000 sensores. $Pfa = 0,01$ .	136

## Índice de tablas

Tabla 19. Sesgo, varianza y error cuadrático para 10, 50, 100, 300, 500 y 1000 sensores. Pesos fijos a 1(detección) y 0,1 (abstención). $Pfa = Pnd = 0,01$ . ....	140
Tabla 20. Sesgo, varianza y error cuadrático para 10, 50, 100, 300, 500 y 1000 sensores. Pesos fijos a 1(detección) y 0,5 (abstención). $Pfa = Pnd = 0,01$ . ....	140
Tabla 21. Sesgo, varianza y error cuadrático para 10, 50, 100, 300, 500 y 1000 sensores, sin limitación en el número de bits transmitidos. $Pfa = Pnd = 0,01$ . ....	142
Tabla 22. Comparativa error cuadrático medio de los distintos escenarios contemplados. $Pfa = Pnd = 0,01$ . ....	143
Tabla 23. Comparativa sesgo, varianza y error cuadrático para la estimación a partir de la SVM $(x_1, y_1)$ y para la estimación siempre que hay blanco $(x_2, y_2)$ . 500 sensores. Variaciones de $Pfa = 0,01$ , $Pnd = 0,01$ , $Pfa = 0,1$ y $Pnd = 0,1$ . ....	145
Tabla 24. Coste unitario del material utilizado en la realización del proyecto.....	155

# Capítulo 1

## Introducción y descripción del problema

### 1.1 Introducción

Este proyecto tiene como origen el estudio de las redes de sensores, y en particular las redes inalámbricas de sensores. La mejora constante en las comunicaciones y el gran espectro de aplicación de los sensores nos invitan a buscar mecanismos de detección, clasificación y estimación que potencien la inferencia en dichas redes. Sin olvidarnos de los modelos clásicos de inferencia, en este proyecto se analizarán formas de inferencia distribuida.

De modo simplificado, se desarrollará un escenario principal en el que se pretende determinar la presencia o no de un blanco en un área monitorizada. La red, dispondrá de un centro de fusión de datos y de múltiples sensores cuya capacidad de comunicación con el centro de fusión es restringida. Ante tales restricciones buscaremos mostrar que existen mecanismos suficientes para, agregando muy poca información de muchos sensores (hasta un solo bit por sensor), implementar con éxito métodos de detección binaria. Como complemento, se propone un estimador sencillo de las posiciones con las que recrear la trayectoria seguida por el blanco.

Este capítulo se estructura de manera que los contenidos se dividen en 4 secciones. La Sección 1.2 recoge una breve descripción del problema mientras que por su parte las Secciones 1.3 y 1.4, detallan las fases del desarrollo y los medios empleados. Por último, la Sección 1.5 está dedicada a describir la estructura de la memoria.

## 1.2 Descripción del problema

El objetivo fundamental de este proyecto es analizar los límites estadísticos de la inferencia distribuida en redes con un centro de fusión introduciendo varios teoremas sobre dichas limitaciones que se trataran de comprobar posteriormente de forma empírica. En base a ese objetivo principal, se proponen los siguientes objetivos parciales:

- Conocer en profundidad las redes inalámbricas de sensores: evolución histórica, elementos principales, topología, aplicaciones, etc.
- Estudiar los mecanismos clásicos de detección, clasificación y estimación mediante la agregación de toda la información disponible.
- Comprender el funcionamiento de las maquinas de vectores soporte (SVM).
- Analizar las restricciones que aparecen al tratar de establecer detectores, clasificadores y estimadores que nos permitan trabajar de forma distribuida.
- Desarrollar un escenario práctico donde comprobar en qué medida esas limitaciones afectan al rendimiento del detector y mostrar una primera aproximación de cómo podríamos aprovechar la información obtenida para estimar el resultado.

## 1.3 Fases del desarrollo

De forma paralela a los objetivos parciales se han ido completando las distintas fases de desarrollo, muchas de las cuales se han solapado temporalmente:

- Lectura y documentación sobre redes de sensores inalámbricas.
- Lectura y documentación sobre detección, clasificación y estimación.
- Lectura y documentación sobre el problema concreto de inferencia distribuida.
- Implementación del escenario básico en Matlab.
- Lectura y documentación sobre detección, clasificación y estimación distribuida.
- Lectura y documentación sobre SVM (Máquina de Vectores Soporte).
- Implementación del detector básico en Matlab.
- Análisis de los primeros resultados de detección.
- Implementación de mejoras en los detectores.
- Análisis de los resultados mejorados de detección.
- Implementación del estimador básico.
- Análisis de los resultados de estimación.
- Redacción de los resultados y conformado de la memoria.

## 1.4 Medios empleados

Para realizar este proyecto los medios empleados son, en cuanto al hardware: un ordenador personal, una impresora y material de oficina. Y en cuanto al software, la licencia del sistema operativo Windows 7 Enterprise, la licencia de MATLAB 7.6.0 (R2008a) y el paquete ofimático Microsoft Office 2007.

Además, se ha utilizado la librería LIB-SVM, de libre distribución, que contiene funciones utilizadas para el manejo de la SVM y que se encuentra disponible en <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

## 1.5 Estructura de la memoria

Para facilitar la lectura de la memoria, se incluye a continuación un breve resumen de cada capítulo.

El siguiente capítulo trata en primer lugar sobre las redes de sensores en general: su evolución, los componentes y la arquitectura de cada nodo y las topologías de las redes. Posteriormente se concentra en las redes inalámbricas, para culminar proporcionando varios ejemplos sobre aplicaciones.

El Capítulo 3 se concentra en la detección, clasificación y estimación. Se analiza el modo de trabajo frente a problemas de detección/clasificación y se expone un breve ejemplo de comunicación binaria que nos ayuda a introducir los detectores de máxima verosimilitud, de máxima probabilidad a posteriori y a definir la probabilidad de error.

Adicionalmente, se detallan otros criterios para establecer decisores, como el de Neyman-Pearson, el de Bayes y el criterio de Minimax, y se definen conceptos como el estadístico suficiente o el test de hipótesis compuestas.

En cuanto a la estimación, las secciones se dividen en estimación bayesiana, y estimación de máxima verosimilitud. Dentro de la estimación bayesiana, se profundiza en el estimador de mínimo error cuadrático medio (MMSE) y en el de máxima probabilidad a posteriori (MAP).

En el Capítulo 4 se desarrolla un modelo general para el aprendizaje y la inferencia distribuida y se trata de explicar la diferencia entre modelos paramétricos y no paramétricos. Como ejemplos se detallan dos escenarios concretos: uno en el que una red “ad-hoc” gestiona la información localmente y otro que considera un centro de fusión de datos. Por último, se analizan las limitaciones estadísticas del aprendizaje distribuido

## Capítulo 1: Introducción y descripción del problema

originadas por el hecho de no disponer de toda la información en todos los puntos del espacio.

El Capítulo 5 se divide en tres bloques: en el primero se describe el escenario práctico que vamos a estudiar: los sensores, el área, la trayectoria, las distribuciones de potencia y el ruido. En el segundo se analiza el detector en sí mismo y todo el proceso que requiere la obtención del modelo, con su correspondiente entrenamiento, test y validación. Y por último, en el tercero de los bloques se describe el método de estimación escogido y la forma de evaluar la calidad del estimador.

El Capítulo 6 recoge los resultados obtenidos de poner en práctica lo especificado en el capítulo previo. De este modo, se muestran en primer lugar el despliegue de los sensores en el área, las distribuciones de las señales y la función de núcleo o “Kernel” que se utilizará en el detector. Posteriormente se describen los dos escenarios desplegados, uno donde el sensor decide en función de un solo umbral de detección relacionado con la probabilidad de falsa alarma y otro donde el sensor dispone de dos umbrales que acotan una región de incertidumbre en la decisión. Una vez establecidos los escenarios, se generan los datos y se evalúan los resultados del detector proporcionado por la SVM (“Support Vector Machine”) en términos de precisión, sensibilidad y especificidad para distintos conjuntos de datos (entrenamiento, test y validación). Para completar el análisis del clasificador binario se prueba con una función de núcleo Gaussiana, frente a la habitualmente utilizada (Lineal).

Para completar el Capítulo 6 se adjuntan los resultados obtenidos para la estimación de las posiciones. Se consideran tres escenarios básicos: el primero donde no se contempla la abstención por parte del sensor, y donde todos los sensores que detectan contribuyen a la estimación de la posición en la misma proporción. El segundo donde sí se contempla la abstención y se les asigna a todos los sensores que se abstienen un peso idéntico menor al asignado a los que detectan. Y el último donde no se consideran limitaciones en la información transmitida, de modo que la contribución de cada sensor que se abstiene a la posición estimada depende de la potencia recibida por el sensor. El bloque de estimación se cierra con el estudio del comportamiento de los estimadores frente a la variación en las probabilidades de error y otro acerca de las variaciones del radio de referencia utilizado para considerar los sensores como aptos para contribuir a la estimación.

El Capítulo 7 recoge las conclusiones del proyecto y algunas posibles líneas futuras de investigación. Por último, el Apéndice A recoge el presupuesto necesario para llevar a cabo este proyecto.

# Capítulo 2

## Redes de sensores

### 2.1 Introducción

A lo largo de este capítulo se puede encontrar un estudio breve sobre todo lo concerniente a las redes de sensores. Desde el origen y su evolución, pasando por sus componentes, arquitectura o topologías de la red y llegando a una muestra de las innumerables aplicaciones. Este pretende ser el paso inicial que nos permita entender la versatilidad de las redes, aumentada si su naturaleza es inalámbrica, y la necesidad de desarrollar mecanismos que nos ayuden a manejar la información que estas captan. Los contenidos del capítulo se separan en siete bloques, los cinco primeros son de carácter general: Evolución histórica, componentes del nodo, arquitectura del nodo, topologías de la red, requisitos de una WSN y posibles retos. Los dos últimos aportan un punto de vista específico y concreto sobre las redes “Ad-hoc” frente a redes con centro de fusión y sobre la diversidad de las aplicaciones.

### 2.2 Evolución histórica

Para todo el mundo son conocidas las redes cableadas de sensores, cuyas funciones han sido tradicionalmente medir magnitudes físicas como niveles de temperatura, de líquidos, humedad, etc. Sin embargo, una nueva generación de sensores se unen para formar las redes inalámbricas, que en lo sucesivo se denotarán con las siglas WSNs (“Wireless Sensor Networks”) y cuya principal diferencia es la inteligencia que las define (estas nuevas redes son capaces de proporcionar una respuesta en función de los datos que van almacenando), sin olvidar el carácter móvil que les aporta la ausencia del cable.

## Capítulo 2: Redes de sensores

El desarrollo de las WSNs ha requerido tecnologías de tres áreas diferentes: los sensores, las comunicaciones y la computación (incluyendo “hardware”, “software” y algoritmos) [1]. Los avances en la fabricación de microchips, los estándares en las comunicaciones inalámbricas y las nuevas funcionalidades de los “routers” y otros dispositivos de red, sin olvidar la continua evolución del software, han conseguido multiplicar el potencial de los sensores independientemente de la tecnología inalámbrica que utilicen: IEEE 802.11 (WiFi), Bluetooth, RFID (“Radio Frequency IDentification”), UWB (“Ultra Wide Band”), etc.

A continuación se describe su evolución histórica que, como ocurre con muchas de las tecnologías actuales, tiene como origen y motor de desarrollo las aplicaciones militares. Uno de los primeros ejemplos son los sistemas de espionaje y vigilancia con sensores acústicos, que se desarrollaron durante la denominada Guerra Fría [2]. En estos inicios militares las redes adoptaron una estructura de procesamiento jerárquico, donde cada nivel iba procesando los eventos de interés hasta llegar al usuario, interviniendo incluso en muchos casos operadores humanos con un papel clave en el sistema. Todo estaba centrado en satisfacer las necesidades de cada misión: procesado de las citadas señales acústicas, interpretación, seguimiento, fusión de datos, etc. Existía por tanto una especialización de las redes y una falta de automatización.

En la década de los 80 comienza el que puede denominarse estudio moderno de las redes de sensores, con la utilización de configuraciones distribuidas. Sirva de ejemplo el programa DSN (“Distributed Sensor Networks”), llevado a cabo por DARPA (“Defense Advanced Research Projects Agency”), aprovechando el despliegue ya realizado de la semilla de Internet y las consecuentes mejoras de las comunicaciones. En esta época se desarrollaron los minicomputadores, los módems, Ethernet se convirtió en un protocolo popular y así se llegó a definir la siguiente lista de componentes básicos de una DSN [3]:

- Sensores acústicos.
- Protocolos de alto nivel, que enlazaban los procesos, trabajando en aplicaciones comunes en una red que compartía recursos.
- Técnicas de procesado y algoritmos que incluían herramientas para la autolocalización del sensor.
- Software distribuido, sistemas distribuidos modificables de modo dinámico y un lenguaje de diseño específico.

Este desarrollo era paralelo al de la Inteligencia Artificial, también patrocinado en gran parte por DARPA, de modo que se incluyeron en dicha lista conceptos de Inteligencia Artificial necesarios para el entendimiento de las señales y técnicas de resolución de problemas distribuidos.

En paralelo con el trabajo de DARPA, se sucedieron avances en distintos campos y a cargo de distintos estamentos, como por ejemplo:

- Carnegie Mellon University (CMU), que se centró en el desarrollo de un sistema operativo para las redes que fuera flexible y de acceso transparente para los recursos distribuidos, algo indispensable para mejorar la tolerancia a fallos [4].
- Massachusetts Institute of Technology (MIT), que se centró en técnicas de procesamiento de señales basadas en el conocimiento, concretamente para el seguimiento de helicópteros usando arrays distribuidos de micrófonos acústicos mediante abstracciones y técnicas de comparación [5].
- También en el MIT se desarrolló el Lenguaje de Procesado de Señal y el Entorno de Computación Interactiva (SPLICE) para el análisis de datos y algoritmos de la mencionada DSN [5].

Estos avances provocaron que los diseñadores de sistemas militares del ejército americano se dieran cuenta de la importancia de las redes, dando origen a armas específicas que empleaban sensores, a los que era necesario proporcionar una arquitectura sólida. Después, armas y sensores se montaban sobre plataformas distintas y los sensores no pertenecían a ningún arma concreta. Fue entonces cuando tomaron una función colaborativa para informar apropiadamente a los soldados que manejaban las armas. Ejemplos concretos son sistemas como “Cooperative Engagement Capability” (CEC) [6], desarrollado por la marina americana, “Fixed Distributed System” (FDS), para batallas submarinas, “Remote Battlefield Sensor System” (REMBASS), “Tactical Remote Sensor System” (TRSS) o “Unattended Ground Sensors” (UGS) [7].

La continuación del proyecto DSN por parte de DARPA dio origen al programa “Sensor Information Technology” (SensIT) [8], que contaba con dos líneas de investigación clave:

- Una enfocada a las técnicas de la red, listas para el despliegue rápido y dinámico de redes en modo “Ad-hoc”, considerando capacidades para voz y datos.
- Otra segunda dirigida al procesamiento de la información de modo útil y fiable aprovechando la computación distribuida.

De este modo transcurrieron las evoluciones hasta alcanzar el siglo XXI, donde los avances previos en computación y comunicación fueron la piedra angular que permitió alcanzar el concepto actual de WSN: sensores baratos y pequeños de carácter micro electro-mecánico (MEMS), que se agrupan formando una red inalámbrica también de bajo coste y con procesadores de mínimo consumo que colaboraban unos con otros de un modo autónomo, o cuasi-autónomo, compartiendo información para optimizar el aprovechamiento de la misma [2].

Como resumen, la siguiente tabla comparativa muestra la evolución de las principales características de las redes de sensores. Es significativo el cambio que se

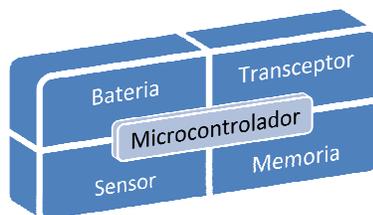
aprecia en el tamaño y peso del nodo, alcanzando dimensiones de una mota de polvo y pesos despreciables, lo cual facilita su uso en sistemas embebidos o su despliegue de forma sencilla e inmediata. La evolución paralela que sufren las baterías y de los sistemas de comunicación P2P forman ineludiblemente parte de la evolución de los sensores y su red.

Características del sensor	1980-1990	1990-2000	2000-2010
Tamaño	Mayor que caja de zapatos grande.	Entre caja de cartas y caja de zapatos.	Mota de polvo.
Peso	Kilogramos.	Gramos.	Despreciable.
Arquitectura del nodo	Separa sensor, procesado y comunicación.	Integra sensor, procesado y comunicación.	Integra sensor, procesado y comunicación.
Topología de la red	P2P, estrella.	Cliente-Servidor, P2P.	P2P.
Alimentación	Grandes baterías de horas, días o semanas de duración.	Baterías AA. Duración de días a semanas.	Solares. Duración de meses a años.
Implementación	Colocados con vehículos, sensores solitarios.	Colocados a mano.	Embebidos.

**Tabla 1.** Evolución histórica de las redes de sensores [2].

## 2.3 Componentes del nodo

Un sistema WSN es, como se ha citado anteriormente, una red con numerosos dispositivos distribuidos espacialmente, que utilizan sensores para analizar diversas condiciones en distintos puntos. Estos dispositivos son unidades autónomas, denominados nodos, que constan siempre, como puede verse en la Figura 1, de un microcontrolador con su correspondiente memoria, una fuente de energía (casi siempre una batería), un transceptor (transmisor-receptor) radio y uno o más elementos sensores [9]. Adicionalmente se pueden encontrar en el nodo actuadores, módulos de movimiento (un motor) u otros bloques específicos más concretos dependientes de la aplicación.



**Figura 1.** Componentes de un nodo de una WSN [9].

De manera conjunta el nodo funciona típicamente como sigue: los dispositivos sensores detectan un cambio en la magnitud física que están midiendo, esa información se la pasan al microcontrolador, que la analiza y gestiona. Por último, el transceptor de cada nodo se utiliza para recibir la información de otros sensores o compartir su propia información con el resto de sensores. A continuación se describen brevemente cada uno de los bloques básicos.

### 2.3.1 Sensores y actuadores

Un sensor es un dispositivo que detecta propiedades físicas, biológicas o químicas de su ambiente y convierte estas propiedades en una señal eléctrica. Por el contrario, un actuador normalmente acepta una señal eléctrica y la convierte en una acción física para operar sobre el medio en el que se encuentra. Las redes inalámbricas que integran sensores y actuadores se conocen como WSNs (“Wireless Sensor and Actuator Networks”) y puede verse una descripción detallada de las mismas en [10]. En el caso concreto de este estudio los dispositivos serán simplemente sensores y no realizarán ninguna acción física sobre el medio.

Los sensores suelen tener dimensiones reducidas, bajo coste e incluir circuitos acondicionadores específicos. Existen sensores para medir prácticamente cualquier magnitud deseada, como por ejemplo [11]:

- Sensores potenciométricos o capacitivos para medir desplazamientos, posiciones o niveles.
- Sensores de efecto Hall, para medir campos magnéticos.
- Sensores de ultrasonido u ópticos para medir distancias o velocidades.
- Sensores piezorresistivos o células de carga, para medir fuerzas y deformaciones.
- Giróscopos y acelerómetros, para medir posiciones, ángulos y velocidades.
- Interferómetros ópticos para medir cambios en el medio.
- Termopares para medir temperaturas.

### 2.3.2 Transceptor

Cada uno de los nodos de un sistema de redes de sensores inalámbricos incluye un módulo que está a cargo de la comunicación inalámbrica con sus compañeros. Los transceptores para WSN se diferencian de los demás en las características que se desarrollan y explotan para solventar las limitaciones que se imponen a los nodos. En concreto, se trata de minimizar el consumo, con modos de ahorro de energía, así como de minimizar su tamaño. En cuanto a este último parámetro, no podemos olvidar que la transmisión se realiza gracias a una antena y que el tamaño de esta dependerá de la frecuencia utilizada en la comunicación. En el caso de WSNs típicamente se suele utilizar una de las bandas reservadas a las aplicaciones industriales, científicas y medicas (bandas

ISM) cuyas frecuencias son de 902-928MHz y 2.4 GHz. Por un lado, a medida que aumenta la frecuencia la longitud de onda se hace más pequeña y el tamaño de la antena podrá ser más pequeño. Por otro lado, el aumento de la frecuencia supone un empeoramiento de las condiciones de transmisión (atenuación y sensibilidad frente a obstáculos). Por lo tanto, la elección de una banda u otra supone un compromiso entre ambos factores.

### **2.3.3 Microcontrolador y memoria**

Un microcontrolador es un circuito integrado o chip que incluye en su interior las tres unidades funcionales de un ordenador: la unidad central de procesamiento, la unidad de entrada/salida (E/S) y la memoria. Adicionalmente se puede añadir a los nodos una memoria externa a la del microcontrolador si es necesario.

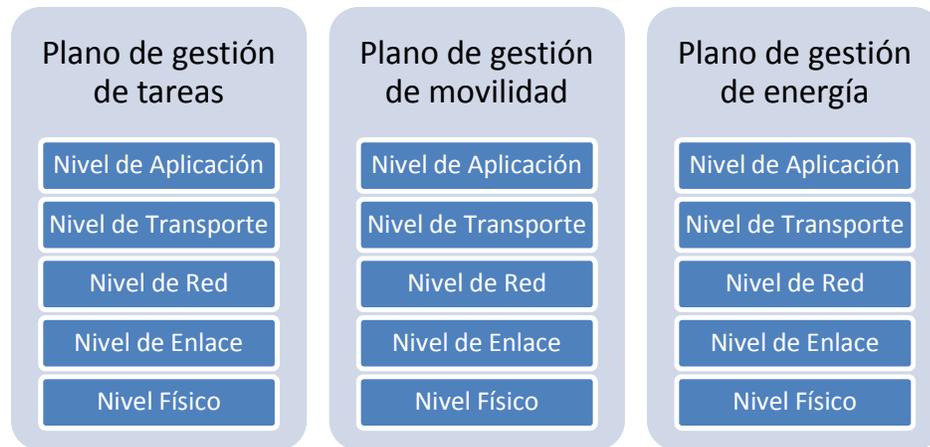
### **2.3.4 Batería y administrador de la energía**

Existen técnicas para hacer buen uso de la energía, obteniéndola a través de diferentes fuentes renovables además de las baterías. El consumo puede llegar a ser una de las principales limitaciones de la red, en cuanto a su tamaño y autonomía, de modo que es un aspecto muy importante a considerar. Por norma general, en las WSNs se desea que los nodos tengan un consumo de energía reducido y que los mismos estén capacitados para soportar condiciones ambientales extremas [12].

## **2.4 Arquitectura de un nodo sensor**

Independientemente del cometido del nodo, este tiene una pila de protocolos que pretende combinar la vigilancia del consumo y el enrutamiento, integrando protocolos de datos y red en un medio inalámbrico y promoviendo los esfuerzos cooperativos entre nodos.

Como se puede ver en la Figura 2, la pila consta de los típicos cinco niveles de la pila TCP/IP de cualquier dispositivo: físico, enlace, red, transporte y aplicación. Además, puede apreciarse que presenta tres planos distintos a monitorizar: el plano de gestión de energía, el de gestión de movilidad y el de gestión de tareas [13].



**Figura 2.** Pila de protocolos típica de un sensor [13].

La **capa física** da solución a las necesidades de técnicas de modulación, transmisión y recepción simples pero robustas.

El **nivel de enlace** es el responsable del multiplexado del flujo de datos, de la detección de la trama de datos, del control de acceso al medio (MAC) y del control de errores. Como el medio es ruidoso y los nodos sensores pueden ser móviles, el protocolo de control de acceso al medio debe estar enfocado a minimizar las colisiones al mismo tiempo que controla el consumo (control lógico del enlace, LLC). Debe asegurar conexiones seguras punto a punto y punto a multipunto en una red de comunicación.

El **nivel de red** cuidará del enrutamiento de los datos para proveerlos al nivel de transporte. Este se diseña en base a los principios de la eficiencia del consumo, de la centralización o no de los datos en la red y del nivel de colaboración y por tanto de tráfico que habrá entre nodos.

La **capa de transporte** se encarga de controlar el flujo si la aplicación lo requiere. Esta capa es especialmente necesaria cuando se prevé que el acceso al sistema vaya a ser realizado a través de Internet u otras redes externas.

Dependiendo del objetivo del sensor, diferentes tipos de aplicaciones software pueden construirse y ser usadas en el **nivel de aplicación**.

En cuanto a los planos, el de **gestión de potencia** sirve para administrar el modo en el que el sensor o nodo utiliza su potencia. Por ejemplo, puede apagarse tras recibir un mensaje de uno de los vecinos, evitando así duplicidad de mensajes, o enviar un aviso “broadcast” si se encuentra con bajo nivel de batería para informar de que no puede participar en las tareas de enrutamiento.

En el **plano de gestión de la movilidad**, se detectan y registran los movimientos del nodo, permitiendo en todo momento el conocimiento de las rutas al origen y las

posiciones propias y de los nodos vecinos. De este modo el nodo puede optimizar la potencia a utilizar en las transmisiones y aplicar algoritmos de procesamiento de señal que requieran datos de localización.

En el **plano de gestión de tareas** se negocian y organizan todas las labores necesarias para el proceso de adquisición de un dato en una región específica. Este plano es necesario para facilitar el trabajo conjunto y hacerlo eficientemente en términos de energía y recursos.

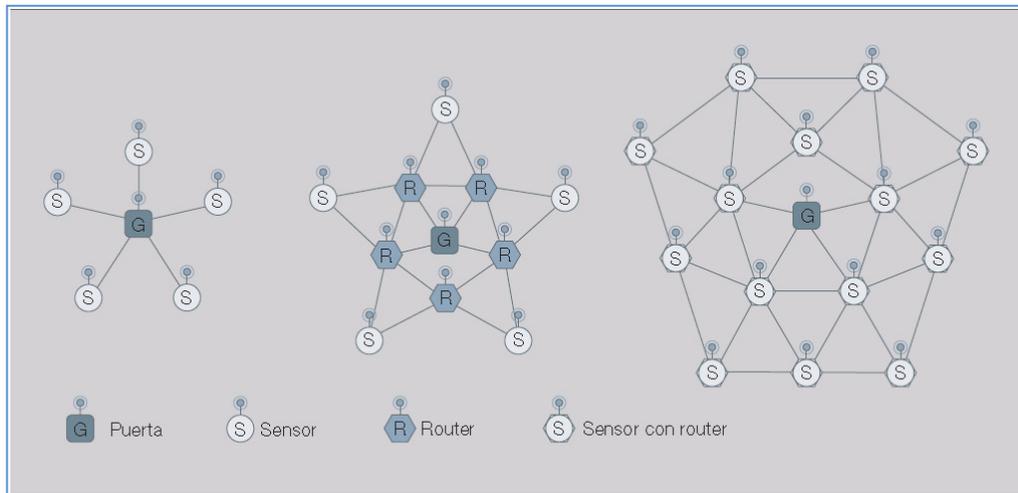
## 2.5 Topologías

En su forma más simple, una WSN puede organizarse en torno a una estación base, o **centro de fusión**, en la que se realiza la recogida de datos que se han tomado en todos los otros dispositivos (nodos) de una manera sincronizada y estructurada.

Las tareas de la red, es decir, el cálculo de las rutas y protocolos de comunicación, se implementan en software y se ejecutan generalmente en microcontroladores de gama baja. Dicha red suele estar configurada para soportar fallos en los nodos y en la comunicación y permite la movilidad de los mismos, debiendo ser por tanto una red dinámica y adaptativa, donde se produce un enrutamiento de información incluso entre nodos heterogéneos.

Las estaciones base son primordiales y críticas, especialmente en los sistemas centralizados, ya que contienen muchos más recursos de cálculo, energía y comunicación, y a menudo actúan como interfaz entre los nodos sensores y el usuario final. La Figura 3 muestra tres topologías típicas de una WSN. Observando la misma se pueden distinguir los siguientes elementos de la red [14]:

- La **G** se corresponde con la pasarela o **“Gateway”**. Es un nodo que podemos considerar principal, que puede tener capacidad de sensor o no, y que va a recibir toda la información de los sensores en cualquiera de los casos. Además, este nodo desarrollará tareas de puente a una red de comunicación cableada o inalámbrica TCP/IP.
- Los **Sensores**, **S**, son obviamente los elementos principales de la red. No debemos olvidar que se trata de nodos con capacidades de procesamiento, alimentación y emisión y recepción de datos. En función de si tienen capacidades de “router” o no, se distinguen en la Figura 3 dos símbolos distintos para este tipo de nodos.
- Los **“Routers”**, **R**, son elementos intermedios que no tienen capacidad de recolectar ningún dato: no funcionan como sensores, sino que ayudan a la comunicación y al intercambio de información entre nodos y con el “gateway”.

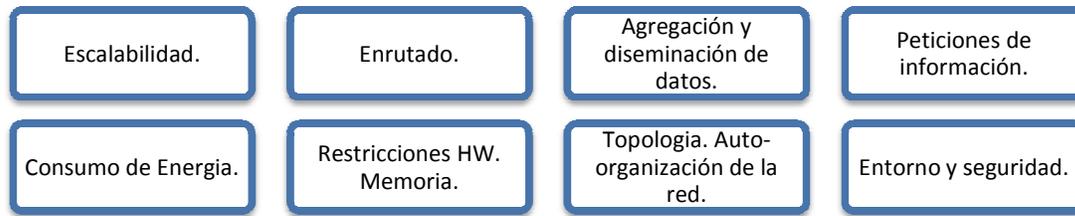


**Figura 3.** Topologías típicas de una WSN[14].

Además, la figura ilustra las topologías más habituales. Como puede apreciarse, la más simple es la situada a la izquierda: no hay ninguna malla y las comunicaciones son directas y punto a punto entre los sensores y el “gateway”. La topología que se sitúa en el centro es la clásica en estrella: los sensores no integran las capacidades del Router (R), de modo que no pueden comunicarse entre ellos y es necesario establecer una conexión punto a punto entre cada sensor y su “router” para facilitar el intercambio de información entre sensores. Esta configuración permite a los nodos mantener la sencillez y no aumentar las exigencias en lo referente al consumo y a la capacidad. La topología de la derecha, por el contrario, tiene una malla densa y considera que los sensores sí integran funciones de enrutamiento. Esto posibilita la comunicación entre ellos de un modo directo, incrementado en este caso el consumo pero facilitando las conexiones.

## 2.6 Requisitos de una WSN y posibles retos

En cualquier red de sensores las dificultades técnicas son considerables en el procesado de los datos, la comunicación y la propia gestión del sensor debido a las potenciales variaciones en el entorno. Adicionalmente a las ya mencionadas deben unirse las impuestas por el carácter inalámbrico y sus restricciones en el ancho de banda y la energía [1], [2], [15]. El diseño de las redes por tanto se realiza considerando factores como los mostrados en la Figura 4 y que se describen brevemente en las secciones sucesivas.



**Figura 4.** Factores que intervienen en el diseño de una WSN.

### 2.6.1 Escalabilidad

Al estar las WSNs formadas por sensores baratos y de pequeño tamaño, pueden acumularse cientos o miles de ellos, convirtiendo la adecuada gestión de la escalabilidad en algo realmente imprescindible. Habitualmente se utilizan técnicas como el “clustering” para resolver este problema [16]. Esta técnica consiste en agrupar sensores en función de algún determinado requisito, como por ejemplo la proximidad, permitiendo que mediante un trabajo local con ese conjunto o “cluster”, todos los sensores incluidos en el mismo se vean representados por un elemento equivalente.

### 2.6.2 Consumo de energía

En muchas aplicaciones los sensores se distribuyen en lugares de difícil acceso, de modo que acceder con determinados servicios resulta complicado. Por eso la duración de la batería puede llegar a limitar la vida de los sensores. Además, en este tipo de redes es frecuente que la aplicación de destino trate de detectar sucesos que ocurren con muy poca frecuencia, de modo que la primera medida que podría tomarse para reducir el consumo es dormir a los nodos cuando no se necesitan, activando por completo la zona próxima al evento cuando este suceda. Si el fenómeno se desplaza, los nodos que estén dentro de un determinado rango de sensibilidad se irán activando y adquiriendo datos, ahorrando el consumo que requieren en tareas de seguimiento y adquisición de datos los nodos que posteriormente pueden resultar poco relevantes por encontrarse alejados del foco del suceso. Este modelo, representado en la Figura 5, se denomina comúnmente “frisbee” y se utiliza típicamente para seguimiento de blancos [1]. Aunque no se ha mencionado, se requiere la existencia de nodos que estén siempre vigilantes y que tengan la capacidad de despertar a los demás si lo consideran necesario.

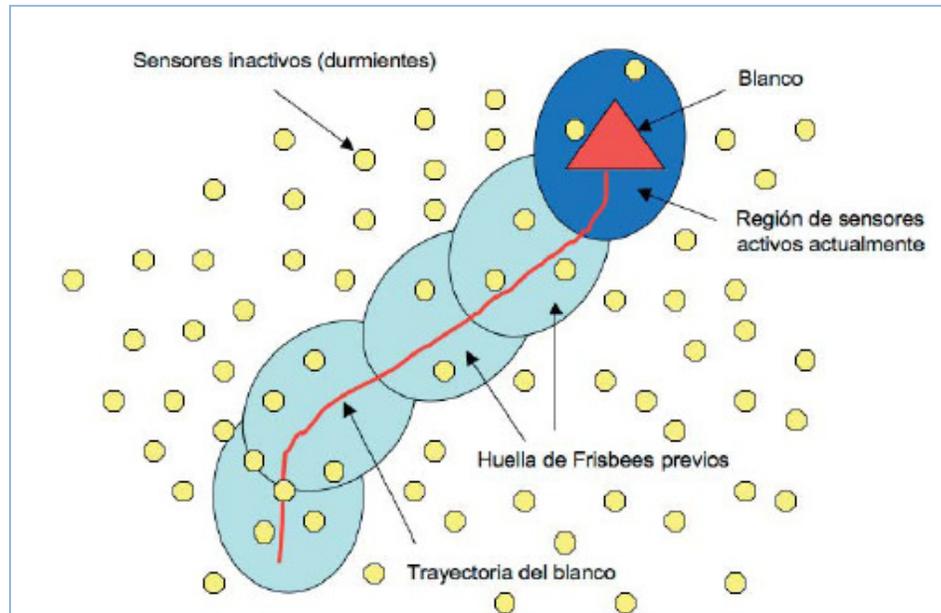


Figura 5. Modelo de ejemplo para el ahorro del consumo [1].

### 2.6.3 Enrutado

Las restricciones en el consumo de energía convierten a los protocolos convencionales de distribución de datos en poco eficientes, dando lugar a diferentes alternativas [17]:

- Los basados en **negociación**, que evitan la redundancia utilizando descriptores antes de transmitir la información.
- Los de **difusión directa**, que se basan en un enrutamiento reactivo originado en el destino.
- Los de **ahorro de energía**, también con mecanismos reactivos pero basados en las probabilidades del consumo energético.
- Los protocolos **multicamino**, que manejan caminos alternativos por si el óptimo no está disponible.

### 2.6.4 Restricciones hardware de memoria

Además de los costes de producción que puedan existir en la elaboración de los nodos de las WSN y de las limitaciones de tamaño, en los nodos sensores típicamente existen importantes restricciones en cuanto a capacidad de almacenamiento de información (esto es, de memoria). Debido a estas, las tablas de enrutamiento, la replicación de datos, la seguridad y todo el tráfico que no corresponda exclusivamente a información útil debe reducirse al máximo para encajar en la pequeña cantidad de memoria que contienen los nodos.

### **2.6.5 Agregación y diseminación automática de datos**

Un elevado número de nodos transmitiendo simultáneamente puede congestionar la red. Por eso resulta imprescindible establecer nodos que van a recibir información de otros, resumiéndola y difundiéndola de nuevo. Este procedimiento se asemeja bastante al que se realiza con técnicas como el “clustering”, que fomenta el tráfico local en el “cluster” [18]. Esto requiere menos recursos, disminuyendo de este modo el tráfico global gracias a la transmisión de la información una vez analizada y resumida.

### **2.6.6 Topología. Auto organización de la red**

Como se ha mencionado con anterioridad, a menudo los lugares donde se despliegan las redes son hostiles y provocan fallos en los nodos. Cada nodo necesita conocer la identidad y la localización de sus vecinos para apoyarse en ellos a la hora de procesar los datos. Habitualmente la topología de las redes es conocida, pero en otras ocasiones la red se construye y modifica en tiempo real. Es aquí donde toman relevancia los algoritmos de localización, permitiendo que la red se reorganice periódicamente de un modo dinámico [19]. Una posible solución pasaría por utilizar receptores GPS, algo que desafortunadamente parece inviable en términos de precio, cobertura y tamaño, por lo que la mayoría de las veces se recurre al uso de estaciones base de referencia junto con algoritmos de multilateración. La multilateración o posicionamiento hiperbólico es una técnica que, empleando los tiempos de llegada de una misma señal radioeléctrica a un conjunto de antenas, consigue el posicionamiento tridimensional de una estación móvil mediante un mínimo de 4 antenas [19].

### **2.6.7 Peticiones de información**

Un campo de sensores puede llegar a considerarse una base de datos distribuida donde cada sensor almacena una característica determinada. Dichas características son adquiridas de un modo dinámico del entorno bajo estudio en lugar de ser introducidas por un operario o usuario de la base de datos. Por eso en ocasiones puede resultar importante utilizar una interfaz interactiva que permita consultar dichos valores. Para la estructuración de dichas bases de datos habitualmente se consideran dos tipos de direccionamiento: uno donde la petición se envía a una región específica de la red, y otro donde la petición va dirigida a un nodo individual de la red [2].

### **2.6.8 Entorno y seguridad**

El citado despliegue de WSNs en territorios a menudo hostiles, incluso tras las líneas enemigas en el ámbito militar, hace indispensable desarrollar técnicas de baja latencia, resistentes y seguras. La latencia es un término utilizado para definir la suma de

retardos temporales acumulados en la red de comunicación y que depende típicamente del tamaño del paquete que se envía y de los “buffers” de almacenamiento. Además, los sensores deben pasar lo más desapercibido posible para evitar ser descubiertos, con el fin de evitar ataques de intrusión o suplantación de identidad dentro de la red [20].

## 2.7 Redes Ad-hoc frente a redes con centro de fusión

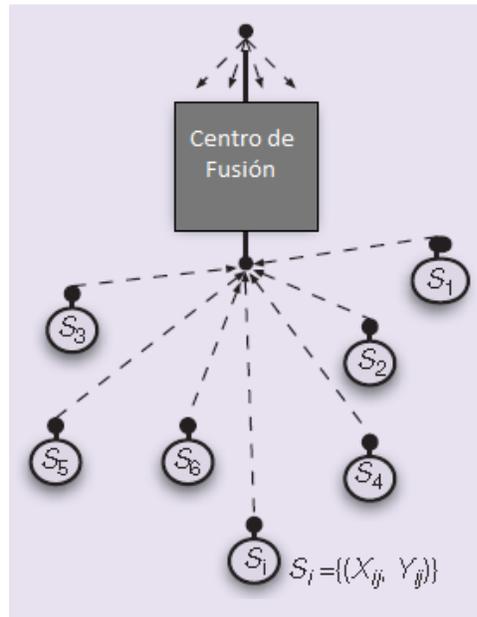
A continuación se detallan las dos topologías básicas de una WSN desde el punto de vista de recolección y procesamiento de información, así como sus posibles ventajas e inconvenientes. En cualquiera de los dos casos, los nodos colaboran para recolectar y procesar los datos con la intención de generar información útil. Como puede imaginarse, bastantes aspectos técnicos y parte del éxito del modelo dependerán del grado de compartición de información y de cómo se fusiona toda esa información recibida desde otros nodos.

No debe olvidarse que el objetivo principal es la estimación o detección de determinados eventos de interés y no simplemente la comunicación. Para mejorar el rendimiento es muy útil combinar los datos de distintos sensores, siendo en general mejores los resultados cuanto mayor sea el número de sensores utilizado. Como contrapunto, se sacrifican más recursos y es ahí donde hay que tratar de encontrar el equilibrio, utilizando además reglas simples de decisión y estimación que sean robustas a pesar de ser subóptimas [21].

### 2.7.1 Redes con centro de fusión

En esta sección vamos a concentrarnos en la red paralela de la Figura 6. Esta arquitectura es especialmente relevante en redes en las que el objetivo fundamental es la recolección de datos. El elemento principal es el **centro de fusión** o **estación base**, que puede ser modelado como un nodo adicional con mayor ancho de banda, disponibilidad de energía y/o capacidad de cómputo.

A modo de recordatorio, cabe destacar que cada nodo en el grafo representa a un sensor y su información observada localmente. Imaginemos, por ejemplo, que los datos capturados se corresponden con la temperatura para una posición en el plano y un instante de tiempo dados. El centro de fusión recibe cada una de las observaciones locales y construye un estimador global del campo de temperaturas, que varía continuamente. Como se aprecia, los nodos pueden estar distribuidos aparentemente sin ninguna restricción en torno al centro de fusión, que tiene la capacidad de generar mensajes “broadcast” a todos los nodos.

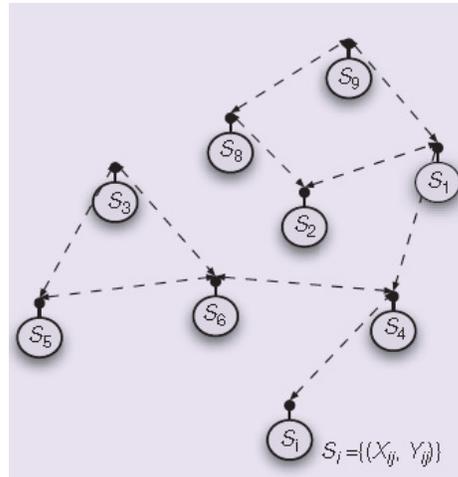


**Figura 6.** Topología de una WSNs con centro de fusión [21].

En este entorno surgen cuestiones como la siguiente: ¿cómo se produce el aprendizaje cuando las restricciones que afectan a la comunicación limitan el acceso a los datos? Estudiando a simple vista el modelo del ejemplo, la configuración va a obligar a los sensores a enviar todos sus datos al centro de fusión, lo que acarreará elevados costes en términos de energía y ancho de banda, ya que esto implica llevar a cabo una conexión punto a punto unidireccional desde cada nodo al centro. Para paliar esos costes, sería posible designar pequeños “clusters” para enviar los datos. Si el número de nodos es pequeño y tanto los sensores como la información se escogen de un modo adecuado, entonces esta estrategia podría ser efectiva a la hora de optimizar el rendimiento del aprendizaje, reduciendo el mencionado coste en la comunicación. Más detalles e información en profundidad sobre estas técnicas se aportarán en la Sección 4.4.

## 2.7.2 Redes Ad-hoc con procesamiento interno

En la Figura 7 se expone una nueva arquitectura que puede describirse como “ad hoc”. En este caso la topología es dinámica y probablemente no está definida antes de su despliegue. Aunque podrían incluir un centro de fusión, estas redes de sensores son más autónomas que las de la configuración anterior, lo que les permite tomar decisiones independientemente del centro de fusión.



**Figura 7.** Topología de una WSNs sin centro de fusión [21].

En esta ocasión la atención se centra en cuestiones como: ¿Puede la comunicación local entre sensores ser explotada para permitir un aprendizaje colaborativo eficiente? Si volvemos al ejemplo utilizado en la red con centro de fusión, los sensores cercanos entre sí adquirirán valores de la temperatura similares. Para estimar modelos globales a partir de estos datos locales, este intercambio de información local debe utilizarse en el aprendizaje con algoritmos distribuidos. Tal y como ocurría en la topología previa, se podrá encontrar más información sobre estos mecanismos en la Sección 4.3.

## 2.8 Aplicaciones

El conjunto de aplicaciones donde tienen cabida las WSNs es realmente amplio y está abierto a nuevas ideas y tendencias. Como se mencionaba anteriormente, su origen es militar, y las aplicaciones van desde grandes redes de vigilancia acústica oceánica a pequeñas redes para detección de objetivos en tierra. Sin embargo, el bajo coste ha facilitado, junto con la mejora de las redes de comunicaciones, la incorporación de estos a muchas aplicaciones civiles, desde infraestructuras de seguridad a control industrial. A continuación se enumeran e introducen algunas de ellas [22].

### 2.8.1 Infraestructuras de seguridad

En general las aplicaciones de seguridad se enmarcan en la lucha contra el terrorismo y el control de edificios y construcciones, como plantas de energía o centros de comunicaciones, que son susceptibles de ser atacados. Redes de vídeo, acústicas y todo tipo de sensores se despliegan entorno a estas dependencias con el fin de detectar rápidamente cualquier amenaza, ya sea química, nuclear, biológica o de cualquier otra índole. En este caso es posible disminuir la probabilidad de falsas alarmas y no

detecciones recopilando los datos de múltiples sensores y aprovechando el carácter inalámbrico para obtener mayor flexibilidad en las redes.

Fuera del ámbito antiterrorista, destacan otros dispositivos, como los comunes sensores antiincendios y las señales lumínicas indicadoras de las salidas de emergencia, obligatorias en los grandes edificios. La combinación de ambos sistemas de modo inalámbrico podría permitir mostrar al usuario el mejor camino de manera dinámica para escapar en una situación de peligro [23]. Por último, las redes de sensores están tomando también relevancia en la ingeniería civil por su utilidad en el control de estructuras y vibraciones (por ejemplo en puentes) para la prevención y detección de posibles daños [24], [25].

### **2.8.2 Monitorización del entorno y el hábitat**

Con la actual controversia sobre el cambio climático, la monitorización del medioambiente se ha convertido en una de las aplicaciones con más futuro y donde más margen de mejora se presenta. En este tipo de aplicaciones las variables que se controlan (como temperatura, humedad, presión, o contenido de CO<sub>2</sub>) se extienden habitualmente a lo largo de grandes regiones, donde las redes de sensores inalámbricos pueden obtener mejores resultados con un coste menor que el que implicaría una red cableada [26] .

A modo de ejemplo, el Amazonas, considerado el pulmón del planeta, monitoriza el tráfico aéreo a través de radares y sensores que permiten crear imágenes de lo que allí ocurre. Además, de forma paralela se monitoriza el comportamiento de algunas especies allí protegidas [27].

Pero no sólo puede monitorizarse el entorno exterior, sino que, por ejemplo, también se puede controlar el gasto energético en recintos cerrados mediante redes de sensores y actuadores para la temperatura, la luz, la ventilación y el estado de las ventanas y puertas, evitando innecesarios procesos de calentamiento o refrigeración del hábitat [22].

### **2.8.3 Aplicaciones a la agricultura**

En este campo las redes de sensores se despliegan con la intención principal de mejorar la eficiencia y el crecimiento de los cultivos. Un claro ejemplo es el estudio realizado por Intel e Intel Research Berkeley Labs acerca de una red de sensores desplegada sobre un viñedo [28]. Originalmente este sistema estaba enfocado a encontrar y controlar el microclima más adecuado y útil. En consecuencia, se equiparon los nodos con sensores de temperatura y humedad. Sin embargo, la aportación de estos datos no resultó relevante para los agricultores y se decidió extender dicha red de sensores al proceso completo de elaboración, desde el cultivo hasta su comercialización,

monitorizando la temporalización de cada tarea: seguimiento de la poda, control de plantaciones con productos químicos y plaguicidas, etc. En resumen, la aplicación se extiende al manejo de cultivos, reduciendo los costes y tratando de aumentar la calidad del producto final.

#### **2.8.4 Sensores en el entorno industrial**

A lo largo de la historia la industria ha buscado reducir el coste, mejorar el rendimiento de las máquinas y facilitar su mantenimiento. Monitorizando la salud de las mismas, en términos de vibraciones o niveles de lubricación, introduciendo sensores donde los humanos no pueden llegar, estas mejoras son posibles. En este sentido, un gran avance para la implantación de las redes de sensores en la industria fue la definición del estándar P1451 [29], “Smart Transducer Interface Standard” por parte del IEEE (“**Institute of Electrical and Electronics Engineers**”) y el NIST (Instituto Nacional de Estándares y Tecnología), que permitió la proliferación de sensores “plug-and-play” en todo el entorno industrial [14].

#### **2.8.5 Sensores en el control de tráfico**

El aumento de la calidad de las carreteras y del número de vehículos que circulan por las mismas hace imprescindible el control del tráfico, ya que no sólo la velocidad sino el flujo de los mismos se controlan a través de los clásicos semáforos. También puede detectarse un determinado nivel de congestión o incidentes mediante la monitorización con cámaras o con sensores de paso en la carretera. La asociación en redes de todos estos sensores permitiría llegar a elaborar un mapa del tráfico, que podría utilizarse por los usuarios para evitar atascos o escoger rutas alternativas [30].

#### **2.8.6 Apoyo logístico**

El control de inventario no es un problema trivial para las grandes compañías, y a medida que estas se expanden por el mundo el problema de controlar sus activos se distribuye y complica. Es ahí donde las redes de sensores, unidas al uso de los identificadores de RF (RFIDs), toman relevancia y ayudan al seguimiento de los mencionados activos.

Una aplicación concreta dentro de este área es la desarrollada por (BP) British Petroleum y Accenture Technology Labs para controlar la gestión del almacenamiento de sus barriles de petróleo, atendiendo a su localización y contenido y alertando en caso de incompatibilidad con otros líquidos o elementos próximos que puedan producir reacciones químicas que puedan alterar la calidad del producto [31]. Otros ejemplos son los del supermercado británico Tesco PLC, que controla de este modo su inventario, o el

de petroleras como Shell Oil Co, que controla las condiciones del bombeo en sus estaciones, entre una infinidad de ejemplos [22].

### **2.8.7 Aplicaciones centradas en los humanos**

Las recientemente denominadas ciencias de la salud se benefician también de los sensores inalámbricos a la hora de monitorizar pacientes. Las WSN permiten hacer un seguimiento en las primeras etapas de enfermedades como el Alzheimer. Por ejemplo los nodos pueden ser usados para recordar y controlar los tratamientos de los pacientes, así como el movimiento de los mismos y la proximidad de las futuras visitas [22].

Entre otros ejemplos, podemos destacar la capacidad de los sensores para ser implantados en órganos humanos y comunicarse de un modo inalámbrico con sistemas de computación externos para controlar determinados procesos de monitorización. A otro nivel se han utilizado recientemente los propios juguetes infantiles a los que se les incorporan sensores, para desarrollar tareas de análisis de comportamientos de los niños [32].

### **2.8.8 Aplicaciones en robótica**

Muchas de las aplicaciones que aúnan sensores y robótica se concentran en seguir o detectar caminos o campos. Generalmente el robot es el que da carácter móvil al sensor, que permite determinar a su vez cual es el camino a seguir o su localización [33]. Pero también se utilizan estos sensores para mantener y configurar el hardware, detectando posibles anomalías en el comportamiento del dispositivo o incluso controlando los recursos energéticos.

En el campo de las aplicaciones médicas, se han desarrollado por ejemplo redes de sensores capaces de controlar la aceleración y el movimiento de la mano del cirujano a través de un guante repleto de sensores como el que vemos en la Figura 8, algo imprescindible para trabajar en entornos simulados o virtuales, que disminuyen el carácter invasivo en las operaciones, acortando los tiempos de recuperación [34].



**Figura 8.** Imagen del prototipo de guante con sensores y movimientos detectados [34].

## 2.9 Discusión

Después de considerar tanto la extensa cantidad de aplicaciones ya existentes como la versatilidad de los sensores, no queda duda alguna de que, apoyados en un elemento básico como puede ser un sensor y con el impulso reciente y constante de las tecnologías inalámbricas, las WSNs son un instrumento moldeable con el que obtener información de cualquier entorno de interés.

Como hemos visto, detrás de cada sensor y red hay unas necesidades en relación al manejo de la información y a la gestión de los recursos disponibles que precisan ser analizadas e implementadas adecuadamente. Estas necesidades estarán relacionadas con las topologías, los mecanismos para intercambiar la información, el consumo, el entorno, la seguridad requerida, e incluso las restricciones económicas. Todo ello puede provocar que, aunque en ocasiones esta implementación podría ser genérica, dada la variedad de los escenarios existentes, cada problema requiera una solución original y adecuada.

Por último, apartando lo “físico” del problema, no podemos olvidar que, a partir de la información que estamos captando y mediante un tratamiento adecuado, debemos extraer conclusiones y conocimiento en un formato entendible para cualquier usuario. La masividad de las redes enriquece la información, pero puede convertirse en ocasiones en redundante o excesiva. En cualquier caso, son imprescindibles mecanismos para manejarla y es en esa relación donde se enlazan los conceptos clave de este proyecto el aprendizaje e inferencia a partir de la información obtenida en dichas redes de sensores inalámbricas.



# Capítulo 3

## Detección, clasificación y estimación en WSN

### 3.1 Introducción

A lo largo de este capítulo vamos a analizar las tres clases de problemas a los que nos podremos enfrentar cuando tratemos de dar forma a la información que extraemos de la red de sensores, así como sus posibles soluciones. Básicamente se diferencian en función del rango de valores que se pueden obtener en su salida: son la detección, la clasificación y la estimación. A menudo la detección y la clasificación se consideran un único tipo de problema, ya que en ambos casos la variable es discreta: esta puede tomar dos valores en el caso de la detección o un número mayor aunque finito en un caso de clasificación. Si la variable es continua el problema se considera de estimación.

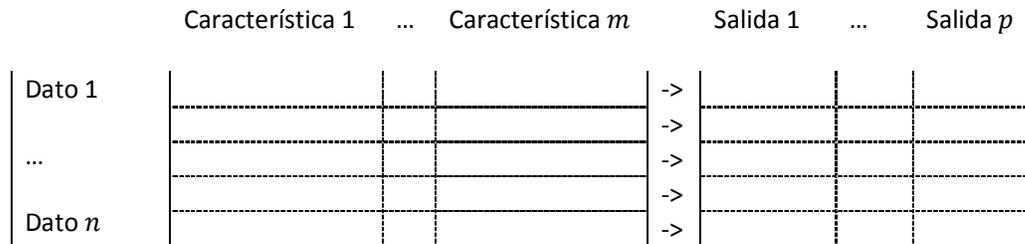
Para terminar de catalogar el problema vamos a revisar la posible naturaleza de los datos disponibles, y cómo los podemos ordenar. Inicialmente consideraremos que cada nodo sensor tomará un valor de una determinada característica y proporcionará una salida acorde a la misma. Trabajaremos entonces con un par de datos que, si fuesen de una dimensión, denominaríamos  $(x, y)$ , donde  $x$  indica la característica medida e  $y$  la función de salida asociada a la misma. Es posible que el mismo experimento se repita  $n$  veces, dando lugar entonces a  $n$  pares de datos:  $(x_i, y_i)_{i=1}^n$ .

Supongamos ahora que un mismo sensor puede medir distintas características, por ejemplo humedad, temperatura y velocidad del viento. En tal caso todas se pueden agrupar en un vector de dimensión  $m$ , que en este caso sería 3.

Imaginemos la posibilidad de que la salida tampoco sea un único parámetro sino varios, por ejemplo la probabilidad de lluvia y la de nieve. En dicho caso de forma análoga

al anterior, las salidas pueden agruparse en un vector, de dimensión  $p$ , que no necesariamente debe coincidir con  $m$  y que en nuestro caso sería 2.

En resumen, los datos se pueden expresar como  $(\mathbf{x}_i, \mathbf{y}_i)_{i=1}^n$ , donde  $\mathbf{x}_i \in \mathbb{R}^{m \times 1}$  e  $\mathbf{y}_i \in \mathbb{R}^{p \times 1}$ , y para un caso general podríamos ordenarlos matricialmente como se muestra en la Figura 9, donde en cada caso, a partir de las  $m$  características de entrada, obtenemos los  $p$  valores de salida.



**Figura 9.** Disposición matricial de los datos.

En este trabajo, típicamente consideraremos que la dimensión de  $\mathbf{y}_i$  es 1 ( $p = 1$ ), de modo que la variable de salida es sólo una en función de todas las entradas, y denotaremos las matrices de datos con mayúsculas y en negrita.

Con esta descripción inicial podemos empezar a detallar los problemas y las soluciones, primero de los casos de detección y clasificación y posteriormente de regresión. Dado que la implementación práctica se ha enfocado más hacia la detección que a la regresión el análisis tiene un carácter más exhaustivo en la primera. De este modo, los contenidos del capítulo se han organizado en dos grandes bloques: detección y estimación.

En la Sección 3.2 trataremos sobre la detección y clasificación. Para comenzar se expone un breve ejemplo de comunicación binaria que nos ayuda a introducir los detectores de máxima verosimilitud, de máxima probabilidad a posteriori y a definir la probabilidad de error. Adicionalmente se detallan otros criterios para establecer decisores como el de Neyman-Pearson, el de Bayes, y el criterio Minimax y se definen conceptos como el estadístico suficiente o el test de hipótesis compuestas.

En cuanto a la estimación, tratada en la Sección 3.3, los contenidos se dividen en estimación bayesiana, y estimación de máxima verosimilitud. Dentro de la estimación bayesiana, se profundiza en el estimador de mínimo error cuadrático medio y en el de máxima probabilidad a posteriori.

## 3.2 Detección/Clasificación

A menudo surge confusión entre los términos detección y decisión. Esto es debido a que el origen de estos problemas en el campo de la ingeniería se remonta a aplicaciones como la del radar, utilizado para detectar blancos, mientras que la rama de las matemáticas que se utiliza para dar solución a los mismos es la teoría de la decisión. No obstante, a lo largo del documento ambos términos, detección y decisión, serán considerados sinónimos: decidir entre dos hipótesis es equivalente resolver un problema de detección de presencia de blanco o ausencia de blanco. Una vez realizada esta aclaración, a continuación se describen cuales son las tareas que componen un proceso de detección o clasificación.

### 3.2.1 Modo de trabajo general

Como se dejó entrever anteriormente, detección y clasificación, se han agrupado bajo el mismo epígrafe, puesto que comparten el carácter discreto de la variable objeto de estudio y la mayoría de las técnicas utilizadas en una pueden extenderse para el uso en la otra. Imaginemos que un sistema de comunicación utiliza un alfabeto de más de dos símbolos para formar el mensaje, como ocurre por ejemplo en una modulación M-PSK o M-QAM. Esto implica una detección de más de dos niveles, donde cada nivel corresponde a una hipótesis distinta sobre el dato de origen. Cuando la variable de salida puede tomar un número de valores concretos y finitos, el problema se cataloga como clasificación y puede enunciarse como encontrar una función de  $\mathbf{X}$  e  $\mathbf{Y}$ , que nos proporcione una salida discreta en el rango  $\{0,1,2, \dots, N - 1\}$  [35]:

$$C: (\mathbf{X}, \mathbf{Y}) \xrightarrow{\text{Donde } \mathbf{X} \in \mathbb{R}^{m \times n} \text{ e } \mathbf{Y} \in \mathbb{R}^{p \times n}} \{0,1,2, \dots, N - 1\}$$

A partir de lo anterior, puede considerarse la detección o decisión como un caso particular de clasificación en el que las opciones que puede tomar como resultado la variable de salida se reducen a 2, típicamente 0 (no detección o ausencia de blanco) y 1 (detección o presencia de blanco) o -1 y +1:

$$D: (\mathbf{X}, \mathbf{Y}) \xrightarrow{\text{Donde } \mathbf{X} \in \mathbb{R}^{m \times n} \text{ e } \mathbf{Y} \in \mathbb{R}^{p \times n}} \{0,1\}$$

En cualquier caso, podemos definir dos etapas básicas en el diseño de un detector o clasificador, la fase de entrenamiento y la fase de test o aplicación [36]. Ambas fases se representan en la Figura 10.

Durante la fase de entrenamiento el clasificador recibe los patrones a detectar a través del vector de características ( $\mathbf{Y}$ ) y de las clases etiquetadas correspondientes ( $\mathbf{X}$ ). Se obtienen unos parámetros denominados pesos,  $\mathbf{W}$  (del inglés, “weights”), que definen las funciones de discriminación o fronteras entre unas regiones de clasificación y otras. Estas

regiones no se solapan y pueden ser puntos, curvas, superficies o hipersuperficies en base a las dimensiones del espacio de entrada.

Posteriormente, durante la fase de test, se aplica el detector/clasificador entrenado sobre un nuevo vector de observaciones disponibles para tener una estima de las etiquetas desconocidas.

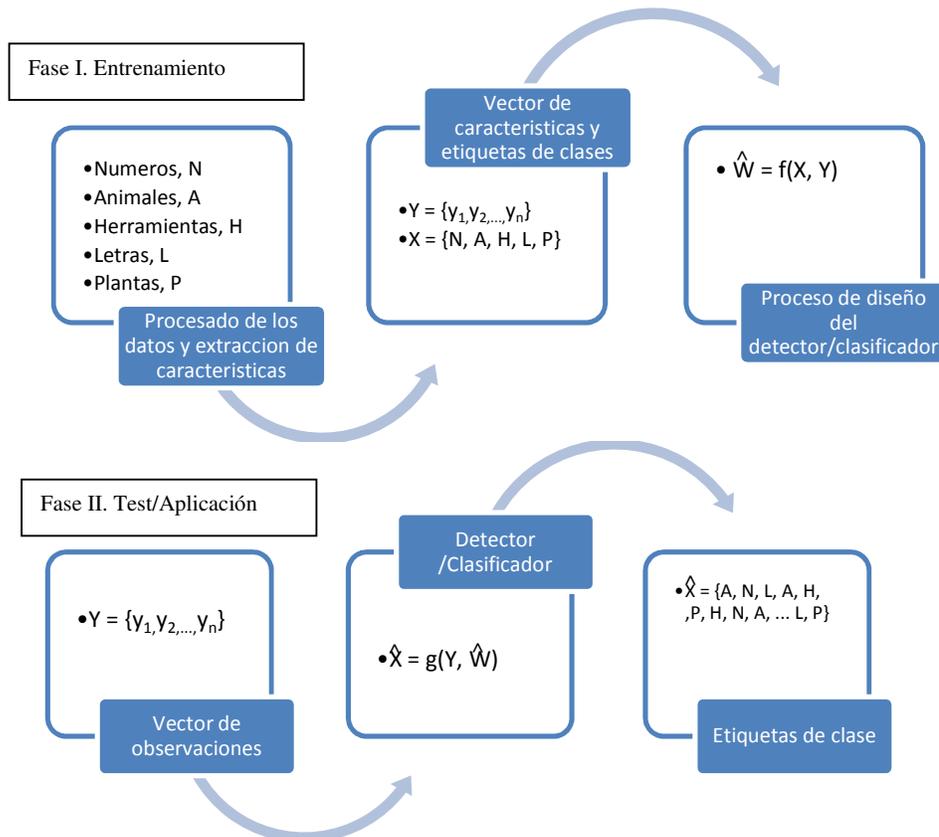


Figura 10. Etapas en el diseño de un clasificador.

Existen muchos criterios diferentes para el diseño de reglas de detección o clasificación: de máxima verosimilitud, Neyman-Pearson, Minimax, de máxima probabilidad a posteriori, etc. A continuación se describen brevemente algunos de los más significativos para el caso concreto de la detección, haciéndose extensibles a problemas de clasificación.

### 3.2.2 Un problema simple: comunicación binaria

Las comunicaciones digitales están íntimamente relacionadas con la detección. Para demostrarlo se desarrolla este simple ejemplo en el que suponemos que el mensaje  $x$  puede tomar valores  $x_0$  o  $x_1$ . Como no conocemos cual es el mensaje que va a ser enviado, modelamos  $x$  como una variable aleatoria que puede tomar dichos valores con probabilidades  $P_0$  y  $P_1 = 1 - P_0$  respectivamente.

El canal de transmisión puede añadir un ruido aleatorio  $w$ , que típicamente se considera Gaussiano y que en este sencillo ejemplo consideramos de media cero y con una varianza  $\sigma_w^2$ . Entonces la variable aleatoria resultante presentada como dato es  $y = x + w$ . A partir de la señal recibida ( $y$ ), conociendo las formas de las distintas opciones que puede tomar el mensaje ( $x$ ), queremos decidir cuál es la señal original enviada. Para convertirlo en un problema de detección de la señal basta con convertir uno de los valores en 0 y otro en 1, de modo que el problema sea equivalente a detectar si se recibe o no señal.

### 3.2.2.1 Detector ML

Si nos adentramos en el análisis estadístico tendremos, tanto para  $x_0$  como para  $x_1$ , una densidad de probabilidad Gaussiana [37]:

$$p_y(y|x) = (2\pi\sigma_w^2)^{-1/2} \exp\left(-\frac{(y-x)^2}{2\sigma_w^2}\right) \quad (3.1)$$

A esta función se le denomina, **función de verosimilitud** dado que, para cada valor concreto de  $y$ , nos indica como de verosímil es que dicha observación haya sido generada por un determinado valor de  $x$ .

A la hora de estudiar los clasificadores para nuestra variable, vamos a comenzar considerando el caso más general, en el que no conocemos las probabilidades a priori de  $x_0$  y  $x_1$ , denominadas  $P_0$  y  $P_1$  respectivamente. En este caso, procederíamos realizando un cociente de las funciones de verosimilitud:

$$\begin{aligned} \lambda(y) &= \frac{p_y(y|x_1)}{p_y(y|x_0)} \\ &= \frac{(2\pi\sigma_w^2)^{-1/2} \exp\left(-\frac{(y-x_1)^2}{2\sigma_w^2}\right)}{(2\pi\sigma_w^2)^{-1/2} \exp\left(-\frac{(y-x_0)^2}{2\sigma_w^2}\right)} \\ &= \exp\left(-\frac{(y-x_1)^2}{2\sigma_w^2} + \frac{(y-x_0)^2}{2\sigma_w^2}\right) \end{aligned} \quad (3.2)$$

Donde vamos a denominar a  $\lambda(y)$  factor de verosimilitud (“**Likelihood Ratio**”) y su valor se va a utilizar para plantear el primer detector: el **test de ratio de verosimilitudes (LRT, “Likelihood Ratio Test”)**. En este test, el factor  $\lambda(y)$  se compara con el umbral,  $\gamma = 1$ , para obtener la regla de decisión:

$$\hat{x}_{ML} = \begin{cases} x_1, & \lambda(y) > 1 \\ x_0, & \lambda(y) < 1 \end{cases} \quad (3.3)$$

Donde el subíndice ML hace referencia a que se trata del decisor de **máxima verosimilitud** (“**Maximum Likelihood**”).

Nótese que  $\lambda(y) > 1$  si y solo si  $p_y(y|x_1) > p_y(y|x_0)$ , mientras que  $\lambda(y) < 1$  si y solo si  $p_y(y|x_0) > p_y(y|x_1)$ . Por tanto,  $\hat{x}_{ML}$  nos aporta el valor de  $x$ , ( $x_0$  o  $x_1$ ) cuya función de verosimilitud sea mayor dada la observación.

Se conoce además que el logaritmo es una función creciente monótona, de modo que el LRT también se puede expresar como:

$$\hat{x}_{ML} = \begin{cases} x_1, & \ln(\lambda(y)) > 0 \\ x_0, & \ln(\lambda(y)) < 0 \end{cases} \quad (3.4)$$

Operando con el logaritmo:

$$\ln(\lambda(y)) = -\frac{(y-x_1)^2}{2\sigma_w^2} + \frac{(y-x_0)^2}{2\sigma_w^2} \quad (3.5)$$

Y si realizamos la comparación con 0 obtenemos:

$$(y-x_0)^2 - (y-x_1)^2 > 0 \quad (3.6)$$

Desarrollando los cuadrados llegamos a la regla de decisión final:

$$y > \frac{(x_0 + x_1)}{2} \quad (3.7)$$

Expresado de modo equivalente como sigue:

$$\hat{x}_{ML} = \begin{cases} x_1, & y > \frac{x_0 + x_1}{2} \\ x_0, & y < \frac{x_0 + x_1}{2} \end{cases} \quad (3.8)$$

Dado que en nuestro caso concreto,  $x_0 = 0$  y  $x_1 = 1$ , obtendríamos como resultando el siguiente decisor:

$$\hat{x}_{ML} = \begin{cases} 1, & y > 1/2 \\ 0, & y < 1/2 \end{cases} \quad (3.9)$$

### 3.2.2.2 Detector MAP

El objetivo del decisor pasa siempre por tomar la mejor opción entre los dos posibles mensajes, esto es, elegir el valor de  $x$  que tiene mayor probabilidad de haber sido enviado en función del valor recibido,  $y$ . En el caso binario esto significa escoger  $x_1$  si  $p_x(x_1|y) > p_x(x_0|y)$  o  $x_0$  en el caso contrario, escogiendo aleatoriamente  $x_1$  o  $x_0$  si las probabilidades son iguales. Haciendo uso de la regla de Bayes y asumiendo conocidas las probabilidades a priori  $P_0$  y  $P_1$ , las **funciones de densidad de probabilidad a posteriori** para  $x_0$  y  $x_1$  se pueden escribir como:

$$p_x(x_1|y) = p_y(y|x_1) \frac{P_1}{p_y(y)} \quad (3.10)$$

$$p_x(x_0|y) = p_y(y|x_0) \frac{P_0}{p_y(y)} \quad (3.11)$$

Si, análogamente al proceso previo, realizamos el cociente entre las dos:

$$\lambda(y) = \frac{p_x(x_1|y)}{p_x(x_0|y)} = \frac{p_y(y|x_1)P_1}{p_y(y|x_0)P_0} > 1 \quad (3.12)$$

Este criterio nos da como resultado el decisor de máxima probabilidad a posteriori o **MAP (“Maximum a posteriori probability”)**:

$$\hat{x}_{\text{MAP}} = \begin{cases} 1, & \lambda(y) > \frac{P_0}{P_1} \\ 0, & \lambda(y) < \frac{P_0}{P_1} \end{cases} \quad (3.13)$$

La aplicación del logaritmo neperiano facilita, como en el caso previo, el trabajo con las funciones exponenciales, de modo que el decisor se pueden reescribir como:

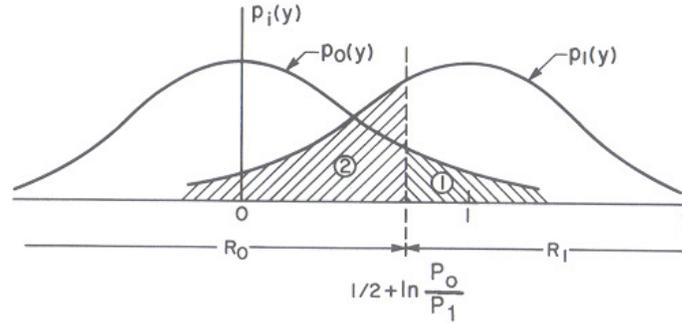
$$\hat{x}_{\text{MAP}} = \begin{cases} x_1, & y > \frac{\sigma_w^2}{x_1 - x_0} \ln\left(\frac{P_0}{P_1}\right) + \frac{x_0 + x_1}{2} \\ x_0, & y < \frac{\sigma_w^2}{x_1 - x_0} \ln\left(\frac{P_0}{P_1}\right) + \frac{x_0 + x_1}{2} \end{cases} \quad (3.14)$$

Si sustituimos de nuevo los valores de nuestro ejemplo,  $x_0 = 0$  y  $x_1 = 1$ , obtenemos:

$$\hat{x}_{\text{MAP}} = \begin{cases} 1, & y > \sigma_w^2 \ln\left(\frac{P_0}{P_1}\right) + \frac{1}{2} \\ 0, & y < \sigma_w^2 \ln\left(\frac{P_0}{P_1}\right) + \frac{1}{2} \end{cases} \quad (3.15)$$

Podemos comprobar que el caso de máxima verosimilitud, en el que no conocemos las probabilidades a priori, es equivalente a suponer que todas las situaciones son equiprobables, de manera que si introducimos  $P_0 = P_1$  en  $\hat{x}_{MAP}$ , alcanzamos el resultado formulado en (3. 9) coincidiendo con  $\hat{x}_{ML}$ .

En la Figura 11, se representan las distribuciones y el umbral, en función de las probabilidades a priori, para el caso concreto de  $x_0 = 0$ ,  $x_1 = 1$ , y  $\sigma_w^2 = 1$ .



**Figura 11.** Regiones de probabilidad de error para un test de hipótesis binario [37].

En este caso, el umbral que separa las dos regiones de decisión,  $R_0$  y  $R_1$ , y delimita las áreas marcadas como (1) o (2) es el siguiente:

$$y_{threshold} = \ln\left(\frac{P_0}{P_1}\right) + \frac{1}{2} \quad (3. 16)$$

Si estudiamos la Figura 11, si  $P_0 > P_1$ , entonces el caso  $x_0$  es más probable que el caso  $x_1$ ,  $\ln\left(\frac{P_0}{P_1}\right) > 0$  y el umbral se desplazaría a la derecha. En el caso contrario, si  $P_0 < P_1$ , entonces  $\ln\left(\frac{P_0}{P_1}\right) < 0$ , la probabilidad de falsa alarma aumenta y la de no detección disminuye y el umbral se desplazaría a la izquierda. Si  $P_0 = P_1$ , ambos casos son equiprobables y el umbral coincidirá con el punto donde se cruzan amabas Gaussianas.

### 3.2.2.3 Probabilidad de error

Volviendo a la Figura 11, las citadas áreas (1) y (2) están relacionadas con las probabilidades de error en sus dos posibles formas:

- Que la detección sea  $x_1$  cuando el símbolo transmitido originalmente era  $x_0$ .
- Que la detección sea  $x_0$  cuando el símbolo transmitido originalmente era  $x_1$ .

Si establecemos las hipótesis como  $H_0$  (transmitir  $x_0$ ) y  $H_1$  (transmitir  $x_1$ ), y las decisiones como  $D_0$  (decidir  $x_0$ ) y  $D_1$  (decidir  $x_1$ ), para los casos correspondientes de no

detección y detección, podemos definir las probabilidades de error para el detector MAP de la siguiente manera.

**Probabilidad de falsa alarma**, corresponde con la probabilidad de decidir  $D_1$  cuando la hipótesis correcta es la  $H_0$ :

$$\begin{aligned}
 P_{fa} &= P(D_1|H_0) \\
 &= P(y > y_{threshold}|H_0) \\
 &= \int_{y_{threshold}}^{\infty} (2\pi\sigma_w^2)^{-\frac{1}{2}} \exp\left(-\frac{(y-x_0)^2}{2\sigma_w^2}\right) dy \\
 &= \int_{\frac{1}{\alpha}\ln\left(\frac{P_0}{P_1}\right)+\frac{\alpha}{2}}^{\infty} (2\pi)^{-\frac{1}{2}} \exp\left(-\frac{u^2}{2}\right) du
 \end{aligned} \tag{3.17}$$

**Probabilidad de no detección**, que corresponde con la probabilidad de decidir  $D_0$  siendo  $H_1$  la hipótesis correcta.

$$\begin{aligned}
 P_{nd} &= P(D_0|H_1) \\
 &= P(y \leq y_{threshold}|H_1) \\
 &= \int_{-\infty}^{y_{threshold}} (2\pi\sigma_w^2)^{-\frac{1}{2}} \exp\left(-\frac{(y-x_1)^2}{2\sigma_w^2}\right) dy \\
 &= \int_{-\infty}^{\frac{1}{\alpha}\ln\left(\frac{P_0}{P_1}\right)-\frac{\alpha}{2}} (2\pi)^{-\frac{1}{2}} \exp\left(-\frac{u^2}{2}\right) du
 \end{aligned} \tag{3.18}$$

En ambos casos,  $\alpha = \frac{|x_1-x_0|}{\sigma_w}$ , es un parámetro que está directamente relacionado con la relación señal a ruido SNR (“Signal to Noise Ratio”).

Sin embargo, en un sistema de comunicaciones no estamos interesados en separar las probabilidades de error de cada tipo. Dado que los sucesos son exclusivos, podemos escribir la probabilidad de error total como:

$$P_e = P(D_1|H_0)P_0 + P(D_0|H_1)P_1 \tag{3.19}$$

### 3.2.3 El criterio de Neyman-Pearson

Los decisores ML o MAP tienen como ventaja el hecho de minimizar la probabilidad de error. Frente a esta ventaja, el principal inconveniente es la forma que se establece el umbral que separa las regiones de decisión. El ratio liga las probabilidades de falsa alarma y no detección e impide modificarlas sin modificar previamente las FDPs o las probabilidades a priori, parámetros que suelen ser establecidos de modo permanente e inamovible por el propio sistema de comunicaciones.

En determinadas aplicaciones, como el radar o el sonar, donde no es esencial controlar la probabilidad de error sino minimizar la probabilidad de falsa alarma, otros criterios como el de Neyman-Person pueden ser más apropiados. Según este, se elige una probabilidad de falsa alarma tan grande como estemos dispuestos a tolerar y bajo esta limitación trataremos de buscar la minimización del error.

Si lo enunciarnos matemáticamente, el objetivo es encontrar una regla de decisión  $D(y)$  tal que:

$$\min_{D(y)} P(D_0|H_1) \quad (3.20)$$

Sujeto a que:

$$P(D_1|H_0) \leq P_{fa} \quad (3.21)$$

Donde  $P_{fa}$  es un valor de probabilidad de falsa alarma establecido a priori por el usuario.

El problema puede enunciarse trabajando con la probabilidad de detección,  $P(D_1|H_1)$ , de modo que el objetivo pasa a ser la maximización de la probabilidad de detección manteniendo la probabilidad de falsa alarma por debajo de un nivel especificado a priori. Puesto que  $1 - P(D_0|H_1) = P(D_1|H_1)$ , entonces el problema se puede reformular como:

$$\max P(D_1|H_1) = \max \int_{R_1} p_y(y|x_1) dy \quad (3.22)$$

Sujeto a:

$$P(D_1|H_0) = \int_{R_1} p_y(y|x_0) dy \leq P_{fa} \quad (3.23)$$

Dado que el espacio completo de  $y$  se separa en  $R_1$  (región crítica en la que rechazamos  $H_0$ ) y el resto, la pregunta ahora es ¿qué puntos de  $y$  deben ser incluidos en  $R_1$ ? La respuesta es bien conocida [37]: aquellos puntos de  $y$  para los cuales  $p_y(y|x_1)$  contribuye en mayor proporción a incrementar  $P(D_1|H_1)$  de lo que contribuye  $p_y(y|x_0)$  a incrementar  $P(D_1|H_0)$ . Fruto de esa comparativa, se llega a la siguiente regla de decisión:

$$\hat{x}_{N-P} = \begin{cases} x_1, & p_y(y|x_1) > \lambda_0 p_y(y|x_0) \\ x_0, & p_y(y|x_1) < \lambda_0 p_y(y|x_0) \end{cases} \quad (3.24)$$

Además, en el caso de  $p_y(y|x_1) = \lambda_0 p_y(y|x_0)$ , la selección entre cualquiera de las dos hipótesis es aleatoria con idéntica probabilidad. Donde la constante  $\lambda_0$ , dependiente de la probabilidad de falsa alarma fijada como límite se convierte en la clave del decisor, permitiéndonos formularlo de modo análogo al ratio de verosimilitudes:

$$\lambda(y) = \frac{p_y(y|x_1)}{p_y(y|x_0)} > \lambda_0 \quad (3. 25)$$

De esta forma, podemos reescribir la regla de decisión de la siguiente manera:

$$\hat{x}_{N-P} = \begin{cases} x_1, & \lambda(y) > \lambda_0 \\ x_0, & \lambda(y) < \lambda_0 \end{cases} \quad (3. 26)$$

La principal ventaja que nos ofrece el criterio de Neyman-Pearson es la garantía de un valor máximo de probabilidad de falsa alarma, sacrificando el hecho de obtener la mínima probabilidad de error. Ese sacrificio implícito es debido a la dependencia que existe entre probabilidad de falsa alarma y probabilidad de no detección, las dos fuentes de error. Cualquier intento de mejorar una de ellas conlleva un detrimento de la contraria, abandonando de este modo el caso óptimo con respecto a la probabilidad de error.

Nótese que, fijando  $\lambda_0 = 1$  se obtendría el detector ML (3. 3), mientras que usando  $\lambda_0 = \frac{P_0}{P_1}$  el detector obtenido es equivalente al MAP (3. 13).

### 3.2.3.1 Ejemplo para el caso Gaussiano

El criterio de Neyman-Pearson se convertirá en capítulos posteriores en la piedra angular del cálculo de los umbrales de potencia de nuestro escenario práctico, pero aquí simplemente vamos a continuar con el caso Gaussiano de comunicación binaria, comparando de este modo los resultados con los detectores previos.

Si suponemos un nivel de falsa alarma tolerado de  $\alpha$ , (esto es, fijamos  $P(D_1|H_0) \leq \alpha$ ), podemos reescribir la probabilidad de falsa alarma como

$$\begin{aligned} P_{fa} &= P(D_1|H_0) \\ &= P(y > y_{threshold}|H_0) \\ &= \int_{y_{threshold}}^{\infty} (2\pi\sigma_w^2)^{-\frac{1}{2}} \exp\left(-\frac{(y-x_0)^2}{2\sigma_w^2}\right) dy \\ &= 1 - \Phi\left(\frac{y_{threshold} - x_0}{\sigma_w}\right) \end{aligned} \quad (3. 27)$$

Donde  $\Phi$  es la función de distribución de una Gaussiana de media 0 y varianza 1 [35]. En tal caso, el umbral expresado en función del nivel de probabilidad de falsa alarma  $\alpha$ , resulta:

$$y_{threshold} = \lambda_0 = \sigma_w \Phi^{-1}(1 - \alpha) + x_0 \quad (3.28)$$

Aplicando las condiciones de nuestro caso concreto de comunicación binaria ( $x_0 = 0, x_1 = 1$ ) el clasificador resulta:

$$\hat{x}_{N-P} = \begin{cases} 1, & y > \sigma_w \Phi^{-1}(1 - \alpha) \\ 0, & y < \sigma_w \Phi^{-1}(1 - \alpha) \end{cases} \quad (3.29)$$

De esta manera, el decisor depende de la probabilidad de falsa alarma,  $\alpha$  y de la función  $\Phi^{-1}$  que se corresponde con la inversa de  $\Phi$  [35].

### 3.2.4 El criterio de Bayes

Un criterio muy útil, y que se explica a continuación como una generalización de los anteriores, es el de Bayes. Este puede establecerse en términos del coste de las diversas salidas, que debe ser asignado por el usuario del proceso de decisión. Una vez asignado, se trata de calcular el coste medio de utilizar un proceso de decisión particular y escoger el que produce un menor coste medio.

A la nomenclatura establecida previamente para las hipótesis y decisiones vamos a sumar un nuevo concepto: el coste. Definiremos  $C_{ij}$  como el coste asociado a elegir  $D_j$ , siendo  $H_i$  la hipótesis verdadera. Dicho coste puede ser positivo, negativo o nulo. Parece razonable que un coste negativo, o beneficio, esté asociado a una decisión correcta, mientras que un coste positivo, o pérdida estará asociado a una decisión incorrecta. En términos generales se asume que el coste de una decisión incorrecta es mayor que el de una decisión correcta (i.e.,  $C_{ij} > C_{ii} \forall i, j$  con  $i \neq j$ ).

Una vez introducido el concepto del coste, supongamos que el problema sigue siendo binario y sólo hay dos posibles hipótesis en el test. En tal caso, el coste medio o el riesgo de cada decisión se formula de la siguiente manera:

$$C = P_0 [C_{00}P(D_0|H_0) + C_{10}P(D_1|H_0)] + P_1 [C_{01}P(D_0|H_1) + C_{11}P(D_1|H_1)] \quad (3.30)$$

Donde  $P_0$  y  $P_1$  son las probabilidades a priori de las dos hipótesis, y el resto se corresponde con las distintas probabilidades enunciadas a continuación.

Probabilidad de falsa alarma:

$$P(D_1|H_0) = \int_{R_1} p_y(y|x_0)dy \quad (3.31)$$

Probabilidad de detección:

$$P(D_1|H_1) = \int_{R_1} p_y(y|x_1)dy \quad (3.32)$$

Probabilidad de rechazo correcto:

$$P(D_0|H_0) = 1 - P(D_1|H_0) \quad (3.33)$$

Probabilidad de no detección:

$$P(D_0|H_1) = 1 - P(D_1|H_1) \quad (3.34)$$

La utilización de estas cuatro ecuaciones nos permite reescribir el coste medio como:

$$C = P_0C_{00} + P_1C_{01} + \int_{R_1} [P_0(C_{10} - C_{00})p_y(y|x_0) - P_1(C_{01} - C_{11})p_y(y|x_1)] dy \quad (3.35)$$

El objetivo es escoger la región  $R_1$  de modo que se minimice el coste medio. Esto es, decidir qué puntos del espacio de los datos deberían ser incluidos en la región de integración correspondiente a la hipótesis  $H_1$ . Es de destacar que los coeficientes que acompañan a  $p_y(y|x_0)$  y  $p_y(y|x_1)$  en la integral son positivos, dado que se presupone que el coste de fallar es mayor que el coste de acertar, (esto es,  $C_{10} > C_{00}$  y  $C_{01} > C_{11}$ ). En consecuencia, para conseguir minimizar el coste medio debemos incluir todos los puntos que hacen el segundo término de la integral mayor que el primero. Esto significa que

$$\int_{R_1} [P_1(C_{01} - C_{11})p_y(y|x_1)]dy > \int_{R_1} [P_0(C_{10} - C_{00})p_y(y|x_0)] dy \quad (3.36)$$

De este modo, reduciremos más el valor de  $C$  comparándolo con el caso en el que no incluyésemos ese término. En consecuencia, el criterio de Bayes de reducir el coste medio nos lleva de nuevo a un test de cociente de verosimilitudes similar al obtenido en el criterio de Neyman-Pearson, el MAP y el ML, con un umbral dependiente de los costes y las probabilidades a priori:

$$\lambda(y) = \frac{p_y(y|x_1)}{p_y(y|x_0)} > \frac{P_0(C_{10} - C_{00})}{P_1(C_{01} - C_{11})} = \lambda_0 \quad (3.37)$$

Y el decisor:

$$\hat{x}_{\text{Bayes}} = \begin{cases} x_1, & p_y(y|x_1) > \lambda_0 p_y(y|x_0) \\ x_0, & p_y(y|x_1) < \lambda_0 p_y(y|x_0) \end{cases} \quad (3.38)$$

Para concluir, vamos a centrarnos en el ejemplo particular usado anteriormente: un sistema de comunicaciones binario. En estos sistemas normalmente no existe una razón para penalizar una elección errónea de  $H_0$  de un modo distinto a una elección errónea de  $H_1$ , por eso vamos a suponer las siguientes restricciones:

$$\begin{aligned} C_{10} &= C_{01} = c \\ C_{00} &= C_{11} = 0 \end{aligned} \quad (3.39)$$

En cuyo caso el coste medio obtenido es:

$$C = c[P_0P(D_1|H_0) + P_1P(D_0|H_1)] = cP_e \quad (3.40)$$

Donde la probabilidad de error es  $P_e$  y  $c$  es una constante que permite que la estrategia de minimizar el coste se convierta en otra equivalente correspondiente a minimizar la probabilidad de error. En tal caso la regla de decisión ahora se formula como sigue, tomando la elección de  $H_1$  si:

$$\lambda(y) = \frac{p_y(y|x_1)}{p_y(y|x_0)} > \frac{P_0}{P_1}$$

Como puede verse, este detector coincide con el test que maximiza las probabilidades a posteriori de las dos hipótesis, esto es, el decisor MAP (3.13).

En definitiva, el criterio de Bayes aporta flexibilidad, al definir el decisor en función de otros parámetros denominados costes que se asocian a su vez a cada una de las combinaciones entre las decisiones e hipótesis existentes.

### 3.2.5 Criterio Minimax

En este último caso se consideran disponibles los costes asignados a las decisiones pero no se dispone de información suficiente para establecer las probabilidades a priori de las hipótesis. La estrategia entonces consiste en utilizar el criterio de Bayes y asignar las probabilidades a priori de tal modo que los efectos de una asignación errónea se minimicen. Esto es, asumimos que ignoramos dichas probabilidades y actuamos de modo que se minimicen las consecuencias. Partiendo del coste medio utilizado en el criterio de Bayes, (3.30) consideremos la estrategia Minimax para el caso binario con las hipótesis descritas hasta ahora (3.35).

El objetivo en este caso es minimizar el máximo valor posible del riesgo del criterio de Bayes teniendo en cuenta la posible variación de la variación de  $P_0$  tal y como se formula en (3. 37).

Para ellos debemos derivar el coste con respecto a  $P_0$ , recordando que  $P_1 = 1 - P_0$ , con lo que obtenemos [35]:

$$\begin{aligned} \frac{dC_{min}}{dP_0} &= C_{00} - C_{01} + \int_{R_1} [(C_{10} - C_{00})p_y(y|x_0) + (C_{01} - C_{11})p_y(y|x_1)]dy \\ &= C_{00} - C_{01} + (C_{10} - C_{00})P(D_1|H_0) + (C_{01} - C_{11})P(D_1|H_1) \end{aligned} \quad (3. 41)$$

Si igualamos a cero la derivada encontraremos el valor de  $P_0$  que minimiza el máximo riesgo de Bayes:

$$C_{00} - C_{01} + (C_{10} - C_{00})P(D_1|H_0) + (C_{01} - C_{11})P(D_1|H_1) = 0 \quad (3. 42)$$

Operando:

$$C_{00}(1 - P(D_1|H_0)) + C_{10}P(D_1|H_0) = C_{01}(1 - P(D_1|H_1)) + C_{11}P(D_1|H_1) \quad (3. 43)$$

Y teniendo en cuenta que  $P(D_0|H_0) = 1 - P(D_1|H_0)$  y  $P(D_0|H_1) = 1 - P(D_1|H_1)$ :

$$C_{00}P(D_0|H_0) + C_{10}P(D_1|H_0) = C_{01}P(D_0|H_1) + C_{11}P(D_1|H_1) \quad (3. 44)$$

Por último, introduciendo el resultado obtenido en la expresión general del coste  $C$  (3. 35):

$$\begin{aligned} C &= P_0[C_{00}P(D_0|H_0) + C_{10}P(D_1|H_0)] + P_1[C_{00}P(D_0|H_0) + C_{10}P(D_1|H_0)] \\ &= (P_0 + P_1)[C_{00}P(D_0|H_0) + C_{10}P(D_1|H_0)] \\ &= C_{00}P(D_0|H_0) + C_{10}P(D_1|H_0) \\ &= C_{01}P(D_0|H_1) + C_{11}P(D_1|H_1) \end{aligned} \quad (3. 45)$$

Así, el coste medio obtenido coincide con el correspondiente a que la hipótesis  $H_0$  sea verdadera y que a su vez es el mismo que el logrado si la hipótesis  $H_1$  es verdadera. Es decir, el coste medio permanece constante independientemente de cual sea la hipótesis verdadera.

Para asentar este criterio se expone un ejemplo numérico con el que al mismo tiempo se introduce la posibilidad de que existan un número de hipótesis y de decisiones posibles mayor. La Tabla 2 contiene los costes de tomar cada decisión bajo cada hipótesis para un dato recibido. Si observamos los costes máximos asociados a una elección errónea para cada clase, estos son 4, 5 y 5 (subrayados). La clasificación siguiendo el criterio Minimax sería “Clase 1”, porque es la que produce un coste máximo menor al decantarse por ella, considerando que pudiese haber sido cualquiera de las otras hipótesis.

Clase\Hipótesis	H1	H2	H3	Máximo
Clase 1	<u>0</u>	<u>4</u>	<u>3</u>	<b>4 (Mínimo Máximo)</b>
Clase 2	2	0	<u>5</u>	5
Clase 3	<u>5</u>	1	0	5

**Tabla 2.** Tabla de costes correspondientes a cada clasificación según hipótesis (Minimax).

Para completar el ejemplo vamos a comprobar qué ocurre siguiendo una estrategia Bayesiana pura, suponiendo las hipótesis equiprobables y manteniendo los costes que habíamos fijado en la Tabla 2. Si promediamos los costes de decidir cada clase para cada una de las hipótesis, tal y como se adjunta en la Tabla 3, la clase para la que se minimiza la función de coste es la 3. El resultado es una clase distinta a la clase 1 obtenida con el criterio de Minimax, dando una muestra de cómo es posible, para un mismo escenario, obtener decisores distintos según el criterio aplicado.

Clase\Hipótesis	H1	H2	H3	Coste
Clase 1	0	4	3	7/3
Clase 2	2	0	5	7/3
Clase 3	5	1	0	<b>6/3</b>

**Tabla 3.** Tabla de costes correspondientes a cada clasificación según hipótesis (Bayesiano).

### 3.2.6 Estadístico suficiente

Después de considerar diferentes tipos de detectores, en esta sección se detalla un concepto que permite simplificar las decisiones o clasificaciones gracias a la observación de determinadas características que contienen toda la información útil necesaria para la detección.

Imaginemos que estamos interesados en una variable aleatoria,  $y$ , cuya función de densidad  $p_y(y|\theta)$  depende del parámetro  $\theta$ , y supongamos que se encuentra una función  $s(y)$  tal que:

$$p_y(y|\theta, s) = p_y(y|s) \tag{3.46}$$

Esto es, la probabilidad condicionada al parámetro  $\theta$  y a la función  $s(y)$  únicamente depende en realidad de la función  $s(y)$ . En tal caso,  $s(y)$  se denomina **estadístico suficiente** para  $\theta$ , porque la información sobre el parámetro de interés  $\theta$  es aportada por completo por  $s(y)$ , que típicamente tiene una dimensión menor a la de  $y$ , facilitando así el trabajo.

Como consecuencia, la FDP a posteriori necesaria para la detección se puede expresar de modo equivalente como:

$$p_{\theta}(\theta|y, s) = \frac{p_y(y|\theta, s)p_{\theta}(\theta|s)}{p_y(y|s)} = p_{\theta}(\theta|s) \quad (3.47)$$

Por lo tanto, si pretendiésemos establecer un cociente de verosimilitudes para discriminar entre dos hipótesis:

$$\begin{aligned} H_0: \theta &= \theta_0 \\ H_1: \theta &= \theta_1 \end{aligned} \quad (3.48)$$

El resultado dependería exclusivamente del estadístico suficiente:

$$\lambda(y, s) = \frac{p_{y,s}(y, s|\theta_1)}{p_{y,s}(y, s|\theta_0)} = \frac{p_s(s|\theta_1)}{p_s(s|\theta_0)} = \lambda(s) \quad (3.49)$$

Por último, nótese que, aunque el desarrollo de esta sección se haya realizado para el detector Bayesiano, se puede extender de forma análoga a todos los detectores o clasificadores enunciados en este capítulo.

### 3.2.6 Test de hipótesis compuestas

En ocasiones la densidad de probabilidad bajo cada hipótesis no está completamente especificada. Por ejemplo, supongamos que estamos interesados en contrastar dos hipótesis:

$$\begin{aligned} H_0: y &\sim N(0, \sigma_w^2) \\ H_1: y &\sim N(A, \sigma_w^2) \end{aligned} \quad (3.50)$$

Donde  $A$  y  $\sigma_w^2$ , tienen un valor desconocido. Si realizamos un test de verosimilitud buscando el decisor:

$$\begin{aligned}
 \lambda(y) &= \frac{p_y(y|x_1)}{p_y(y|x_0)} \\
 &= \frac{\exp\left(-\frac{(y-A)^2}{2\sigma_w^2}\right)}{\exp\left(-\frac{y^2}{2\sigma_w^2}\right)} \\
 &= \exp\left(-\frac{A^2 - 2yA}{2\sigma_w^2}\right)
 \end{aligned} \tag{3.51}$$

El cociente, que depende de la varianza y de  $A$ , no puede ser calculado. En estos casos las hipótesis, que dependen de parámetros desconocidos, se denominan **hipótesis compuestas** [38]. Dado que dichos parámetros son desconocidos, el primer paso para resolver este tipo de situaciones es estimarlos.

Si el parámetro que se desconoce (supongamos que es  $\alpha$ ) es aleatorio, hay dos formas de actuar:

- La primera permite, si se conocen las funciones de densidad condicionadas a cada hipótesis,  $p(\alpha|H_i)$ , eliminar la dependencia del parámetro. Esto se consigue integrando y permite seguir minimizando el coste medio como en casos anteriores.

$$p(x|H_i) = \int_{\Lambda_i} p(x|\alpha, H_i) p(\alpha|H_i) d\alpha \tag{3.52}$$

Donde  $\Lambda_i$  denota el rango de posibles valores de  $\alpha$  para la hipótesis  $H_i$ .

- La segunda contempla el hecho de estimar las funciones de densidad desconocidas mediante el uso de histogramas, ventanas de Parzen o cualquier otro método [39].

Por otro lado, si el parámetro que se desconoce es determinista, se podría recurrir a estrategias del tipo Minimax o estimar el parámetro bajo cada hipótesis para construir un test bayesiano con dichas estimaciones, como se hace por ejemplo en los tests de verosimilitud generalizados (GLRT):

$$\Lambda_G(x) = \begin{matrix} D_1 \\ \frac{p(x|\hat{y}_1, H_1)}{p(x|\hat{y}_0, H_0)} > \\ D_0 \end{matrix} \eta \tag{3.53}$$

Donde  $\hat{y}_1$ , es la estima ML de  $y$  (que depende del parámetro desconocido  $\alpha$ ) suponiendo que la hipótesis correcta es  $H_1$ , e  $\hat{y}_0$  es la estima ML de  $y$  suponiendo que la hipótesis correcta es  $H_0$ .

### 3.2.7 Clasificación

Después de repasar los diferentes tipos de decisores, vamos a considerar a partir de ahora que la variable discreta objeto de estudio tiene un rango de salida superior a dos opciones, es decir, que existen más de 2 clases. Todos los conceptos descritos previamente, como el estadístico suficiente o las hipótesis compuestas, son extrapolables a este nuevo escenario, así como todas las técnicas utilizadas (ML, MAP, Neyman-Pearson, Bayes y Minimax). A continuación se muestra como ejemplo la extensión del un test Bayesiano al caso con múltiples hipótesis alternativas. Sin restar importancia al resto de clasificadores se detalla un único ejemplo, puesto que la implementación práctica del proyecto está enfocada hacia un escenario de detección.

#### 3.2.7.1 Test con múltiples hipótesis alternativas

Si extendemos la estrategia de Bayes considerando ahora  $m$  hipótesis, el coste se formula como:

$$C = \sum_{i=1}^m \sum_{j=1}^m C_{ij} P(D_i | H_j) P_j \quad (3.54)$$

Donde:

- $C_{ij}$ : Coste de decidir  $D_i$  siendo la hipótesis correcta  $H_j$ .
- $P(D_i | H_j)$ : Probabilidad de decidir  $D_i$  dada la hipótesis  $H_j$ .
- $P_j$ : Probabilidad a priori de la hipótesis  $H_j$ .

Como en el caso del decisor, buscamos una regla para clasificar que minimice el coste. Ahora  $R_i$  es la región del espacio de  $y$  en la que se toma la decisión  $D_i$  para la hipótesis  $H_i$ , esto es, en la que se decide correctamente la hipótesis  $H_i$ .

Dado que las regiones de decisión deben incluir todo el espacio de los datos,  $R$ , y son regiones disjuntas,  $R_i \cap R_j = \emptyset, \forall i \neq j$ , (cada punto,  $y$ , puede corresponder exclusivamente a una decisión), podemos escribir:

$$R = \bigcup_{i=1}^m R_i \quad (3.55)$$

Sabiendo además que:

$$R_j = R \setminus \bigcup_{\substack{i=1 \\ i \neq j}}^m R_i \quad (3.56)$$

A partir de estas definiciones:

$$\begin{aligned} P(D_j|H_j) &= \int_{R_j} p_y(y|x_j) dy \\ &= \int_R p_y(y|x_j) dy - \sum_{\substack{i=1 \\ i \neq j}}^m \int_{R_i} p_y(y|x_j) dy \\ &= 1 - \sum_{\substack{i=1 \\ i \neq j}}^m P(D_i|H_j) \end{aligned} \quad (3.57)$$

Usando esta última expresión, el coste puede reescribirse de tal modo que:

$$C = \sum_{j=1}^m P_j \left\{ C_{jj} + \sum_{\substack{i=1 \\ i \neq j}}^m (C_{ij} - C_{jj}) \int_{R_i} p_y(y|x_j) dy \right\} \quad (3.58)$$

Donde, en el sumatorio en  $i$ , el término  $i = j$  puede ser incluido formalmente, dado que su valor es cero. Reorganizando la ecuación anterior llegamos a:

$$C = \sum_{j=1}^m C_{jj} P_j + \sum_{i=1}^m \int_{R_i} \left\{ \sum_{j=1}^m P_j (C_{ij} - C_{jj}) p_y(y|x_j) dy \right\} \quad (3.59)$$

Si asumimos que el coste de una decisión incorrecta es siempre mayor que el de la decisión correcta, el primer término es fijo y el segundo es el que queremos minimizar, asignando cada dato,  $y$ , a la región  $R_i$  a la que contribuye menos. Esto es, se desea minimizar la siguiente cantidad, que denominaremos **función de coste**:

$$\lambda_i = \sum_{j=1}^m P_j (C_{ij} - C_{jj}) p_y(y|x_j) \quad (3.60)$$

El problema puede resultar difícil de resolver en un caso general. Sin embargo, un caso especial, y el más sencillo, es aquel en el que las decisiones correctas tienen coste cero y todas las decisiones incorrectas tienen el mismo coste, tomado a menudo como la unidad:

$$C_{ij} = \begin{cases} 1, & i \neq j \\ 0, & i = j \end{cases} \quad (3.61)$$

Para tal caso el mínimo de la función de coste resulta:

$$\begin{aligned} \lambda_i &= \sum_{\substack{j=1 \\ i \neq j}}^m P_j p_y(y|x_j) \\ &= \sum_{\substack{j=1 \\ i \neq j}}^m p(y, x_j) \\ &= p_y(y) \sum_{\substack{j=1 \\ i \neq j}}^m P(H_j|y) \end{aligned} \quad (3.62)$$

Además, dado que  $p_y(y)$  es independiente de  $i$ , nos permite reducirlo a elegir la hipótesis  $H_i$  para la cual se minimiza:

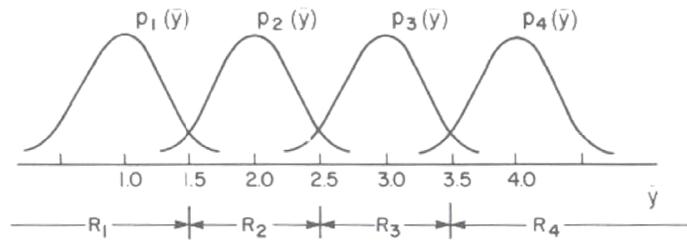
$$\sum_{\substack{j=1 \\ i \neq j}}^m P(H_j|y) = 1 - P(H_i|y) \quad (3.63)$$

De modo equivalente: se escoge la hipótesis  $H_i$  para la cual la probabilidad a posteriori  $P(H_i|y)$  es máxima, proporcionando el siguiente resultado para el coste medio:

$$\begin{aligned} C &= \sum_{j=1}^m P_j \sum_{\substack{j=1 \\ i \neq j}}^m P(D_i|H_j) \\ &= \sum_{j=1}^m P_j P_e(H_j) \\ &= P_e \end{aligned} \quad (3.64)$$

Donde la  $P_e(H_j)$  es la probabilidad de error dada la hipótesis  $H_j$  y  $P_e$  la probabilidad de error total para todo el conjunto de hipótesis.

Para completar la información anterior, a continuación se muestra un ejemplo concreto con cuatro hipótesis. En la Figura 12, se representan las formas de las 4 densidades de probabilidad condicionadas. A simple vista podemos comprobar que sus medias son 1, 2, 3 y 4 respectivamente, con probabilidades a priori que supondremos iguales a  $1/4$ .



**Figura 12.** Funciones de densidad condicionadas y regiones de clasificación [37].

Para encontrar las regiones de clasificación siguiendo las pautas anteriormente descritas, deberíamos fijarnos en cuál de las 4 fdps condicionadas  $[P(H_1|y), P(H_2|y), P(H_3|y), P(H_4|y)]$  toma un valor mayor en cada región. De acuerdo con esta observación, los umbrales que marcan la frontera entre clases se obtienen simplemente a partir de los valores de  $y$  para los que se igualan las probabilidades. Por ejemplo, el umbral que separa las regiones  $R_1$  y  $R_2$  coincide con  $y = 1.5$ , valor de  $y$  para el cual  $p_1(y) = p_2(y)$  o coincidiendo con nuestra nomenclatura  $(P(H_1|y) = P(H_2|y))$ . Ídem para el resto, ya que las hipótesis son equiprobables y todos los costes asociados a un error son idénticos. Lo mismo ocurre para los costes asociados a cada acierto.

### 3.3 Estimación

La segunda clase de problemas que vamos a encontrarnos es la estimación. La principal diferencia con el anterior problema radica en que, en lugar de tratar con un conjunto finito y discreto de clases o hipótesis, el objetivo ahora es encontrar una función de  $\mathbf{X}$  e  $\mathbf{Y}$  que nos proporcione una salida continua. Siendo  $\mathbf{X}$  el conjunto de valores reales de las características observadas e  $\mathbf{Y}$  las propias medidas realizadas en un entorno determinado con ruido.

$$E: (\mathbf{X}, \mathbf{Y}) \xrightarrow{\text{Donde } \mathbf{X} \in \mathbb{R}^{m \times n} \text{ e } \mathbf{Y} \in \mathbb{R}^{p \times n}} S \subseteq \mathbb{R}$$

Existen dos versiones diferentes de este problema En la primera el vector  $\mathbf{x}$  puede estar formado por un conjunto de parámetros deterministas aunque desconocidos relacionados con los datos  $\mathbf{y}$ . Estos parámetros se tratan como constantes no aleatorias con valores desconocidos dando origen a estimadores como mínimos cuadrados, (LS, “Least Squares”) o el de máxima verosimilitud, (ML, “Maximum Likelihood”).

En el segundo caso  $\mathbf{x}$  e  $\mathbf{y}$  son variables aleatorias cuya relación es modelada a partir de una función de densidad de probabilidad conjunta, dando origen a estimadores Bayesianos. Este segundo caso es en el que nos concentraremos.

### 3.3.1 Tipos de problemas de estimación

Independientemente del carácter de los parámetros, el objetivo de la aplicación nos permite también clasificar el problema de estimación como regresión, filtrado o predicción. De este modo, si el objetivo es encontrar una función que se ajuste sobre los datos obtenidos, estaremos entonces en un caso de **regresión**. Como reseña, el primer modelo de regresión utilizado fue el publicado por Legendre en 1805 y por Gauss en 1809: el método de Mínimos Cuadrados [40]. Este establecía un modelo lineal que relacionaba la variable dependiente con la variable observada. También es posible encontrar modelos de regresión que, a pesar de no ser lineales, puedan linealizarse, y por supuesto modelos no lineales. Por ejemplo, considérese un modelo de regresión tal que  $y_i = \alpha_1 \exp(\sqrt{\alpha_2 x_i} + \alpha_3 x_i^2) + \varepsilon_i$ , donde  $\varepsilon_i$  es el ruido de observación. Entonces, la función de regresión  $m(x, \underline{\alpha}) = \alpha_1 \exp(\sqrt{\alpha_2 x} + \alpha_3 x^2)$  no es lineal ni se puede transformar en lineal fácilmente [41].

En otras ocasiones el objetivo es estimar un valor actual, estamos entonces frente a un problema de **filtrado**. El filtro de Kalman es probablemente el tipo más común de filtro, omnipresente en las radios de FM, así como en la mayoría de los equipos de comunicaciones electrónicas. Su propósito es utilizar las medidas que se observan en el tiempo, que contienen ruido y otras inexactitudes, para producir valores que tienden a estar más cerca de los verdaderos valores de las mediciones. Dicho filtro estima la incertidumbre del valor que se supone verdadero y calcula una media ponderada de los estimados como verdaderos y de los medidos, asignando más peso a los que tengan menor incertidumbre. De este modo, la media ponderada produce una menor incertidumbre que cada uno de los valores por separado.

El último caso posible es aquel en que la estimación se realiza para valores futuros, en cuyo caso se cataloga el problema como de **predicción**. No obstante, la línea que separa unos casos de otros es muy delgada y habitualmente los mismos modelos de regresión pueden utilizarse para el filtrado o la predicción valores futuros con ligeras modificaciones.

### 3.3.2 Estimación Bayesiana

#### 3.3.2.1 Un estimador Bayesiano general

Para comenzar, consideremos un vector de observaciones  $\mathbf{y}$  cuyo contenido es un conjunto de variables aleatorias. Dicho vector está relacionado de algún modo con otro vector de parámetros  $\mathbf{x}$  cuyos valores nos gustaría conocer. Esa relación está definida por la densidad de probabilidad conjunta  $p(\mathbf{x}, \mathbf{y})$ , a partir de la cual buscamos un algoritmo que nos indique cómo calcular una aproximación para  $\mathbf{x}$ . Es decir, deseamos construir una función de los datos tal que  $\hat{\mathbf{x}}(\mathbf{y}) \approx \mathbf{x}$ .

Independientemente del algoritmo, vamos a cometer errores. Como el resultado del estimador Bayesiano es una variable aleatoria, el error también lo será:

$$\boldsymbol{\varepsilon} = \boldsymbol{x} - \hat{\boldsymbol{x}}(\boldsymbol{y}) \quad (3.65)$$

Además, como ya se ha explicado anteriormente, cometer ese error lleva asociado un coste, que se puede definir mediante una función escalar: la función de coste. En el caso concreto de la estimación Bayesiana, queremos minimizar el coste medio, de modo que:

$$\hat{\boldsymbol{x}}(\boldsymbol{y}) = \underset{\hat{\boldsymbol{x}}}{\operatorname{argmín}} E\{C(\boldsymbol{\varepsilon})\} \quad (3.66)$$

Donde  $E\{\cdot\}$  denota el operador esperanza matemática, tomada con respecto a  $p(\boldsymbol{x}, \boldsymbol{y})$ . El primer paso para encontrar una solución es convertir el problema de seleccionar una función que minimice  $\hat{\boldsymbol{x}}(\boldsymbol{y})$ , en otro problema equivalente de cálculo ordinario cuyo objetivo es la minimización de  $\hat{\boldsymbol{x}}$ . Si trabajamos con el coste medio:

$$\begin{aligned} E\{C(\boldsymbol{\varepsilon})\} &= E\{C(\boldsymbol{x}, \hat{\boldsymbol{x}}(\boldsymbol{y}))\} \\ &= \int_Y \int_X C(\boldsymbol{x}, \hat{\boldsymbol{x}}(\boldsymbol{y})) p_{\boldsymbol{x}, \boldsymbol{y}}(\boldsymbol{x}, \boldsymbol{y}) d\boldsymbol{x} d\boldsymbol{y} \\ &= \int_Y \left[ \int_X C(\boldsymbol{x}, \hat{\boldsymbol{x}}(\boldsymbol{y})) p_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} \right] p_{\boldsymbol{y}}(\boldsymbol{y}) d\boldsymbol{y} \end{aligned} \quad (3.67)$$

Como la función  $p_{\boldsymbol{y}}(\boldsymbol{y})$  es no negativa, el coste medio se minimizará si para cada valor fijo de  $\boldsymbol{y}$ , seleccionamos un  $\hat{\boldsymbol{x}}(\boldsymbol{y})$  que minimice la integral mas interna. Es decir:

$$\begin{aligned} \hat{\boldsymbol{x}}_{\text{Bayes}} &= \underset{\hat{\boldsymbol{x}}}{\operatorname{argmín}} E\{C(\boldsymbol{x}, \hat{\boldsymbol{x}})|\boldsymbol{y}\} \\ &= \underset{\hat{\boldsymbol{x}}}{\operatorname{argmín}} \int_X C(\boldsymbol{x}, \hat{\boldsymbol{x}}(\boldsymbol{y})) p_{\boldsymbol{x}|\boldsymbol{y}}(\boldsymbol{x}|\boldsymbol{y}) d\boldsymbol{x} \end{aligned} \quad (3.68)$$

Donde la esperanza  $E\{C(\boldsymbol{x}, \hat{\boldsymbol{x}})|\boldsymbol{y}\}$  se corresponde con la media condicionada sobre  $\boldsymbol{x}$  para un  $\boldsymbol{y}$  dado, coincidiendo con la integral entre corchetes. A partir de este caso general, dependiendo de la función de coste vamos a tener una serie de estimadores particulares. A continuación, se describen los dos métodos más utilizados: el de mínimo error cuadrático medio (MMSE) y el de máxima probabilidad a posteriori (MAP).

### 3.3.2.2 Estimador de Mínimo Error Cuadrático Medio (MMSE)

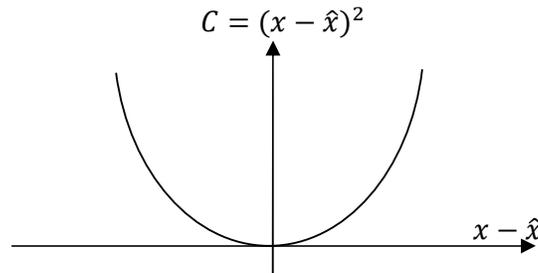
El primero de los estimadores derivados de la media condicionada es el MMSE, (“Minimum Mean Squared Error”). Como su propio nombre indica este es un estimador que minimiza el error cuadrático medio esperado. En este caso, la función de coste es cuadrática:

$$C(\mathbf{x} - \hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{A} (\mathbf{x} - \hat{\mathbf{x}}) \quad (3.69)$$

Donde  $\mathbf{A}$  es cualquier matriz simétrica definida positiva, es decir, una matriz hermítica cuadrada que cumple cualquiera de las siguientes condiciones equivalentes [42]:

- Para cualquier vector no nulo  $\mathbf{z} \in \mathbb{R}^n$ :  $\mathbf{z}^T \mathbf{A} \mathbf{z} > 0$ .
- Todos los autovalores  $\lambda_i$  de  $\mathbf{A}$  son positivos.
- La función  $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{A} \mathbf{y}$  define un producto interno en  $\mathbb{R}^n$ .
- Todos los determinantes de los menores principales de  $\mathbf{A}$  son positivos. Donde se definen los menores principales de  $\mathbf{A}$  como los determinantes de un conjunto de submatrices cuadradas de  $\mathbf{A}$ , y se considera  $T$  como transposición.

Si nos centramos en el caso de una dimensión, gráficamente, dicha función se representaría como se muestra en la Figura 13:



**Figura 13.** Función de coste cuadrática para el estimador MMSE [37].

En cualquier caso, si retomamos la expresión general y si derivamos  $C(\mathbf{x} - \hat{\mathbf{x}})$  con respecto a  $\hat{\mathbf{x}}$  e igualamos a cero obtenemos, para buscar los posibles máximos o mínimos:

$$-2 \int_{\mathbf{X}} \mathbf{A} (\mathbf{x} - \hat{\mathbf{x}}) p_{\mathbf{x}}(\mathbf{x}|\mathbf{y}) d\mathbf{x} = \mathbf{0} \quad (3.70)$$

Si operamos:

$$\mathbf{A} \hat{\mathbf{x}} \int_{\mathbf{X}} p_{\mathbf{x}}(\mathbf{x}|\mathbf{y}) d\mathbf{x} = \mathbf{A} \int_{\mathbf{X}} \mathbf{x} p_{\mathbf{x}}(\mathbf{x}|\mathbf{y}) d\mathbf{x} \quad (3.71)$$

Haciendo uso del hecho de que  $\mathbf{A}$  es definida positiva, y por tanto tiene inversa, llegamos al estimador:

$$\hat{\mathbf{x}}_{\text{MMSE}} = \int_{\mathbf{X}} \mathbf{x} p_{\mathbf{x}}(\mathbf{x}|\mathbf{y}) d\mathbf{x} = E\{\mathbf{x}|\mathbf{y}\} \quad (3.72)$$

Para comprobar que minimiza el error cuadrático calculamos la segunda derivada:

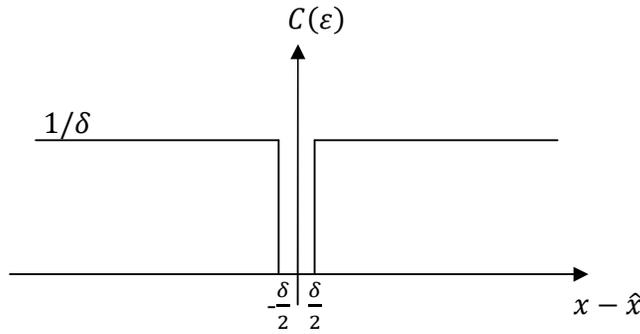
$$\left(\frac{\partial}{\partial \hat{\mathbf{x}}}\right)\left(-2 \int_{\mathbf{x}} \mathbf{A}(\mathbf{x} - \hat{\mathbf{x}}) p_{\mathbf{x}}(\mathbf{x}|\mathbf{y}) d\mathbf{x}\right) = 2\mathbf{A} \int_{\mathbf{x}} p_{\mathbf{x}}(\mathbf{x}|\mathbf{y}) d\mathbf{x} = 2\mathbf{A} > \mathbf{0} \quad (3.73)$$

Como la matriz  $\mathbf{A}$  es definida positiva estamos ante un mínimo. Si además la matriz coincide con la identidad, el estimador minimiza la varianza del error ya que:

$$C(\boldsymbol{\varepsilon}) = \boldsymbol{\varepsilon}^T \boldsymbol{\varepsilon} = (\mathbf{x} - \hat{\mathbf{x}})^T (\mathbf{x} - \hat{\mathbf{x}}) \quad (3.74)$$

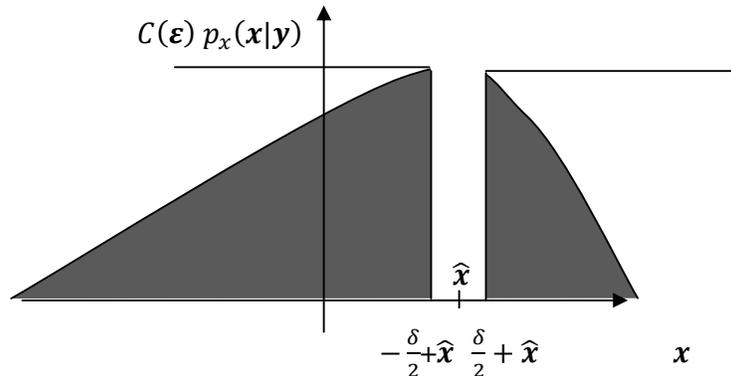
### 3.3.2.3 Estimador de Máxima Probabilidad a Posteriori (MAP)

Otro caso particular de estimador Bayesiano es el MAP. La función de coste para el caso unidimensional es ahora uniforme, pero con un cero en el origen, (esto es en  $x = \hat{x}$ ) como muestra la Figura 14. El estudio de la misma debe realizarse considerando un  $\delta$  muy pequeño ( $\delta \rightarrow 0$ ), sin llegar a ser cero, de modo que la función de coste es muy elevada e idéntica para cualquier valor de la diferencia ( $x - \hat{x}$ ), excepto para el caso en el que  $\hat{x}$  es igual a  $x$ .



**Figura 14.** Función de coste uniforme para el estimador MAP [37].

Gráficamente, podríamos decir que el intervalo de valor de coste 0, debe coincidir con el máximo de la  $p_{\mathbf{x}}(\mathbf{x}|\mathbf{y})$ , para minimizar el resultado de la integral, tal y como se muestra en la Figura 15.



**Figura 15.** Producto coste uniforme por probabilidad condicionada.

La formulación correspondiente parte del producto del coste y la probabilidad:

$$E\{C(\hat{\mathbf{x}} - \mathbf{x})|\mathbf{y}\} = \int_{-\infty}^{\infty} C(\hat{\mathbf{x}} - \mathbf{x}) p_x(\mathbf{x}|\mathbf{y}) d\mathbf{x} \quad (3.75)$$

Si separamos la integral en las dos regiones distintas de cero:

$$E\{C(\hat{\mathbf{x}} - \mathbf{x})|\mathbf{y}\} = \frac{1}{\delta} \left[ \int_{-\infty}^{\hat{\mathbf{x}} - \frac{\delta}{2}} p_x(\mathbf{x}|\mathbf{y}) d\mathbf{x} + \int_{\hat{\mathbf{x}} + \frac{\delta}{2}}^{\infty} p_x(\mathbf{x}|\mathbf{y}) d\mathbf{x} \right] \quad (3.76)$$

De modo equivalente, si lo expresamos en función del valor de la probabilidad en dicho intervalo:

$$\begin{aligned} E\{C(\hat{\mathbf{x}} - \mathbf{x})|\mathbf{y}\} &= \frac{1}{\delta} \left[ 1 - \int_{\hat{\mathbf{x}} - \frac{\delta}{2}}^{\hat{\mathbf{x}} + \frac{\delta}{2}} p_x(\mathbf{x}|\mathbf{y}) d\mathbf{x} \right] \\ &\approx \frac{1}{\delta} [1 - \delta p_x(\hat{\mathbf{x}}|\mathbf{y})] \\ &\approx \frac{1}{\delta} - p_x(\hat{\mathbf{x}}|\mathbf{y}) \end{aligned} \quad (3.77)$$

Como puede comprobarse, la esperanza del coste condicionado es muy elevada y se minimiza cuando se maximiza  $p_x(\mathbf{x}|\mathbf{y})$ , lo cual se enuncia como sigue:

$$\hat{\mathbf{x}}_{\text{MAP}} = \underset{\mathbf{x}}{\operatorname{argmax}} p_x(\mathbf{x}|\mathbf{y}) \quad (3.78)$$

Como  $p_x(\mathbf{x}|\mathbf{y}) = p_y(\mathbf{y}|\mathbf{x})p_x(\mathbf{x})/p_y(\mathbf{y})$  por la regla de Bayes y  $p_y(\mathbf{y})$  no involucra a  $\mathbf{x}$ , el estimador MAP también se puede formular como:

$$\hat{\mathbf{x}}_{\text{MAP}} = \underset{\mathbf{x}}{\operatorname{argmax}} p_y(\mathbf{y}|\mathbf{x}) p_x(\mathbf{x}) \quad (3.79)$$

En el caso de que  $\mathbf{x}$  e  $\mathbf{y}$  sean variables aleatorias Gaussianas, la densidad a posteriori también es Gaussiana y su máximo coincide con su media. En este caso, los estimadores MMSE y MAP coincidirán. En general, esto ocurrirá siempre que  $p_y(\mathbf{y}|\mathbf{x})$  sea unimodal y simétrica.

### 3.3.3 Estimador de Máxima Verosimilitud (ML)

El último de los estimadores que vamos a detallar es el de máxima verosimilitud. Formalmente, ahora se considera que  $\mathbf{x}$  es determinista aunque desconocido y, para introducirlo, vamos a apoyarnos en el estimador anterior (3.79). Si suponemos ahora que

$p_x(\mathbf{x})$  es una función que se expande ampliamente con respecto a  $p_y(\mathbf{y}|\mathbf{x})$ , de forma que la FDP a priori no es informativa, es decir, no hay valores de  $\mathbf{x}$  que tengan una probabilidad de ocurrencia a priori mayor que el resto, entonces el máximo del producto coincide con el máximo del factor con  $p_y(\mathbf{y}|\mathbf{x})$ , dando origen a otro estimador por sí mismo, el estimador de máxima verosimilitud de  $\mathbf{x}$ .

$$\hat{\mathbf{x}}_{\text{ML}} = \underset{\mathbf{x}}{\operatorname{argmax}} p_y(\mathbf{y}|\mathbf{x}), \quad (3.80)$$

La principal virtud del estimador ML es que no requiere información a priori, simplemente toma  $p_y(\mathbf{y}|\mathbf{x})$ , y lo maximiza. Por tanto, suele utilizarse, cuando no se conoce ninguna información estadística a priori o en casos en los que el estimador Bayesiano es difícil de calcular. Para obtenerlo sólo es necesario saber cómo se obtiene el dato  $\mathbf{y}$  a partir de una señal específica  $\mathbf{x}$ , información que a menudo está fácilmente disponible.

## 3.4 Discusión

El capítulo que se cierra con esta sección es un repaso a la teoría de detectores, clasificadores y estimadores. Un repaso más profundo en determinadas ocasiones (detección) y superficial en otras (estimación) que resultarán menos relevantes en el desarrollo práctico del proyecto. La primera lección aprendida es la distinción de cada tipo de problema según el rango de su salida. De este modo, si la variable es discreta y tiene 2 clases es un decisor (o detector), si tiene más, estamos ante un problema de clasificación. En última instancia, si la variable es continua, la herramienta que necesitamos es un estimador.

Posteriormente se han recorrido los detectores, desde la situación donde se dispone de menos información hasta la situación en la que se dispone de más información. Así, a partir del test de funciones de verosimilitud, donde no conocemos más que dicha función de verosimilitud hemos llegado a los detectores MAP y posteriormente generalizado al caso Bayesiano, donde sí se conocen las probabilidades a priori de cada hipótesis. Estos decisores, que nos proporcionan la probabilidad de error mínima, no son siempre los más adecuados, por ese motivo se ha expuesto el de Neyman-Pearson, que garantiza una determinada probabilidad de falsa alarma. Por último, se ha presentado el criterio Minimax, que trata de minimizar el riesgo máximo. Entre todos estos conceptos se han intercalado los test de hipótesis compuestas y la ventaja que supone encontrar un estadístico suficiente.

A continuación de todos estos detectores, aplicables a la clasificación, se ha realizado una extensión concreta para clasificación exponiendo un test de múltiples hipótesis alternativas para el caso Bayesiano.

Para finalizar, en el estudio de la regresión hemos seguido el camino inverso, partiendo de la estimación Bayesiana genérica, apoyada en funciones de coste. El objetivo de minimizar el coste medio conocido un parámetro nos ha permitido recorrer los distintos estimadores sin más que cambiar la función de coste a minimizar. De este modo, una función cuadrática nos proporcionaba el estimador MMSE, y una función de coste uniforme con un cero en el origen nos permitía introducir el estimador MAP. A partir del mismo, suponiendo que la función de probabilidad a priori de la variable no es informativa, hemos llegado al estimador ML, aunque formalmente este último considera las variables como parámetros deterministas aunque desconocidos.



# Capítulo 4

## Detección/Estimación distribuida en WSNs con centro de fusión

### 4.1 Introducción

Después de describir las redes inalámbricas de sensores y las alternativas para tratar la información en función del carácter del parámetro que se pretende a la salida, (detectores o decisores, clasificadores, y estimadores), es el turno de enlazar ambas materias, en lo que será la antesala del escenario concreto en el que se desarrolla el proyecto. De forma breve, podemos anticipar que los contenidos se dividen como sigue:

En la Sección 4.2 se realiza una breve exposición sobre el aprendizaje e inferencia distribuida. Dentro de la misma, se parte del modelo general para expresar las diferencias entre los métodos paramétricos y no paramétricos, y se culmina todo este bloque con un ejemplo concreto.

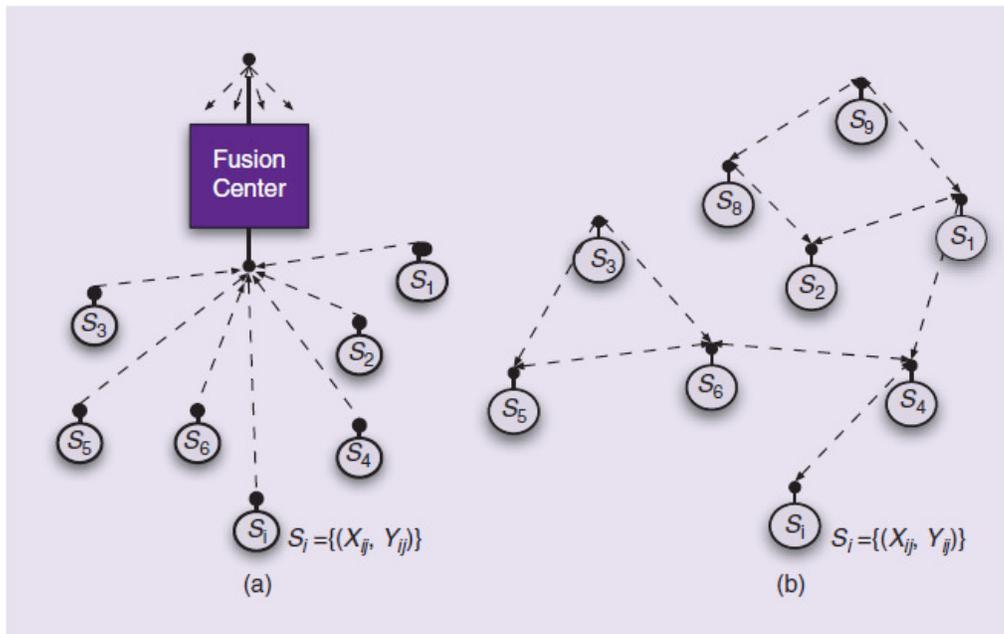
En las Secciones 4.3 y 4.4 se detallan dos formas alternativas de manejar la información en WSNs. Una primera en la que no existe un centro de fusión y la propia red “Ad-hoc” maneja la información nodo a nodo, y una segunda topología en la que la red inalámbrica cuenta con un centro de fusión. En este segundo caso, la sección se ha desarrollado a modo de introducción, puesto que coincide con el escenario práctico y se le dedican posteriormente capítulos completos.

En la Sección 4.5 se analizan los límites estadísticos a considerar en la inferencia distribuida y se introducen varios teoremas sobre dichas limitaciones. Estos teoremas, que trataremos posteriormente de comprobar de forma empírica, afirman que existen mecanismos suficientes para, agregando muy poca información de muchos sensores (hasta solo un bit por sensor), implementar con éxito métodos de detección binaria. De manera similar se trabajará en el análisis de teoremas enunciados para la estimación.

## 4.2 Aprendizaje e inferencia distribuido

### 4.2.1 Un modelo general

En este apartado vamos a desarrollar un modelo general para el aprendizaje distribuido que nos ayude a formular el problema. Dentro del mismo se pueden definir  $m$  sensores, cada uno de los cuales adquiere un conjunto de medidas (considerados datos de entrenamiento). Imaginemos dos configuraciones posibles, tal y como se muestra en la Figura 16.



**Figura 16.** (a) Red paralela con un centro de fusión. (b) Red “Ad-hoc” con procesamiento interno [21].

En la Figura 16, cada nodo en el grafo representa un sensor y su información observada localmente, mientras que cada línea de puntos representa un enlace inalámbrico. Destaca especialmente el centro de fusión en la Figura 16 (a), que puede modelarse como un nodo adicional, aunque podría disponer de mayor capacidad energética, de computación y de comunicación con el resto de sensores. Los modelos no consideran ninguna hipótesis sobre la topología de la red, pero el éxito del aprendizaje puede depender de las propiedades de la misma.

Profundizando sobre las **redes paralelas**, apreciamos que los sensores se comunican directamente y de modo exclusivo con el centro de fusión. Su objetivo fundamental será recolectar datos, pero la cuestión más relevante en este caso es: ¿cómo se produce ese aprendizaje cuando las restricciones que afectan a la comunicación limitan el acceso al dato?

Frente al caso anterior, en la Figura 16 (b) se encuentra una segunda arquitectura, que puede describirse como “**Ad-hoc**”. Su topología es dinámica y probablemente no está definida antes de su despliegue. En ocasiones estas redes pueden incluir un centro de fusión, pero en este caso los sensores son más autónomos y capaces de tomar decisiones independientemente del centro de fusión. Aquí la cuestión que surge es ¿cómo puede la comunicación local o entre sensores ser explotada para permitir un aprendizaje colaborativo eficiente?

Antes de resolver esta y otras cuestiones hay que resaltar un hecho importante. Inspirada por la disponibilidad creciente de conjuntos de datos, un área activa de estudio del aprendizaje máquina se concentra en el escalado de los algoritmos ya existentes para manejar las bases de datos masivas [43][44][44], la célebre metodología de “divide y vencerás”. Existen dos modos fundamentales de abordar este problema:

- La descomposición en pequeños grupos de sensores del conjunto total, paralelizando el aprendizaje mediante la asignación de distintos procesadores o agentes a cada subconjunto. En este aprendizaje paralelo las restricciones que puedan afectar a la comunicación aparecen como parámetros a ser ajustados.
- Asumiendo que los sensores entrenan de un modo local a los estimadores antes de transmitir una descripción completa de las reglas aprendidas al centro de fusión. Este modelo, que se conoce como “aprendizaje de la población”, deriva la responsabilidad final al centro de fusión, que debe observar la respuesta de la red para inferir una regla más precisa. Estos métodos, están relacionados con la aproximación paramétrica y se construye siguiendo el lema de obtener modelos “probablemente aproximadamente correctos” (PAC) [21].

A raíz de los métodos de aprendizaje de la población aparecen contribuciones al aprendizaje distribuido en WSNs con centro de fusión. En este caso, un conjunto de sensores aporta una predicción a un agente central (centro de fusión), que recibe estos pronósticos y genera su propia predicción ponderando las predicciones de los expertos (sensores anteriores). Una vez aprendida la que denominamos “verdad”, el agente pierde toda su información y en sucesivos intentos actualiza los pesos de la ponderación teniendo en cuenta el rendimiento ofrecido por cada experto [21]. El número de iteraciones se acota comparando el rendimiento que ofrece la nueva ponderación obtenida con la inicialmente considerada como verdad. Este tipo de mecanismos se utilizan por ejemplo en problemas de agregación.

## 4.2.2 Modelos paramétricos y no paramétricos

La unión de las capacidades de las WSNs y de la fortaleza del tratamiento estadístico da como resultado la renovación del interés en la inferencia descentralizada. Esto es, conseguir que cada sensor sea capaz de generar su propia regla o su propia manera

de tratar la información en función de lo que observa, de modo que no sea necesario reunir toda la información de la red antes de generar una decisión o clasificación.

Este hecho nos permitiría trabajar directamente en el propio entorno en el que se recoge la información. Habitualmente los estudios sobre esta materia se han centrado en dos grandes bloques en función de la forma de manejar la información. Si hay un centro de fusión, se enfocan en:

- Determinar cómo la capacidad del canal sensor-centro de fusión limita la calidad de los estimadores [45].
- Determinar la sensibilidad al retraso, las reglas de decisión del sensor y las estrategias de fusión en canales bajo limitaciones de ancho de banda y no fiables [46].

Si no lo hay, los estudios se centran en otros aspectos tales como:

- Caracterizar el rendimiento de redes grandes en función de las restricciones que implica la centralización [47].
- Desarrollar algoritmos de paso de mensajes a través de los cuales se puedan computar estimadores óptimos globales con comunicaciones locales entre sensores [48].

Casi todos los estudios realizados hasta la fecha se basan en modelos paramétricos, como son las SVM (“Support Vector Machine”) [49], [50], [51] o las redes neuronales [52], [53], donde los citados parámetros pueden ser considerados propios de un modelo que se presupone (es decir, se conoce la forma de la función, pero no por completo) y necesitan una estimación previa a través del entrenamiento. Nótese que la catalogación como método paramétrico o no de las SVM depende de si el autor considera los parámetros que se obtienen del entrenamiento como parámetros del modelo o si los considera datos de entrenamiento que vuelven a utilizarse en la predicción, en cuyo caso podría identificarse como un modelo no paramétrico. Para aclarar la diferencia entre métodos paramétricos y no paramétricos, veamos la siguiente comparativa.

En primer lugar se detallan los modelos **paramétricos**, para los que vamos a asumir como disponibles:

**$x$** : Variable independiente.

**$y$** : Observaciones.

**$w$** : Ruido

**$\theta$** : Parámetros.

Que se relacionan entre sí de modo que  $y = f(x, \theta) + w$ . Si los tratamos de forma vectorial,  **$y$**  se puede considerar como una función de  **$x$**  y  **$\theta$**  cuya forma es conocida, por

ejemplo polinómica. Donde los valores que toman los parámetros  $\theta$ , que en este caso coincidirían con los coeficientes del polinomio, son desconocidos y necesitan ser estimados a partir de un conjunto de datos de entrenamiento formado por las observaciones ( $\mathbf{y}$ ) y los valores de la variable independiente ( $\mathbf{x}$ ):

$$(\mathbf{x}, \mathbf{y}) \rightarrow \hat{\theta} \quad (4.1)$$

Si suponemos una regresión mediante mínimos cuadrados, entonces la estima de los parámetros para el modelo polinómico vendría determinada por:

$$\hat{\theta} = \arg \min_{\theta} \left\{ \sum_{i=1}^N \left( y_i - \sum_{k=0}^M \hat{\theta}_k x_i^k \right)^2 \right\} \quad (4.2)$$

Una vez hallados los parámetros  $\hat{\theta}$ , los datos de entrenamiento son innecesarios, y para cualquier valor de la variable independiente  $\tilde{\mathbf{x}}$ , podemos obtener la estimación sin más que aplicar el modelo con los parámetros obtenidos tras el entrenamiento:

$$\hat{y} = f(\tilde{\mathbf{x}}; \hat{\theta}) = \sum_{k=0}^M \hat{\theta}_k \tilde{x}^k \quad (4.3)$$

Donde  $\tilde{\mathbf{x}}$  son los puntos de test y  $\hat{\theta} = [\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_M]^T$ .

En contraposición con los modelos paramétricos, los **modelos no paramétricos** no realizan ningún tipo de suposición previa sobre la forma de la función, y se trata de estimarla a partir de la aplicación de propiedades generales de la función de densidad de probabilidad y de los propios datos. Ejemplos típicos de modelos no paramétricos son las redes Bayesianas [54] y los procesos Gaussianos [55]. Sus principales inconvenientes son la necesidad de una elevada memoria y una alta carga computacional en la aplicación, tanto durante la fase de entrenamiento o aprendizaje como durante la de test o aplicación.

Si asumimos que:

$\mathbf{x}$ : Variable independiente.

$\mathbf{y}$ : Observaciones.

$\theta$ : Hiperparámetros.

La estimación de los hiperparámetros se realiza para cada muestra a partir de las observaciones y de la variable independiente tal y como se expresa en (4.1). Para ello, los métodos no paramétricos se apoyan habitualmente en una función discriminante o “Kernel”  $K(\mathbf{x}, \tilde{\mathbf{x}})$ , que actúa como una medida de semejanza entre dos puntos cualesquiera.

Típicamente,  $\mathbf{x}$  será un punto del conjunto de entrenamiento mientras que  $\tilde{\mathbf{x}}$ , puede ser un punto de entrenamiento o de test.

Como hemos mencionado previamente, en un modelo no paramétrico no se asume una dependencia funcional concreta, sino que se trata de estimar esta. Por ejemplo, para un Proceso Gaussiano se modelan los datos como un proceso estocástico con una FDP Gaussiana para cualquier número finito de muestras como sigue:

$$\mathbf{y} \sim \text{GP}(\mu(\mathbf{x}; \Phi), K(\mathbf{x}, \tilde{\mathbf{x}}; \Psi)) \quad (4.4)$$

Donde  $\mu$  y  $K$  son las funciones de media y auto-covarianza del proceso y en este caso los hiperparámetros  $\theta$ , están formados tal que  $\theta = [\Phi \ \Psi]$ .

Si una vez entrenado el modelo (esto es, halladas las estimas de los hiperparámetros  $\hat{\Phi}$  y  $\hat{\Psi}$ ) se desea encontrar el valor de  $\hat{\mathbf{y}}$  en un conjunto nuevo de puntos,  $\tilde{\mathbf{x}}$ , esta estima vendría dada por:

$$\hat{\mathbf{y}} = \mu(\tilde{\mathbf{x}}; \hat{\Phi}) + K(\tilde{\mathbf{x}}, \mathbf{x}; \hat{\Psi})K(\mathbf{x}, \mathbf{x}; \hat{\Psi})^{-1} \mathbf{y} \quad (4.5)$$

Como puede comprobarse, estamos ante una función que en todo momento va a depender de los datos de entrenamiento, tanto para la creación del modelo como para la evaluación del mismo. Esto es, los datos de entrenamiento son necesarios para realizar la predicción, al contrario que ocurría en los modelos no paramétricos. En cada muestra es ineludible evaluar el producto de las dos funciones de Kernel, cuyas dimensiones dan origen a matrices de tamaño no despreciable y que acarrearán un incremento en la mencionada carga computacional frente a la sencillez del estimador obtenido mediante mínimos cuadrados.

Desde una perspectiva teórica, los modelos paramétricos permiten el estudio exhaustivo de muchas cuestiones fundamentales en la inferencia bajo ciertas condiciones. Por el contrario, la escasez de información acerca de la relación entre  $x$  e  $y$  en algunos entornos donde se despliegan las WSN hace que resulte aconsejable el uso de modelos no paramétricos. En realidad, si no se conocen en profundidad los datos, los modelos no paramétricos son más robustos y presentan mayor capacidad de adaptación a cualquier entorno [21]. Sin embargo, si la forma funcional asumida es la correcta, un método paramétrico siempre proporcionará mejor resultado que uno no paramétrico.

#### 4.2.2.1 El núcleo o Kernel

Tanto las máquinas de vectores soporte (SVM) como los Procesos Gaussianos (GP), requieren del uso de funciones de similitud que denominamos función de núcleo o “Kernel”. Su papel principal es analizar los patrones de entrenamiento disponibles y

encontrar relaciones entre los datos, permitiendo a las herramientas anteriormente citadas trabajar en un espacio distinto al de las observaciones y facilitando así la medida de la similitud entre las entradas. Existen infinitud de tipos de “Kernel” según la función que utilizan para medir esa similitud. En el presente trabajo únicamente se consideran los siguientes ejemplos:

1) **“Naive”**:

Este Kernel, necesita una distancia  $r_n$  de referencia y la norma entre los dos vectores de entrada  $\mathbf{x}$  y  $\tilde{\mathbf{x}}$ :

$$K(\mathbf{x}, \tilde{\mathbf{x}}) = 1_{\{\|\mathbf{x}-\tilde{\mathbf{x}}\| \leq r_n\}}(\mathbf{x}, \tilde{\mathbf{x}}) \quad (4.6)$$

De forma equivalente, si la norma es menor que la distancia de referencia, entonces  $K(\mathbf{x}, \tilde{\mathbf{x}}) = 1$ . Si no,  $K(\mathbf{x}, \tilde{\mathbf{x}}) = 0$ . Esto es, se fija un radio de influencia en torno a  $\tilde{\mathbf{x}}$  y se establecen los datos disponibles como dentro o fuera de ese radio.

$$K(\mathbf{x}, \tilde{\mathbf{x}}) = \begin{cases} 1, & \|\mathbf{x} - \tilde{\mathbf{x}}\| \leq r_n \\ 0, & \|\mathbf{x} - \tilde{\mathbf{x}}\| > r_n \end{cases} \quad (4.7)$$

2) **Lineal**:

En este caso, la medida de similitud vendrá proporcionada por el producto escalar de la información disponible y la nueva información incógnita.

$$K(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbf{x}^T \tilde{\mathbf{x}} \quad (4.8)$$

Si los datos están normalizados (Esto es,  $\|\mathbf{x}\| = \|\tilde{\mathbf{x}}\| = 1$ ), entonces podremos garantizar que cuanto mayor es  $K(\mathbf{x}, \tilde{\mathbf{x}})$ , más parecido es  $\tilde{\mathbf{x}}$  a  $\mathbf{x}$ , midiéndose dicho parecido en este caso en términos de orientación espacial seguida por los vectores  $\tilde{\mathbf{x}}$  y  $\mathbf{x}$ .

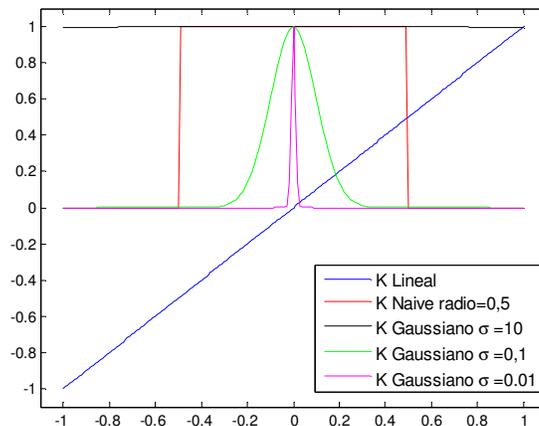
3) **Gaussiano**:

Esta función también depende de la norma y de un parámetro  $\sigma$  que modifica la exigencia del núcleo, es decir, establece el orden de magnitud a partir del cual la norma aporta valores despreciables a la salida del núcleo. En este caso, el “Kernel” viene dado por la siguiente fórmula:

$$K(\mathbf{x}, \tilde{\mathbf{x}}) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2\right) \quad (4.9)$$

El resultado no es tan abrupto como el proporcionado por el núcleo “Naive”, donde se fijaba el resultado de la función a 0 o 1. En este caso, la función siempre proporciona un valor mayor que cero que decrece exponencialmente conforme disminuye la similitud entre los datos  $\mathbf{x}$  y  $\tilde{\mathbf{x}}$ , medida en términos de su distancia euclídea.

Para completar la comparativa entre las tres funciones, la Figura 17, representa la forma de las mismas en una sola dimensión. Como puede apreciarse las tres funciones tienen características distintas y aportan distintos valores según la similitud entre  $\mathbf{x}$  y  $\tilde{\mathbf{x}}$ . Además, si nos centramos en el Kernel Gaussiano, a medida que se aumenta  $\sigma$  la función se ensancha, proporcionando valores no nulos de salida para un margen mayor de la norma, es decir, suavizando las restricciones de similitud.



**Figura 17.** Comparativa de las funciones de Kernel: Lineal, “Naive” y Gaussiana para una dimensión.

### 4.2.3 Un ejemplo concreto

Para ilustrar como de relevante es la inferencia descentralizada y discutir los retos, es importante apoyarse en un ejemplo. Supongamos que el espacio  $\mathbf{X}$  modela el conjunto de posibles medidas observables por los sensores en una red inalámbrica (por ejemplo, la posición sobre el plano y el tiempo de un elemento de modo que  $x \in \mathbf{X} = \mathbb{R}^2 \times \mathbb{R}^+$ ) e  $y$  representa el espacio de las medidas (por ejemplo,  $y \in \mathbf{Y} = \mathbb{R}^+$  para medidas de temperatura absoluta en grados Kelvin). Supongamos también que la aplicación requiere la temperatura en un punto concreto del espacio-tiempo, independientemente de la existencia o no de un centro de fusión. Para darle carácter desconocido a priori a las coordenadas y a la temperatura usaremos un modelo de variables aleatorias,  $(\mathbf{X}, \mathbf{Y})$ , definido por una distribución conjunta  $p_{x,y}$  que podríamos obtener de la correlación espacio-temporal del campo de temperaturas. Si dicha distribución puede ser asumida a priori como conocida, entonces el estimador podría obtenerse utilizando las herramientas habituales para modelos paramétricos. En caso contrario, recurriríamos a un modelo no paramétrico.

Hasta ahora, tal y como ocurre en el aprendizaje centralizado, se ha dejado a un lado el proceso de adquisición de datos y no se tienen en cuenta las restricciones en la comunicación, que pueden limitar el acceso del que aprende a los datos. Este hecho es una herencia de los métodos clásicos, basados en la consideración hipotética de que todos los datos de entrenamiento están disponibles todo el tiempo y en cualquier punto donde sean necesarios. Sin embargo, en las WSNs la energía y el ancho de banda requeridos para evaluar constantemente las medidas de los sensores pueden llegar a ser prohibitivamente elevados, de modo que utilizar estrategias de aprendizaje centralizado limitará la vida de las baterías y en última instancia el potencial de las WSN. Entonces, surgen las siguientes preguntas:

- ¿Hay métodos más eficientes en términos de la comunicación para el aprendizaje distribuido?
- ¿Podemos diseñar algoritmos que respeten unas determinadas restricciones de energía y ancho de banda?

Antes de proporcionar una respuesta en las próximas secciones, cabe contemplar la simplicidad del ejemplo expuesto, que no debería enmascarar su generalidad. Por un lado,  $X$  representará, más que una simple posición, un conjunto de variables de entrada independientes de diferente índole. Por otro lado,  $Y$  puede modelar cualquier variable de interés, desde la fuerza medida por una galga a la intensidad acústica detectada por un sensor. En estas circunstancias, cada sensor buscará una regla de decisión que permita predecir una medida de salida usando los datos de entrada, y el centro de fusión (si existe) tendrá la suya propia, considerando la información observada a través de la red de sensores con las limitaciones que esta imponga.

### **4.3 Aprendizaje distribuido en WSNs ad-hoc con procesado en la propia red**

A lo largo de esta sección se describe muy brevemente como se puede llevar a cabo un aprendizaje global eficiente y efectivo apoyándose exclusivamente en comunicaciones locales, y por tanto mediante el intercambio de información entre sensores vecinos. La clave es intuitiva: la comunicación local es más eficiente, puesto que necesita menos energía y ancho de banda que una comunicación global [21].

Los métodos que facilitan este aprendizaje se conocen como algoritmos de paso de mensaje y han sido ampliamente estudiados y utilizados en infinidad de campos. Su uso es imprescindible en sistemas distribuidos sin centro de fusión, dado que no existe un nodo central permita intercambiar información entre los sensores. Para simplificar la explicación, consideraremos un problema de estimación de mínimos cuadrados, de modo que el problema de optimización se resuelve como:

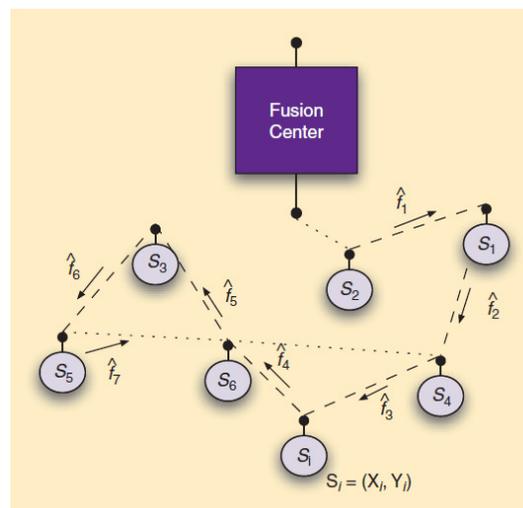
$$f = \arg \min_{f \in H_k} \left[ \frac{1}{m} \sum_{i=1}^m (f(\mathbf{x}_i) - \mathbf{y}_i)^2 + \lambda \|f\|_{H_k}^2 \right] \quad (4.10)$$

Donde  $H_k$  es un espacio de Hilbert [56] y donde el término  $\lambda \|f\|_{H_k}^2$  es un término de regularización para evitar sobreajustes.

En una WSN la solución directa de este problema resulta inviable, ya que el conjunto de datos está distribuido por la red y no en un sólo centro. Sin embargo, a continuación se citan, a modo de ejemplo, dos algoritmos que permitirán resolverlo de manera aproximada en un entorno distribuido [21]. El primero se basa en el uso del **método del subgradiente incremental** y requiere, de un modo simplificado, de los siguientes pasos:

- Inicializar de los sensores.
- Establecer una ruta a través de la red que recorra todos los sensores.
- Ejecutar el algoritmo de subgradiente incremental en el primer, dando origen a un estimador local.
- Ejecutar sucesivamente en cada nodo el algoritmo de subgradiente incremental con los datos disponibles y el estimador recibido del nodo anterior.

El proceso de recorrer la ruta se repite y aunque los datos disponibles no cambian, sí lo hace el estimador recibido del sensor anterior, que va adaptándose a lo que opina toda la red aprovechando el hecho de que las medidas de los sensores están disociadas. Gráficamente puede apreciarse el mecanismo en la Figura 18, pudiéndose encontrar más información en [57], [58], [59], [60].

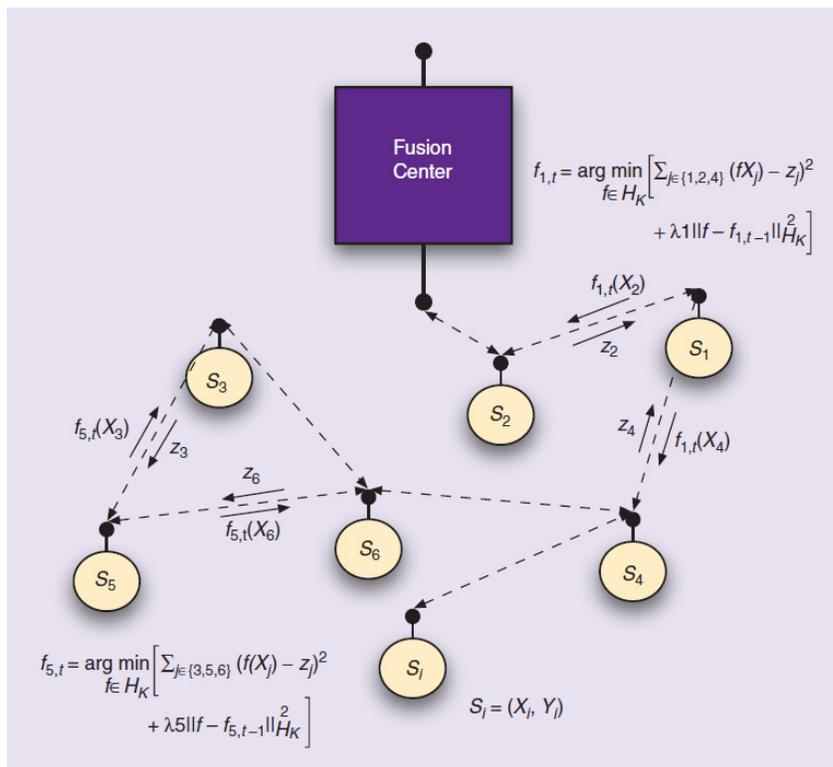


**Figura 18.** Entrenamiento distribuido basado en subgradiente incremental [21].

El segundo de los mecanismos de entrenamiento considerados se apoya en la compartición local e iterativa de datos, no de funciones completas. Si retomamos el escenario del campo de temperatura, los pasos que se siguen son:

- Inicialización de los sensores.
- Petición de información de los sensores vecinos por parte de cada nodo.
- Generación del estimador global con los datos obtenidos en cada sensor.
- Devolución de la estimación de la temperatura que le corresponde a cada vecino como respuesta a la temperatura observada que éste le había enviado.
- Actualización: Repetición de las tres acciones previas.

El estimador no se genera siempre en el mismo nodo, ni en el mismo instante, sino de modo alternativo tanto espacial como temporalmente. Todo el proceso se muestra gráficamente en la Figura 19 y puede encontrarse más información en [61], [62].



**Figura 19.** Entrenamiento distribuido basado en proyecciones alternadas [21].

Cualquiera de los dos métodos representa una alternativa al aprendizaje con centro de fusión, que es el objetivo fundamental de este proyecto y del que hablaremos en profundidad en la siguiente sección.

## 4.4 Aprendizaje distribuido en WSNs con un centro de fusión

En esta sección vamos a concentrarnos en la red paralela de la Figura 16(a). Siguiendo con el ejemplo anterior, donde los datos capturados por cada sensor representaban una temperatura para una posición en el plano y un instante de tiempo dado. Se considera en esta ocasión que toda la información es enviada al centro de fusión para construir un estimador global del campo de temperaturas, que varía continuamente. Estudiando a simple vista el modelo, la configuración va a obligar a los sensores a enviar todos sus datos al centro de fusión, lo que acarreará elevados costes en términos de energía y ancho de banda.

Para paliar esos costes es posible designar pequeños subconjuntos de nodos (“clusters”) para enviar los datos. Si el número de nodos es pequeño y tanto los sensores como la información se escogen de un modo adecuado, entonces esta estrategia podría ser efectiva a la hora de optimizar el rendimiento del aprendizaje, reduciendo al tiempo el mencionado coste en la comunicación.

Por ejemplo, podríamos dividir los sensores en diferentes subgrupos y asignar a cada uno un cabeza de “cluster” o elemento principal. Este último debe recoger la información de los sensores de su conjunto y dado que la distancia entre los sensores del “clúster” y la cabeza es en teoría pequeña, se minimizan los posibles inconvenientes y se mejora la eficiencia en la comunicación. El elemento principal debe filtrar toda la información recibida y enviar al centro de fusión un resumen que podría incluir una regla aprendida de modo local o un subconjunto de datos relevantes.

Esta configuración en “clusters” se ha considerado frecuentemente en el trabajo paramétrico para detección y estimación [43]. Otros autores consideran la configuración en “clusters” para dirigir la localización de los sensores en la red [63]. En dicho estudio el espacio de observación,  $\mathbf{X} = \mathbb{R}^2$ , modela puntos en un terreno plano y la salida corresponde a 0 o 1 en función de si el punto de la entrada pertenece a una región convexa del terreno o no. Los datos de entrenamiento se adquieren del subconjunto de sensores (estaciones base), cuyas posiciones se estiman utilizando varias medidas físicas.

Alternativamente podríamos limitar la capacidad de la comunicación de los sensores, en lugar de agrupar los datos entorno a una cabecera de “cluster”, para disminuir los recursos necesarios. De este modo, todos los sensores transmitirían su información directamente al centro de fusión, con la principal restricción de que esa información sea la mínima posible, llegando a considerarse un solo bit por sensor. Esta última medida es la que consideraremos posteriormente en el Capítulo 5, donde se podrá encontrar información más completa y detallada sobre la misma.

## 4.5 Límites estadísticos del aprendizaje distribuido

Después de concretar los escenarios, vamos a estudiar sus posibles limitaciones estadísticas que aparecen de forma concreta en uno de ellos. El escenario clave a partir de ahora es el que considera un centro de fusión en el que trataremos de encontrar formas de inferencia considerando restricciones en las capacidades de los canales de comunicación. Todo este análisis de las limitaciones estadísticas tiene como origen el Teorema de Stone a partir del que se enuncian otros teoremas concretos aplicados al mencionado escenario clave. Pero para poder continuar debemos introducir el concepto de la **consistencia estadística**.

### **DEFINICIÓN 1 (CONSISTENCIA ESTADÍSTICA):**

*Se dice que un estimador es consistente si el valor proporcionado por el mismo tiende al valor real del parámetro estimado conforme aumenta el número de datos disponibles.*

Ligado a este término, el teorema de Stone es un resultado seminal en el ámbito del reconocimiento de patrones que establece la existencia de reglas de aprendizaje consistentes universalmente [64]. Esto es, se confirma que hay reglas de aprendizaje que nos aseguran que vamos a conseguir un modelo cada vez más fiel al real, cuanto mayor sea el número de ejemplos de entrenamiento de los que dispongamos. El citado teorema de Stone se desarrolla a continuación.

### **TEOREMA 1 (STONE):**

*Si suponemos:*

1) *La información de los sensores  $\mathbf{s}_i$  es independiente e idénticamente distribuida, con  $(\mathbf{x}_i, \mathbf{y}_i) \sim P_{XY} \forall i \in \{1, \dots, m\}$ .*

2) *Definimos  $L_n \triangleq \mathbf{E}\{l(g_n(\mathbf{x}), \mathbf{y}) | \mathbf{s}_n\}$  donde  $g_n(\mathbf{x})$  es un estimador de  $\mathbf{E}\{Y|X\}$  construido como sigue:*

$$g(\mathbf{x}) = g(\mathbf{x}, \mathbf{s}_n) = \begin{cases} \frac{\sum_{i=1}^n K(\mathbf{x}, \mathbf{x}_i) \mathbf{y}_i}{\sum_{i=1}^n K(\mathbf{x}, \mathbf{x}_i)}, & \sum_{i=1}^n K(\mathbf{x}, \mathbf{x}_i) > 0 \\ 0, & \text{resto} \end{cases}$$

Donde  $K(\mathbf{x}, \tilde{\mathbf{x}})$  es la función de núcleo “Naive”:

$$K(\mathbf{x}, \tilde{\mathbf{x}}) = \begin{cases} 1, & \|\mathbf{x} - \tilde{\mathbf{x}}\| \leq r_n \\ 0, & \|\mathbf{x} - \tilde{\mathbf{x}}\| > r_n \end{cases}$$

Entonces, si  $r_n \rightarrow 0$  y  $nr_n^d \rightarrow \infty$ , siendo  $d$  la dimensión del espacio de entrada, la regla proporcionada por el “Kernel” es universalmente consistente bajo el criterio del error cuadrático. Esto es, siendo  $l(y, y') = |y - \tilde{y}|^2$ :

$$E\{L_n\} \rightarrow E\{l(E\{Y|X\}, Y)\}$$

Sea cual sea la distribución  $P_{XY}$  siempre que  $E\{Y^2\} < \infty$

Nótese que el Teorema de Stone básicamente nos indica que un estimador construido a partir de un “Kernel” extremadamente simple como el “Naive” puede llegar a proporcionarnos un error tan bajo como el estimador óptimo dado por  $E\{Y|X\}$ , cuando el número de sensores tiende a infinito más rápidamente de lo que el radio usado en la función núcleo tiende a cero.

Los trabajos clásicos para encontrar estas reglas se han ocupado de configuraciones centralizadas, asumiendo que toda la información de las bases de datos de entrenamiento está disponible para un procesador único. Sin embargo, a lo largo de los últimos años se han intentado caracterizar los límites del aprendizaje distribuido con un centro de fusión superponiendo varios modelos simples de comunicación al modelo clásico aprendizaje supervisado. Por ejemplo, en [65] se supone que cada sensor toma solo un ejemplo de entrenamiento  $\mathbf{s}_i = \{(\mathbf{x}_i, \mathbf{y}_i)\}$  y que la comunicación se modela como sigue:

- Cuando el centro de fusión desea tener una nueva observación en un punto  $\mathbf{x}$ , difunde dicha localización a toda la red en busca de información. En ese momento las restricciones en el ancho de banda y la energía limitan a cada sensor a responder como mucho con 1 bit.
- Cada sensor particular elige si responder o no a la petición de información del centro de fusión. Si decide responder, el sensor transmite 0 o 1 basado en su algoritmo de decisión local.
- Una vez observada la respuesta de la red por parte del centro de fusión, este combina toda la información para crear un estimador global de  $\mathbf{y}$ .

En cuanto a los enlaces, el centro de fusión tiene un canal de tipo “broadcast” hacia cada sensor para realizar las peticiones de información sobre  $\mathbf{x}$ , mientras que cada sensor tiene un enlace inalámbrico punto a punto de sentido único hacia el centro de fusión que permite enviar 1 bit.

El primer caso que vamos a analizar no considera la abstención como opción, de modo que la respuesta a la petición de información por parte del centro de fusión de cada sensor debe ser 0 o 1. (En otros casos podría considerarse, -1 o 1) ¿Pueden los sensores transmitir información suficiente al centro de fusión en este escenario para que exista la consistencia universal en el aprendizaje distribuido sin abstención? La respuesta se obtiene en los dos siguientes teoremas [21].

**TEOREMA 2 (DETECCIÓN SIN ABSTENCIÓN):**

*Si suponemos:*

1) *La información de los sensores  $\mathbf{s}_i$  es independiente e idénticamente distribuida, con  $(\mathbf{x}_i, \mathbf{y}_i) \sim P_{XY} \forall i \in \{1, \dots, m\}$ .*

2) *Cada sensor conoce  $m$ , el tamaño de la red de sensores.*

*Entonces, **existen** reglas de decisión para cada sensor y para el centro de fusión que garantizan la consistencia universal del aprendizaje distribuido sin abstención para el problema de detección o decisión binaria bajo el criterio de la minimización de la probabilidad de error.*

En otros términos, existe un número de sensores para la red que nos permitirá aportar al centro de fusión tanta información obtenida localmente como necesite para tomar una decisión global. Incluso transmitiendo un único bit por sensor, esa decisión será más certera conforme aumenta el tamaño de la red, de modo que, conforme  $m \rightarrow \infty$ , el detector global tenderá al óptimo independientemente de cuál sea  $P_{XY}$ .

Si seguimos inmersos en el escenario sin abstención, el siguiente paso una vez confirmada que la detección es posible, es cuestionarse que ocurre con la estimación.

**TEOREMA 3 (ESTIMACIÓN SIN ABSTENCIÓN):**

*Si suponemos:*

1) *La información de los sensores  $\mathbf{s}_i$  es independiente e idénticamente distribuida con  $(\mathbf{x}_i, \mathbf{y}_i) \sim P_{XY} \forall i \in \{1, \dots, m\}$ .*

2) *Cada sensor conoce  $m$ , el tamaño de la red de sensores.*

3) *La regla del centro de fusión satisface un conjunto de condiciones de regularidad [21].*

Entonces **no existe** una regla para el centro de fusión que garantice la consistencia universal del aprendizaje distribuido sin abstención para cualquier distribución de  $P_{XY}$  con  $E\{Y^2\} < \infty$  para el problema de la estimación bajo el criterio de minimización del error cuadrático medio.

En otros términos, no existe un número de sensores para la red que nos permita aportando al centro de fusión un bit por sensor generar un estimador global próximo al óptimo.

La no obligatoriedad de votar abre la puerta a un nuevo escenario donde el canal de subida tiene una capacidad ligeramente mayor a la indicada por el único bit transmitido, ya que se considera que el sensor aporta información al centro de fusión incluso cuando el sensor se abstiene de votar.

La cuestión de nuevo es: ¿pueden los sensores comunicar información suficiente al centro de fusión para hacer posible la consistencia universal del aprendizaje? El siguiente teorema contesta a la pregunta en escenario concreto de para aprendizaje distribuido con abstención [21], [65].

**TEOREMA 4 (DETECCIÓN Y ESTIMACIÓN CON ABSTENCIÓN):**

Supongamos que:

- 1) La información de los sensores  $\mathbf{s}_i$  es independiente e idénticamente distribuida (i.i.d) con  $(\mathbf{x}_i, \mathbf{y}_i) \sim P_{XY} \forall i \in \{1, \dots, m\}$ .
- 2) Cada sensor conoce  $m$ , el tamaño de la red de sensores.

Entonces, **existen** reglas de decisión para cada sensor y para el centro de fusión que garantizan la consistencia universal del aprendizaje distribuido con abstención para el problema de detección binaria bajo el criterio de minimización de la probabilidad de error y para cualquier distribución de  $P_{XY}$  con  $E\{Y^2\} < \infty$  para el problema de estimación bajo criterio de minimización del error cuadrático medio.

Ahora cada regla de decisión del sensor puede verse como la selección de uno de tres posibles estados: abstención, votar 0 o votar 1. Con esta observación el teorema puede interpretarse como que son suficientes  $\log_2(3)$  bits por sensor para que se produzca la consistencia universal en el aprendizaje bajo este modelo, algo que no ocurría para el escenario sin abstención para el problema de la estimación.

En general los Teoremas 2-4 establecen límites fundamentales para el aprendizaje distribuido en WSNs, estudiando el hecho de si se cumple o no el teorema de Stone para entornos distribuidos con un nodo central. Sin embargo, la aplicabilidad de estos resultados puede estar limitada por lo adecuado o no del modelo en la práctica. Aspectos claves que deben considerarse son [21]:

- Los datos de entrenamiento observados por una red de sensores pueden no ser independientes e idénticamente distribuidos (i.i.d.).
- En el problema de estimación, el dato puede estar corrompido por ruido correlado.
- La regla de decisión local, con la que los sensores adquieren el dato, puede ser distinta de la utilizada en el proceso que realiza el centro de fusión.
- Los sensores pueden estar desplegados de modo uniforme a lo largo del área de observación, mientras que el centro de fusión puede estar centrado en una zona particular, recibiendo niveles de potencia distintos con distintos niveles de fiabilidad.

## 4.6 Discusión

Este capítulo que se cierra con esta sección comenzaba con un ejemplo que aunaba las redes inalámbricas de sensores con el aprendizaje distribuido. Esto hace que los contenidos del mismo se conviertan en un nexo de los dos caminos recorridos hasta ahora y el origen de la única línea que se seguirá a partir de este momento.

Una vez expuesto un ejemplo del potencial y la necesidad del aprendizaje distribuido, este capítulo también ha servido para comprender la distinción entre los modelos paramétricos y no paramétricos. Esta comparativa nos ha llevado al primer contacto con una herramienta fundamental a partir de este instante, la SVM, e indirectamente con las funciones de Kernel.

Posteriormente se han comparado dos escenarios de aprendizaje distribuido alcanzando las siguientes conclusiones:

- En uno de ellos, concentrado en las redes “Ad-hoc” sin centro de fusión, la red gestiona en modo local la información de los sensores y los nodos tienen capacidad como para extender una regla de decisión de forma global.
- En el otro, se consideran las redes paralelas con centro de fusión, y va a recoger toda la atención a partir de ahora. En él, los nodos tendrán su propia regla de decisión local y se comunicarán paralelamente y de modo exclusivo con el centro de fusión para transmitirle su decisión. Será tarea de este último acumular toda esa información y ser capaz de realizar un estimador global.

El carácter inalámbrico de la red, limita los recursos de cada nodo, por eso nos concentraremos en comprobar si existen reglas de detección y estimación tratando de consumir lo menos posible. Como se detalla en los teoremas enunciados es posible la consistencia universal en problemas de detección y estimación distribuida con abstención y también en casos de detección distribuida sin abstención.

# Capítulo 5

## Descripción de los métodos usados en Detección/Estimación

### 5.1 Introducción

Para poner en práctica los conocimientos que hemos desarrollado sobre las WSNs y sobre los sistemas de aprendizaje distribuido, necesitamos describir un entorno concreto sobre el que comprobar la teoría. Así en la Sección 5.2 vamos a tratar de acotar el funcionamiento de los sensores en cuanto a la potencia de señal recibida, el ruido y a describir las trayectorias tipo que consideraremos en nuestros experimentos.

En la Sección 5.3 vamos a describir el sistema de detección desplegado en el escenario escogido para un blanco que atraviesa, con una determinada probabilidad, un área de dimensiones conocidas por el usuario, donde estarán distribuidos los sensores. Un factor entorno al que girará el estudio es el número de sensores desplegados, que afectará al comportamiento del modelo de aprendizaje. El trabajo que se desarrolla puede dividirse en dos bloques: uno donde se detallan las reglas a nivel local (Sección 5.3.1) y otro donde se profundiza en las tareas llevadas a cabo por el centro de fusión (Sección 5.3.2).

El centro de fusión va a realizar su trabajo apoyándose en las SVM, sobre las que profundiza en la Sección 5.3.2.1, donde también se hablará sobre las dos funciones de “Kernel” escogidas: Lineal y Gaussiana.

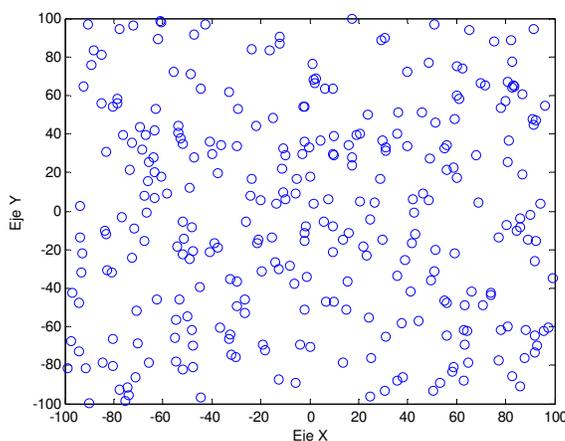
El capítulo se completa con los contenidos sobre estimación (Sección 5.4), englobados en tres subsecciones: las dos primeras, donde se analiza el algoritmo encargado de reconstruir la trayectoria que sigue el blanco en caso de que este aparezca en el área. Y una tercera donde se recoge la forma de estudiar la calidad del estimador.

## 5.2 El escenario

Al considerar la palabra aprendizaje y lo que representa, con frecuencia asumimos que todo lo que aprendemos es bueno y beneficioso, olvidando que el aprendizaje más básico reside en la imitación, independientemente del carácter de la conducta. Es por tanto esencial que ese modelo que vamos a imitar sea lo más fiel a la realidad posible, esto es, que las trayectorias que vamos a usar para entrenar a nuestra máquina, la forma en la que aparecen y la variedad de las mismas estén controladas. El hecho de restringir las trayectorias a su forma lineal y a un conjunto de tipos concretos no parece muy realista, pero se ha escogido para tratar de maximizar el uso de los sensores de la red. Una trayectoria que recorra el borde del área monitorizada no aporta información eficaz para el modelo de aprendizaje. Además de las trayectorias, como parte fundamental del escenario vamos a describir los sensores, la señal que reciben y cómo se modela.

### 5.2.1 Los sensores

El funcionamiento de la red se basa en que cada uno de los citados sensores, recibirá una señal fruto de la presencia del blanco y del ruido, a partir de la que tomará su propia decisión, que posteriormente compartirá con el centro de fusión. Con todos los datos que el centro reciba, y mediante un entrenamiento previo, veremos si el centro será capaz de recrear la trayectoria que está siguiendo el blanco. En la Figura 20 podemos ver cómo sería una distribución uniforme de 300 sensores en un área monitorizada cuya desviación máxima sobre el origen se estableciese en  $\pm 100\text{m}$  para cada coordenada, esto es, una superficie total igual a  $40000\text{ m}^2$ . Aunque en algunas aplicaciones podría tener sentido el uso de otras distribuciones, (por ejemplo, Gaussiana centrada en un punto del área monitorizada), la distribución uniforme es la única considerada en lo sucesivo para las posiciones de los sensores.



**Figura 20.** Ejemplo de distribución aleatoria uniforme de 300 sensores sobre un área sobre un área de  $200\text{m} \times 200\text{m}$ .

### 5.2.1.1 Potencia recibida y umbrales de clasificación

La potencia que recibe cada sensor va a tener dos fuentes. La primera corresponde al ruido en ausencia de señal, y tendrá una distribución Gaussiana con media cero y una varianza que se establece a partir de la relación señal a ruido (SNR) deseada. Y la segunda corresponde a la señal recibida del blanco y su potencia es inversamente proporcional a la distancia, combinándose con la primera cuando existe un blanco en el área.

Matemáticamente podríamos decir que existen dos hipótesis bajo las que se conforman las siguientes distribuciones:

**Hipótesis  $H_0$** , (ausencia de blanco):

$$y_o = w \quad (5.1)$$

Donde  $w$  es una variable aleatoria Gaussiana de media cero y varianza  $\sigma_w^2$

$$w \sim N(0, \sigma_w^2) \quad (5.2)$$

De modo que la señal recibida por el sensor en este caso sigue la misma distribución:

$$y_o \sim N(0, \sigma_w^2) \quad (5.3)$$

Si consideramos el término de potencia:

$$P_0 = E\{w^2\} = \sigma_w^2 \quad (5.4)$$

**Hipótesis  $H_1$** , (presencia de blanco):

$$y_1 = x + w \quad (5.5)$$

En este caso, la suma de las señales da como resultado una variable aleatoria Gaussiana, cuya media corresponde con la señal dependiente de la distancia al sensor (si la asumimos con determinista aunque desconocida) y cuya varianza continúa siendo la del ruido:

$$y_1 \sim N(x, \sigma_w^2) \quad (5.6)$$

Si consideramos la potencia, asumiendo  $x$  y  $w$  independientes:

$$P_1 = E\{(x + w)^2\} = \sigma_x^2 + \sigma_w^2 \quad (5.7)$$

Donde  $x$  se obtiene como:

$$x = \sqrt{P_x(d)} \quad (5.8)$$

Y a su vez  $P_x(d)$ , es la potencia recibida a una distancia  $d$  y se obtiene a partir de la fórmula para la transmisión de potencia en espacio libre de Friis [66]:

$$P_x(d) = \frac{P_t G_t G_r \lambda^\gamma}{(4\pi d)^\gamma} = \frac{P_t G_t G_r \lambda^\gamma}{(4\pi d_0)^\gamma} \left(\frac{d_0}{d}\right)^\gamma \quad (5.9)$$

De modo que resulta:

$$P_x(d) = P_x(d_0) \left(\frac{d}{d_0}\right)^{-\gamma} \quad (5.10)$$

Donde consideraremos  $2 \leq \gamma \leq 4$ , para la constante de propagación.

Adicionalmente, se lleva a cabo una normalización del área, esto es, una normalización de cada coordenada en base a su máxima desviación. El objetivo es que el rango de esta variable se encuentre entre los recomendados para el trabajo con la SVM (-1,1). Esta normalización provoca que el factor de potencia recibida para un metro se convierta en el factor de potencia recibido para un metro normalizado por la máxima dimensión, y la ecuación se ve modificada del siguiente modo:

$$P_x(d) = P_x(d_0) \left(\frac{d_{max} d}{d_0 d_{max}}\right)^{-\gamma} \quad (5.11)$$

Si consideramos que  $\tilde{d}$  es  $d/d_{max}$ :

$$P_x(\tilde{d}) = \frac{P_x(1)}{d_{max}^\gamma} \tilde{d}^{-\gamma} \quad (5.12)$$

Esto nos permite explorar, en el caso de ser necesario, el comportamiento del sistema con distintos valores de la potencia recibida a un metro  $P_x(1)$  y de la constante de pérdidas,  $\gamma$ , que inicialmente se establecen en 1mW y en 2 respectivamente.

De este modo quedan identificadas las dos fuentes y su formulación. Antes de continuar con el análisis de los umbrales de detección, es necesario fijar la varianza del ruido y mostrar que su elección es la adecuada para conservar la coherencia en el escenario que estamos desplegando.

### 5.2.1.2 La varianza del ruido

La varianza del ruido está ligada con la relación señal a ruido (SNR) que establecemos para cada sensor:  $SNR = P_{señal}/P_{ruido}$ . Si trabajamos en decibelios, como es habitual tratándose de potencias:

$$SNR(dB) = P_{señal}(dBw) - P_{ruido}(dBw) \quad (5.13)$$

Si consideramos a continuación que la potencia de la señal es exclusivamente debida a la presencia del blanco:

$$P_{señal}(dBw) = 10 \log_{10}(P_x(1)) - 10\gamma \log_{10}(d_{max}) - 10\gamma \log_{10}(\tilde{d}) \quad (5.14)$$

Por otra parte, de la distribución del ruido podemos obtener:

$$P_{ruido}(dBw) = 10 \log_{10}(\sigma_w^2) \quad (5.15)$$

Si consideramos, como se citó previamente, que en nuestro escenario son fijos los siguientes valores,  $d_{max} = 100m$ ,  $P_x(1) = 10^{-3}W$  y  $\gamma = 2$ , entonces la potencia de señal únicamente depende de la distancia normalizada  $\tilde{d}$ :

$$P_{señal}(dBw) = -70 - 20 \log(\tilde{d}) \quad (5.16)$$

Con las dos formulas previas, podemos relacionar la varianza del ruido con la SNR:

$$SNR(dB) = -70 - 20 \log(\tilde{d}) - 10 \log(\sigma_w^2) \quad (5.17)$$

Despejando la varianza:

$$\sigma_w^2 = 10^{[SNR(dB)+70+20 \log(\tilde{d})]/10} \quad (5.18)$$

Si despejamos para el caso en el que SNR=30 dB para un metro por ejemplo:

$$10 \log(\sigma_w^2) = -30 - 70 + 40 = -60 \quad (5.19)$$

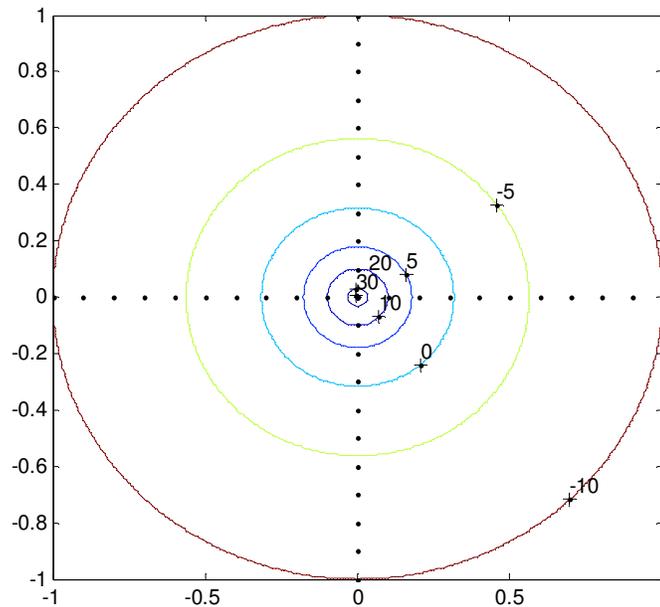
La varianza del ruido resulta ser:

$$\sigma_w^2 = 10^{-6} \quad (5.20)$$

Con esos niveles fijados podemos calcular la SNR para 20m:

$$SNR(dB) = -70 - 20 \log(0.2) + 60 \approx -70 + 13.97 + 60 \approx 3.97 \quad (5.21)$$

Para completar el estudio, en la Figura 21, se muestran las curvas de nivel para la SNR que supondría en cada punto del espacio la presencia de un blanco situado en el centro del área. A modo de referencia, la SNR es menor que 5 dB si el radio es mayor de 0,18 que una vez deshecha la normalización supone un alcance de 18 m.

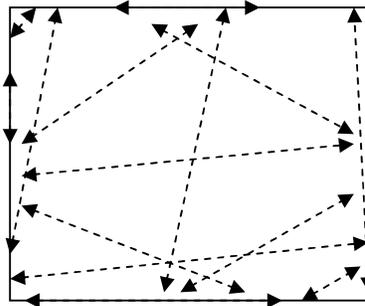


**Figura 21.** Curvas de nivel de SNR (dB) obtenidas supuesto un blanco en el centro del área normalizada. Se representan desde el centro hacia fuera 30, 20, 10, 5, 0, -5 y -10 decibelios.

Como ha podido comprobarse, el valor de la varianza está directamente ligado al de la relación señal a ruido, ambos son parámetros del sistema que definen de manera significativa el entorno en el que se va a desarrollar el proyecto. Modificarlos afecta a los umbrales que estableceremos para la detección, y por tanto al número de sensores del total que van a detectar al blanco, complicando tanto por defecto como por exceso la estimación de la trayectoria.

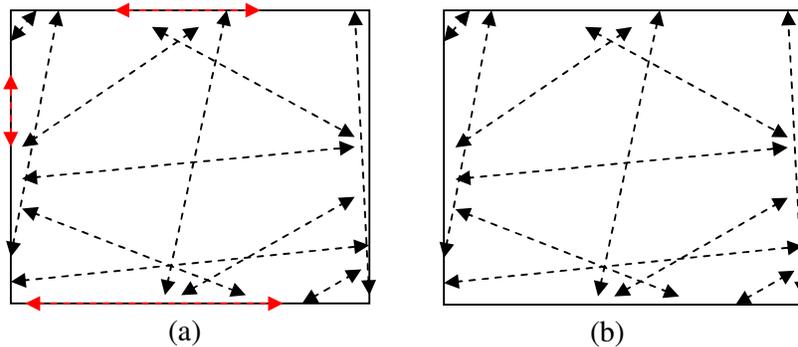
## 5.2.2 Las trayectorias

Con el fin de tener las trayectorias bajo control, se han ido barajando tres formas de generar trayectorias, que van desde la más general a la más concreta, sin desviarse en ningún caso del carácter lineal de las mismas ni del número de saltos, definido en 19 para dar lugar a 20 posiciones. La primera de las opciones implementadas considera independientes las probabilidades de iniciar y finalizar la trayectoria en lo que se refiere a cada una de las cuatro fronteras, permitiendo incluso trayectorias sobre los bordes. La Figura 22, muestra la variedad de trayectorias que podrían generarse y permite comprender la dificultad que entrañaría entrenar la SVM con dicha diversidad.



**Figura 22.** Trayectorias generadas de modo aleatorio sin restricciones.

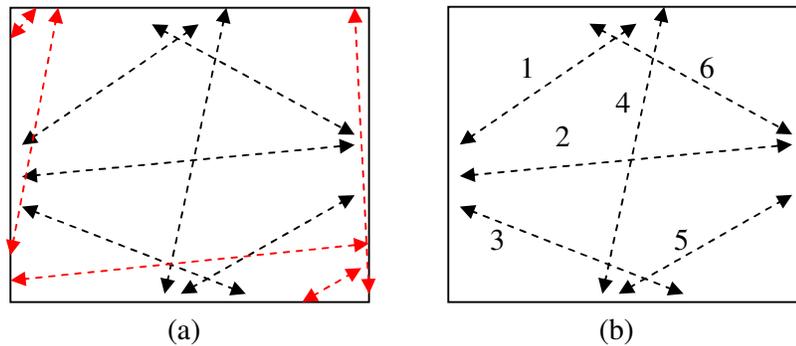
La segunda restricción considera que es imposible que el blanco entre y abandone el área por el mismo lateral del área monitorizado, eliminando así un cierto número de trayectorias que se incluían anteriormente.



**Figura 23.** Trayectorias contempladas y trayectorias que se excluyen (en rojo) por comenzar y terminar en la misma frontera (a). Trayectorias contempladas tras la restricción (b).

En la tercera versión se establecen todas las posibles trayectorias y se fuerza la presencia de cada tipo en la misma proporción en el conjunto de datos de entrenamiento. Esto hace que el sistema esté igualmente preparado para detectar cada una, evitando el sobreajuste en beneficio de una determinada trayectoria. Además, se restringen los inicios y finales de trayectorias a coordenadas cercanas al centro de cada eje, evitando que se generen para el entrenamiento trayectorias de saltos muy pequeños y muy corto recorrido

por situarse el inicio y el fin cercanos a una de las esquinas del área monitorizada. De este modo, definimos 6 trayectorias tipo con un objetivo claro: tratar de alcanzar el máximo el uso de toda la red.



**Figura 24.** Trayectorias contempladas tras la primera restricción y trayectorias que se excluyen (en rojo) por no cruzar el área de modo significativo (a). 6 Trayectorias supervivientes y que se contemplan en el escenario desplegado (b).

## 5.3 Detección

A lo largo de esta sección vamos a analizar el trabajo necesario tanto en los sensores como en el centro de fusión. El objetivo es obtener reglas de detección locales en función de los niveles de potencia recibidos por el sensor y una regla de detección global apoyada en la información local recibida por el centro.

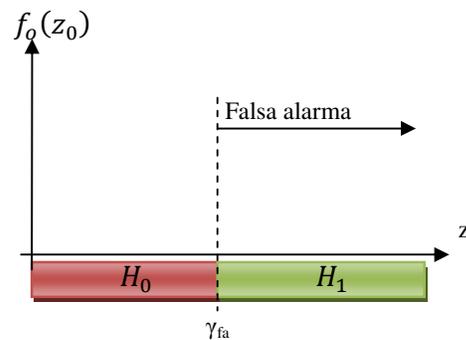
### 5.3.1 Trabajo con los sensores

El primer paso en el proceso de detección es hallar los niveles de potencia que marcan la detección para cada uno de los sensores. A estos niveles los llamaremos a partir de ahora umbrales y van a estar asociados a las probabilidades de falsa alarma y de no detección.

#### 5.3.1.1 Los umbrales

Alcanzamos un punto clave en el desarrollo práctico. Es aquí donde comienza la diferenciación entre los dos escenarios con los que vamos a trabajar, uno de ellos sin región de incertidumbre, y el segundo con dos niveles de potencia entre los cuales el sensor se abstiene, al considerar que su información puede ser confusa.

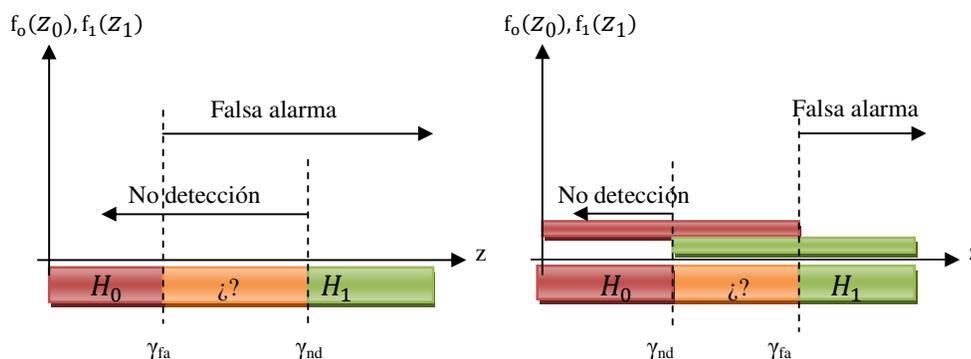
La Figura 25 trata de representar de un modo gráfico cómo vamos a obtener el único umbral necesario en base a la probabilidad de falsa alarma tolerada, en el que será nuestro modelo sin abstención.



**Figura 25.** Escenario con umbral único dependiente de la probabilidad de falsa alarma.

Con esta regla local, cada sensor decidirá si considera que hay un blanco o no en su entorno cuando el nivel de potencia que recibe esté por encima o por debajo del citado umbral único.

Por el contrario, en el modelo con abstención de la Figura 26 podemos comprobar cómo el hecho de que existan dos umbrales provoca que haya dos situaciones en las que podamos considerar la información como confusa. En la izquierda, si el umbral de falsa alarma está por debajo del obtenido en función de la probabilidad de no detección, tendremos unos niveles de potencia intermedios en los que no podremos confirmar presencia o ausencia de blanco. Sin embargo en la derecha, la confusión viene por el hecho de cruzarse los umbrales y provocar que, para determinados niveles de potencia, podamos considerar que hay simultáneamente presencia y ausencia de blanco.



**Figura 26.** Dos configuraciones posibles del escenario formado a partir de dos umbrales.

Como puede apreciarse, en las figuras previas independientemente de considerar uno o dos umbrales, estos van a estar estrechamente relacionados con dos funciones de distribución de potencia  $f_0(z_0)$  y  $f_1(z_1)$ , que se obtienen tras una transformación de variables aleatorias conocidas. Antes de relacionar los umbrales y las probabilidades, vamos a analizar estas transformaciones.

Supongamos que tenemos una variable aleatoria conocida  $x$ , tal que:

$$x \sim f_X(x) \quad (5.22)$$

Y otra segunda variable aleatoria  $y$ , que se relaciona con la primera del siguiente modo:

$$y = g(x) \sim f_Y(y) \quad (5.23)$$

Donde  $g(x)$  puede no ser invertible. Esto es, pueden existir múltiples funciones que cumplan la siguiente ecuación:

$$x_i = g^{-1}(y) \quad (5.24)$$

Donde el subíndice  $i = 1, \dots, N$  indica a cada uno de los valores que puede cumplir (5.24) donde  $N$  es el número de muestras.

Entonces, el Teorema sobre la transformación de variables aleatorias nos dice que sus funciones de densidad se relacionan de modo que [67]:

$$f_Y(y) = \sum_{i=1}^M f_X(x_i) \left| \frac{dx_i}{dy} \right| \quad (5.25)$$

Para la primera de las transformaciones concretas que hemos realizado, partiendo de la potencia recibida en ausencia de blanco:

$$z_0 = y_0^2 = w^2 \quad (5.26)$$

Si operamos,

$$y_0 = g^{-1}(z) = \pm\sqrt{z_0} \quad (5.27)$$

Introduciéndolo en la expresión previa:

$$\begin{aligned} f_o(z_0) &= \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{(\sqrt{z_0})^2}{2\sigma_w^2}\right) \left| \frac{1}{2\sqrt{z_0}} \right| + \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{(\sqrt{z_0})^2}{2\sigma_w^2}\right) \left| -\frac{1}{2\sqrt{z_0}} \right| u(z_0) \\ &= \frac{1}{\sqrt{2\pi z_0 \sigma_w^2}} \exp\left(-\frac{z_0}{2\sigma_w^2}\right) u(z_0) \end{aligned} \quad (5.28)$$

De este modo resulta la primera de las distribuciones que estábamos buscando, y que depende exclusivamente del ruido. Donde  $u(z_0)$ , se corresponde con la función escalón [68]:

$$u(z_0) = \begin{cases} 1, & z_0 > 0 \\ 0, & z_0 < 0 \end{cases} \quad (5.29)$$

Para la otra distribución, fruto de la suma de la señal recibida y el ruido, obtenemos:

$$z_1 = y_1^2 = (x + w)^2 \quad (5.30)$$

Donde  $x$  y  $P_x(d)$  vienen dadas por (5.8) y (5.12).

Si operamos análogamente al caso previo y obtenemos:

$$f_1(z_1|x) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \left[ \exp\left(-\frac{(\sqrt{z_1} - x)^2}{2\sigma_w^2}\right) \left| \frac{1}{2\sqrt{z_1}} \right| + \exp\left(-\frac{(-\sqrt{z_1} - x)^2}{2\sigma_w^2}\right) \left| -\frac{1}{2\sqrt{z_1}} \right| \right] u(z_1) \quad (5.31)$$

Reagrupando los términos:

$$f_1(z_1|x) = \frac{1}{\sqrt{8\pi z_1 \sigma_w^2}} \left[ \exp\left(-\frac{(\sqrt{z_1} - x)^2}{2\sigma_w^2}\right) + \exp\left(-\frac{(\sqrt{z_1} + x)^2}{2\sigma_w^2}\right) \right] u(z_1) \quad (5.32)$$

Conocidas  $f_o(z_0)$  y  $f_1(z_1|x)$ , podemos centrarnos en encontrar los umbrales de potencia en función de las probabilidades de no detección o de falsa alarma.

### 5.3.1.2 Umbral único

Inicialmente vamos a contemplar un umbral único, que estará ligado con la probabilidad de falsa alarma. Esto es, considerando que no hay blanco trataremos de controlar que número de detecciones a causa del ruido vamos a tolerar.

Podemos relacionar el citado umbral con la potencia tal y como se hacía en el Capítulo 3:

$$\begin{aligned} P_{fa} &= \int_{\gamma_{fa}}^{\infty} f_o(z_0) dz_0 \\ &= \int_{\gamma_{fa}}^{\infty} \frac{1}{\sqrt{2\pi z \sigma_w^2}} \exp\left(-\frac{z_0}{2\sigma_w^2}\right) dz_0 \end{aligned} \quad (5.33)$$

Dada la complejidad para integrar (5. 33), podemos trabajar de un modo equivalente con la distribución del ruido, que sabemos que es una Gaussiana, de modo que:

$$\begin{aligned}
 P_{fa} &= P_r\{z_0 > \gamma_{fa}\} \\
 &= P_r\{|y_0| > \sqrt{\gamma_{fa}}\} \\
 &= 2 P_r\{y_0 > \sqrt{\gamma_{fa}}\}
 \end{aligned} \tag{5. 34}$$

Y por tanto la probabilidad de falsa alarma puede reescribirse como:

$$\begin{aligned}
 P_{fa} &= 2 \int_{\sqrt{\gamma_{fa}}}^{\infty} \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{y_0^2}{2\sigma_w^2}\right) dy_0 \\
 &= \operatorname{erfc}\left(\sqrt{\frac{\gamma_{unico}}{2\sigma_w^2}}\right)
 \end{aligned} \tag{5. 35}$$

Donde  $\operatorname{erfc}(x)$  se corresponde con la función de error complementaria que se enuncia como sigue [69]:

$$\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} \exp(-t^2) dt \tag{5. 36}$$

Y a partir de (5. 35) podemos despejar el umbral en función de la probabilidad de falsa alarma:

$$\gamma_{unico} = 2\sigma_w^2(\operatorname{erfc}^{-1}(P_{fa}))^2 = \gamma_{fa} \tag{5. 37}$$

Esta ecuación es la que utilizaremos a partir de ahora para el diseño del umbral, y a la poste del sistema de detección.

### 5.3.1.3 Modelo con abstención

En este segundo escenario vamos a considerar dos umbrales, que pueden o no cruzarse, y que se obtienen en función de la probabilidad de falsa alarma y de no detección. El primero de ellos es el utilizado en el escenario anterior, dependiente de la probabilidad de falsa alarma, y se corresponde con (5. 37). Para el segundo de los umbrales, vamos a apoyarnos en la probabilidad de no detección y en la distribución obtenida anteriormente,  $f_1(z_1)$ :

$$P_{nd} = \int_0^{\gamma_{nd}} f_1(z_1) dz_1 \tag{5. 38}$$

En este caso, existe una dependencia clara de la distancia entre el blanco y el sensor que, para evitar confusión en la nomenclatura en el trabajo con el diferencial de la integral, pasamos de llamar  $d$  a llamar  $l$ . En consecuencia,  $P_{nd}$  se puede expresar como:

$$P_{nd} = \int_0^{\gamma_{nd}} \int_0^{\infty} f_1(z_1|l) f_L(l) dl dz_1 \quad (5.39)$$

Donde  $f_1(z_1|l) = f_1(z_1|x)$  y viene dado por (5.32) y  $f_L(l)$  es la fdp a priori de la distancia entre blancos y sensores. De forma análoga, podríamos tratar de aplicar el razonamiento usado para el umbral anterior manejando la distribución de la combinación de ambas señales, considerando que el resultado debería ser:

$$\begin{aligned} P_{nd} &= P_r\{z_1 < \gamma_{nd}\} \\ &= P_r\{|y_1| < \sqrt{\gamma_{nd}}\} \\ &= P_r\{-\sqrt{\gamma_{nd}} < y_1 < \sqrt{\gamma_{nd}}\} \end{aligned} \quad (5.40)$$

En consecuencia, podemos redefinir la probabilidad de no detección como:

$$P_{nd} = \int_{-\sqrt{\gamma_{nd}}}^{\sqrt{\gamma_{nd}}} \int_0^{\infty} f_1(y_1|l) f_L(l) dl dy_1 \quad (5.41)$$

Donde,

$$f_1(y_1|l) = \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{(y_1 - x(l))^2}{2\sigma_w^2}\right) \quad (5.42)$$

Por lo tanto, el problema se reduce a encontrar la función de distribución de la distancia  $f_L(l)$  y calcular las integrales. Es aquí donde surgen los primeros inconvenientes, haciendo necesario el trabajo con aproximaciones. De entre las posibles soluciones se ha considerado la siguiente, apoyada en el trabajo empírico. Supongamos por ejemplo, que se despliegan 300 sensores, entonces tendremos, para una determinada posición aleatoria, 300 distancias y 300 muestras de potencias recibidas en función de cada distancia. Si repetimos esa operación un número elevado de veces, tomando puntos aleatorios con los que generar distancias a los sensores siempre fijos, podremos estimar  $f_L(l)$  como una suma ponderada de deltas desplazados en función de cada una de las distancias, esto es:

$$f_L(l) \approx \sum_{i=1}^N \frac{1}{N} \delta(l - l_i) \quad (5.43)$$

Desde el punto de vista práctico, si realizamos la siguiente aproximación:

$$\int_0^{\infty} f_1(y_1|l) f_L(l) dl \approx \frac{1}{N} \sum_{i=1}^N f_1(y_1|l_i) \quad (5.44)$$

Y la probabilidad de no detección resulta como sigue:

$$\begin{aligned}
 P_{nd} &\approx \frac{1}{N} \sum_{i=1}^N \int_{-\sqrt{\gamma_{nd}}}^{\sqrt{\gamma_{nd}}} f_1(y_1|l_i) dy_1 \\
 &= \frac{1}{N} \sum_{i=1}^N \int_{-\sqrt{\gamma_{nd}}}^{\sqrt{\gamma_{nd}}} \frac{1}{\sqrt{2\pi\sigma_w^2}} \exp\left(-\frac{(y_1 - x(l_i))^2}{2\sigma_w^2}\right) dy_1
 \end{aligned} \tag{5.45}$$

Integrando llegamos a la probabilidad de no detección:

$$P_{nd} \approx \frac{1}{2N} \sum_{i=1}^N \left[ \operatorname{erf}\left(\frac{\sqrt{\gamma_{nd}} + x(l_i)}{\sqrt{2\sigma_w^2}}\right) + \operatorname{erf}\left(\frac{\sqrt{\gamma_{nd}} - x(l_i)}{\sqrt{2\sigma_w^2}}\right) \right] \tag{5.46}$$

Donde [69]:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2) dx \tag{5.47}$$

Obviamente no podemos despejar de modo trivial el umbral de no detección en función de la probabilidad, como ocurría con la probabilidad de no detección. Para solucionar este problema plantearemos una función de coste cuadrática cuyo mínimo coincide con la probabilidad de no detección deseada:

$$J = \left( P_{nd} - \frac{1}{2N} \sum_{i=1}^N \left[ \operatorname{erf}\left(\frac{\sqrt{\gamma_{nd}} + x(l_i)}{\sqrt{2\sigma_w^2}}\right) + \operatorname{erf}\left(\frac{\sqrt{\gamma_{nd}} - x(l_i)}{\sqrt{2\sigma_w^2}}\right) \right] \right)^2 \tag{5.48}$$

De modo que  $\gamma_{nd}$  es el argumento que minimiza la función  $J$ .

### 5.3.1.4 El radio de alcance

A continuación vamos a describir la forma de encontrar un parámetro que, sin ser fundamental para el desarrollo de la detección, puesto que es obtenido indirectamente, proporciona una dimensión práctica de lo que representan dichos límites de potencia.

Para el escenario en el que el umbral es único y está asociado a la probabilidad de falsa alarma, el radio se obtiene igualando la potencia que se obtendría en ausencia de ruido con el mencionado umbral. Esto es,

$$P_x(r') = \gamma_{fa} \tag{5.49}$$

Si desarrollamos ambas expresiones:

$$\frac{P(1m)}{d_{max}^2} (r')^{-2} = 2\sigma_w^2 [erfc^{-1}(P_{fa})]^2 \quad (5.50)$$

Es necesario recordar que  $r'$  sería el radio normalizado, que la constante de pérdidas es establecida por defecto en 2, y que el valor de la potencia recibida de referencia se toma a distancia un metro,  $P(1m)$ . De este modo podemos relacionar el radio con el umbral y a su vez con la probabilidad de falsa alarma sin más que operar:

$$\begin{aligned} r' &= \frac{r}{d_{max}} \\ &= \sqrt{\frac{P(1m)}{2\sigma_w^2 [erfc^{-1}(P_{fa})]^2 d_{max}^2}} \end{aligned} \quad (5.51)$$

Finalmente, eliminando la dependencia con  $d_{max}$ , obtenemos:

$$r = \sqrt{\frac{P(1m)}{2\sigma_w^2 [erfc^{-1}(P_{fa})]^2}} \quad (5.52)$$

Como podemos comprobar, la varianza del ruido vuelve a jugar un papel clave en el escenario. En este caso la varianza marca el orden de magnitud de la distancia, y la variación en torno a esa magnitud es la que depende de la probabilidad de falsa alarma. Estableciendo en un valor fijo de 1mW la potencia a un metro y en  $10^{-6}$  la varianza del ruido podemos comprobar la variación del radio de alcance en función de la probabilidad de falsa alarma. Como puede apreciarse en la Tabla 4, el radio aproximadamente se duplica si aumentamos la probabilidad de falsa alarma tolerada de 0,005 a 0,2.

Valores Pfa	Radio (m)
0,005	11,26
0,01	12,28
0,05	16,13
0,1	19,22
0,2	24,68

**Tabla 4.** Variación del radio de detección con respecto a la probabilidad de falsa alarma tolerada.

En un modelo equivalente, habitualmente a distancias por debajo o coincidentes con ese radio se consideraría detectado el blanco, mientras que a distancias por encima del mismo se consideraría no detectado. Como es lógico, cuanto menor es la tolerancia a falsas alarmas menor es el radio de alcance, que limitará cuales son los sensores que detectan blanco y cuáles no. Si se cierra el radio, implica subir el nivel de potencia necesario para detectar, evitando de este modo muchas detecciones incorrectas a causa del ruido.

Como puede verse en la Figura 27, para el escenario con dos umbrales tendremos dos niveles de potencia, que se traducirán en dos distancias de referencia: una a partir de la cual se considera detección,  $r_{int}$ , y otra a partir de la cual se considera no detección,  $r_{ext}$ , y un rango de distancias intermedias que forman la región de incertidumbre o de abstención.

Como el radio de detección puede ser mayor que el de no detección, el sistema establece los radios como:

$$\begin{aligned} r_{int} &= \min\{r_d, r_{nd}\} \\ r_{ext} &= \max\{r_d, r_{nd}\} \end{aligned} \quad (5.53)$$

Mientras que el radio obtenido en base a la falsa alarma no se ve modificado, el cálculo del radio de no detección no es inmediato y requiere del trabajo empírico, y por eso se desarrollará en el capítulo de resultados.

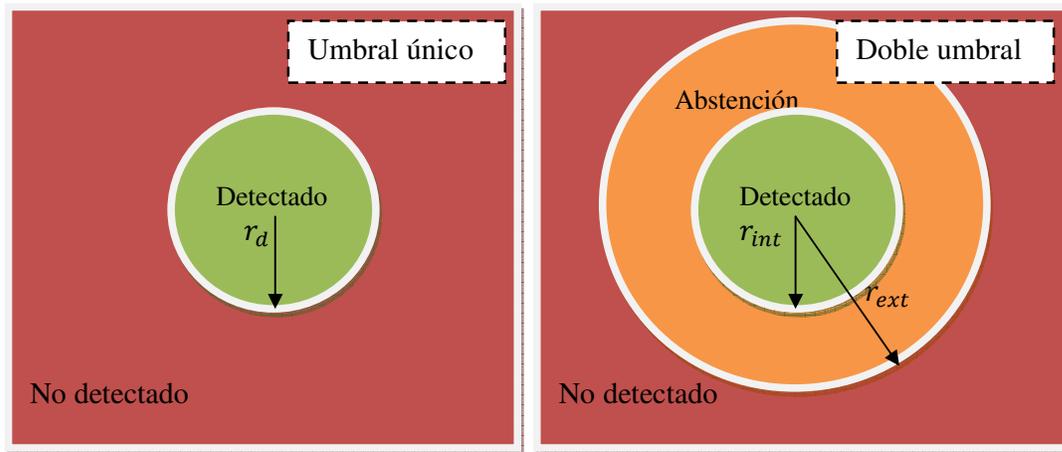


Figura 27. Comparativa de radios de detección y no detección para los dos escenarios posibles.

### 5.3.2 Detección en el centro de fusión

Podemos considerar la distribución de los sensores y el cálculo de los umbrales como una carga inicial que hay que realizar en el sistema, esto es, como un “set up” de los parámetros del entorno. Una vez conocidas las reglas de decisión local, es necesario trabajar con toda la información disponible en el centro de fusión para obtener la mayor precisión posible a la hora de decidir de forma global. En el centro de fusión, en lugar de tratar de desarrollar un detector sencillo, usando algún criterio como el de Neyman-Pearson, se va a recurrir a métodos de aprendizaje máquina, en concreto a las máquinas de vectores soporte (SVM). Es el momento por tanto de trabajar metódicamente en la generación de los datos, puesto que el modelo SVM utilizado así lo requiere. Por ello,

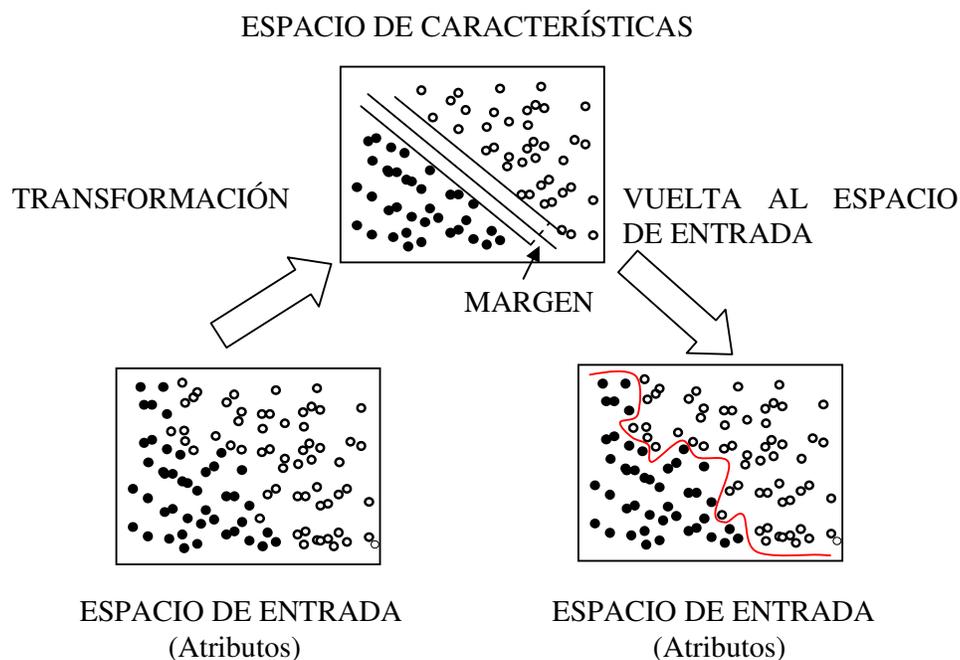
antes de continuar es recomendable conocer más sobre las máquinas de vectores soporte para ajustar los datos al formato que esta herramienta necesita.

### 5.3.2.1 SVM

Las máquinas de vectores soporte o “Support Vector Machines” (SVMs) son máquinas que llevan a cabo la clasificación mediante la construcción de un hiperplano N-dimensional que separa los datos de forma óptima en dos categorías. En la literatura relacionada con las SVM, las variables con las que contamos para realizar la predicción se denominan atributos. Dichos atributos son transformados no linealmente y pasan a denominarse características (“features”), pudiendo agruparse formando vectores que son utilizados para definir los hiperplanos de separación, de tal manera que los casos con una categoría de la variable de destino se encuentren en un lado de dicho hiperplano y los casos de la otra clase se encuentran en el otro [49], [50].

Por tanto, la SVM se apoya en dos procesos de forma paralela:

- Transformar las observaciones a otro espacio y establecer los vectores soporte (y llevarlos de vuelta al espacio de entrada), como se muestra de modo gráfico en la Figura 28.
- Maximizar de la distancia mínima entre dos observaciones de clases distintas, tal y como se ilustra en la Figura 29.



**Figura 28.** Esquema de trabajo de una SVM: transformación de los datos a otra dimensión donde se trabaja y crea la frontera lineal y transformación de vuelta al espacio de entrada [51].

Aunque no es el objeto principal de estudio de este proyecto, dado que las funciones de la SVM se toman de la librería de libre distribución LIBSVM[70], es conveniente conocer los principales parámetros libres de la SVM. Estos son el coste (C), fruto del problema de optimización citado debajo, y los asociados a cada Kernel.

Independientemente del Kernel, el objetivo de la SVM es producir un modelo en base a los datos de entrenamiento que sea capaz de predecir las etiquetas o hipótesis en función de los atributos de entrada. Así pues, dado un conjunto de parejas de instancias y etiquetas  $(\mathbf{x}_i, \mathbf{y}_i)$  para  $i = 1, \dots, l$ . Donde  $\mathbf{x}_i \in \mathbb{R}^n$  e  $\mathbf{y}_i \in \{1, -1\}$ , entonces la máquina de vectores soporte tratará de resolver el siguiente problema de optimización [71]:

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l \xi_i \quad (5.54)$$

Sujeto a:

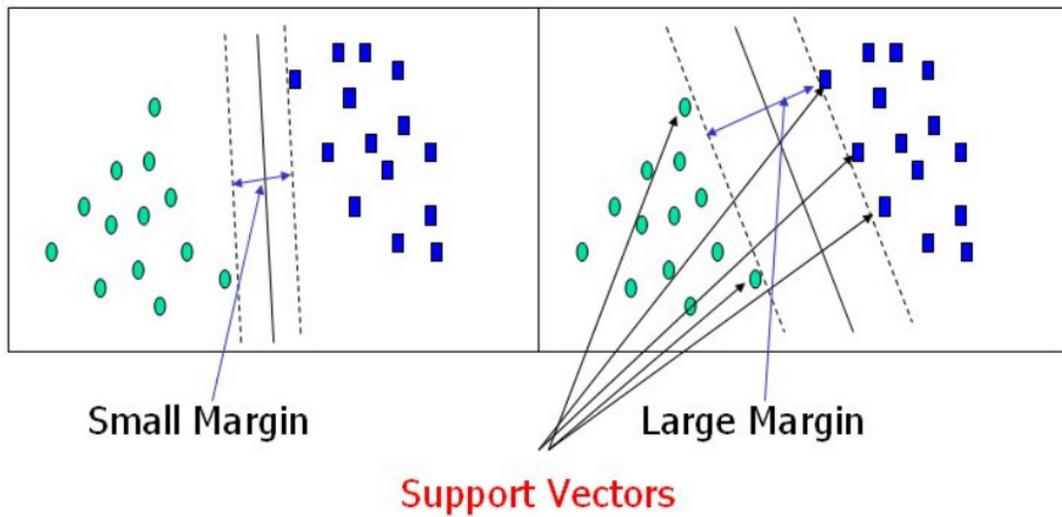
$$\begin{aligned} \mathbf{y}_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \end{aligned} \quad (5.55)$$

Donde  $\phi(\mathbf{x}_i)$  denota un conjunto de transformaciones no lineales del espacio de entrada al espacio de características,  $\xi_i$  son una serie de “slack variables” que tienen en cuenta puntos no separables y el parámetro C controla la relación entre la complejidad de la máquina y el número de puntos no separables.

Antes de considerar los hiperplanos con N dimensiones, analicemos un ejemplo con dos dimensiones. Supongamos que queremos realizar una clasificación y nuestros datos son continuos en la entrada y binarios en la salida, tal y como se representa en la Figura 29 (clase 1: oval verde, clase 2: cuadrado azul).

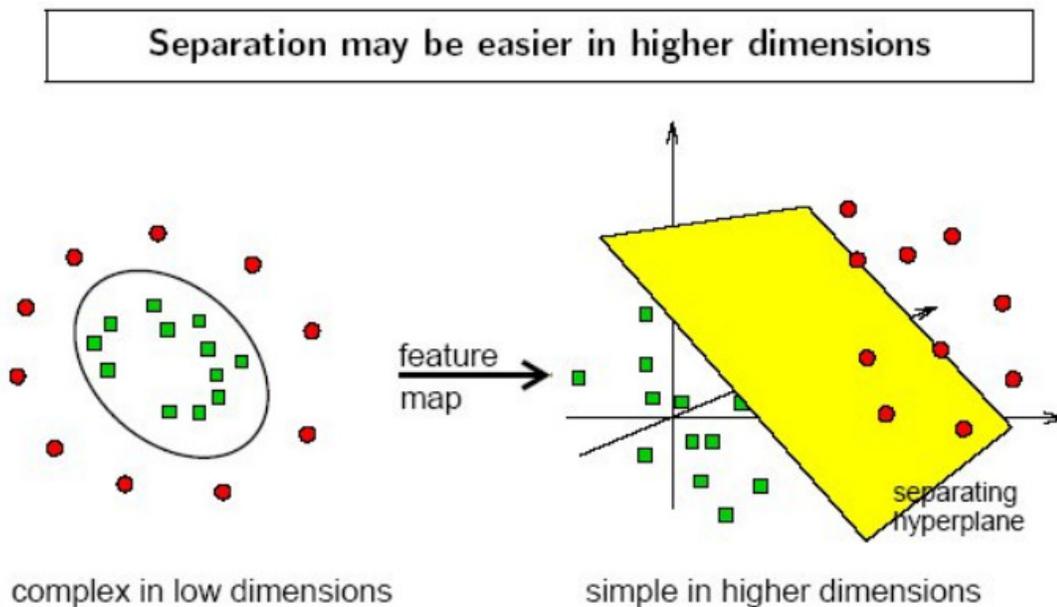
En el ejemplo idealizado, los valores quedan completamente separados y el análisis SVM nos proporciona un hiperplano de una dimensión (una línea) que actúa de frontera. Sin embargo, hay infinitas líneas que permitirán representar dicha frontera y la cuestión ahora pasa por encontrar cual es la mejor. Las líneas de puntos son paralelas a la frontera y nos ayudan a marcar la distancia entre la misma y los puntos más cercanos, el margen. Los puntos (que serían vectores si fueran de mayor dimensión) que delimitan dicho margen son los anteriormente llamados vectores de soporte (“Support Vectors”). La idea es maximizar ese margen y por ello en la Figura 29, la mejor frontera sería la de la derecha.

Desafortunadamente la mayoría de los análisis no son de clasificación binaria, ni cuentan con sólo dos variables de entrada. Además, la representación no suele estar tan perfectamente agrupada. A medida que agregamos otras variables de predicción (atributos) los datos se pueden representar en el espacio N-dimensional y separarse con un hiperplano de dimensión N-1.



**Figura 29.** Comparativa de los márgenes obtenidos por distintas fronteras (SVM) [53].

Si la frontera de separación no fuese lineal habría que ajustar las curvas correspondientes a los datos. Como ya se ha mencionado, en lugar de buscar directamente dichas curvas, las SVM utilizan funciones “Kernel” para proyectar los datos en otro espacio diferente, probablemente de mayor dimensión, donde los hiperplanos sí pueden ser utilizados para realizar la separación de un modo sencillo, tal y como se ilustra en la Figura 30. En la misma podemos ver como la transformación a otro espacio permite una separación mediante un hiperplano (amarillo) en lugar de trazar curvas complejas.



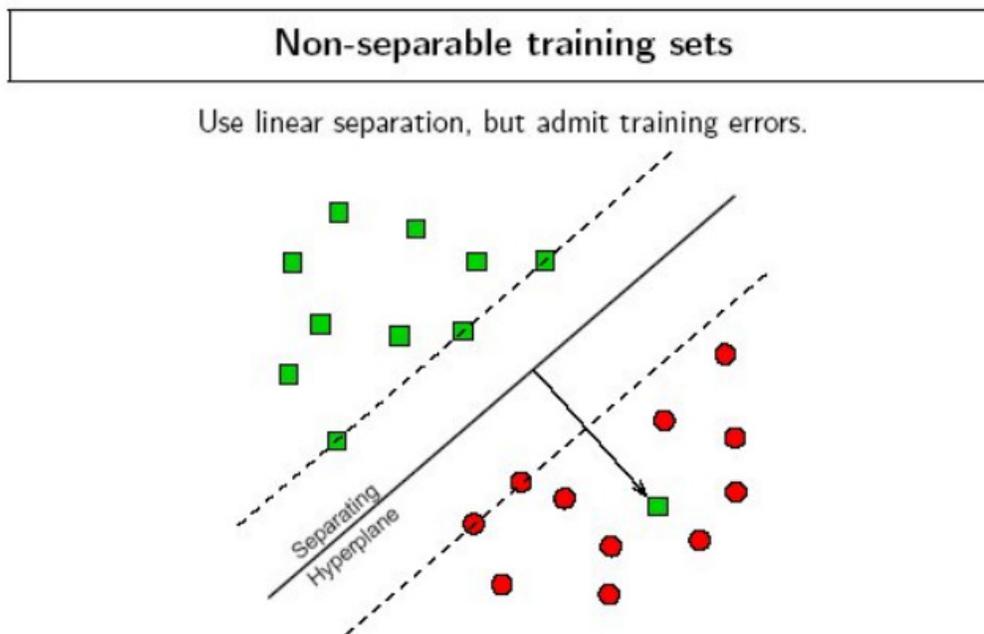
**Figura 30.** Transformación a un espacio de mayor dimensión para facilitar la separación[51].

Como mencionábamos con anterioridad, la transformación requiere de una función de núcleo o “Kernel”. Existen multitud de ellas, pero entre las más utilizadas destacan la función lineal y la de base radial (RBF) o Gaussiana, cuyas Ecuaciones son (4. 8) y (4. 9) respectivamente.

La precisión de un modelo SVM depende en gran medida de los parámetros de dichas funciones “Kernel”. Dos métodos para encontrar los valores óptimos de los parámetros pueden ser la búsqueda de patrones o la búsqueda por zonas. Si realizamos la búsqueda por zonas, se asignan unos rangos específicos y se van evaluando dichos parámetros en una rejilla. Por el contrario, si utilizamos una búsqueda de patrones, se inicia en un centro y se hacen búsquedas en ambas direcciones con un determinado salto para cada parámetro: si el modelo mejora se mueven los centros a los nuevos puntos y se repite el proceso. Si no, el tamaño del salto se reduce y se busca de nuevo. La condición de parada se alcanza cuando el salto es menor que una tolerancia especificada.

Aun así, es posible que no se alcance la separación perfecta, o que el modelo tenga tantas dimensiones que no generalice bien para otros conjuntos de datos. Tendremos entonces dos problemas a los que debemos hacer frente: el subajuste o sobreajuste y el hecho de que los conjuntos no sean completamente separables.

Si el conjunto **no es claramente separable**, como el de la Figura 31, la SVM permite una cierta flexibilidad en la separación de las categorías, pero ello conlleva un incremento en el coste (C en (5. 54)). Típicamente el coste es la suma de todas las distancias desde el margen a los puntos mal clasificados [51], que en la citada figura correspondería con el cuadrado verde que se encuentra rodeado de círculos rojos.



**Figura 31.** Los conjuntos de datos de entrenamiento no separables acarrear un coste [51].

En cuanto al **subajuste o sobreajuste**, podemos encontrar ejemplos de cada situación en la izquierda y derecha de la Figura 32 respectivamente. El caso intermedio, y por tanto deseado, sería el mostrado en el recuadro inferior. Para evaluar el posible sobreajuste o subajuste se utilizan datos de validación, que nos permiten comprobar el comportamiento del modelo con un conjunto de datos independiente del utilizado para encontrar los vectores soporte.

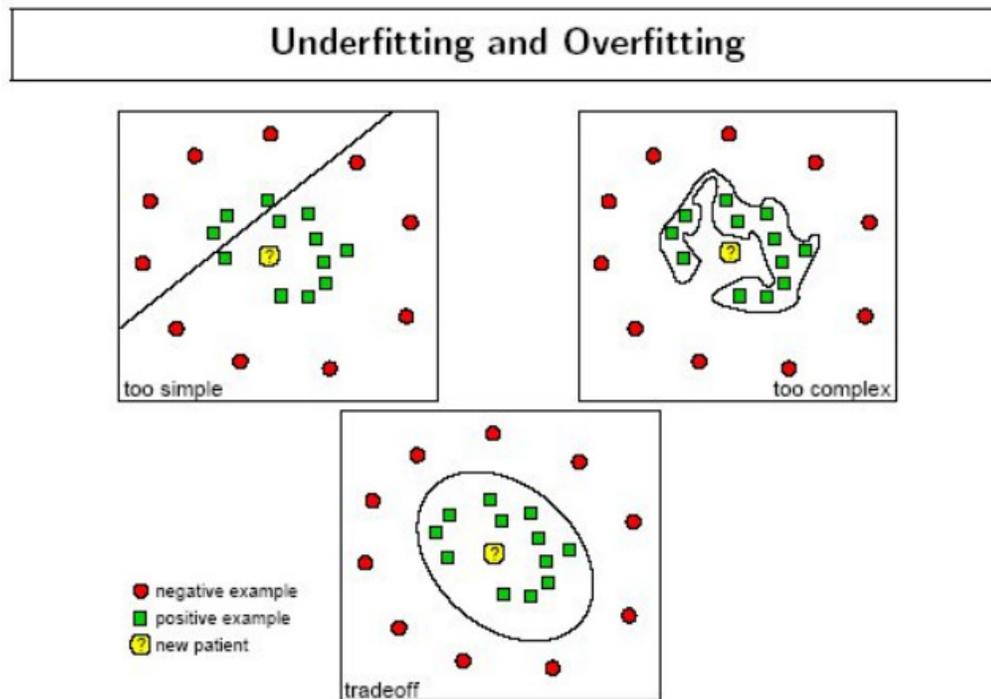


Figura 32. Ejemplos de sobreajuste y subajuste del modelo [51].

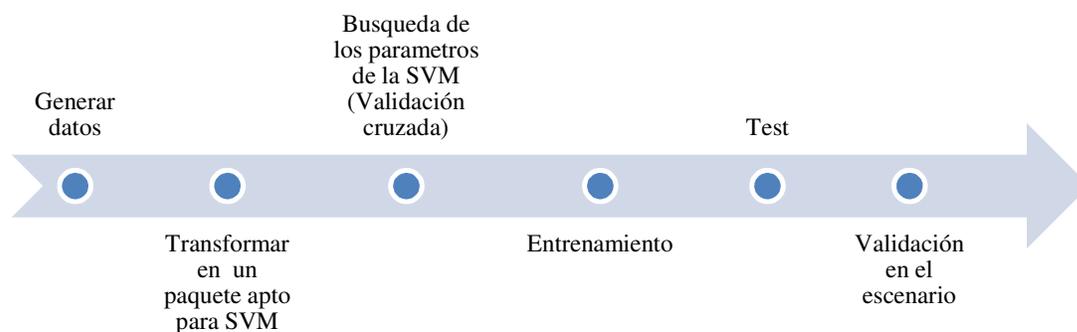
### 5.3.2.2 El procedimiento seguido

En este apartado se ordenan y especifican las distintas fases del trabajo desarrollado para encontrar el modelo y a la postre el decisor global [72]:

- 1) **Generación de datos:** En esta fase inicial, a partir del escenario descrito se obtienen las posiciones de los sensores y su decisión local.
- 2) **Transformación en un paquete apto para SVM:** En esta etapa la información recogida localmente debe ser tratada de acuerdo con las necesidades de la SVM y la librería de software que vamos a utilizar.
- 3) **Búsqueda de los parámetros de la SVM. (Validación cruzada):** La SVM requiere de una función de núcleo, en nuestro caso será inicialmente el “Kernel” Lineal y posteriormente el Gaussiano. Para entrenar a la máquina de vectores soporte y escoger los parámetros del núcleo que maximizan la precisión nos apoyaremos en la validación cruzada.

- 4) **Entrenamiento:** Una vez obtenidos los parámetros que maximizan la precisión, debemos crear un modelo considerando todos los datos del conjunto de entrenamiento. En esta fase es en la que se encuentran los vectores soporte que definen la frontera de decisión.
- 5) **Test:** Aprovechando el conjunto de datos de test, en esta fase se comprueba si los valores de precisión, sensibilidad y especificidad no varían significativamente o si por el contrario se ha producido un sobreajuste a la hora de generar el modelo.
- 6) **Validación en el escenario:** En esta última fase se generan nuevos datos considerando una probabilidad de aparición del blanco. Los datos nos servirán para cotejar los resultados en términos de precisión. Además, los almacenaremos para disponer de ellos en la tarea de estimación.

De modo gráfico, las fases se muestran en la Figura 33.



**Figura 33.** Fases del trabajo con la SVM.

A continuación vamos a realizar un análisis detallado de cada una de ellas.

### 5.3.2.2.1 Generación de datos

La idea es generar la misma proporción de datos con ausencia de blanco que con presencia de blanco, ya que en el caso contrario la frontera de decisión podría estar sesgada, penalizando aquella clase con un menor número de muestras disponibles. Si así fuera, existen diversas técnicas capaces de corregir este sesgo [73]. Sin embargo, en este caso el problema se soluciona modificando lo que definimos a lo largo de la simulación completa como la probabilidad de aparición de blanco. Si dicha probabilidad de aparición se establece a cero, podemos generar tantos datos como necesitemos de forma continua en ausencia de blanco. Si, por el contrario, dicha probabilidad se establece a 1, siempre habrá presencia de un blanco, simulando tantas trayectorias nuevas como necesitemos. Concretamente, hay que generar datos suficientes para entrenar el modelo y para comprobar que el mismo ha sido satisfactorio. De acuerdo con esta premisa y retomando el caso práctico que nos concierne, se consideran los siguientes tipos de datos.

- El primero se corresponde con cada par de coordenadas que forman la posición en la trayectoria.
- El segundo se corresponde con la etiqueta real que le asignamos en la generación, 0 si el dato proviene de ausencia de blanco y 1 si el dato es obtenido cuando hay presencia de blanco.
- El tercero se corresponde con un vector de longitud coincidente con el número de sensores y que almacena la decisión local de cada uno de ellos.
- Adicionalmente se conoce una matriz con las posiciones de los sensores, que se obtiene durante la iniciación del sistema.

Con el fin de obtener un número suficiente de datos y mantener el compromiso con la complejidad de entrenamiento del sistema, se ha decidido trabajar con 6 bloques de datos de cada tipo coincidiendo con la cantidad de trayectorias diferentes identificadas (ver Figura 24). Cada conjunto tendrá longitud 40 (múltiplo de la longitud de la trayectoria), es decir cada conjunto de datos obtenido en presencia de blanco se corresponderá con 2 trayectorias completas distintas seguidas por el blanco.

#### **5.3.2.2.2 Obtener un paquete apto para la SVM**

Para cualquier escenario, lo ideal sería poder cuantificar los valores disponibles y asignar un código binario a cada valor, pero en determinadas situaciones el número de categorías es tan grande que es prácticamente imposible cuantificarlo, tal y como ocurre con las posiciones de los sensores. Si recordamos, en la configuración inicial se mencionaba el escalado de las posiciones, dividiendo por la desviación máxima tolerada en cada eje con la finalidad de que todas las coordenadas estén entre -1 y 1. Para mantener en dicho rango todos los datos, trataremos las etiquetas asignadas en presencia y ausencia de blanco para que pasen a la siguiente correspondencia: -1 ausencia de blanco, 1 presencia de blanco. De este modo, para cada instante la posición (dos dimensiones) de cada sensor, junto con su decisión local, así como la etiqueta global de presencia o ausencia de blanco son entregados al centro de fusión y todos tienen el mismo rango. Si consideramos  $N$  sensores, la cantidad de datos que este último recibe es  $3N+1$ .

En la Figura 34 se representa cómo se agrupan los datos generados para formar un paquete apto para la SVM y para su entrenamiento mediante validación cruzada. A partir de los bloques generados en presencia y ausencia de blanco; un tercio de cada tipo se separa para dedicarlos en exclusiva a comprobar el rendimiento del modelo en la fase de test. El resto se combinan en bloques mixtos de modo que los datos estén balanceados 50% “presencia” (azul) y 50% de “ausencia” (rojo).

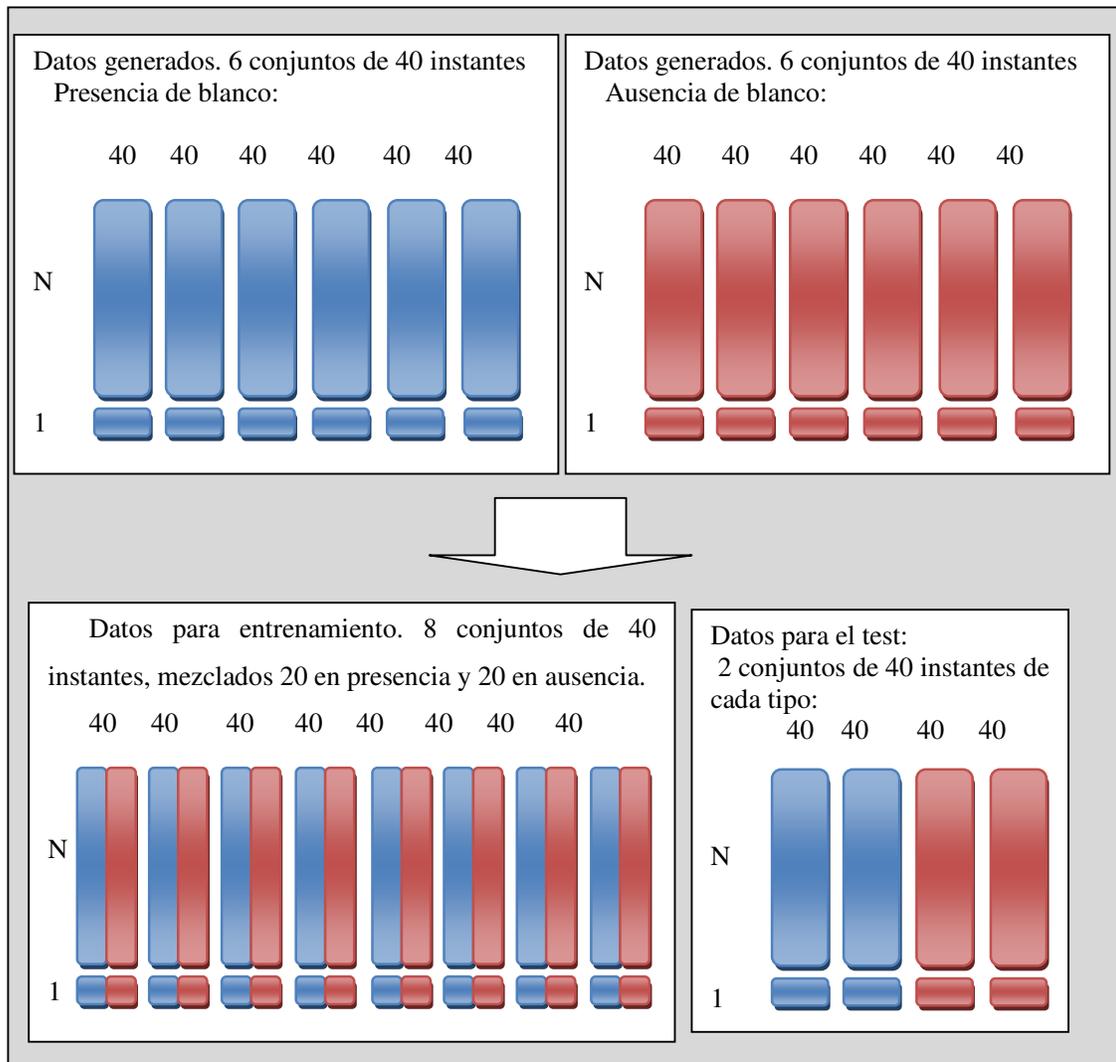


Figura 34. Agrupamiento de los datos disponibles para entrenamiento y test.

### 5.3.2.2.3 Búsqueda de los parámetros de la SVM. (Validación cruzada)

En la Sección 5.3.2.1 se mencionaban las dos tareas paralelas que realiza la SVM: por un lado medir la similitud entre los datos mediante las funciones “Kernel” y por otro proyectar los datos en otro espacio diferente al de las observaciones, posiblemente de mayor dimensión, donde los hiperplanos sí pueden ser utilizados para realizar la separación de un modo más sencillo. Fruto del problema de optimización descrito en dicha sección aparece la dependencia del modelo con nuestro parámetro C. Este será el único parámetro libre si el “Kernel” que usamos en la SVM es el lineal, expuesto previamente en (4. 8), y que recordamos ahora:  $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$

Además, podemos permitirnos utilizar este Kernel, porque el número de características y muestras disponibles es elevado y es posible que no sea necesario trasladar los datos a una dimensión mayor.

Así pues, como tarea previa al entrenamiento propiamente dicho, es necesario realizar una validación cruzada que nos permita encontrar el valor del parámetro C (en el caso del “Kernel” lineal) que proporcione los mejores resultados posibles en términos de precisión, sensibilidad y especificidad. A continuación se definen estos tres parámetros, que resultan fundamentales en lo sucesivo [70]:

La **sensibilidad** se corresponde con el cociente entre las detecciones correctas de blanco y el total de casos con presencia de blanco, lo identificaremos con el símbolo  $\eta$  y su formulación es la siguiente:

$$\begin{aligned}\eta &= P\{\hat{y} = 1|y = 1\} \\ &= 1 - P\{\hat{y} = -1|y = 1\} \\ &= 1 - P_{nd}\end{aligned}\tag{5.56}$$

La **especificidad** se corresponde con el cociente entre los casos de ausencia de blanco detectados y el total de casos sin blanco, lo identificaremos con el símbolo  $\mu$  y su formulación es la siguiente:

$$\begin{aligned}\mu &= P\{\hat{y} = -1|y = -1\} \\ &= 1 - P\{\hat{y} = 1|y = -1\} \\ &= 1 - P_{fa}\end{aligned}\tag{5.57}$$

La **precisión** por su parte, se corresponde con el cociente entre el número total de aciertos y el número total de casos disponible, lo identificaremos con el símbolo  $\rho$  y su formulación es la siguiente:

$$\rho = 1 - P_e\tag{5.58}$$

Además, puesto que:

$$\begin{aligned}P_e &= P\{\hat{y} = 1|y = -1\}P\{y = -1\} + P\{\hat{y} = -1|y = 1\}P\{y = 1\} \\ &= (1 - P\{\hat{y} = -1|y = -1\})P\{y = -1\} + (1 - P\{\hat{y} = 1|y = 1\})P\{y = 1\} \\ &= P_{fa}P\{y = -1\} + P_{nd}P\{y = 1\}\end{aligned}\tag{5.59}$$

O de forma equivalente:

$$P_e = (1 - \mu)P\{y = -1\} + (1 - \eta)P\{y = 1\}\tag{5.60}$$

Finalmente, si las probabilidades de presencia o ausencia de blanco son iguales,  $P\{y = 1\} = P\{y = -1\} = \frac{1}{2}$ , como ocurre en nuestro caso, la probabilidad de error resulta:

$$\begin{aligned} P_e &= \frac{P_{fa} + P_{nd}}{2} \\ &= 1 - \frac{\mu + \eta}{2} \end{aligned} \quad (5.61)$$

Y la relación entre sensibilidad, especificidad y precisión:

$$\rho = \frac{\mu + \eta}{2} \quad (5.62)$$

Volviendo al problema de la validación cruzada, dado que el valor del coste inicial es desconocido, habitualmente se propone para la validación cruzada un conjunto de costes tal que barra distintos órdenes de magnitud,  $[2^{-5} \ 2^{-4} \dots 2^4 \ 2^5]$ . Es frecuente, en función de los resultados, acotar el rango del coste y realizar una segunda “búsqueda fina”, pero esa tarea corresponde al siguiente capítulo [72].

La citada validación cruzada se apoya en una estrategia “leave-one-out”, y en nuestro caso concreto consiste, como puede verse en la Figura 35, en repetir para cada valor propuesto para el coste el siguiente proceso, aprovechando los 8 bloques de datos mixtos recogidos en la Figura 34.

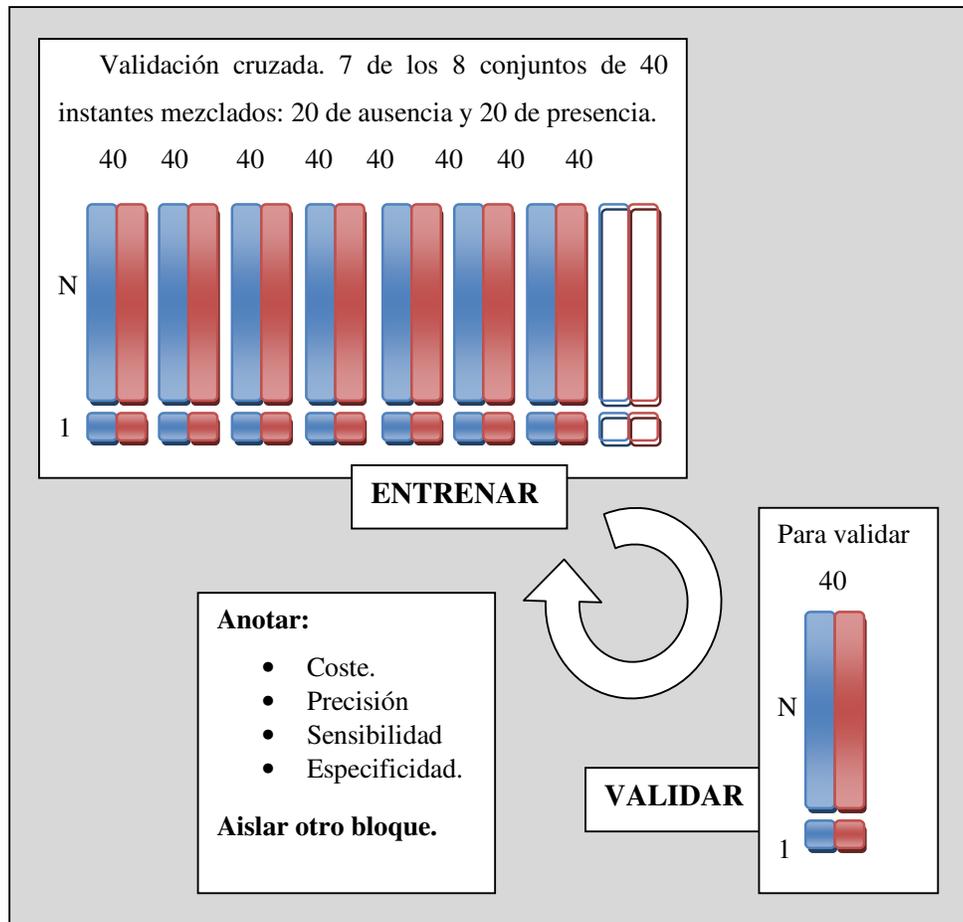
- Tomar cada uno de los 8 bloques y aislarlo.
- Entrenar al modelo con los 7 restantes.
- Comprobar el modelo obtenido con el aislado, anotando el coste, la precisión, la sensibilidad y la especificidad.

En la Figura 35, N representa el número de sensores y coincide con las N decisiones locales (una por sensor). Además, la SVM considera también la etiqueta global (tamaño 1).

El valor de sensibilidad, especificidad y precisión para cada coste correspondiente se obtiene ponderando el valor proporcionado en cada iteración del “leave-one-out” con igual peso. Al concluir el barrido tenemos una sensibilidad, especificidad y precisión para cada coste. Entonces, si establecemos como objetivo maximizar la precisión, bastaría con seleccionar el coste que nos ha proporcionado el mayor valor de precisión.

Por último, para los casos en los que el número de sensores es pequeño es posible que se considere la alternativa de utilizar un núcleo RBF, que recordamos de (4.9):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$$



**Figura 35.** Proceso de validación cruzada mediante la técnica “Leave-one-out”.

Mediante el uso de este “Kernel” la SVM permite obtener fronteras de decisión no lineales, pero requiere, además del coste, derivado del problema previamente comentado en la Sección 5.3.2.1, otro parámetro adicional,  $\gamma$ , que está inversamente relacionado con la varianza en la forma de la Gaussiana del núcleo.

$$\gamma = \frac{1}{2\sigma^2} \tag{5.63}$$

El proceso de entrenamiento y validación es análogo al del “Kernel” lineal con la excepción de que la búsqueda de dos parámetros óptimos obliga a una búsqueda “en una rejilla” de dos dimensiones, produciendo una matriz donde se almacenan pares de costes y valores de  $\gamma$ . El rango recomendado de búsqueda para  $\gamma$  es  $[2^{-15} \ 2^{-14} \dots \ 2^2 \ 2^3]$  [72].

La varianza  $\sigma$  modifica el carácter suave o pronunciado de la función, tal y como veíamos en la Figura 17. Valores bajos de  $\sigma$  producen una caída muy rápida de la medida de similitud entre dos puntos, pudiendo resultar en un problema de sobreajuste (overfitting).

### 5.3.2.2.4 Entrenamiento

Una vez realizada la validación cruzada, que nos permite evaluar el comportamiento de los hiperparámetros de la SVM y nos proporcione una primera medida de la precisión, sensibilidad y especificidad que podrá aportar nuestro modelo, es el momento de generar dicho modelo. Para ello vamos a utilizar todos los datos disponibles para entrenamiento (8 bloques) y una función de la librería SVM [72]. Concretamente la función `svmtrain`, que necesitará conocer el tipo de SVM (binario o no), el tipo de “Kernel” (Lineal, Gaussiano u otro) y mas parámetros de configuración opcionales y que pueden verse en [70]. Este proceso se ilustra en la Figura 36.

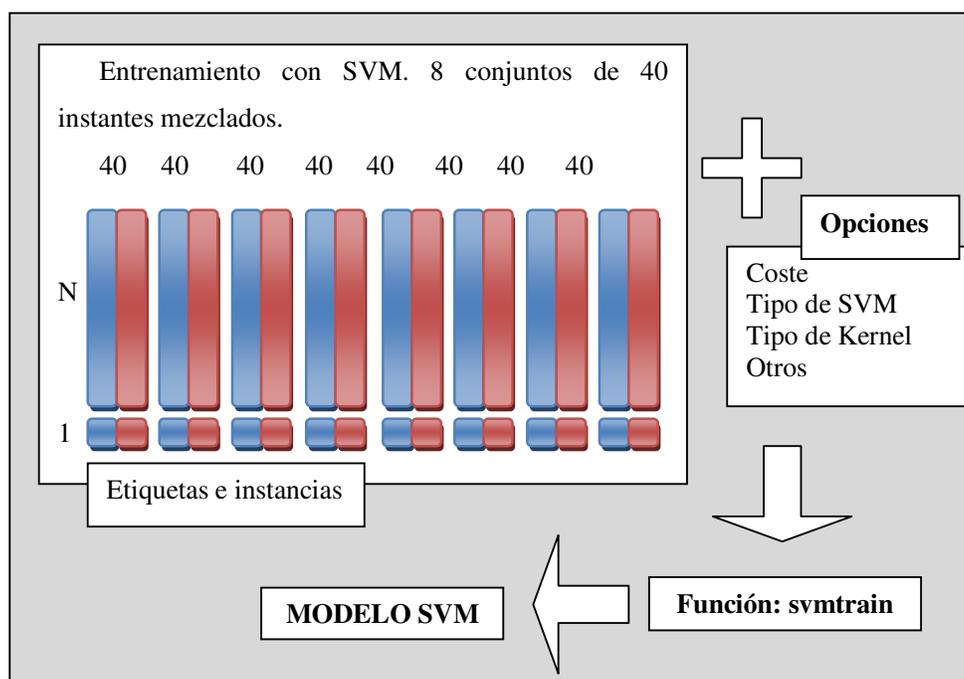


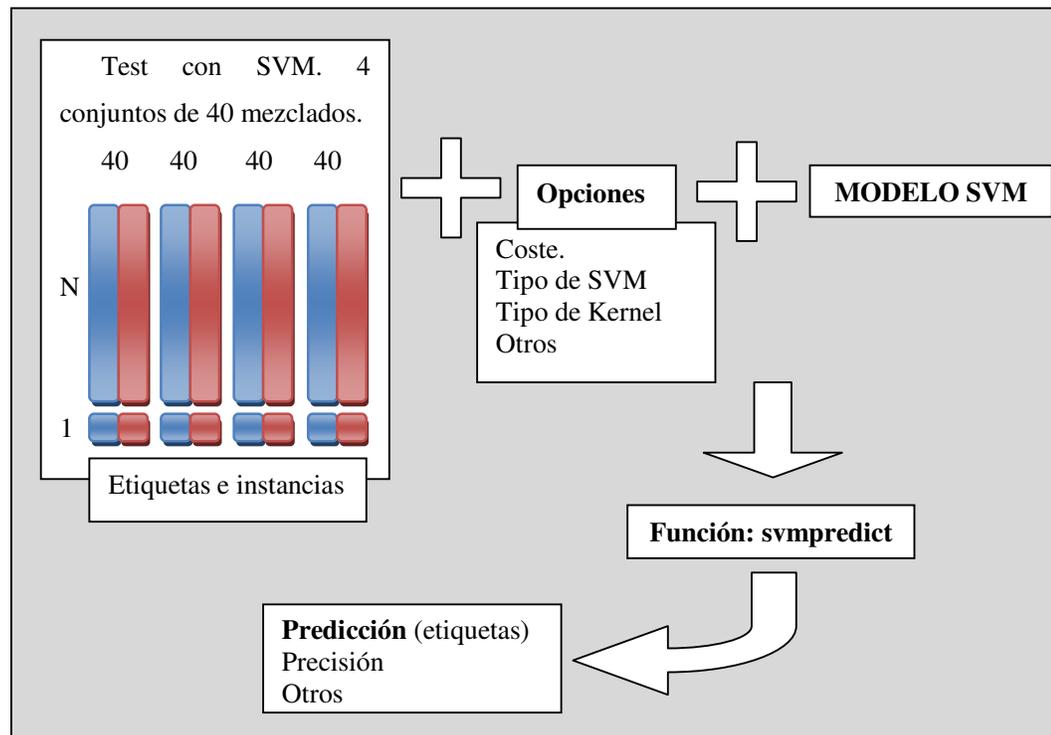
Figura 36. Generación del modelo

El modelo resultante es una estructura de datos que contiene:

- Parámetros. En el caso de ser el “Kernel” lineal sólo  $C$  y en el caso de ser el “Kernel” Gaussiano, también  $\gamma$ .
- Número de clases.
- Los vectores de soporte (SV).
- Número total de vectores soporte.
- El valor de  $b$  en cada regla de decisión del tipo:  $\mathbf{w}^T \phi(\mathbf{x}_i) + b$
- La etiqueta de cada clase SVM.  $(-1, 1)$
- El número de SV para cada clase.
- Los coeficientes de los SV en las reglas de decisión

### 5.3.2.2.5 Test

Una vez generado el modelo, es el momento de comprobar sus prestaciones con datos distintos a los empleados para generarlo. Para esta fase necesitaremos los datos previamente reservados para el test y nos apoyaremos en este caso en otra función de la librería SVM [72], cuyo objetivo es realizar las predicciones `svmpredict`. El procedimiento completo se comprende mejor observando la Figura 37. En dicha figura se consideran los datos reservados en la fase de generación: 4 conjuntos de 40 instancias, fruto de la combinación de 20 instancias creadas en ausencia de blanco (rojo) y otras 20 instancias creadas en presencia de blanco (azul). N se corresponde de nuevo con el número de sensores (y las decisiones locales de cada uno) y el 1 con la etiqueta real. Además, la función `svmpredict` requiere especificaciones como son: el tipo de SVM (binaria o no) que se está utilizando y el Kernel (Lineal o Gaussiano). También necesita los parámetros establecidos como óptimos a partir de la validación cruzada.

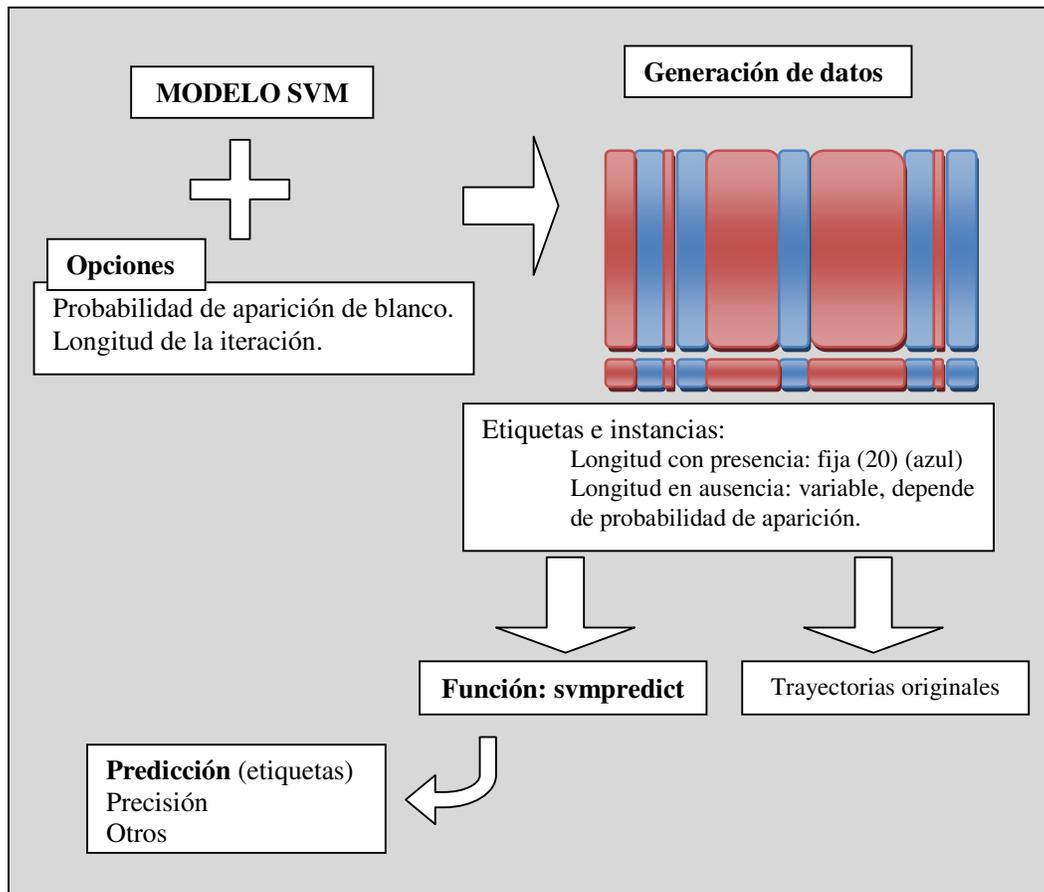


**Figura 37.** Uso del modelo para generar predicciones a partir de los datos de test.

Con las etiquetas originales disponibles para el conjunto de datos de test y las predicciones obtenidas podemos calcular la precisión, sensibilidad y especificidad, comprobando si son razonablemente parecidos a los que habíamos obtenido en la fase de entrenamiento. Como en el caso anterior, un estudio más completo de la función y sus parámetros puede encontrarse en [70].

### 5.3.2.2.6 Validación en el escenario

El proceso concluye con la utilización del modelo creado dentro del escenario completo que simulamos, tal y como se describe en la Figura 38. Aunque se mantienen parámetros como los tipos de trayectorias, el número de sensores y todos los parámetros considerados de inicialización, en este caso la probabilidad de aparición del blanco se establecerá a un valor determinado, así como la longitud de la ejecución. El número de iteraciones debe ser finito y de un valor adecuado para dar lugar a una cantidad de datos suficientemente grande.



**Figura 38.** Proceso de validación de datos en el escenario completo.

Un aspecto a considerar en referencia a la generación de los datos de validación es la longitud de las ráfagas de datos. De este modo, la longitud de las franjas en azul sigue siendo fija (20) porque corresponden con los saltos de una trayectoria completa, es decir, presencia de blanco. Y los tramos en rojo, correspondientes a la ausencia de blanco varían su longitud en función del número de casos que transcurren hasta la aparición de un nuevo blanco.

Con todos esos datos generados y gracias a la función de la librería SVM `svmpredict`, tendremos de nuevo las predicciones de la SVM y por supuesto las etiquetas originales, que nos permitirán obtener los parámetros de precisión, sensibilidad y especificidad.

De esta forma, al final de las tareas de generación, entrenamiento, test y validación del modelo, podremos comparar el comportamiento del detector global que nos proporciona la SVM para distintos conjuntos de datos. Si el modelo trabaja adecuadamente y no se sobreajusta, los valores de precisión, sensibilidad y especificidad no deberían diferir mucho de unos conjuntos a otros.

En esta última fase todos los datos generados para la validación, incluyendo las trayectorias originales de los blancos que van apareciendo y las decisiones locales de cada sensor en cada instante, son almacenados para ser utilizados posteriormente en la regresión. Podríamos considerar que todos esos datos son enviados de forma independiente al centro de fusión, que se encargará de reconstruir las posiciones del blanco, en el caso de que exista, en cada instante de tiempo.

A la hora de realizar las simulaciones, el proceso descrito, incluyendo la generación del modelo, debería repetirse un número determinado de veces con el fin de comprobar la consistencia de los resultados.

## 5.4 Estimación

En esta sección se estudia cómo, una vez que el centro de fusión ha recolectado los datos de los sensores y ha determinado la presencia de un blanco, se produce la inferencia de su posición. Esto es, cómo se trabaja con la información que recibe el centro de fusión y se trata de reconstruir las trayectorias que siguen los blancos. Analizaremos la calidad del estimador y comprobaremos cómo evolucionan el error, el sesgo y la varianza conforme se modifican las probabilidades de error toleradas y el número de sensores.

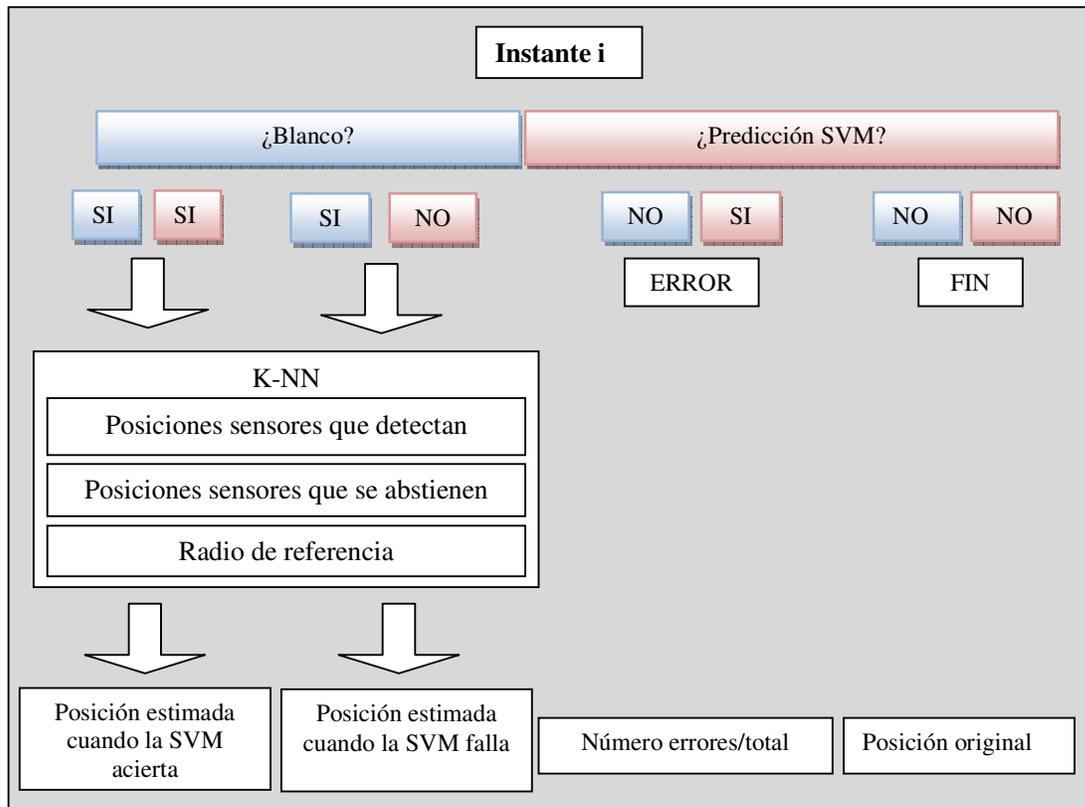
Aunque lo que compararemos son las trayectorias, el trabajo se realiza posición a posición, instante a instante, apoyándonos en los datos disponibles, principalmente en la etiqueta original y en la predicción de la SVM para conseguir estimar, en cada iteración, una posición que nos permita recrear posteriormente las trayectorias de forma completa.

El proceso para cada posición, que se describe en la Figura 39, considera una serie de condicionantes que se listan a continuación, expuestos de derecha a izquierda en la misma:

- Si no hay blanco y la SVM así lo considera, no hay tarea ni posición que estimar.

- Si la etiqueta original es de ausencia de blanco, pero la SVM considera que este sí existe, se considerará el caso como un error.
- Si por el contrario, la etiqueta original es de presencia de blanco y la predicción de la SVM no detectaba blanco, la posición va a estimarse en base a las detecciones locales para poder comprender por qué la detección global ha sido incorrecta.
- En el caso de que la predicción SVM y la situación real coincidan en que hay presencia de blanco, se estimará la posición considerando, dos conjuntos de posiciones para cada iteración. Uno de ellos correspondiente a todos los sensores que detectaron el blanco y otro con todos los sensores que se abstuvieron, ponderando adecuadamente cada uno de los conjuntos en la medida final.

Conviene aclarar que, la Figura 39 recorre de forma gráfica todas las opciones contestando a las dos preguntas situadas en la parte alta: ¿Hay blanco? ¿Cuál es la predicción de la SVM? Cada respuesta se proporciona en un cuadro con el color de la pregunta.



**Figura 39.** Proceso de regresión en función de la detección del blanco y la situación real.

El hecho de comprobar cómo sería la posición estimada en los casos en los que originalmente hay blanco pero la SVM predice que no hay blanco se justifica al considerar que el blanco no puede desaparecer del área monitorizada en medio de una trayectoria. Por tanto, es preferible estimar una posición si conocemos que el blanco está en el área (aunque

no sea muy fiable), que no estimarla. La parte fundamental de la tarea es la que se lleva a cabo por parte del algoritmo de estimación derivado del K-NN, que se detalla a continuación.

### 5.4.1 Método escogido para la estimación de la posición

Antes de comenzar a describir el método utilizado, conviene recordar que el objetivo principal es estimar una posición en base a las posiciones de los sensores que detectan y se abstienen. Para ello vamos a partir de un algoritmo que nos permite agrupar a todas las contribuciones, el K-NN. Este simple algoritmo de aprendizaje se corresponde con las siglas en inglés de “K-Nearest Neighbors”, y se define típicamente como una herramienta para tareas de clasificación y regresión local, que agrupa los K ejemplos de entrenamiento más cercanos. Como puede verse en la Figura 40, el primer paso consiste en considerar que:

- Todos los sensores que detecten el blanco (verde) tienen el mismo peso sobre la ponderación (1).
- Todos aquellos que se abstienen (naranja), contribuirán a la posición estimada con un peso menor,  $\alpha$ , (por ejemplo 0,5).
- Los sensores que no detecten contribuirán con peso 0, es decir, no contribuirán.

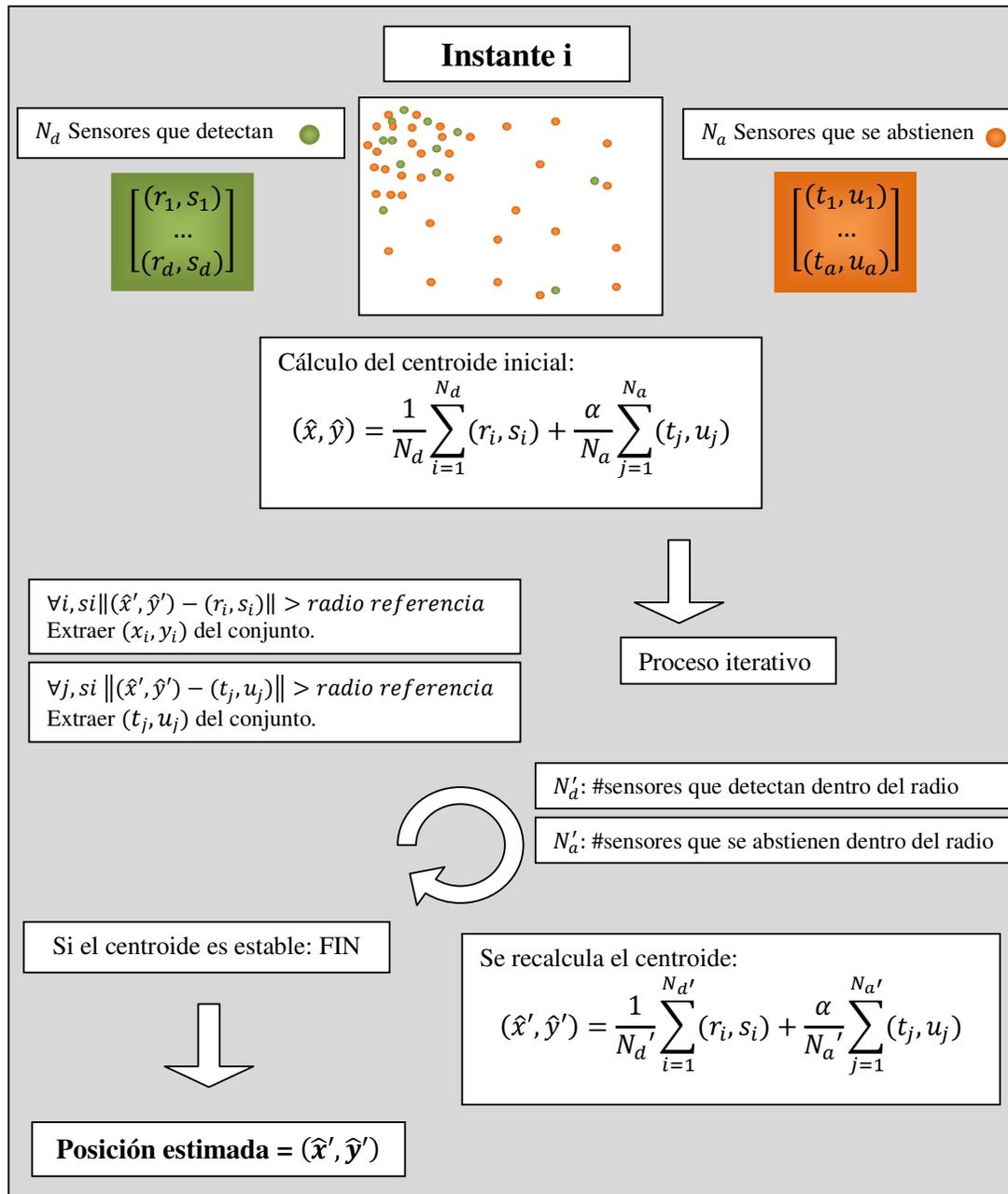
Matemáticamente se formula como sigue:

$$(\hat{x}, \hat{y}) = \frac{1}{N_d} \sum_{i=1}^{N_d} (r_i, s_i) + \frac{\alpha}{N_a} \sum_{j=1}^{N_a} (t_j, u_j) \quad (5.64)$$

Donde  $N_d$  es el número de sensores que detectan,  $N_a$  el número de sensores que se abstienen dentro del radio y  $(r_i, s_i)$  y  $(t_j, u_j)$  sus respectivas posiciones.

Si nos concentramos en la Figura 40, después de calcular esa posición inicial, se trata de mejorar la estimación de forma iterativa, considerando para el cálculo del nuevo centroide sólo a los sensores más cercanos. Para establecer qué sensores son considerados como vecinos, es necesaria una distancia de referencia. Esa distancia es un parámetro variable de nuestro experimento y vamos a asociarlo al radio de alcance estudiado en el apartado 5.3.1.4. Cabe recordar que este radio depende directamente de la probabilidad de error tolerada y del nivel de ruido.

En esa búsqueda iterativa del centroide, se escoge como condición de parada la estabilidad en el número de sensores dentro del citado radio de acción, esto es, que no se elimine ningún sensor del conjunto de los considerados vecinos en la iteración previa. En el momento en el que el centroide es estable, este se considera como posición estimada. Todo el proceso se repite cada vez que la etiqueta real indica presencia de blanco.



**Figura 40.** Proceso interno del K-NN. Estimación de una posición a partir de dos bloques de sensores y un radio de referencia.

En función de los resultados, existen varias modificaciones que podrían suponer mejoras a los mismos. Varias de ellas se listan a continuación:

- 1) Considerar sólo los sensores que detectan blanco, ignorando la contribución que podrían realizar los que se abstienen.

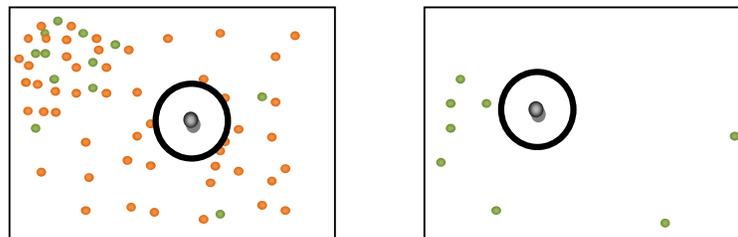
- 2) Eliminar sistemáticamente del conjunto de sensores con el que calculamos el centroide aquel que diste más del mismo, esto es, el vecino más lejano. Esto nos ayudaría a dejar de considerar los datos atípicos o detecciones lejanas al blanco.
- 3) Considerar que los pesos de las contribuciones de los sensores que han detectado y que se abstuvieron sea proporcional a la distancia al centroide, en lugar de considerar pesos fijos.
- 4) Establecer la distancia de referencia en base a múltiplos o fracciones del citado radio de alcance.
- 5) Modificar la condición de parada del algoritmo, de modo que, en lugar de que el número de sensores sea estable, considere un número de sensores mínimo (por ejemplo, un porcentaje del total). De este modo nos aseguraremos de que la posición se estima en base a varios sensores, disminuyendo el riesgo de eliminar sistemáticamente a todos los sensores menos el último.

En el escenario de un sólo umbral no habrá abstenciones, y por tanto ninguna posición en dicho conjunto, dejando el cálculo del centroide en función exclusivamente de los sensores que detectan localmente al blanco, tal y como indica la modificación 1). Salvo por este detalle, el mecanismo del algoritmo es exactamente el mismo.

## 5.4.2 Modificaciones al método

Para el segundo escenario en el que consideramos la abstención, vamos a desarrollar una variante del método en previsión de problemas que podamos encontrar, como el detallado a continuación.

Si se considera a todos los sensores en la primera estimación, esta posición inicial puede llegar a condicionar por completo el rendimiento del algoritmo. Dada la gran cantidad de sensores que se abstienen y la distribución de los sensores, la posición estimada puede no estar cerca de donde existe la mayoría de los sensores, provocando incluso que el número de sensores contenidos en el radio sea cero. El problema podría corresponderse con cualquiera de las situaciones gráficamente expuestas en la Figura 41, donde los puntos verdes son detecciones, los puntos naranjas abstenciones y el negro el centroide, acompañado de su radio de acción.



**Figura 41.** Caso de no estimación a pesar de detección correcta.

Para tratar de evitar este comportamiento se proponen las siguientes modificaciones al algoritmo:

- Rectificar la evolución del algoritmo para eliminar del cálculo iterativo del centroide las posiciones más lejanas a este, estén o no dentro del radio de acción.
- Modificar la condición de parada. Originalmente el radio de acción sólo se considera para la condición de parada: el procedimiento se detiene cuando el número de sensores permanece estable en dos iteraciones consecutivas. En la versión propuesta, el algoritmo se detiene cuando el número de sensores contenidos dentro del radio de alcance es menor o igual que el 10% de todos los sensores desplegados.

Por último, añadiremos otra modificación para poder comparar los resultados obtenidos estableciendo los pesos homogéneos, frente al caso en el que no se consideran limitaciones en la comunicación. Esta alternativa supone que es posible cuantificar los niveles de potencia recibida, de forma que los pesos no homogéneos se obtienen como sigue:

$$w_i = \frac{(P_i - \gamma_{\min})}{(\gamma_{\max} - \gamma_{\min})} \quad (5.65)$$

Donde  $P_i$  es la potencia recibida por el sensor y:

$$\begin{aligned} \gamma_{\min} &= \min\{\gamma_{fa}, \gamma_{nd}\} \\ \gamma_{\max} &= \max\{\gamma_{fa}, \gamma_{nd}\} \end{aligned} \quad (5.66)$$

En consecuencia la estimación de la posición se ve modificada resultando ahora:

$$(\hat{x}', \hat{y}') = \frac{1}{N_{a'}} \sum_{i=1}^{N_{a'}} (r_i, s_i) + \frac{1}{N_{a'}} \sum_{j=1}^{N_{a'}} w_j (t_j, u_j) \quad (5.67)$$

### 5.4.3 Error, esperanza y sesgo

La medida de calidad que usaremos para evaluar el rendimiento del estimador será el error cuadrático medio (MSE). Si definimos las posiciones reales y estimadas como:

$$\mathbf{x} = [x_1, x_2]^T \quad (5.68)$$

$$\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2]^T \quad (5.69)$$

Entonces, el MSE se obtiene como:

$$\text{MSE}(\hat{\mathbf{x}}) = E\{(\hat{\mathbf{x}} - \mathbf{x})^T (\hat{\mathbf{x}} - \mathbf{x})\} \quad (5.70)$$

Que suele descomponerse de la siguiente manera:

$$\text{MSE}(\hat{\mathbf{x}}) = b(\hat{\mathbf{x}})^T b(\hat{\mathbf{x}}) + \text{Var}\{\hat{\mathbf{x}}\} \quad (5.71)$$

Donde  $b(\hat{\mathbf{x}})$  se corresponde con el sesgo del estimador, que nos da la medida de error sistemático:

$$\begin{aligned} b(\hat{\mathbf{x}}) &= E\{(\hat{\mathbf{x}} - \mathbf{x})\} \\ &= E\{\hat{\mathbf{x}}\} - \mathbf{x} \end{aligned} \quad (5.72)$$

Si este fuese nulo, nuestro estimador sería insesgado. En el caso contrario, esta medida nos permitiría rectificar nuestro estimador para aproximarle al caso real restándole el sesgo (en el caso de ser conocido).

Por otra parte, nos apoyaremos en la varianza, formulada tal que:

$$\begin{aligned} \text{Var}\{\hat{\mathbf{x}}\} &= E\{(\hat{\mathbf{x}} - E\{\hat{\mathbf{x}}\})^T (\hat{\mathbf{x}} - E\{\hat{\mathbf{x}}\})\} \\ &= E\{\hat{\mathbf{x}}^T \hat{\mathbf{x}}\} - E\{\hat{\mathbf{x}}^T\}E\{\hat{\mathbf{x}}\} \end{aligned} \quad (5.73)$$

La varianza nos proporciona una medida de la dispersión del error. Obviamente es deseable una menor varianza, que se traduciría en una mayor calidad del estimador. Típicamente el objetivo es encontrar un estimador con el menor sesgo posible y la menor varianza posible. Sin embargo, en ocasiones existe una relación tal entre el sesgo y la varianza que relajar las condiciones en el sesgo podría proporcionarnos valores menores de la varianza.

## 5.5 Discusión

A lo largo de este capítulo hemos desgranado las claves del desarrollo práctico que vamos a llevar a cabo para extraer los resultados. Se describen dos escenarios, uno más sencillo donde se establecen un nivel de potencia en función de la probabilidad de falsa alarma, que marcará el umbral de detección, y otro más complejo donde se introduce un segundo nivel de potencia sujeto a la probabilidad de no detección. Entre ambos umbrales se establece una zona de incertidumbre donde el sensor local se abstiene de votar, proporcionando al centro de fusión de este modo una tercera categoría de datos.

El escenario está sujeto a diversas características fijas, como son:

- Área de monitorización.
- Trayectorias tipo.

## Capítulo 5: Descripción de los métodos usados en Detección/Estimación

- Constante de pérdidas en la transmisión de potencia.
- Nivel de potencia de referencia a un metro.
- Relación señal a ruido.
- El numero de sensores desplegados en el área.

De estos parámetros fijos vamos a extraer, de forma más o menos directa, otros parámetros como son:

- Radios de alcance.
- Varianza del ruido.

Con todo lo anterior, el primer paso en la fase de detección es establecer los umbrales de potencia para cada detector local, apoyándonos en el criterio de Neyman-Pearson y fijando para cada uno de forma alternativa los valores de:

- La probabilidad de falsa alarma.
- La probabilidad de no detección.

Con el estimador local diseñado, la regla de detección global se basa en máquinas de vectores soporte. En este caso se debe escoger el Kernel de la SVM y optimizar sus parámetros con el objetivo de maximizar la precisión.

A partir de los datos y los parámetros se evalúa la respuesta del modelo proporcionado por la máquina de vectores soporte en términos de precisión, sensibilidad y especificidad, obtenidos para distintos conjuntos de datos de entrenamiento.

Fruto de la validación del escenario, dispondremos de datos para la estimación, una tarea con la que completar el proyecto y para la cual se sugiere un método de regresión derivado del K-NN. Calibraremos el comportamiento del estimador obtenido en función del error cuadrático medio (MSE), que puede verse como una combinación de su sesgo y su varianza.

# Capítulo 6

## Resultados

### 6.1 Introducción

Llega el momento de aportar resultados, números, así como de exponer las dificultades encontradas y las soluciones propuestas, poniendo especial atención en dos parámetros fundamentales: el número de sensores desplegados en la red y las probabilidades de falsa alarma y no detección toleradas por el sistema.

El primer bloque de resultados, que se encuentra en la Sección 6.2, recoge los aspectos comunes a los dos escenarios que se han desarrollado (el área y los sensores, la distribución de las señales y el “Kernel” escogido).

En las Secciones 6.3 y 6.4 se muestran los resultados obtenidos en el despliegue de cada uno de los escenarios: toda la información relacionada con los umbrales y las reglas de decisión locales. A partir de estos umbrales se analizan, en la Sección 6.5, las trayectorias tipo y las proporciones de datos generados para los experimentos.

Más adelante, en la Sección 6.6, se muestran los resultados del detector global en términos de precisión, sensibilidad y especificidad para unas probabilidades de error establecidas. Como complemento, se aportan en la Sección 6.7, más resultados del detector global, obtenidos sustituyendo la función de “Kernel” Lineal utilizada en la SVM por una Gaussiana.

Por último, la Sección 6.8 está dedicada al análisis del estimador obtenido para cada uno de los escenarios tratados: con abstención y sin abstención. Dicha sección se completa con una comparativa para el caso en el que no existen limitaciones en la cantidad de información transmitida y con dos análisis más sobre la evolución de los estimadores en función de la probabilidad de error y del radio de referencia.

## 6.2 Aspectos comunes a los dos escenarios

Vamos a considerar como primer escenario el correspondiente al clasificador binario, y como segundo escenario el que considera la región de incertidumbre. Sin embargo, muchos elementos de ambos son comunes. El área monitorizada es la misma y los sensores desplegados también. Lo mismo ocurre con la distribución de las señales y con el núcleo de la máquina de vectores soporte (SVM). En esta sección se describen todos esos elementos comunes.

### 6.2.1 El área y los sensores

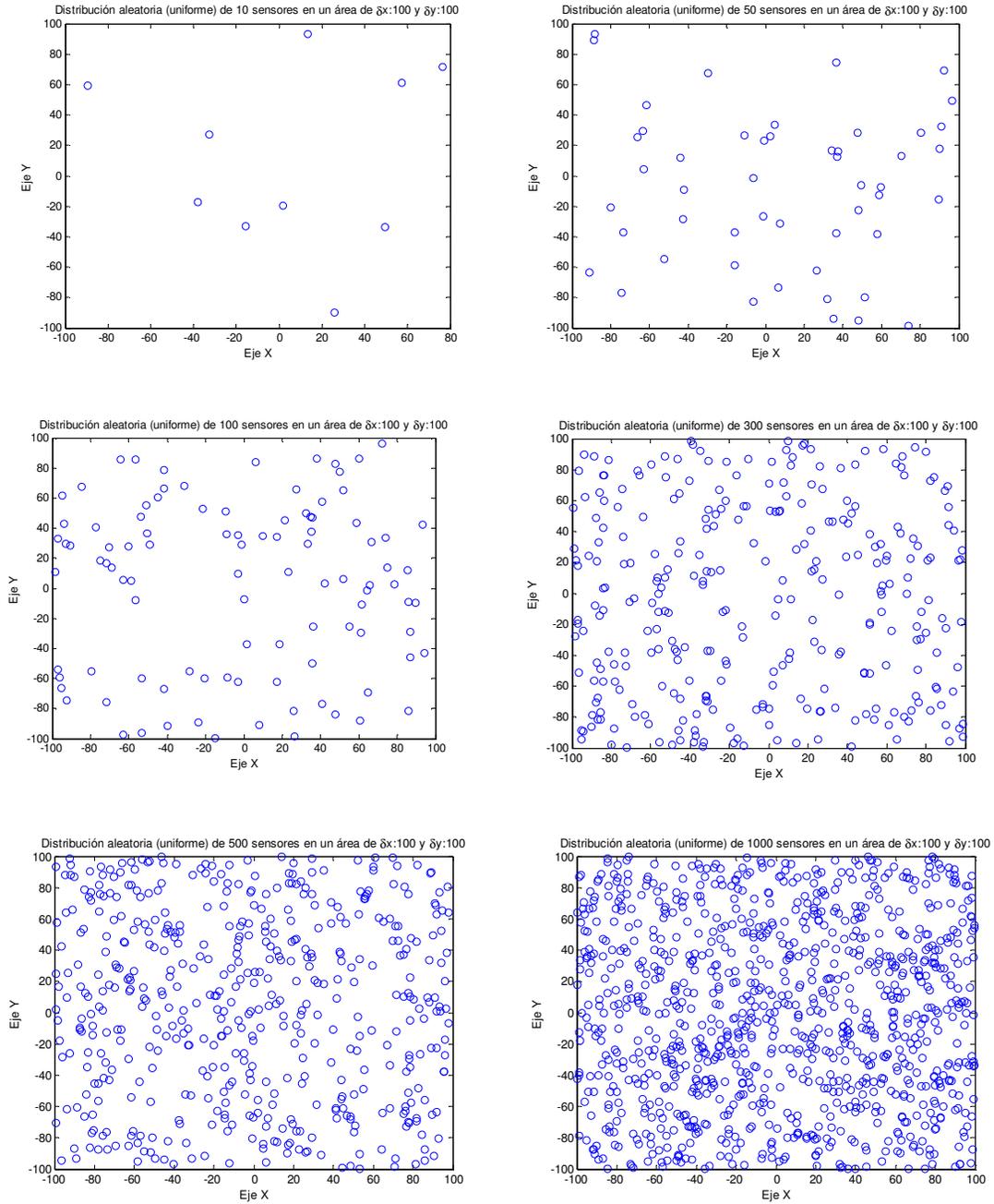
A lo largo de todo el proyecto va a considerarse un área monitorizada de dimensiones fijas. Haciendo coincidir el centro de coordenadas con el centro del área monitorizada, estas dimensiones son de 200m x 200m (para un total de 40000 m<sup>2</sup> de superficie).

El número de sensores es un parámetro fundamental, pues nos va a permitir comprobar cómo las reglas de decisión extraídas del aprendizaje mejoran conforme aumenta el número de fuentes de información, en este caso, el número de sensores. Concretamente van a considerarse los siguientes valores: 10, 50, 100, 300, 500 y 1000.

En la Figura 42 se muestra un ejemplo de la distribución sobre el área de los sensores en cada caso. Como puede verse, a medida que la cantidad de sensores aumenta estos cubren una porción mayor de área total de un modo más uniforme, siendo obvio el incremento en la densidad de sensores. Esto va a suponer que el centro de fusión dispondrá de más información local, mejorando sus prestaciones a la hora de detectar de modo global.

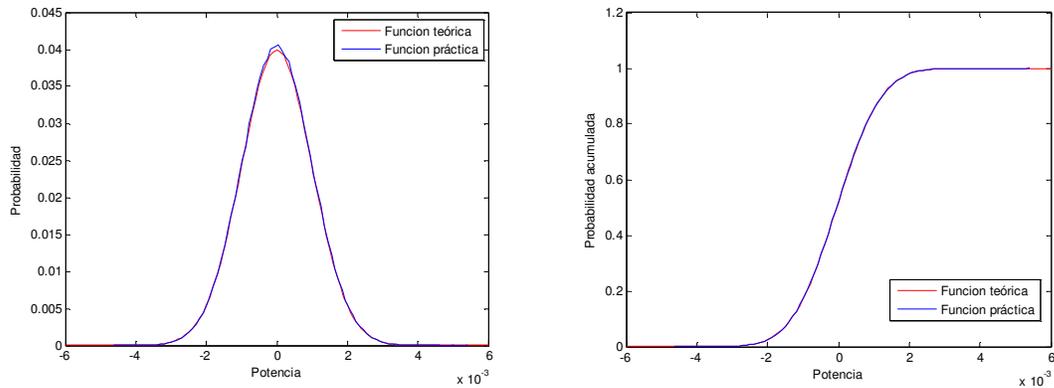
### 6.2.2 Distribuciones de las señales

Como se explicaba en la Sección 5.2.1.1, se han considerado dos hipótesis acerca de la potencia recibida por cada sensor: una primera en la que consideramos ausencia de blanco (se considera únicamente el ruido) y otra segunda fruto de la combinación de la potencia recibida en función de la distancia al blanco presente y el ruido.



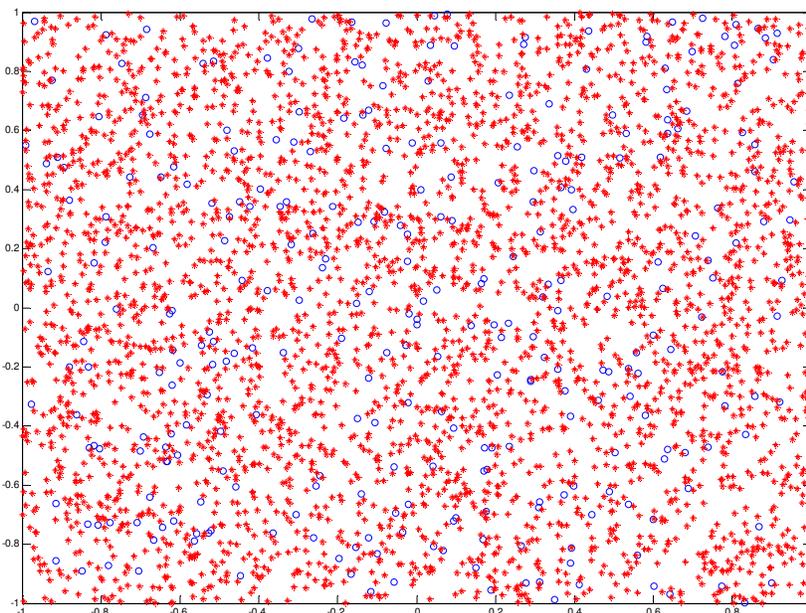
**Figura 42.** Distribución aleatoria de 10, 50, 100, 300, 500 y 1000 sensores sobre el área.

Para calcular las funciones de densidad y de distribución de probabilidad de la potencia del ruido las muestras no dependen de la distancia y la forma en que se producen es conocida: una Gaussiana centrada en cero y con una varianza establecida adecuadamente en base a la SNR para dar forma a un escenario coherente. Apoyándonos en un simple histograma y sumando las probabilidades obtenemos la Figura 43, en la que se puede apreciar la coincidencia entre la fórmula teórica (5.3) y la estimación empírica.



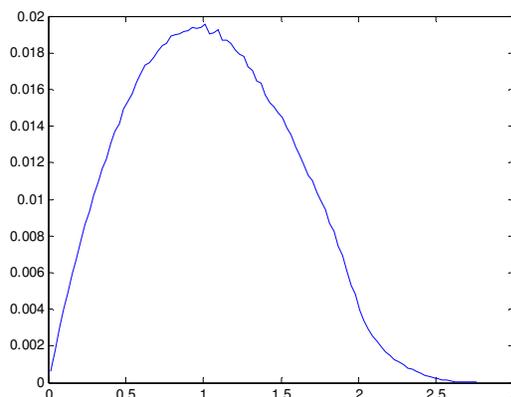
**Figura 43.** Funciones de densidad de probabilidad (izquierda) y de distribución de probabilidad (derecha) de la potencia de ruido. Teórica (rojo) frente a empírica (azul).

Bajo la segunda hipótesis considerada, el sensor recibe una potencia que depende del ruido y de la distancia que lo separa del blanco, tal y como expresa (5. 6). En este caso, para generar las muestras tomaremos las posiciones de los sensores como establecidas por el usuario (fijas) y aleatoriamente se establecen las posiciones de los puntos de medida, dando origen a un escenario como el mostrado en la Figura 44, donde se exponen 300 sensores posicionados aleatoriamente (en azul) y  $\lfloor 10^6/300 \rfloor$  posiciones de medida, también escogidas aleatoriamente.



**Figura 44.** Distribución de los 300 sensores en el área (azul) y los puntos aleatorios para el cálculo empírico de las distribuciones (rojo).

En la Figura 45 se representa la función de densidad de probabilidad de las distancias, donde puede apreciarse como las medidas están normalizadas y observarse que la mayor de las distancias consideradas es aproximadamente 2,8. Este resultado es coherente con la máxima distancia teórica,  $\sqrt{2^2 + 2^2} = \sqrt{8} \approx 2,828$ , y equivaldría a tener el blanco y el sensor en esquinas opuestas. Nótese que en este caso no se dispone de una FDP teórica con la que comparar.



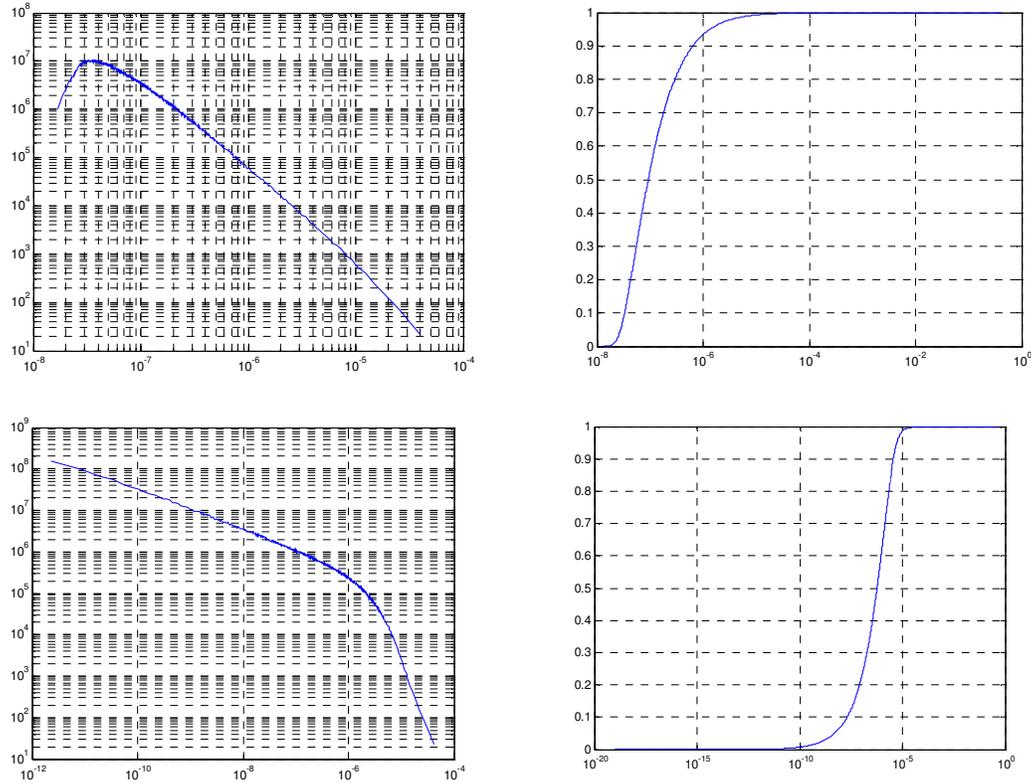
**Figura 45.** Función de densidad de probabilidad de la distancia.

A partir de la distribución de las distancias, siguiendo un procedimiento análogo al realizado para el caso del ruido, se obtienen las funciones de densidad y de distribución de probabilidad de la potencia que se muestran en la Figura 46 para los dos escenarios siguientes:

- 1) Potencia recibida por el sensor sin mezclar con ruido.
- 2) Potencia recibida por el sensor mezclada con ruido.

Si analizamos las gráficas de la izquierda en la Figura 46, correspondientes a la función de densidad de probabilidad, podemos comprobar cómo ambos ejes están representados en escala logarítmica. Este paso es necesario, puesto que la mayoría de las medidas corresponden a potencias similares y todas de muy bajo valor, pero existen otras distancias menores que aportan valores de potencia mucho más elevado, complicando la representación en una escala lineal.

Si nos concentramos en el caso sin ruido, en la parte superior, podríamos decir que los niveles de potencia más probables se encuentran entre  $2,6 \cdot 10^{-8}$  y  $6,3 \cdot 10^{-8}$  W. Además, se aprecia cómo, a medida que aumenta la potencia, su probabilidad decae linealmente en escala logarítmica, esto es, la probabilidad realmente decae de modo exponencial. El hecho de tener una distancia máxima (la diagonal del área considerada), provoca la existencia de un valor mínimo de potencia por debajo del cual la probabilidad es 0, este valor aproximado es  $1,6 \cdot 10^{-8}$ .



**Figura 46.** Funciones de densidad de probabilidad (izquierda) y de distribución de probabilidad (derecha) de la potencia recibida en función de la distancia para ausencia (arriba) y presencia (abajo) de ruido.

Si por el contrario analizamos el caso con ruido, podemos comprobar que la forma cambia, sobre todo para valores pequeños de la potencia. En esta función de densidad, no existe un mínimo de potencia acotado por la distancia máxima, y la probabilidad de las medidas de ruido es más elevada que las asociadas a la presencia de blanco. El cambio en la pendiente coincide con  $10^{-6}$ , el valor de la varianza del ruido, e indica que para valores de potencia superiores a éste, la contribución de la presencia del blanco empieza a ser muy superior al ruido.

En las gráficas de la derecha de la Figura 46, donde sólo se aplica la escala logarítmica a la coordenada x, se representan las funciones de distribución de probabilidad, que nos confirman las conclusiones obtenidas previamente:

- En el caso sin ruido la probabilidad acumulada empieza a crecer a partir de  $10^{-8}$ , como consecuencia de esa distancia máxima. Además, la mayoría de las medidas se concentran en un rango aproximado de tres décadas  $[2 \cdot 10^{-8}, 2 \cdot 10^{-5}]$ .
- En el caso que considera la combinación de las dos fuentes de potencia, la probabilidad acumulada empieza a crecer aproximadamente a partir de  $10^{-21}$ , concentrándose la mayoría de las potencias en un rango aproximado de 6 décadas,  $[2 \cdot 10^{-11}, 2 \cdot 10^{-5}]$ .

### 6.2.3 El Kernel

El último punto de coincidencia entre los dos escenarios es la función del “Kernel” que vamos a utilizar en la SVM. Inicialmente se utilizará el “Kernel” lineal, y la elección del mismo se justifica en base al número de características y muestras disponibles. En nuestro caso esta cantidad es suficientemente elevada como para que no sea a priori necesario trasladar los datos a una dimensión mayor. En cualquier caso, posteriormente se considera el “Kernel” Gaussiano en ambos escenarios, especialmente para situaciones donde el número de sensores es pequeño.

## 6.3 Despliegue del primer escenario: clasificador binario

En este caso el detector que hemos implementado en primer lugar, denominado también clasificador binario, ha fijado el umbral de detección a partir de un nivel preestablecido de probabilidad de falsa alarma. Para poder comparar cómo afecta la variación de dicha probabilidad al comportamiento del sistema se alternarán a lo largo de todo el proyecto dos valores:  $P_{fa} = 0,1$  y  $P_{fa} = 0,01$ .

De la sección 5.3.1.2, recuérdese que la probabilidad de falsa alarma y el umbral de potencia vienen relacionadas por (5. 35):

$$P_{fa} = \operatorname{erfc} \left( \sqrt{\frac{\gamma_{fa}}{2\sigma_w^2}} \right)$$

Sin más que igualar ese nivel de potencia a la fórmula que relaciona la potencia recibida en función de la distancia, podemos obtener una medida del radio de influencia del blanco, tal y como se muestra en la Tabla 5. Hemos de recalcar que la potencia que influye en el cálculo de este umbral es sólo fruto del ruido y para nada depende de la distancia.

Probabilidad de falsa alarma	Umbral único (mW)	Radio detección(m)
0,1	$2,705 \cdot 10^{-3}$	19,225
0,01	$6,635 \cdot 10^{-3}$	12,277

**Tabla 5.** Umbrales de potencia y radios de detección en función de la probabilidad de falsa alarma  $P_{fa}$ .

Como puede verse, una restricción en el número de falsas alarmas se traduce en un aumento de la potencia necesaria para detectar la presencia del blanco, buscando mayor seguridad. De forma indirecta esto se traduce en un radio menor de alcance.

## 6.4 Despliegue del segundo escenario: considerando abstención

Ofrecerle la posibilidad a cada sensor de abstenerse si considera por sí mismo que su información no es útil es una manera alternativa de introducir una opción más en el clasificador local. El hecho de tener 3 regiones provoca la necesidad de tener dos fronteras y por tanto dos criterios para encontrarlas. Para obtener el primero de los umbrales se procederá como en el caso previo, calculándolo a partir de un determinado nivel de probabilidad de falsa alarma. Para encontrar el segundo umbral vamos a fijar la probabilidad de no detección que, de forma análoga, tomará a lo largo de todo el proyecto los siguientes valores:  $P_{nd} = 0,1$  y  $P_{nd} = 0,01$ .

Revisando la Sección 5.3.1.3 vemos que la relación existente entre la probabilidad de no detección y el umbral de potencia viene dada por (5. 39):

$$P_{no\ deteccion} = \int_0^{\gamma_{nd}} \int_0^{\infty} f_1(z_1|l) f_L(l) dl dz_1$$

Donde  $f_L(l)$  es la función de distribución de la distancia entre el sensor y el blanco, que hemos aproximado como un sumatorio de deltas dada su complejidad, llegando finalmente a formular la probabilidad de no detección como (5. 46):

$$P_{nd} \approx \frac{1}{2N} \sum_{i=1}^N \left[ \operatorname{erf} \left( \frac{\sqrt{\gamma_{nd}} + x(l_i)}{\sqrt{2\sigma_w^2}} \right) + \operatorname{erf} \left( \frac{\sqrt{\gamma_{nd}} - x(l_i)}{\sqrt{2\sigma_w^2}} \right) \right]$$

Como esta ecuación no permitía resolución analítica de forma trivial, a diferencia del caso de la probabilidad de falsa alarma, nos apoyamos en un procedimiento empírico para encontrar el segundo umbral. Concretamente, buscaremos el mínimo de la siguiente función (5. 48):

$$J = \left[ P_{nd} - \frac{1}{2N} \sum_{i=1}^N \left[ \operatorname{erf} \left( \frac{\sqrt{\gamma_{nd}} + x(l_i)}{\sqrt{2\sigma_w^2}} \right) + \operatorname{erf} \left( \frac{\sqrt{\gamma_{nd}} - x(l_i)}{\sqrt{2\sigma_w^2}} \right) \right] \right]^2$$

Como ejemplo, en la Figura 47 y en la Figura 48, se representa  $J$  para los dos valores de probabilidad establecidos.

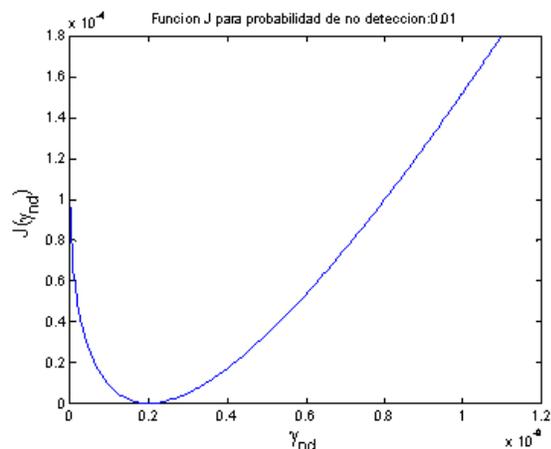


Figura 47. Representación de la función  $J$  para  $P_{nd} = 0,01$ .

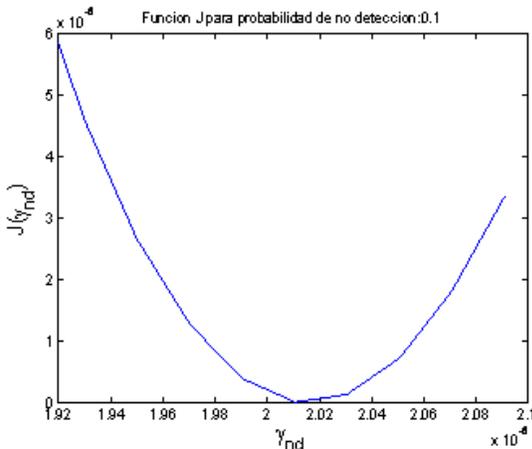


Figura 48. Representación de la función  $J$  para  $P_{nd} = 0,1$ .

La búsqueda del mínimo, dada la forma cuadrática de la función, se inició apoyándose en un algoritmo de descenso de gradiente. Sin embargo, probablemente debido a la magnitud de los resultados, se encontraron dificultades numéricas al tratar de resolver este problema. En consecuencia, se recurrió a encontrar el mínimo mediante una búsqueda de rejilla, recorriendo inicialmente un rango exponencial  $[10^{-12}, 10^{-11}, \dots, 10^{-7}, 10^{-6}]$  y profundizando en un rango más fino en torno al valor obtenido como mínimo. Esta búsqueda nos proporciona un umbral de potencia, que llamamos umbral de no detección que se lista en función de la probabilidad de no detección en la Tabla 6:

Prob. No detección	$\gamma_{nd}$
0,1	$2,011 \cdot 10^{-8}$
0,01	$2,011 \cdot 10^{-10}$

Tabla 6. Mínimos de la función  $J$  para distintos valores de probabilidad de no detección  $P_{nd}$ .

Por su parte, la Tabla 7 muestra, para valores idénticos de probabilidad de falsa alarma y no detección, los dos umbrales de potencia calculados previamente y los radios equivalentes:

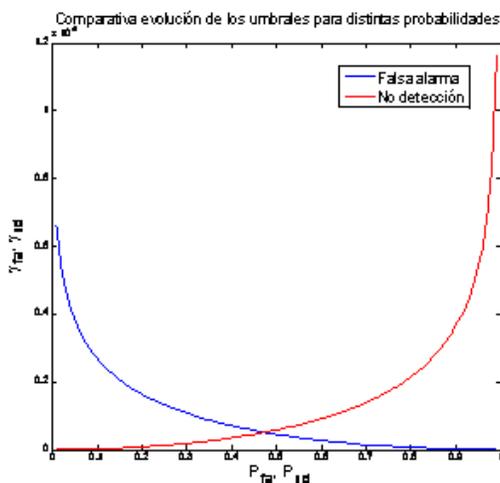
Prob. Falsa alarma Prob. No detección	Umbral no detección (mW)	Umbral detección(mW)	Radio no detección(m)	Radio detección(m)
0,1	$2,011 \cdot 10^{-5}$	$2,706 \cdot 10^{-3}$	223,022	19,225
0,01	$2,011 \cdot 10^{-7}$	$6,635 \cdot 10^{-3}$	2229,944	12,277

**Tabla 7.** Umbrales de potencia y distancias equivalentes.

El hecho de que el umbral de no detección sea tan bajo hace presagiar que muchos de los datos estarán clasificados en la zona de abstención. Un dato relevante es que el radio equivalente en metros a partir del cual consideramos certera la no detección, es dos o tres órdenes de magnitud superior al radio descrito en el primer escenario, obtenido a partir de la probabilidad de falsa alarma y que permanece como radio de detección.

Para completar el estudio en la Figura 49, se muestra la evolución de los umbrales en función de distintos valores de las probabilidades de falsa alarma o no detección, siempre manteniendo constante la varianza del ruido.

En el punto de cruce, para valores de probabilidad de falsa alarma y no detección de 0,47 aproximadamente los dos umbrales coincidirían, tomando un valor de potencia aproximado de  $5 \cdot 10^{-4} mW$ . Este valor para el umbral de potencia que no resulta práctico en ningún caso, puesto que probabilidades de falsa alarma y no detección tan elevadas no tienen sentido en este tipo de escenarios. Nótese que aunque a lo largo de todo el proyecto se consideran valores iguales para las probabilidades de falsa alarma y no detección, estas podrían escogerse con un valor distinto (por ejemplo:  $P_{fa} = 0,1$  y  $P_{nd} = 0,3$ ).



**Figura 49.** Comparativa umbrales frente a probabilidades de no detección y falsa alarma.

Despliegue del segundo escenario: considerando abstención

Además, tal y como se describió en la Sección 5.3.1.3, existen dos posibilidades, que delimitan la región de incertidumbre por exceso o defecto de información en el sensor, y que se corresponden con:

$$\begin{aligned} \gamma_{nd} &> \gamma_{fa} \\ \gamma_{nd} &< \gamma_{fa} \end{aligned} \tag{6.1}$$

En nuestro caso, la región de incertidumbre hace referencia a los niveles de potencia para los que el sensor va a considerar su información como confusa. La considerará confusa porque no es un nivel de potencia suficientemente bajo como para no ser detectado en caso de presencia de blanco ni suficientemente alto como para corresponderse con una muestra de ruido en caso de ausencia de blanco. Gráficamente, el clasificador queda como se muestra en la Figura 50, en la que puede verse como la región de incertidumbre es mayor para probabilidades menores. Esto es, si las condiciones son más restrictivas en lo concerniente a la tolerancia a errores, la zona donde el rango de potencias para los que el sensor va a abstenerse es mayor.

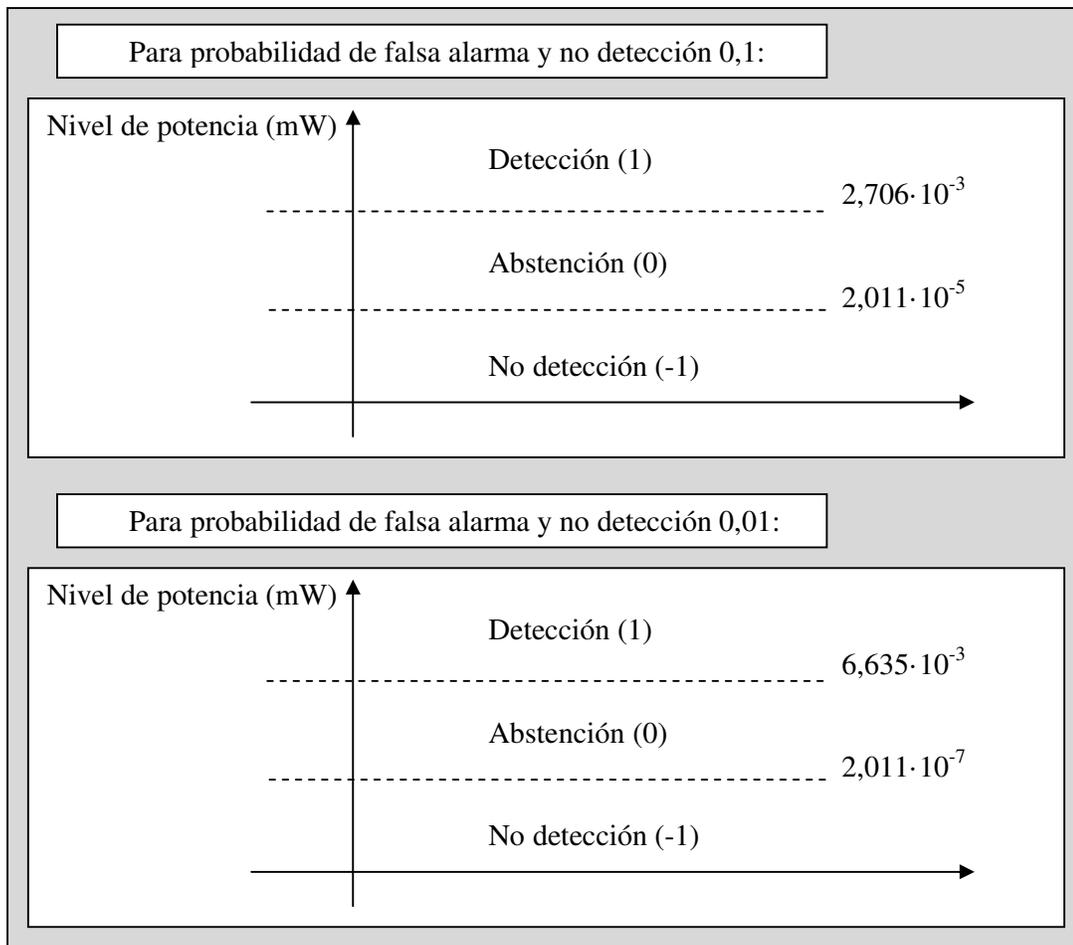


Figura 50. Umbrales para los clasificadores con abstención.

Un hecho significativo es que los resultados obtenidos para los umbrales son los mismos independientemente del número de sensores utilizados, excepto en situaciones con muy pocos sensores (<50), donde pueden variar los decimales sin hacerle el orden de magnitud. Esto es fruto de la distribución no uniforme de los sensores a lo largo de toda el área para una realización concreta cuando existen pocos sensores.

## 6.5 Generando los datos

Conocidos ambos escenarios sólo falta fijar cómo son las trayectorias que va a seguir el blanco cuando aparezca en el área monitorizada. Como se explicaba en la Sección 5.2.2, en este proyecto se han reducido las posibles trayectorias a seis y se ha creado un mecanismo que permite generar datos en ausencia de blanco mediante la imposición de una probabilidad de aparición del blanco igual a 0.

### 6.5.1 Trayectorias tipo

El hecho diferenciador a la hora de decidir es la aparición de un blanco. Por eso, para el entrenamiento se aportan en cantidades idénticas datos generados en presencia y en ausencia de blanco. En el caso de estar generando datos en presencia de blanco, el sistema está diseñado para que todos los tipos de trayectorias que se muestran en la Figura 51 estén presentes en la misma proporción, para evitar cualquier tipo de sobreajuste. Además, los inicios de trayectoria se marcan con un diamante ( $\diamond$ ) y los finales con un asterisco (\*).

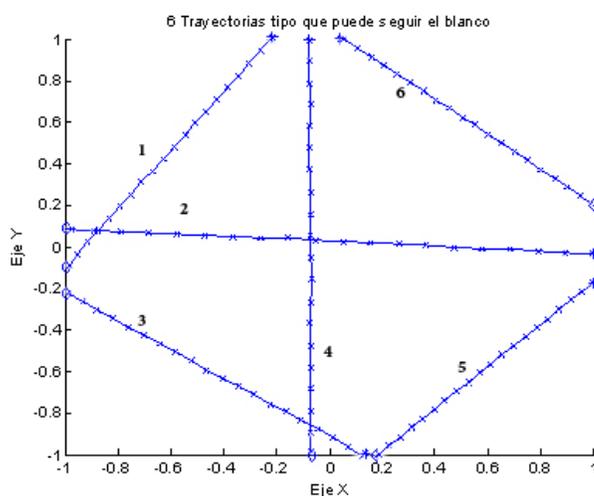


Figura 51. Trayectorias tipo (6) catalogadas en el proyecto.

A modo de recordatorio, se listan las condiciones bajo las que se generan las trayectorias:

- Inicio y final en distinta frontera y en zona próxima al eje:  $\pm 0,25$ .
- Tipo de trayectoria: lineal.
- Numero de saltos del blanco: 19 (20 posiciones).

### 6.5.2 Comparativa de proporciones de datos

El hecho de que balanceemos los datos tratando de construir los conjuntos de entrenamiento para que todas las situaciones estén idénticamente representadas (ausencia y presencia de blanco, con paridad de los distintos tipos de trayectorias), no significa que los sensores detecten o no en la misma proporción. Así, puede ocurrir que aparezca un blanco que sólo sea detectado por el 15 % de los sensores y el 85% restante lo ignore, o puede que en ausencia de blanco un 4% de los sensores detecte blanco. La situación se complica aún más si ofrecemos la posibilidad de abstenerse al sensor.

A continuación se muestra, en la Tabla 8, la proporción de sensores que detectan (%D), no detectan (%ND) o se abstienen (%A), para cada una de las posibles situaciones descritas. Estas proporciones se mantienen para cualquier número de sensores desplegados, variando ligeramente cuando la densidad de sensores es muy baja. Algo lógico, dado que no depende de los mismos, sino de los umbrales, que a su vez no dependen del número de sensores, por ello, se exponen los resultados obtenidos únicamente para el caso de 300 sensores desplegados. Concretamente se estudian dos casos: clasificador binario sin abstención (1) y con abstención (2).

Caso	N=300	Escenario: Ausencia de blanco			Escenario: Presencia de blanco			Escenario: Validación		
		$P_{fa}=P_{nd}$	%D	% ND	%A	%D	% ND	%A	%D	% ND
1	0,10	10,019	89,981	0,000	15,581	84,419	0,000	11,785	88,215	0,000
1	0,01	1,007	98,993	0,000	3,416	96,584	0,000	1,829	98,171	0,000
2	0,10	10,025	11,092	78,883	15,800	9,692	74,508	11,646	10,726	77,628
2	0,01	1,010	1,143	97,847	3,417	1,008	95,575	1,683	1,101	97,216

**Tabla 8.** Distribución real de los datos por clases.

El tamaño de la muestra para calcular las proporciones para situaciones reales de ausencia y presencia de blanco en la Tabla 8 es de: 300 sensores y 6 conjuntos de 20 datos, lo que supone un total de 36000 medidas. En el caso de la validación, la probabilidad de aparición del blanco se establece en 0,02 y los datos sobre los que se calcula la proporción son 500 iteraciones para los 300 sensores, obteniéndose por tanto 150000 muestras. Dado que existe una probabilidad de aparición que puede afectar al número de trayectorias que se completan durante las 500 iteraciones (típicamente entre 9 y 10) ,dentro de este escenario aproximadamente 190 de las 500 iteraciones dan origen a datos generados en presencia de blanco, y entorno a 310 corresponderían a los datos generados en ausencia de

blanco. En ambos casos se promedian los datos a lo largo de 10 repeticiones del experimento completo para una mayor fiabilidad.

El primer detalle de los resultados, que respalda el cálculo de los umbrales es la correspondencia entre los porcentajes de detección y cada probabilidad. Por ejemplo, en el escenario en ausencia de blanco a una probabilidad de detección 0,1 le corresponde proporción aproximada de 10% de detecciones y a una probabilidad de 0,01 le corresponde una proporción del 1%. Lo mismo ocurre para el porcentaje de no detectados en el escenario con abstención en presencia de blanco, probabilidades de no detección de 0,01 y 0,1 producen porcentajes de no detección aproximados del 1% y 10% respectivamente.

Sobre los resultados en sí mismos, podemos concluir que tanto para el escenario 1 (clasificador binario sin abstención) como para el escenario 2 (con abstención) las proporciones de sensores que detectan blanco no se ven modificadas. Esto es lógico, puesto que el umbral que marca la detección es el mismo para ambos escenarios. Indirectamente este resultado implica que el introducir una nueva categoría como es la abstención va a afectar sólo al porcentaje de no detectados, que disminuye. Lo más significativo es la gran cantidad de datos que se agrupan en la categoría de abstención, especialmente cuando la probabilidad de falsa alarma y no detección toman valor 0,01 alcanzando proporciones próximas al 98%. Algo dentro de lo esperado no obstante, puesto que, si se establecen los dos niveles de tolerancia de errores en el 1%, la mayoría de los datos se encuentra en la región de abstención.

Por último, si comparamos también en la Tabla 8 los porcentajes de detectados en los escenarios correspondientes a presencia y ausencia de blanco, como es lógico, cuando existe un blanco, el porcentaje de sensores que detecta se incrementa hasta en un 50% para  $P_{fa} = 0,1$  y  $P_{nd} = 0,1$ , llegando a triplicarse para  $P_{fa} = 0,01$  y  $P_{nd} = 0,01$ .

## 6.6 Precisión, sensibilidad y especificidad

Como ya se indicó en el Capítulo 5, los parámetros que nos sirven para evaluar el rendimiento del modelo global de decisión son la precisión, la sensibilidad y la especificidad. Fruto del proceso de aprendizaje descrito en profundidad en la Sección 5.3.2.2, podemos exponer los resultados obtenidos, para cada conjunto de datos: entrenamiento, test y validación.

El formato en el que se van a expresar los resultados es el siguiente. Para cada uno de los dos escenarios (con abstención y sin abstención) va a fijarse la probabilidad de falsa alarma, y en caso de ser necesario de no detección. Una vez establecidos esos valores, van a ejecutarse las rutinas implementadas para el entrenamiento, el test y la validación del modelo, descritas en las Secciones 5.3.2.2.4, 5.3.2.2.5 y 5.3.2.2.6 respectivamente, para las

cantidades de sensores con las que estamos trabajando (10, 50, 100, 300, 500 y 1000). El proceso va a repetirse 10 veces en todos los casos, para aportar consistencia a los resultados.

### 6.6.1 Comparativa de los dos escenarios para $P_{fa}=P_{nd}=0,01$

Para comenzar el análisis de los resultados, las dos tablas siguientes recogen los datos correspondientes a los escenarios sin abstención y con abstención para valores  $P_{fa} = P_{nd} = 0,01$ , de modo que estamos en la situación más exigente en cuanto a la probabilidad de falsa alarma.

La Tabla 9 muestra (como todas las tablas a partir de esta) una comparativa entre los valores de precisión (P), sensibilidad (S) y especificidad (E) obtenidos para los conjuntos de entrenamiento, test y validación. Adicionalmente, se adjunta una columna a la derecha donde se recoge el coste (C) promediado de las 10 repeticiones obtenido en la fase de validación cruzada de la SVM.

Escenario sin abstención										
	Entrenamiento			Test			Validación			Coste
N	P	S	E	P	S	E	P	S	E	C
10	0,585	0,230	0,940	0,601	0,271	0,930	0,743	0,230	0,923	0,450
50	0,647	0,438	0,855	0,675	0,534	0,816	0,732	0,526	0,824	0,384
100	0,695	0,529	0,861	0,681	0,591	0,771	0,749	0,613	0,811	0,242
300	0,828	0,703	0,954	0,852	0,762	0,942	0,879	0,795	0,922	0,191
500	0,850	0,725	0,975	0,880	0,785	0,975	0,922	0,817	0,975	0,131
1000	0,841	0,681	1,000	0,956	0,925	0,987	0,974	0,937	0,994	0,040

**Tabla 9.** Datos recogidos de precisión (P), sensibilidad (S) y especificidad (E) para entrenamiento, test y validación.  $P_{fa} = 0,01$ .

Por su parte, la Tabla 10, recoge la precisión, sensibilidad y especificidad obtenidas para los conjuntos de datos de entrenamiento, test y validación en el escenario que considera una región de incertidumbre.

Nótese que, en ambos casos, la regla del centro de fusión es más certera a medida que aumenta el número de sensores desplegados. Esto concuerda con los teoremas de la Sección 4.5 para el caso del clasificador binario, confirmando la consistencia universal de la regla de decisión para el aprendizaje distribuido, tanto en casos sin abstención como con abstención (la precisión alcanza valores de hasta 0,97 para los datos de validación).

Además, la principal conclusión que podemos extraer es que al introducir la región de incertidumbre los datos no varían sustancialmente. Aunque parezca una contradicción, los citados teoremas ya anticipaban que, tanto para el caso del clasificador binario como

para el que hemos considerado como segundo escenario (con abstención), existen reglas para cada sensor y para el centro de fusión que garantizan la consistencia universal del aprendizaje. Por lo tanto, en ambos casos los resultados concuerdan con lo esperado a la vista de los teoremas del Capítulo 4.

Escenario con abstención										
N	Entrenamiento			Test			Validación			Coste
	P	S	E	P	S	E	P	S	E	C
10	0,561	0,191	0,931	0,597	0,274	0,921	0,710	0,244	0,931	0,328
50	0,634	0,417	0,851	0,654	0,492	0,816	0,717	0,505	0,810	0,324
100	0,669	0,474	0,864	0,671	0,509	0,832	0,737	0,543	0,826	0,289
300	0,797	0,649	0,944	0,815	0,715	0,915	0,866	0,752	0,921	0,239
500	0,832	0,691	0,973	0,856	0,759	0,952	0,897	0,758	0,945	0,161
1000	0,859	0,725	0,994	0,856	0,712	1,000	0,942	0,847	1,000	0,070

**Tabla 10.** Datos de precisión (P), sensibilidad (S) y especificidad (E) para entrenamiento, test y validación.  $P_{fa} = 0,01$  y  $P_{nd} = 0,01$ .

Analizando las columnas entre sí, es significativo el elevado valor de especificidad que existe en todas las situaciones, en contraposición con el menor valor de sensibilidad, especialmente en los escenarios con un número reducido de sensores. Esto significa que existe una tendencia de nuestro modelo a clasificar como ausencia de blanco de forma sistemática provocando más aciertos en los casos en los que la ausencia es real y menos en los casos de presencia de blanco. Esto puede deberse a que los sensores no están uniformemente distribuidos cuando son pocos y la cantidad de detecciones (tanto por señal como por ruido) es baja en cualquiera de los dos escenarios, haciendo muy pequeño el margen con el que la SVM debe interpretar los datos y tomar su decisión.

### 6.6.2 Comparativa de los dos escenarios para $P_{fa}=P_{nd}= 0,1$

A continuación se muestran los resultados para condiciones menos exigentes en la tolerancia a fallos, es decir,  $P_{fa} = P_{nd} = 0,1$ . De la Tabla 7, sabemos que en este caso la región de abstención es menor y que el umbral de detección es más bajo. En la Tabla 11 se muestran los resultados sin abstención.

Los resultados obtenidos aumentando la probabilidad de falsa alarma tolerada son ligeramente mejores. Esto es debido a que, al reducir el umbral de detección, aumenta el número de sensores que detectan, favoreciendo a la SVM su tarea de separación las clases. Como ocurría con la probabilidad anterior, los valores de precisión, sensibilidad y especificidad no varían sustancialmente del modelo sin abstención al modelo que sí considera la región de incertidumbre, que se muestra en la Tabla 12.

Escenario sin abstención										
N	Entrenamiento			Test			Validación			Coste
	P	S	E	P	S	E	P	S	E	C
10	0,571	0,369	0,772	0,597	0,468	0,727	0,610	0,431	0,707	0,192
50	0,669	0,577	0,761	0,669	0,621	0,717	0,685	0,616	0,722	0,107
100	0,710	0,619	0,801	0,687	0,596	0,777	0,729	0,637	0,763	0,104
300	0,846	0,771	0,921	0,842	0,802	0,882	0,865	0,814	0,890	0,015
500	0,864	0,784	0,944	0,888	0,851	0,925	0,910	0,856	0,931	0,013
1000	0,919	0,837	1,000	0,950	0,925	0,975	0,966	0,935	0,982	0,010

**Tabla 11.** Datos recogidos de precisión (P), sensibilidad (S) y especificidad (E) para entrenamiento, test y validación.  $P_{fa} = 0,1$ .

Escenario con abstención										
N	Entrenamiento			Test			Validación			Coste
	P	S	E	P	S	E	P	S	E	C
10	0,565	0,477	0,652	0,565	0,496	0,634	0,600	0,486	0,654	0,154
50	0,656	0,592	0,720	0,646	0,620	0,671	0,650	0,584	0,680	0,030
100	0,688	0,616	0,762	0,676	0,600	0,752	0,705	0,642	0,733	0,026
300	0,785	0,702	0,868	0,822	0,776	0,867	0,829	0,779	0,847	0,010
500	0,844	0,779	0,909	0,856	0,807	0,905	0,865	0,817	0,881	0,013
1000	0,903	0,831	0,975	0,944	0,912	0,975	0,944	0,917	0,961	0,020

**Tabla 12.** Datos de precisión (P), sensibilidad (S) y especificidad (E) para entrenamiento, test y validación.  $P_{fa} = 0,1$  y  $P_{nd} = 0,1$ .

Si analizamos los niveles de especificidad alcanzados anteriormente para la probabilidad 0,01, podemos comprobar cómo en los casos donde se desplegaban menos sensores, dichos niveles eran muy elevados, llegando a 0,94. El hecho de disminuir el umbral de detección provoca que los valores de sensibilidad y especificidad se equiparen, fruto de la cantidad de sensores que detectan correctamente y del número de ellos que lo hacen por ruido. Esto es, este efecto parece estar relacionado con la proporciones de datos estudiada en la Tabla 8.

## 6.7 El Kernel Gaussiano como alternativa

Los pobres resultados obtenidos con el “Kernel” lineal para redes con pocos sensores, (Precisión=0,56-0,58 para N=10) nos invitan a utilizar un núcleo Gaussiano. La principal ventaja de este “Kernel” es que, a diferencia del lineal, realiza una transformación de los datos a otro espacio distinto donde establece los hiperplanos frontera, para posteriormente trasladarlos de vuelta al espacio de las observaciones, donde la frontera de decisión tomaría una forma no lineal.

Nos centraremos en el estudio de la precisión, la sensibilidad y la especificidad aportadas por uno y otro “Kernel”. Para compararlos, en cada tabla se muestran los resultados obtenidos previamente en el caso lineal y los obtenidos con la nueva función para 10, 50 y 100 sensores. El hecho de utilizar nuevo núcleo introduce una columna donde se recoge el parámetro  $\sigma$  promediado a lo largo de las 10 repeticiones del experimento. La Tabla 13 y la Tabla 14 muestran los valores obtenidos de precisión, sensibilidad y especificidad para los distintos conjuntos de datos (entrenamiento, test y validación) en un escenario sin abstención para probabilidades de falsa alarma de 0,01 y 0,1 respectivamente.

Escenario sin abstención. Kernel Lineal $P_{fa} = 0,01$										
N	Entrenamiento			Test			Validación			Coste
	P	S	E	P	S	E	P	S	E	C
10	0,585	0,230	0,940	0,601	0,271	0,930	0,743	0,230	0,923	0,450
50	0,647	0,438	0,855	0,675	0,534	0,816	0,732	0,526	0,824	0,384
100	0,695	0,529	0,861	0,681	0,591	0,771	0,749	0,613	0,811	0,242

Escenario sin abstención. Kernel Gaussiano $P_{fa} = 0,01$											
N	Train			Test			Validación			Parámetros	
	P	S	E	P	S	E	P	S	E	C	$\sigma$
10	0,578	0,242	0,914	0,609	0,307	0,911	0,703	0,273	0,915	13,10	37,410
50	0,709	0,665	0,753	0,687	0,634	0,741	0,711	0,645	0,747	1,52	6,397
100	0,766	0,727	0,804	0,774	0,729	0,820	0,760	0,704	0,780	0,07	0,075

**Tabla 13.** Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo).  $P_{fa} = 0,01$ .

Escenario sin abstención. Kernel Lineal $P_{fa} = 0,1$										
N	Entrenamiento			Test			Validación			Coste
	P	S	E	P	S	E	P	S	E	C
10	0,571	0,369	0,772	0,597	0,468	0,727	0,610	0,431	0,707	0,192
50	0,669	0,577	0,761	0,669	0,621	0,717	0,685	0,616	0,722	0,107
100	0,710	0,619	0,801	0,687	0,596	0,777	0,729	0,637	0,763	0,104

Escenario sin abstención Kernel Gaussiano $P_{fa} = 0,1$											
N	Entrenamiento			Test			Validación			Parámetros	
	P	S	E	P	S	E	P	S	E	C	$\sigma$
10	0,603	0,582	0,624	0,573	0,572	0,574	0,587	0,567	0,593	1005,10	14,090
50	0,700	0,656	0,744	0,694	0,684	0,705	0,688	0,654	0,701	131,20	0,020
100	0,767	0,804	0,731	0,727	0,707	0,746	0,741	0,704	0,755	1,510	2,301

**Tabla 14.** Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo).  $P_{fa} = 0,1$

Sobre los resultados, podemos apuntar que la variación en líneas generales no es significativa. En concreto, cuando el número de sensores es 10 la respuesta del “Kernel” lineal en validación es mejor. Para 50 sensores los resultados son similares en ambos casos y solo cuando se despliegan 100 sensores, la función de núcleo Gaussiano parece presentar alguna ventaja frente al lineal.

Las Tabla 15 y Tabla 16 muestran los resultados correspondientes a los escenarios con abstención para  $P_{fa} = P_{nd} = 0,01$  y  $P_{fa} = P_{nd} = 0,1$  respectivamente. Las conclusiones son similares a las anteriores: la mínima variación que se produce puede ser fruto de la variabilidad estadística y de la distribución aleatoria de los sensores.

Escenario con abstención. Kernel Lineal $P_{fa} = P_{nd} = 0,01$										
	Entrenamiento			Test			Validación			Coste
N	P	S	E	P	S	E	P	S	E	C
10	0,561	0,191	0,931	0,597	0,274	0,921	0,710	0,244	0,931	0,328
50	0,634	0,417	0,851	0,654	0,492	0,816	0,717	0,505	0,810	0,324
100	0,669	0,474	0,864	0,671	0,509	0,832	0,737	0,543	0,826	0,289

Escenario con abstención. Kernel Gaussiano $P_{fa} = P_{nd} = 0,01$											
	Entrenamiento			Test			Validación			Parámetros	
N	P	S	E	P	S	E	P	S	E	C.	$\sigma$
10	0,589	0,311	0,867	0,611	0,371	0,85	0,674	0,292	0,865	1,5	37,414
50	0,691	0,644	0,737	0,699	0,669	0,729	0,691	0,639	0,713	11,3	2,393
100	0,724	0,696	0,751	0,691	0,656	0,725	0,715	0,670	0,730	1,6	0,222

**Tabla 15.** Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo).  $P_{fa} = 0,01$  y  $P_{nd} = 0,01$ .

Escenario con abstención. Kernel Lineal $P_{fa} = P_{nd} = 0,1$										
	Entrenamiento			Test			Validación			Coste
N	P	S	E	P	S	E	P	S	E	C
10	0,565	0,477	0,652	0,565	0,496	0,634	0,600	0,486	0,654	0,154
50	0,656	0,592	0,720	0,646	0,620	0,671	0,650	0,584	0,680	0,030
100	0,688	0,616	0,762	0,676	0,600	0,752	0,705	0,642	0,733	0,026

Escenario con abstención. Kernel Gaussiano $P_{fa} = P_{nd} = 0,1$											
	Entrenamiento			Test			Validación			Parámetros	
N	P	S	E	P	S	E	P	S	E	C	$\sigma$
10	0,594	0,520	0,667	0,575	0,520	0,630	0,600	0,517	0,637	116,4	0,289
50	0,687	0,694	0,679	0,671	0,674	0,667	0,664	0,709	0,649	4,7	0,103
100	0,728	0,716	0,741	0,711	0,694	0,727	0,735	0,739	0,731	2,7	0,055

**Tabla 16.** Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo).  $P_{fa} = 0,1$  y  $P_{nd} = 0,1$ .

Si comparamos los resultados enfrentando las probabilidades de falsa alarma y no detección para cualquiera de los escenarios, el trabajo realizado por la SVM no refleja dependencia de la probabilidad de error para un número de sensores bajo. Esto es debido a que el modelo aprende suficiente como para obtener los mismos niveles de precisión independientemente de la probabilidad de error con la que se hayan clasificado los datos. Podemos concluir que el nivel de precisión depende más del número de sensores (esto es, cantidad de información disponible) que de la probabilidad de falsa alarma o no detección tolerada, es decir, de la “calidad” de la misma.

Dado que no se aprecian mejoras significativas, y que en el caso de producirse, tienen lugar para 100 sensores, se ha probado el “Kernel” Gaussiano para un número de sensores mayor para cada uno de los escenarios contemplados previamente. Concretamente, se adjuntan los resultados para 500 sensores para los cuatro siguientes escenarios:

- 1) Escenario sin abstención.  $P_{fa} = 0,01$ .
- 2) Escenario sin abstención.  $P_{fa} = 0,1$ .
- 3) Escenario con abstención.  $P_{fa} = P_{nd} = 0,01$
- 4) Escenario con abstención.  $P_{fa} = P_{nd} = 0,1$ .

En la Tabla 17 se comparan los valores obtenidos para el núcleo Lineal y el Gaussiano para 500 sensores. De la misma podemos extraer las siguientes conclusiones:

- La mejora obtenida para la precisión a partir de los datos de entrenamiento y el test sí es significativa, proporcionando valores entre 5%-10% superiores en el caso del núcleo Gaussiano.
- Esa mejora disminuye al aplicar al modelo los datos de validación, estableciéndose aproximadamente en un 4%. Existe un caso atípico (4), donde la mejora apenas es significativa.
- En el escenario sin abstención, los resultados medidos para el modelo proporcionado por la SVM no muestran variación para los dos valores establecidos de probabilidad de error (precisión entorno al 95% para 500 sensores), sino que dependen únicamente del número de sensores.
- Por el contrario, en el escenario con abstención aumentar la probabilidad de error tolerada sí penaliza a los valores alcanzados de precisión. Por ejemplo, la precisión obtenida para los datos de validación cuando  $P_{fa} = P_{nd} = 0,01$ , es del 95,8% mientras que al aumentar las probabilidades de forma que  $P_{fa} = P_{nd} = 0,1$ , la precisión que se obtiene es de 87,4%, lo que supone una reducción de 8,4 puntos porcentuales.

Kernel Lineal										
	Entrenamiento			Test			Validación			Coste
Caso	P	S	E	P	S	E	P	S	E	C
1	0,850	0,725	0,975	0,880	0,785	0,975	0,922	0,817	0,975	0,131
2	0,864	0,784	0,944	0,888	0,851	0,925	0,910	0,856	0,931	0,013
3	0,832	0,691	0,973	0,856	0,759	0,952	0,897	0,758	0,945	0,161
4	0,844	0,779	0,909	0,856	0,807	0,905	0,865	0,817	0,881	0,013

Kernel Gaussiano											
	Entrenamiento			Test			Validación			Parámetros	
Caso	P	S	E	P	S	E	P	S	E	C	$\sigma$
1	0,953	0,950	0,956	0,956	0,937	0,975	0,964	0,944	0,974	0,001	0,013
2	0,950	0,950	0,950	0,937	0,950	0,925	0,954	0,957	0,953	1,000	0,006
3	0,947	0,931	0,962	0,912	0,875	0,950	0,958	0,933	0,966	1,000	0,026
4	0,891	0,925	0,856	0,919	0,975	0,862	0,874	0,979	0,833	0,001	0,013

**Tabla 17.** Comparativa de valores de Precisión (P), Sensibilidad (S) y Especificidad (E) para datos de entrenamiento, test y validación. Kernel lineal (arriba) frente a Kernel Gaussiano (abajo). Red de 500 sensores.

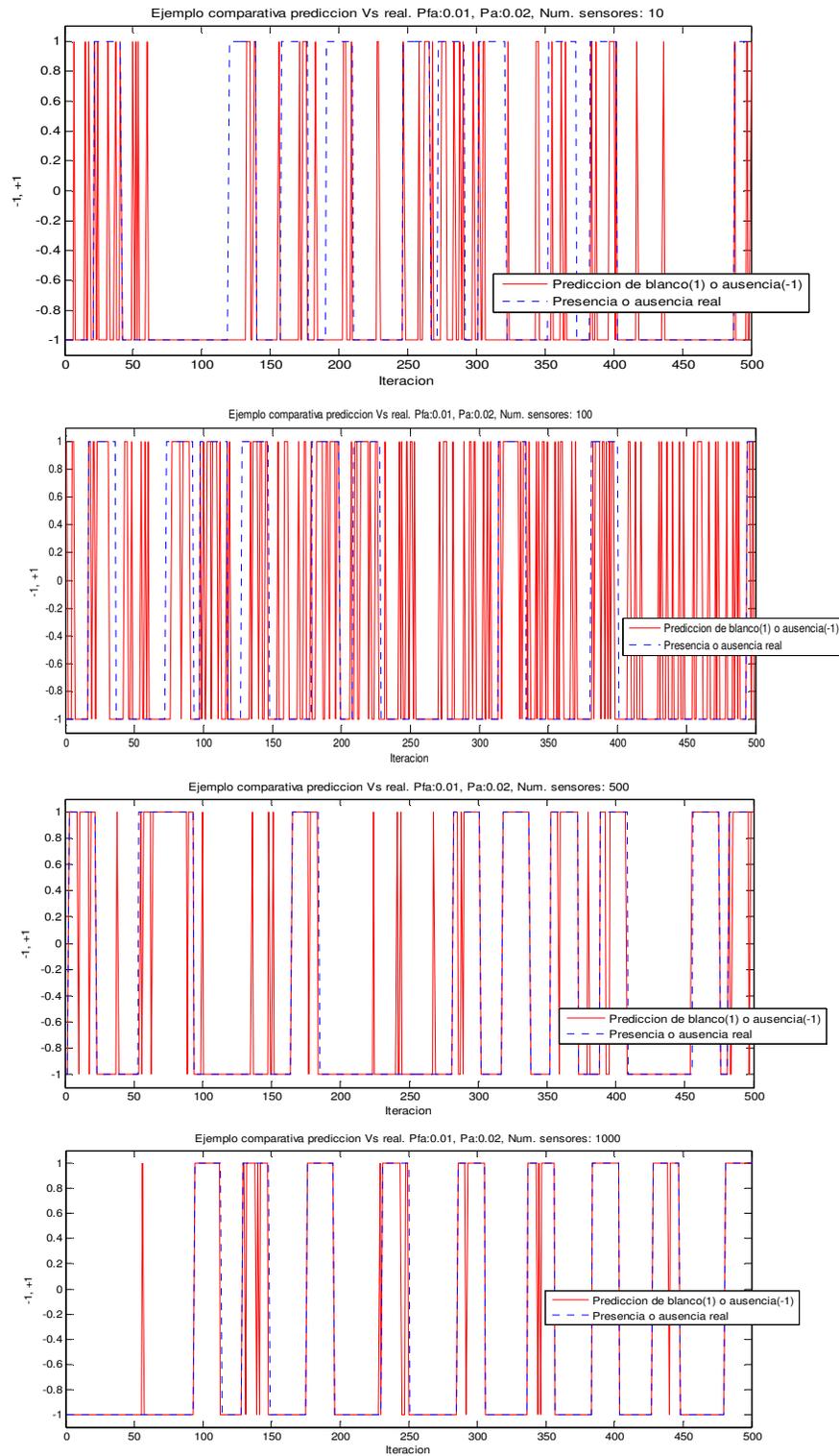
## 6.8 La estimación

Después de probar los teoremas sobre la consistencia universal de las reglas de decisión y clasificación para el aprendizaje distribuido, como complemento a la tarea principal del proyecto (detección), vamos a tratar de aprovechar el escenario creado y la información generada para obtener y evaluar un sencillo estimador de la posición basado en métodos núcleo. No se trata de comprobar la veracidad de los teoremas enunciados en el Capítulo 4 puesto que el escenario no es el mismo, sino de realizar una primera aproximación desde otro enfoque completamente distinto.

### 6.8.1 Estimación sin abstención

El objetivo, que no debemos perder de vista, es estimar para cada instante una posición que podamos comparar con la original. Esa estimación va a depender en este caso de las coordenadas de todos los sensores que detectan el blanco: un conjunto de sensores dentro de una distancia acotada que contribuyen con su medida ponderada por un peso. En el caso sin abstención, los pesos que acompañan a las posiciones de los sensores que detectan se establecen a 1 y para los que no detectan a 0.

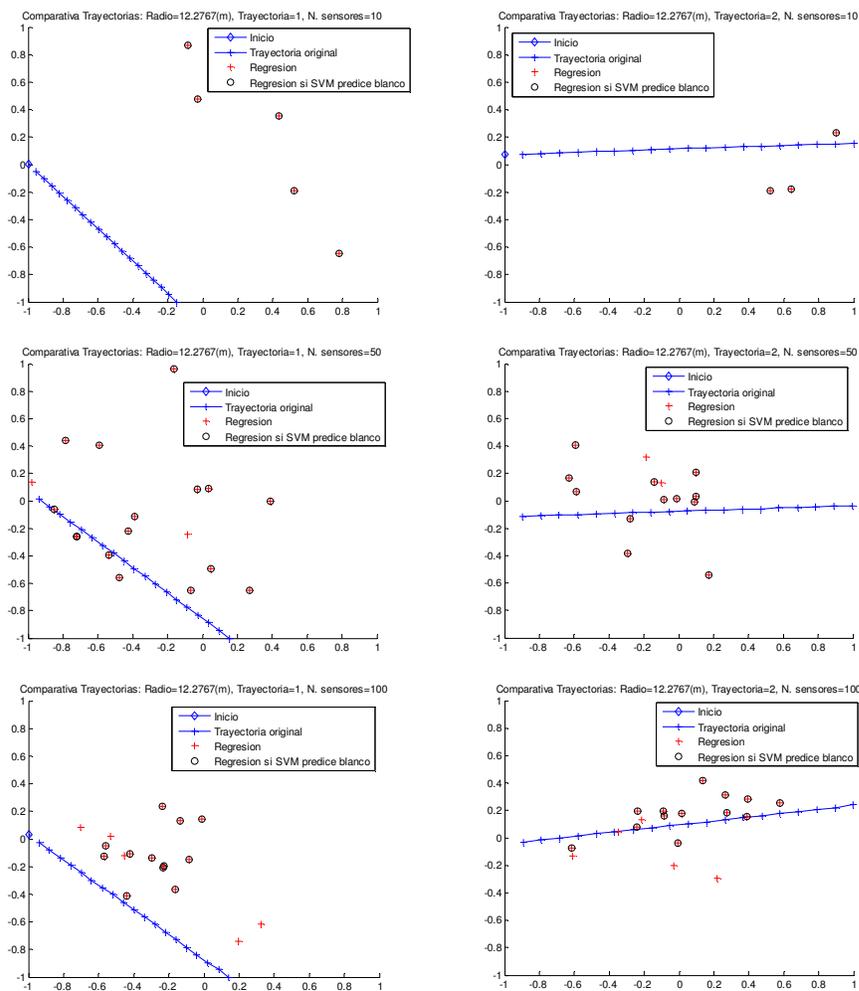
Si se fija la probabilidad de falsa alarma en 0,01 y se aumenta el número de sensores, se obtiene la Figura 52, que nos permite enlazar la sección con los contenidos sobre detección.



**Figura 52.** Comparativa: predicción y presencia real para  $N=10, 100, 500$  y  $1000$ .  $P_{fa} = 0,01$

La Figura 52 representa, para 10, 100, 500 y 1000 sensores, las detecciones (en rojo) frente a la presencia real del blanco (trayectorias completas) marcadas con línea discontinua azul. Observándola, podemos concluir que para 10 sensores las detecciones se realizan en un número muy reducido, para 100 sensores estas aumentan, tanto en casos de presencia de blanco como en ausencia. Para redes con elevado número de sensores, 500 y 1000, los casos de falsas alarmas y no detecciones se reducen, como ya habíamos visto en los resultados sobre detección, la precisión supera el 90%, lo que permite incluso detectar trayectorias completas sin fallo. Otro detalle que podemos observar es que durante la ejecución de duración 500 iteraciones llevada a cabo, llegan a aparecer entre 9 y 10 blancos, que trazan cada uno sus correspondientes trayectorias distintas.

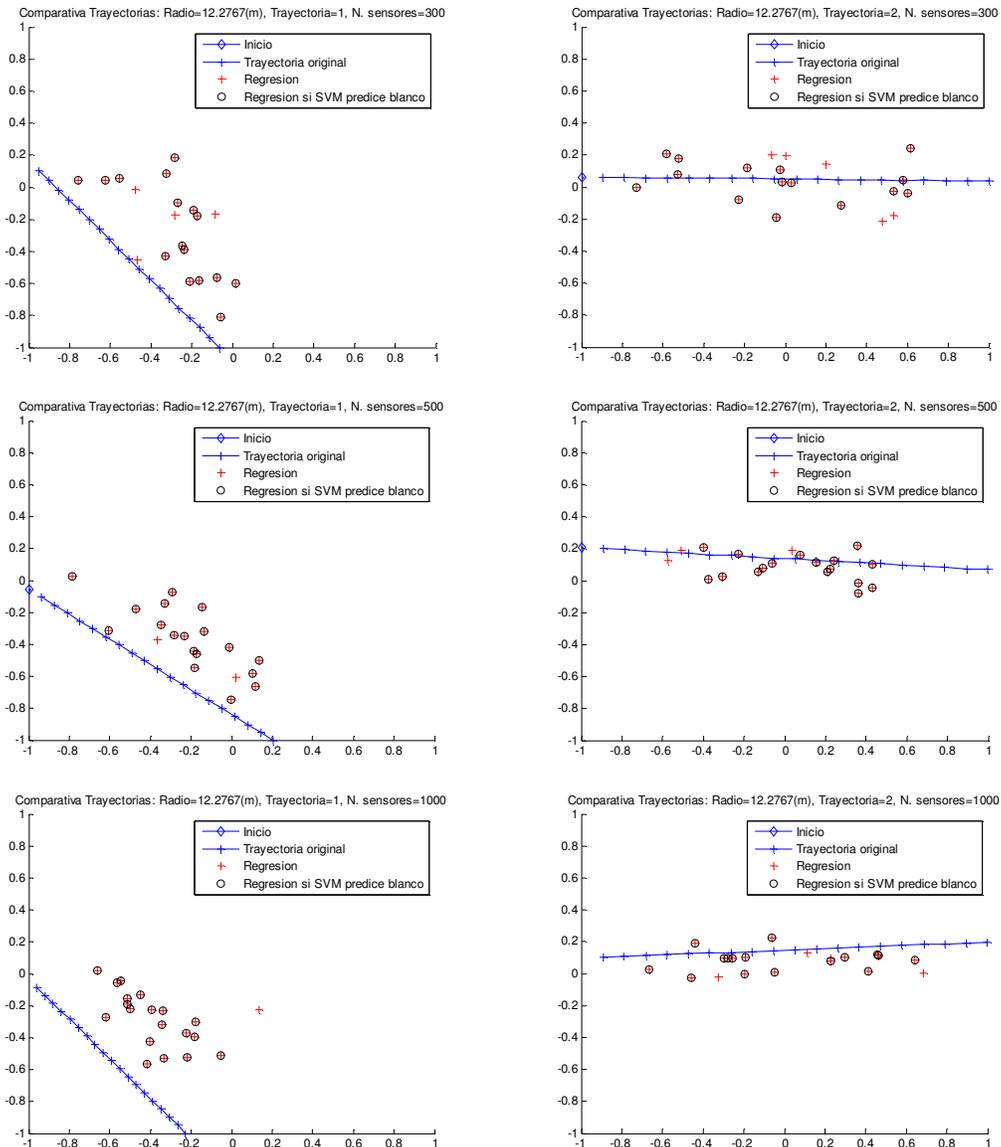
Para no abusar de las figuras, se representarán simplemente un par de trayectorias completas para cada tamaño de la red de sensores que nos sirvan para comprobar los resultados, como se hace en la Figura 53.



**Figura 53.** Comparativa trayectoria real y posiciones estimadas para N=10, 50 y 100 sensores.  $P_{fa} = 0,01$ . 2 trayectorias tipo.

## Capítulo 6: Resultados

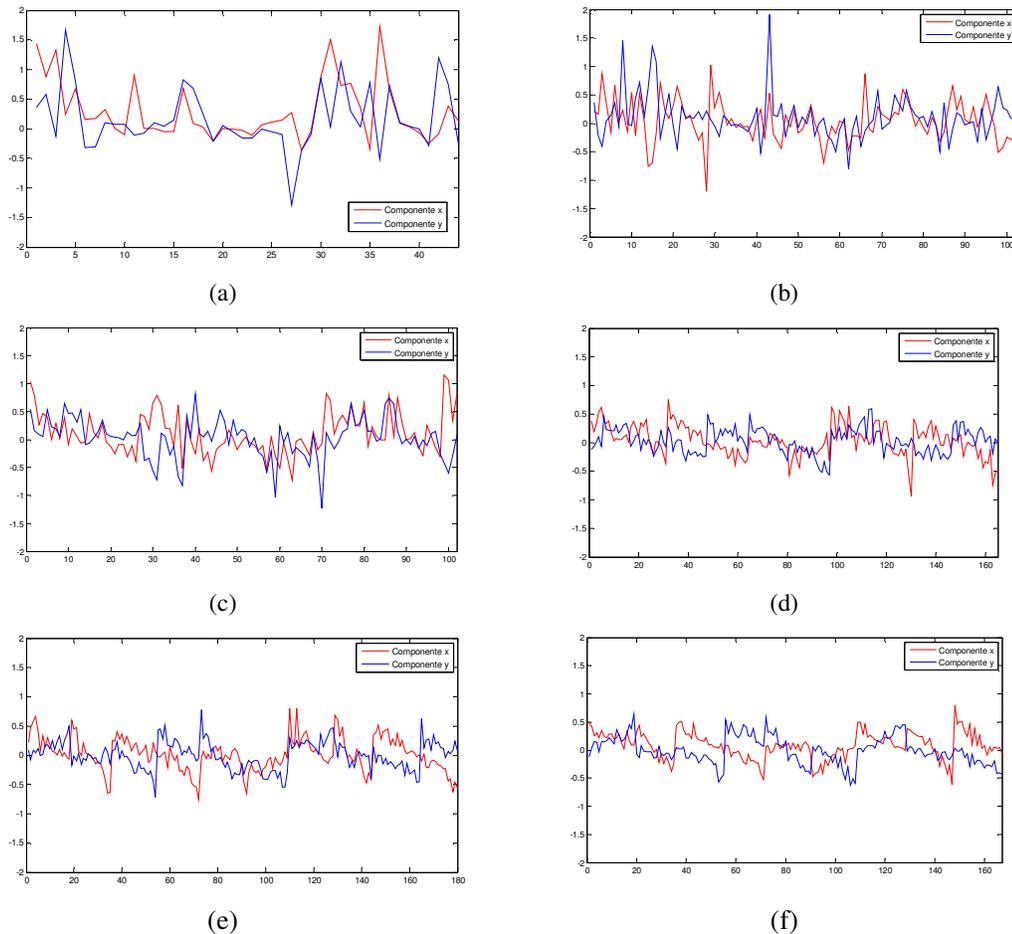
Tanto la Figura 53 como la Figura 54 mostrada a continuación, responden a un tipo de gráficas donde puede verse la trayectoria original en azul, con el diamante ( $\diamond$ ) indicando el inicio, y todos aquellos puntos que resultan de realizar la estimación. Estas posiciones están marcadas con un círculo negro si la predicción del centro de fusión coincide con la presencia de blanco detectada por la SVM, y con el símbolo (+) para aquellas posiciones estimadas ignorando la regla de decisión que haya tomado la SVM. Es decir, aunque no se considere a nivel global presencia de blanco, aprovechando que sabemos con certeza que hay blanco, estimamos con la posición de los sensores que sí lo han detectado a nivel local.



**Figura 54.** Comparativa trayectoria real y posiciones estimadas para  $N=300, 500$  y  $1000$  sensores.  $P_{fa} = 0,01$ . 2 trayectorias tipo.

Para concluir con la Figura 53 y con la Figura 54, podemos destacar que se muestran dos tipos de trayectorias, a la izquierda una que cruza un cuadrante de área total monitorizada y a la derecha una que cruza entre fronteras opuestas del área, haciendo partícipes de la detección a más sensores distribuidos por todo el espacio. Se han escogido estos dos ejemplos porque muestran la diferencia del comportamiento del algoritmo de regresión. Así, para las gráficas de la izquierda todas las posiciones estimadas son desplazadas hacia el centro debido al cálculo inicial del centroide. En las gráficas de la derecha ese sesgo se ve compensado por la presencia en igual proporción de sensores a uno y otro lado de la trayectoria, ya que divide el área monitorizada en dos mitades aproximadamente iguales.

Como complemento, en la Figura 55 se muestra la evolución de las estimaciones de la posición con respecto a la posición original, separadas en sus componentes en x e y. En este caso concreto, se ilustra dicha evolución para 10, 50, 100, y 300, 500, 1000 sensores respectivamente.



**Figura 55.** Diferencia coordenada a coordenada de la posición real con la posición estimada. Para 10(a), 50(b), 100(c), 300(d), 500(e) y 1000(f) sensores. Para  $P_{fa} = 0,01$ .

El primer dato relevante es la longitud de las estimaciones generadas. Si nos fijamos en la Figura 55, para escenarios con un número de sensores reducido (arriba, izquierda), no se estiman tantas posiciones como en los escenarios donde se despliegan más sensores (abajo, derecha). Esto es debido a que el cálculo inicial del centroide puede suponer que posteriormente no haya ningún sensor dentro del radio de acción. Nótese además que cada cambio brusco en la representación coincide con el final de una trayectoria y el inicio de una nueva. El hecho de encontrar ciertos patrones en la evolución de la diferencia, sea la variable que sea, se debe a que los tipos de trayectorias se repiten y siempre se sesgan hacia el centro. Este comportamiento se produce en todos los escenarios y se convierte en el principal hándicap para la estimación de las posiciones. Si analizamos la forma de la función, aparece un pico, que va suavizándose hasta convertirse en otro pico, pero de amplitud opuesta. Esto coincide con el inicio y fin de las trayectorias, y también está asociado al sesgo hacia el centro que se produce: las diferencias entre posiciones van reduciéndose hasta cambiar de signo y posteriormente vuelven a aumentar hasta el final de la trayectoria.

En todas las tablas donde se comparen datos numéricos aparecerán el sesgo (S), la varianza (V) y el error cuadrático (EC) de las variables  $x$  e  $y$  correspondientes a las coordenadas de la posición estimada. Todos los datos numéricos se recogen en la Tabla 18.

Las variables  $\hat{x}_1$  e  $\hat{y}_1$  corresponden con la posición estimada una vez detectado el blanco a nivel global. Por su parte,  $\hat{x}_2$  e  $\hat{y}_2$ , representan una segunda estimación que incluye las posiciones estimadas a partir de las detecciones incluso cuando la presencia real del blanco es cierta pero la SVM no lo considera así. La justificación de esta medida reside en considerar que el sistema estime una posición si se sospecha que existe blanco. Por ejemplo si se conoce que existe un blanco porque se ha estimado con anterioridad presencia de blanco, dado que el mismo no puede desaparecer del área monitorizada en mitad de una trayectoria, podría ser útil estimar una posición aunque quizás no sea tan fiable.

Escenario sin abstención												
N	$\hat{x}_1$			$\hat{y}_1$			$\hat{x}_2$			$\hat{y}_2$		
	S	V	EC									
10	0,297	0,258	0,346	0,163	0,274	0,301	0,234	0,246	0,301	0,198	0,275	0,315
50	0,032	0,146	0,147	0,105	0,147	0,158	0,043	0,192	0,194	0,102	0,168	0,179
100	0,112	0,141	0,153	0,026	0,128	0,128	0,054	0,232	0,235	0,027	0,161	0,161
300	0,034	0,074	0,075	0,030	0,048	0,049	0,052	0,084	0,087	0,019	0,057	0,057
500	0,014	0,081	0,081	-0,003	0,058	0,058	0,011	0,087	0,087	0,001	0,056	0,056
1000	0,082	0,055	0,062	-0,006	0,061	0,061	0,070	0,058	0,063	-0,003	0,061	0,061

**Tabla 18.** Sesgo, varianza y error cuadrático para 10, 50, 100, 300, 500 y 1000 sensores.  $P_{fa} = 0,01$ .

Sobre los resultados en sí mismos, podemos concluir que, al aumentar el número de sensores, tanto el sesgo como la varianza se reducen de forma generalizada, independientemente de la variable que estemos observando. Existe un comportamiento atípico para 50 sensores, puesto que el sesgo de la componente  $x$  se reduce de una manera significativa en contra de la tendencia esperada, pero esto se interpreta como una anomalía de las simulaciones.

Si enfrentamos las componentes de la posición, los sesgos se reducen hasta 0,014 o -0,003 para el caso de  $\widehat{x}_1$  e  $\widehat{y}_1$  estimados en un escenario de 500 sensores. Si desnormalizamos este valor, recordando que la máxima distancia por eje era  $\pm 100m$ , entonces el sesgo tendría un rango de  $\pm 1,4m$ . Con esta información, y dadas las ligeras variaciones que pueden tener su origen en la propia variabilidad estadística, no podríamos afirmar que el comportamiento del estimador sea distinto para una coordenada o la otra, aunque parece repetirse el hecho de que tanto el sesgo como la varianza sean menores en la componente  $y$  que en la componente  $x$ , probablemente supeditado a los tipos de trayectorias que haya seguido el blanco en la fase de validación.

Si continuamos el análisis de la Tabla 18, podemos anticipar que la segunda pareja de variables  $\widehat{x}_2$  e  $\widehat{y}_2$  contempla más estimaciones, concretamente todas las que no son consideradas como presencia de blanco en la detección global.

Además, la varianza parece ser ligeramente mayor, mientras que el sesgo es en unas ocasiones mayor y en otras menor. Esto es lógico, puesto que los errores son más frecuentes, “más graves” y pueden darse a ambos lados de la trayectoria original, compensando en cierto modo el sesgo que se produzca y afectando principalmente a la varianza. Esto conlleva que el error cuadrático medio sea similar al de  $\widehat{x}_1$  e  $\widehat{y}_1$ , aunque de nuevo parece ser ligeramente superior en este caso.

## 6.8.2 Estimación con abstención

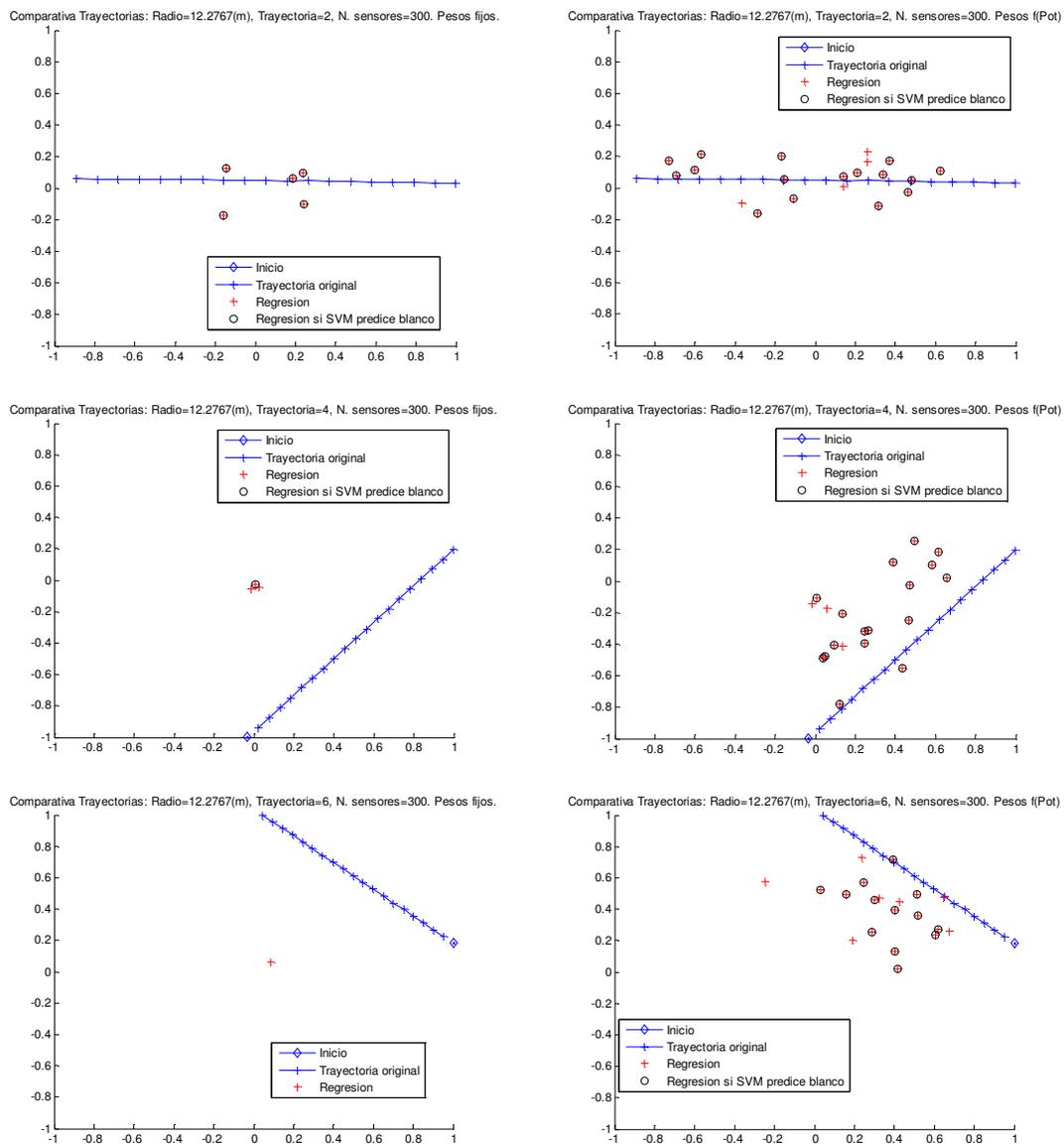
El segundo modelo que se considera, en el que se establece una región de abstención, ha sido descrito previamente en la Sección 5.3.1.3. En este escenario se califica la información aportada por los sensores que se abstienen como relevante pero con una influencia menor que la aportada por los sensores que detectan, asignando a todas las posiciones que influyen en la estimación, un peso fijo.

El primero de los inconvenientes que encontramos con el método de estimación descrito en la Sección 5.4.1 es la escasez de posiciones estimadas. Si aparecen entorno a 9 o 10 trayectorias completas, deberíamos estimar entorno a 190 posiciones. Sin embargo, con dicho método estas se reducen a 30, haciendo inútiles los resultados obtenidos en términos de sesgo y varianza. Si introducimos una de las mejoras descritas en la Sección 5.4.2, modificando la condición de parada, obligaremos a que la posición definitiva se estime en el instante en el que el número de sensores contenidos bajo el radio de acción sea

## Capítulo 6: Resultados

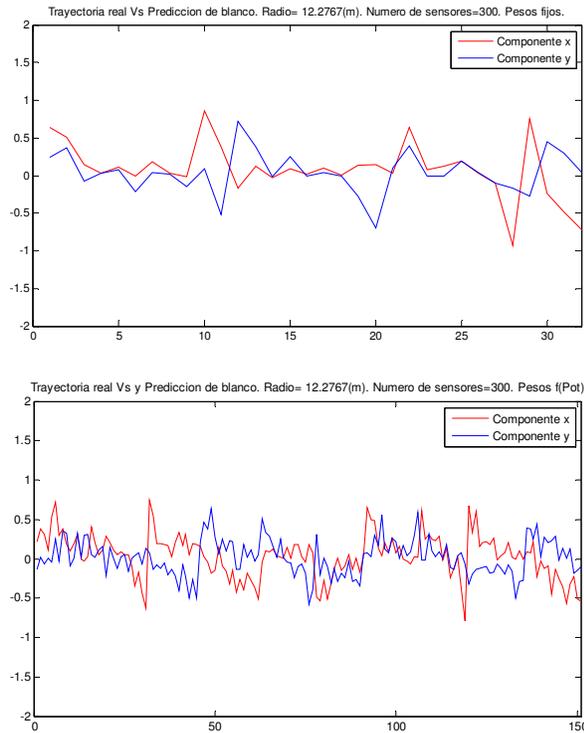
menor o igual que el 10% de la cantidad total de sensores desplegados, evitando de este modo problemas como el explicado en la propia Sección 5.4.2.

De ese modo obtenemos más estimaciones, permitiéndonos analizar el sesgo la varianza y el error cuadrático. Para comprobarlo pueden estudiarse la Figura 56 y la Figura 57. Concretamente en la Figura 56 se muestran a la izquierda las posiciones estimadas con la condición original de parada y a la derecha una vez aplicada la citada mejora. Obviamente, el número de posiciones estimadas aumenta, y también la calidad de las mismas, aunque sin conseguir eliminar el sesgo hacia el centro provocado por la forma de la trayectoria y la distribución de los sensores.



**Figura 56.** Comparativa estimaciones con pesos fijos para distintas condiciones de parada y para 300 sensores, en un escenario con abstención  $P_{fa} = 0,01$  y  $P_{nd} = 0,01$ .

En la Figura 57 se representa la evolución de la diferencia entre la posición original y la estimación componente a componente: en rojo la componente  $x$  y en azul la componente  $y$ . A simple vista podemos comprobar que el número de estimaciones de posiciones ha crecido, concretamente pasa de 33 a 151, aunque en ambos casos su número resulta inferior a las aproximadamente 190 situaciones de presencia de blanco real.



**Figura 57.** Representación de la diferencia entre posiciones de la trayectoria y estimaciones cuando se produce detección del blanco. Pesos fijos para distintas condiciones de parada: original (arriba) y modificada (abajo). Para  $P_{fa} = 0,01$  y  $P_{nd} = 0,01$ .

Una vez confirmada la mejora, se completa el estudio comparando el sesgo, la varianza y el error cuadrático para los casos en los que se establecen los pesos fijos para los sensores que se abstienen en  $\alpha=0,1$  y  $\alpha=0,5$ .

Observando la Tabla 19, obtenida para peso 0,1, volvemos a comprobar cómo la tendencia general es a la disminución del error cuadrático en todas las componentes. Lo mismo ocurre para el sesgo y la varianza, salvo en las dos excepciones comentadas previamente: el sesgo de la componente  $x$  del sesgo para el caso de 50 sensores y el sesgo de las componentes  $x$  e  $y$  para el caso de 1000 sensores. Dos comportamientos atípicos que podrían tener su origen en el tipo de trayectorias consideradas, en la distribución de los sensores o en la propia variabilidad estadística.

Escenario con abstención. Peso fijo 0,1												
N	$\hat{x}_1$			$\hat{y}_1$			$\hat{x}_2$			$\hat{y}_2$		
	S	V	EC									
10	0,156	0,185	0,210	0,264	0,203	0,273	0,138	0,252	0,271	0,220	0,270	0,319
50	0,004	0,118	0,118	0,049	0,170	0,173	0,018	0,202	0,202	0,090	0,230	0,238
100	0,032	0,129	0,130	-0,020	0,115	0,115	0,060	0,235	0,238	-0,013	0,226	0,227
300	0,034	0,081	0,082	0,017	0,053	0,054	0,046	0,081	0,083	0,020	0,060	0,060
500	0,017	0,075	0,075	0,007	0,064	0,064	0,021	0,074	0,075	0,016	0,063	0,064
1000	0,073	0,057	0,062	-0,015	0,056	0,056	0,051	0,061	0,063	-0,014	0,058	0,058

**Tabla 19.** Sesgo, varianza y error cuadrático para 10, 50, 100, 300, 500 y 1000 sensores. Pesos fijos a 1(detección) y 0,1 (abstención).  $P_{fa} = P_{nd} = 0,01$ .

La tendencia descrita en los resultados previos es idéntica a la reflejada en la tabla Tabla 20, donde los pesos para los sensores que se abstienen se establece en 0,5.

Escenario con abstención. Peso fijo 0,5												
N	$\hat{x}_1$			$\hat{y}_1$			$\hat{x}_2$			$\hat{y}_2$		
	S	V	EC									
10	0,201	0,184	0,224	0,400	0,200	0,360	0,191	0,251	0,288	0,361	0,268	0,399
50	0,014	0,117	0,117	0,088	0,170	0,178	0,026	0,201	0,202	0,128	0,230	0,246
100	0,056	0,130	0,133	-0,047	0,115	0,117	0,084	0,234	0,241	-0,041	0,225	0,227
300	0,041	0,083	0,084	0,023	0,055	0,055	0,053	0,082	0,085	0,025	0,061	0,062
500	0,022	0,076	0,076	0,004	0,066	0,066	0,027	0,076	0,076	0,014	0,065	0,065
1000	0,076	0,058	0,064	-0,016	0,057	0,057	0,054	0,062	0,065	-0,015	0,059	0,059

**Tabla 20.** Sesgo, varianza y error cuadrático para 10, 50, 100, 300, 500 y 1000 sensores. Pesos fijos a 1(detección) y 0,5 (abstención).  $P_{fa} = P_{nd} = 0,01$ .

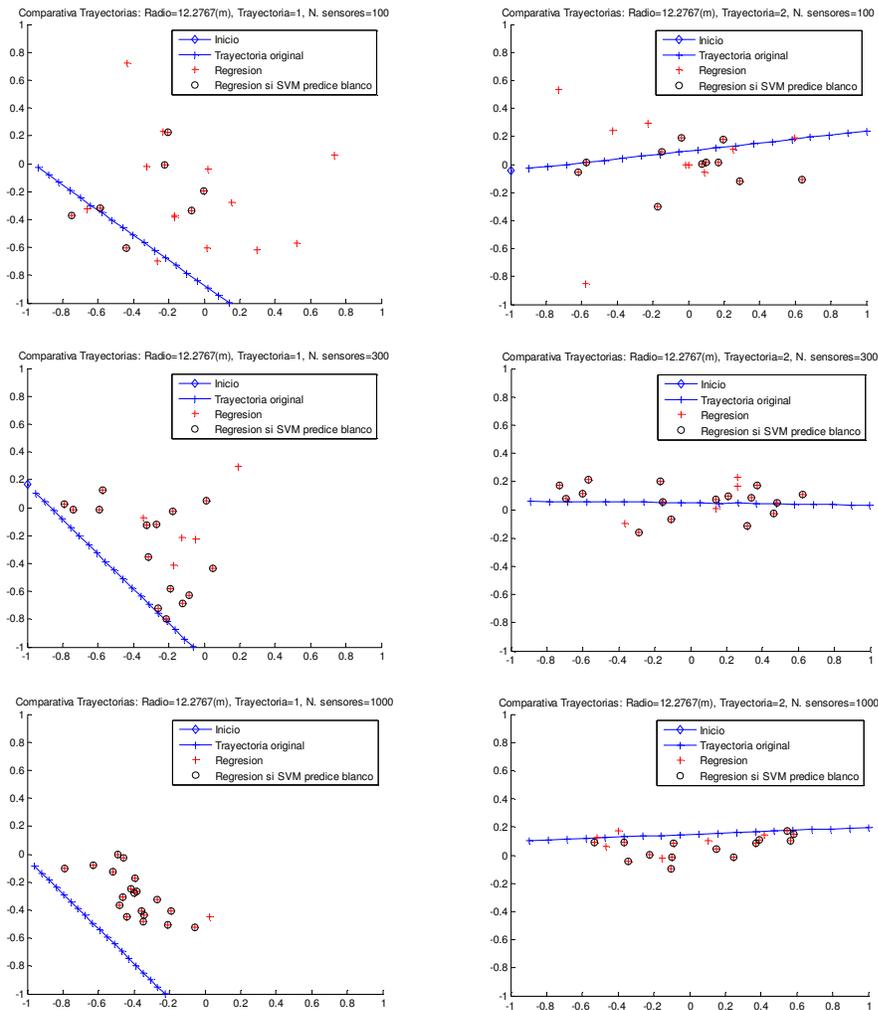
Si enfrentamos los resultados obtenidos para uno y otro valor fijo del peso estaremos comprobando cómo afecta al estimador la medida en que consideramos relevante la información aportada por cada sensor que se abstiene. En general, el hecho de asignar un peso mayor en la estimación de la posición provoca que el error cuadrático, el sesgo y la varianza sean mayores. Esto se explica puesto que la región de abstención engloba a una gran cantidad de sensores, incrementando indirectamente la probabilidad de considerar en esta región sensores que se abstienen por error. Darles más peso en esos casos implica darles más relevancia a los errores en la estimación de la posición frente a los sensores que sí han detectado el blanco correctamente.

Por último, nótese que el caso sin abstención puede verse como un caso particular de este en el que se fija el factor  $\alpha=0$ , de modo que el peso de los sensores que se abstienen sea nulo.

### 6.8.3 Estimación sin limitación en el número de bit transmitidos

A continuación vamos a repetir el proceso previo considerando que las comunicaciones no están limitadas en el número de bits, de forma que el centro de fusión dispondrá del valor de potencia recibido en lugar de la detección local. Esta opción se ha implementado para comprobar las limitaciones del método, siempre en el escenario con abstención.

La condición de parada del algoritmo sigue siendo la misma: considerar que la estimación de la posición es suficiente cuando el número de sensores dentro del radio de acción sea menor o igual que el de 10% de la cantidad total de sensores desplegados. A modo de ejemplo, si se fijan ambas probabilidades en 0,01 podremos comprobar cómo son las estimas sin restricciones en el número de bits transmitidos para dos trayectorias tipo y 100, 300, 500 sensores en la Figura 58.



**Figura 58.** Comparativa trayectoria real y posiciones estimadas para N=100, 300 y 1000, sensores, sin limitación en el número de bits transmitidos.  $P_{fa} = P_{nd} = 0,01$ . 2 trayectorias tipo.

Si comparamos las gráficas de la Figura 58 entre sí volvemos a sacar dos conclusiones principales. Por un lado, aumentar el número de sensores hace en algunos casos las estimaciones se concentren más en la zona de la trayectoria original, evitando posiciones erróneamente estimadas. Por otro lado, en otros casos (trayectorias similares al tipo 1), las posiciones quedan sesgadas por la presencia de muchos más sensores en un lado de la trayectoria.

Para completar la información, en la Tabla 21 se muestran el sesgo, la varianza y el error cuadrático correspondientes al escenario con abstención y sin limitación del número de bits transmitidos al centro de fusión. Las conclusiones extraídas de la misma, son análogas a las de los casos anteriores (Tabla 18, Tabla 19, Tabla 20).

Escenario con abstención												
	$\widehat{x}_1$			$\widehat{y}_1$			$\widehat{x}_2$			$\widehat{y}_2$		
	S	V	EC									
<b>10</b>	0,170	0,189	0,218	0,287	0,197	0,279	0,149	0,248	0,27	0,245	0,258	0,319
<b>50</b>	0,004	0,116	0,116	0,054	0,167	0,170	0,017	0,197	0,197	0,095	0,226	0,235
<b>100</b>	0,033	0,126	0,127	-0,024	0,112	0,112	0,062	0,229	0,232	-0,018	0,223	0,223
<b>300</b>	0,034	0,077	0,078	0,018	0,050	0,051	0,046	0,077	0,079	0,020	0,057	0,057
<b>500</b>	0,016	0,071	0,072	0,006	0,061	0,061	0,021	0,071	0,072	0,015	0,060	0,06
<b>1000</b>	0,070	0,054	0,058	-0,015	0,052	0,052	0,048	0,057	0,06	-0,014	0,054	0,054

**Tabla 21.** Sesgo, varianza y error cuadrático para 10, 50, 100, 300, 500 y 1000 sensores, sin limitación en el número de bits transmitidos.  $P_{fa} = P_{nd} = 0,01$ .

## 6.8.4 Comparación de escenarios

### 6.8.4.1 Comparación de los casos con y sin abstención en función del número de sensores

En esta sección, vamos a comparar el error cuadrático medio obtenido en todos los escenarios para el estimador creado siguiendo las decisiones del detector global proporcionado por la SVM ( $\widehat{x}_1, \widehat{y}_1$ ) en función del número de sensores:

- Sin abstención.
- Con abstención fijando los pesos de los sensores en 0,5.
- Con abstención fijando los pesos de los sensores en 0,1.
- Con abstención fijando los pesos de los sensores en función de la potencia recibida.

La Tabla 22 nos permite apreciar las variaciones en el error cuadrático medio de los 4 escenarios listados. El caso inicial sin abstención resulta ser, excepto para el caso de 1000 sensores, el peor de los estimadores. La explicación reside en la cantidad de información disponible para generar la estimación; los sensores que detectan son pocos en comparación con los que se abstienen y los errores cometidos en la clasificación de los mismos afecta en gran medida a la estimación de la posición.

Por lo tanto, en casi todos los casos error cuadrático asociado al modelo con abstención es ligeramente menor que el del modelo sin abstención. Dentro de los escenarios con abstención, si otorgamos peso 0,5 a las posiciones de los sensores que no votan, y por tanto a su información, el error cuadrático medio es mayor que si le otorgamos peso 0,1. Este hecho nos obliga a considerar que la información de los sensores que se abstienen es útil, pero debe ser ponderada en su justa medida y el factor de ponderación es un factor adicional del sistema que debería optimizarse.

Los mejores resultados son los obtenidos cuando existe libertad en el número de bits transmitidos al centro de fusión (salvo en el caso de 10 sensores), aunque no distan demasiado del caso en el que los pesos son homogéneos y establecidos a 0,1. Este dato, nos permite considerar que estamos cerca de alcanzar la mejor respuesta posible de nuestro algoritmo (disponibilidad de cada potencia recibida por parte del centro de fusión) y que para mejorar los resultados obtenidos habría que realizar modificaciones sustanciales en el mismo o incluso utilizar un método distinto.

Por último, la diferencia entre el error cuadrático de cada componente está sujeto al tipo de trayectoria originada en el escenario de validación, ya que cada tipo afecta en mayor o menor medida al sesgo en una u otra componente. Por ello, si predomina un tipo, el sesgo en una de las componentes puede ser mayor que en la otra y repercutir en el error cuadrático medio.

N	Sin abstención		Con abstención (Pesos =0,5)		Con abstención (Pesos =0,1)		Con libertad de bits transmitidos	
	$\hat{x}_1$	$\hat{y}_1$	$\hat{x}_1$	$\hat{y}_1$	$\hat{x}_1$	$\hat{y}_1$	$\hat{x}_1$	$\hat{y}_1$
10	0,346	0,301	0,224	0,360	0,210	0,273	0,218	0,279
50	0,147	0,158	0,117	0,178	0,118	0,173	0,116	0,170
100	0,153	0,128	0,133	0,117	0,130	0,115	0,127	0,112
300	0,075	0,049	0,084	0,055	0,082	0,054	0,078	0,051
500	0,081	0,058	0,076	0,066	0,075	0,064	0,072	0,061
1000	0,062	0,061	0,064	0,057	0,062	0,056	0,058	0,052

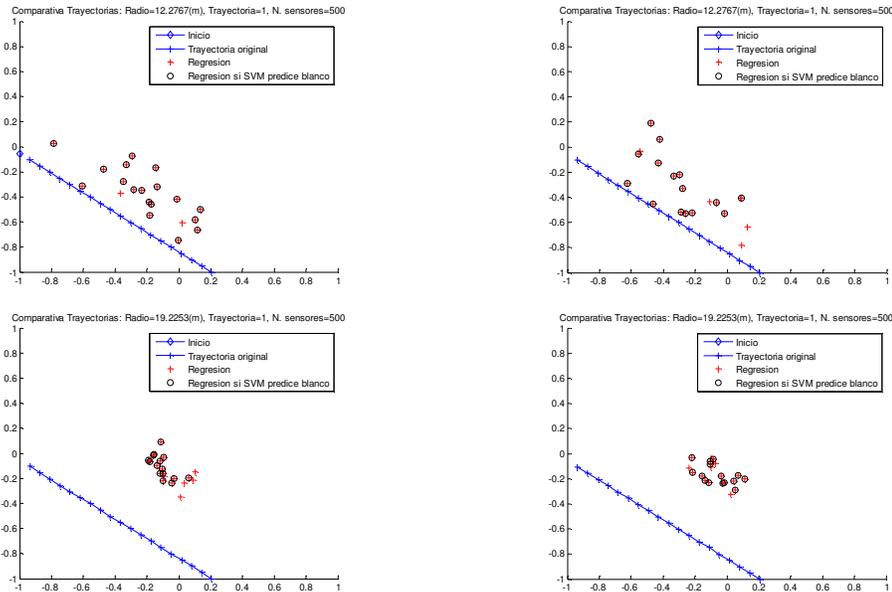
**Tabla 22.** Comparativa error cuadrático medio de los distintos escenarios contemplados.  $P_{fa} = P_{nd} = 0,01$ .

#### 6.8.4.2 Comparación de los casos con y sin abstención en función de las probabilidades de no detección y falsa alarma

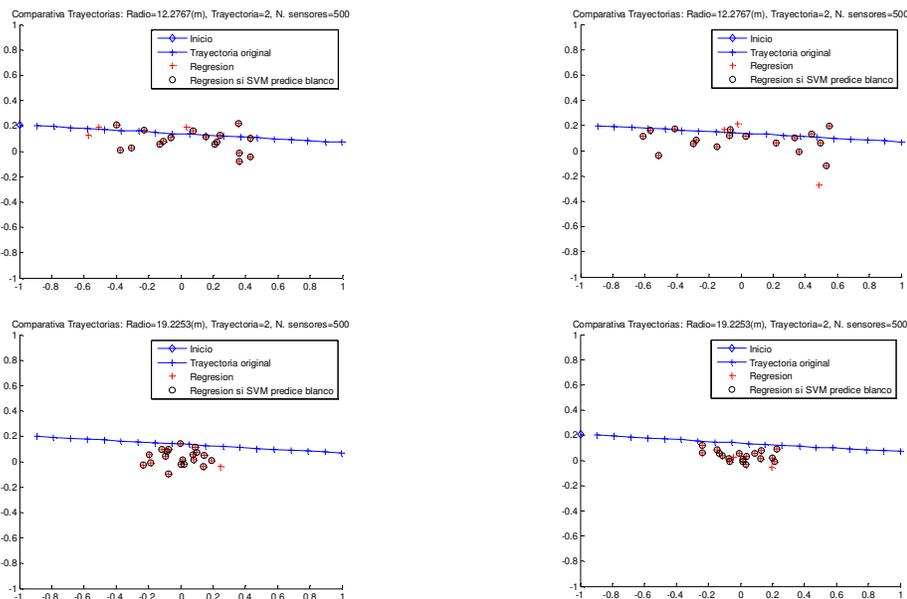
Para completar la información, vamos a fijar el tamaño de la red en 500 sensores y a modificar las probabilidades de error y falsa alarma. En este caso se muestran 4 gráficas por cada figura, todas relacionadas con la misma trayectoria tipo. Así, en la Figura 59, las dos superiores corresponden a probabilidades de falsa alarma y no detección de 0,01 y las dos inferiores con probabilidades a 0,1. Además, las de la derecha se han obtenido en un escenario con abstención, mientras que las de la izquierda dibujan las posiciones obtenidas

## Capítulo 6: Resultados

dado un clasificador binario puro. Para otra posición tipo, que cruza entre fronteras opuestas se obtiene la Figura 60. De forma general, de la Figura 59 y de la Figura 60, podemos obtener como conclusión que el uso de una probabilidad de falsa alarma y no detección mayor provoca que se sesguen los datos aun más hacia el centro.



**Figura 59.** Comparativa de 4 estimaciones de trayectorias distintas para 500 sensores. Sin abstención y  $P_{fa} = 0,01$ , (Arriba izq.). Sin abstención y  $P_{fa} = 0,1$ , (Abajo izq.). Con abstención y  $P_{fa} = P_{nd} = 0,01$ , (Arriba der.). Con abstención y  $P_{fa} = P_{nd} = 0,1$ , (Abajo der.).



**Figura 60.** Comparativa de 4 estimaciones de trayectorias distintas para 500 sensores. Sin abstención y  $P_{fa} = 0,01$ , (Arriba izq.). Sin abstención y  $P_{fa} = 0,1$ , (Abajo izq.). Con abstención y  $P_{fa} = P_{nd} = 0,01$ , (Arriba der.). Con abstención y  $P_{fa} = P_{nd} = 0,1$ , (Abajo der.).

Para finalizar, en la Tabla 23 se comparan los cuatro casos considerados:

- 1) Escenario sin abstención.  $P_{fa} = 0,01$ .
- 2) Escenario sin abstención.  $P_{fa} = 0,1$ .
- 3) Escenario con abstención.  $P_{fa} = P_{nd} = 0,01$
- 4) Escenario con abstención.  $P_{fa} = P_{nd} = 0,1$ .

El hecho de que cada trayectoria afecte a la diferencia con un sesgo de diferente signo, hace el sesgo total de cada variable no sea muy elevado. Por otro lado, es un dato relevante que la varianza y también el propio sesgo son mucho más elevados cuando se relajan las condiciones de probabilidad de error. Como consecuencia, el error cuadrático difiere aproximadamente en una decima al reducir las probabilidades de  $P_{fa} = P_{nd} = 0,1$  a  $P_{fa} = P_{nd} = 0,01$ .

Comparativa para 500 sensores												
Caso	$\widehat{x}_1$			$\widehat{y}_1$			$\widehat{x}_2$			$\widehat{y}_2$		
	S	V	EC									
1	0,014	0,081	0,081	-0,003	0,058	0,058	0,011	0,087	0,087	0,001	0,056	0,056
2	0,034	0,196	0,197	-0,004	0,141	0,141	0,033	0,192	0,193	0,023	0,155	0,155
3	0,016	0,071	0,072	0,006	0,061	0,061	0,021	0,071	0,072	0,015	0,060	0,060
4	0,046	0,183	0,186	0,014	0,159	0,159	0,039	0,187	0,188	0,028	0,158	0,159

**Tabla 23.** Comparativa sesgo, varianza y error cuadrático para la estimación a partir de la SVM  $(\widehat{x}_1, \widehat{y}_1)$  y para la estimación siempre que hay blanco  $(\widehat{x}_2, \widehat{y}_2)$ . 500 sensores. Variaciones de  $P_{fa} = 0,01$ ,  $P_{nd} = 0,01$ ,  $P_{fa} = 0,1$  y  $P_{nd} = 0,1$ .

### 6.8.4.3 Resultados para distintos radios de referencia

A lo largo de esta sección se van a analizar los resultados obtenidos en estimación para distintos radios de referencia fijando el número de sensores en 300. Concretamente frente a los estimadores obtenidos con el radio anteriormente usado (radio de detección), se van a comparar otros obtenidos para el doble de esa distancia y para una distancia infinita, centrándonos en el escenario con abstención.

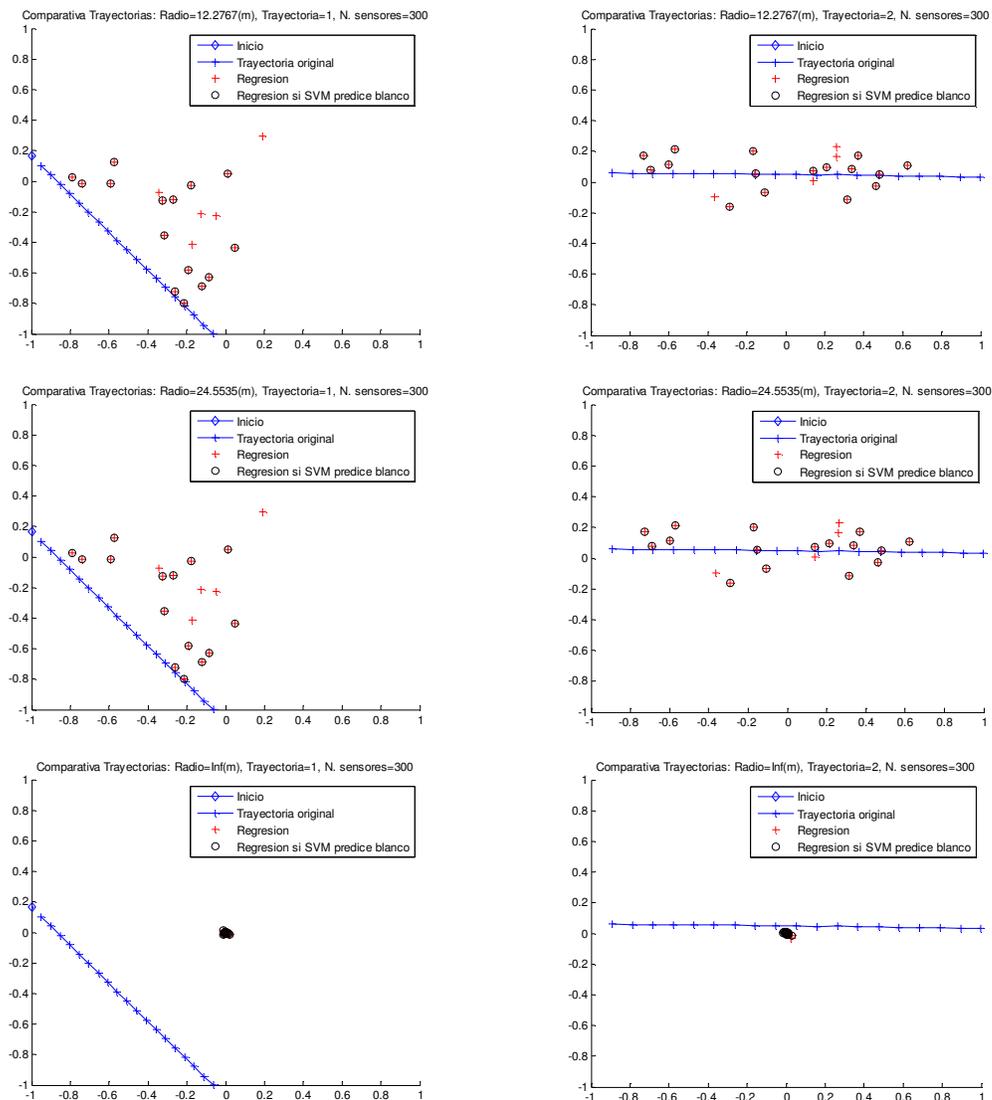
En la Figura 61 se observan las posiciones estimadas para distancias iguales al radio (12,28 m), el doble del radio (24,56 m) y radio infinito. A la derecha se muestran para un tipo de trayectoria y a la izquierda para otro. Revisando detenidamente la misma, se confirma que el comportamiento se mantiene invariante para una distancia que coincide con el radio de detección y con el doble de dicha distancia. Este hecho se debe a dos razones:

- El radio sólo interviene en la condición de parada, de forma que la posición final se estima cuando existe una cantidad de sensores menor o igual al 10% de los sensores desplegados dentro de una circunferencia del radio que estamos variando.

## Capítulo 6: Resultados

- El radio no afecta al cálculo del centroide inicial, ni a la eliminación progresiva de los sensores más alejados.

Por otro lado, en la Figura 61 se observa que las dos recreaciones de las trayectorias para un radio de referencia infinito dan lugar a puntos completamente centrados en el área monitorizada. En realidad esas posiciones coinciden con los centroides que se obtienen inicialmente al considerar todos los sensores que detectan y se abstienen. Obviamente, considerar esas posiciones no tiene sentido ni aporta ningún tipo de información, pero sirve para completar el estudio y entender por qué se produce el sesgo de forma generalizada en la estimación de las posiciones. En consecuencia, establecer una distancia infinita es completamente inútil desde el punto de vista de los resultados.



**Figura 61.** Comparativa de 3 estimaciones de trayectorias distintas para redes de 300 sensores en función de 3 distancias distintas (12,28m, 24,56m y radio infinito) para limitar la influencia de los sensores en la estimación. 2 trayectorias tipo, con abstención,  $P_{fa} = P_{nd} = 0,01$ .

## 6.9 Discusión

Sin llegar a estar divididos físicamente en dos capítulos, los resultados obtenidos en este capítulo provienen de dos procesos distintos: la detección y la estimación.

Inicialmente se evalúan los procesos comunes a los dos escenarios; la representación de los sensores en el área y la elección del Kernel así como las funciones de densidad de probabilidad en presencia y ausencia de blanco, mostrándose la complejidad existente para encontrar una expresión matemática cerrada para el umbral de no detección. Este hecho nos ha obligado a trabajar de forma empírica para encontrar el umbral de potencia asociado a la no detección e indirectamente las reglas de decisión local de los sensores desplegados en el segundo escenario.

Una vez definidas las reglas de decisión local hemos analizado la proporción de los datos generados estableciendo las trayectorias tipo, alcanzando las siguientes conclusiones sobre el clasificador binario:

- La proporción de los sensores que detectan, no detectan o deciden abstenerse varía en función de la probabilidad de falsa alarma y no detección, y del escenario donde se generan los datos (presencia o no de blanco). Estas proporciones afectan a la sensibilidad cuando el número de sensores es reducido, aunque no a la precisión.
- La forma en la que se catalogan los datos no se ve afectada por el número de sensores desplegados en la red, como tampoco se ven afectadas las reglas de decisión local.

A partir de los detectores locales se está en condiciones de obtener un modelo de decisión global, generado a través de la SVM, que ha sido evaluado en términos de precisión, sensibilidad y especificidad. Podemos afirmar que los mismos no varían significativamente de un conjunto de datos a otro (entrenamiento, test y validación) y que incrementan su valor, hasta alcanzar el 99% incluso el 100%, a medida que aumenta el número de sensores desplegados. Como inconveniente, de forma paralela aumenta la carga computacional del centro de fusión.

Otro dato fundamental que extraemos es que la precisión del modelo no se ve afectada por la variación de la probabilidad de error. Esto es, la regla de detección global es igual de precisa independientemente de que la regla de decisión local se establezca para cualquiera de los dos valores tratados de la probabilidad de error y falsa alarma (0,1 y 0,01).

La precisión obtenida para el detector global para redes de un número reducido de sensores (10 a 100) se sitúa entre el 50% y el 75%. Para intentar mejorarla se ha sustituido la función de “Kernel” lineal, por una Gaussiana, que aporta ciertas mejoras, aunque no muy significativas, para los conjuntos de datos de entrenamiento y test.

## Capítulo 6: Resultados

Desafortunadamente dichas mejoras no se trasladan al escenario de validación (precisión hasta un 4% mayor, que se convierte en un 6% para 500 sensores).

Como complemento al objetivo fundamental de analizar el detector global se ha desarrollado un estimador sencillo apoyado en métodos núcleo, tanto para el caso sin abstención como con abstención (donde se ha variado el peso asignado a los sensores que se abstienen)

El cálculo inicial del centroide (posición estimada) considerando las posiciones de todos los sensores que detectan y se abstienen, sesga la estimación hacia el centro del área. Para evitar que ese sesgo impida obtener estimas validas se ha modificado el método de estimación de la posición, eliminando iterativamente el sensor más alejado del centroide y deteniendo la ejecución cuando el 10% de los sensores desplegados o menos estén dentro del radio de acción.

Aunque esta modificación permite mejorar parcialmente los resultados, el sesgo no desaparece, llegándose a la conclusión de que es necesario introducir cambios sustanciales en el algoritmo para mejorar los resultados obtenidos.

# Capítulo 7

## Conclusiones y líneas futuras

### 7.1 Introducción

Como colofón a este proyecto, este último capítulo pretende recoger las principales conclusiones que se extraen de su realización y de los resultados obtenidos. Como la línea de investigación sigue abierta, con muchas variantes y posibles mejoras, también se completa con una sección donde se recogen posibles mejoras y algunas de estas líneas futuras.

### 7.2 Conclusiones

La primera lección que se obtiene de la realización de este trabajo es que existe la necesidad, y por ello la oportunidad, de desarrollar multitud de aplicaciones a partir de la flexibilidad tanto de las redes inalámbricas de sensores como de los propios sensores. Sin embargo, hemos podido observar que existen ciertas limitaciones relacionadas con las topologías, los mecanismos para intercambiar la información, el consumo, la condiciones del entorno, e incluso de las restricciones económicas, que nos obligan a exprimir al máximo los recursos disponibles. Para optimizar el manejo de los mismos, nos apoyaremos en métodos de inferencia y aprendizaje distribuidos, que son capaces de proporcionar decisores, clasificadores y estimadores sin necesidad de disponer de toda la información o de todos los recursos.

En el caso particular que nos ocupa, hemos enfocado esa optimización de recursos hacia la reducción de los bits necesarios en la comunicación entre cada uno de los sensores y el centro de fusión encargado de recoger la información. Esto es, cada nodo sensor lleva a cabo una primera tarea de decisión local, posibilitando así que únicamente sea necesario transmitir de forma paralela la menor cantidad de información posible (hasta un solo bit).

Posteriormente, el centro de fusión llevará a cabo la tarea de decisión global utilizando herramientas más sofisticadas.

Para poder diseñar los citados decisores locales, ha sido necesario un repaso de la teoría clásica de detectores y clasificadores que nos permitiese establecer unas reglas de decisión o clasificación local válidas para todos los sensores. En concreto, el criterio escogido ha sido el de Neyman-Pearson, porque nos garantiza una probabilidad de falsa alarma o de no detección a partir de la que establecer el umbral, o los umbrales, en términos de la potencia detectada.

Si se reduce la información, el centro de fusión también necesita mecanismos para hacer útil toda la que recibe y tratar de conformar su propia regla de detección global. Para ello nos hemos apoyado en una máquina de vectores soporte (SVM), una herramienta que necesita ser configurada y entrenada adecuadamente. En este sentido, es básica la librería de funciones de libre distribución utilizada, LIBSVM, y su documentación.

Para analizar el problema descrito, se ha considerado un escenario concreto: la creación de un detector global de presencia de un objetivo móvil en un área monitorizada donde se despliegan los sensores. Las simulaciones realizadas nos han permitido alcanzar las siguientes conclusiones respecto a los objetivos fundamentales del proyecto:

- Existen reglas de decisión para cada sensor y para el centro de fusión que garantizan la consistencia universal del aprendizaje distribuido, en escenarios con y sin abstención, para el problema de detección binaria implementado en este proyecto. En otros términos, los niveles de certeza en las decisiones aumentan con la densidad de sensores desplegados.
- Dicha garantía de aprendizaje, que se enunciaba para escenarios donde la información era incorrelada, se ha verificado para situaciones donde las muestras generadas sí están correladas.
- El modelo generado para la detección global no depende de la probabilidad de falsa alarma o no detección establecida, al menos para los dos casos considerados, siendo los niveles de precisión obtenidos muy similares en uno y otro caso.
- La elección de la función de núcleo lineal es idónea dada la cantidad de información de la que se dispone y la dimensión del espacio de observación. Al considerarse la función de núcleo Gaussiana y llevarse a cabo una transformación a otro espacio de mayor dimensión, los resultados proporcionados por el detector global son muy similares, y la mejora que aportan es poco significativa independientemente del número de sensores desplegados.

En definitiva, los mecanismos implementados funcionan, y lo hacen cada vez mejor a medida que aumenta el número de sensores, alcanzando niveles de precisión del 90% a partir de 500 sensores para las condiciones preestablecidas a lo largo del proyecto.

Como complemento al objetivo fundamental de analizar el detector global se ha desarrollado un estimador sencillo apoyado en métodos núcleo. En este caso no se ha pretendido demostrar la existencia de mecanismos de estimación en escenarios donde se contemplan limitaciones en el aprendizaje distribuido, sino que se ha tratado simplemente de aprovechar la información disponible y realizar una primera aproximación a la tarea de estimación, con el fin de estudiar si es viable o no una futura aplicación para reconstruir las trayectorias.

Después de modificar heurísticamente el algoritmo de estimación y tras evaluar el sesgo, la varianza y el error cuadrático medio de diversos escenarios en el Capítulo 6, llegamos a la conclusión de que, dados los tipos de trayectorias utilizados, el cálculo inicial del centroide (posición estimada) considerando las posiciones de todos los sensores que detectan y se abstienen sesga la estimación hacia el centro del área. Además, en función de los resultados podemos concluir que:

- Considerar la información aportada por los sensores que se abstienen beneficia al estimador (reduce el error cuadrático medio), aunque estas medidas necesitan ser consideradas de forma adecuada, puesto que asignarles demasiado peso no representa ningún beneficio.
- Un aumento de la probabilidad de error tolerada empeora el error cuadrático medio del estimador.

## **7.3 Posibles mejoras y líneas futuras**

### **7.3.1 Ampliación del tipo de trayectorias**

En la Sección 5.2.2 se detallan las 6 trayectorias tipo utilizadas en este proyecto. Estas han sido escogidas con la intención de maximizar el área que atraviesan, incorporando al proceso de entrenamiento al mayor número de sensores posibles. Sin embargo, el hecho de utilizar trayectorias lineales restringe mucho el tipo de blancos que podrían monitorizarse. Si contemplamos trayectorias cuya forma no fuese lineal o no tuviese un número de saltos establecido, podrían modelarse trayectorias no lineales como las mostradas en la Figura 62 en base a una determinada dirección y aceleración (marcada en azul) y con una componente estocástica o aleatoria en su movimiento que podría ser objeto de estudio en versiones posteriores.

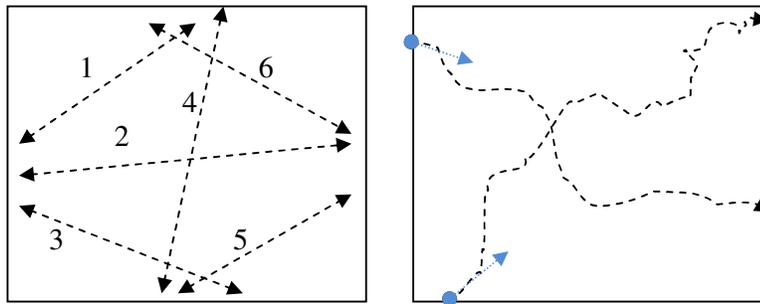


Figura 62. Trayectorias actuales (izquierda) y propuestas para líneas futuras (derecha).

### 7.3.2 El canal binario simétrico

A lo largo de todo el proyecto se ha supuesto que el canal entre cada uno de los sensores y el centro de fusión está libre de errores. Esto es, el centro de fusión recibe la información tal y como la transmite cada sensor. En futuras líneas de investigación podría considerarse el hecho de modelar el canal con alguna probabilidad de error. En el caso más sencillo podría utilizarse un canal binario simétrico BSC, como el de la Figura 63.

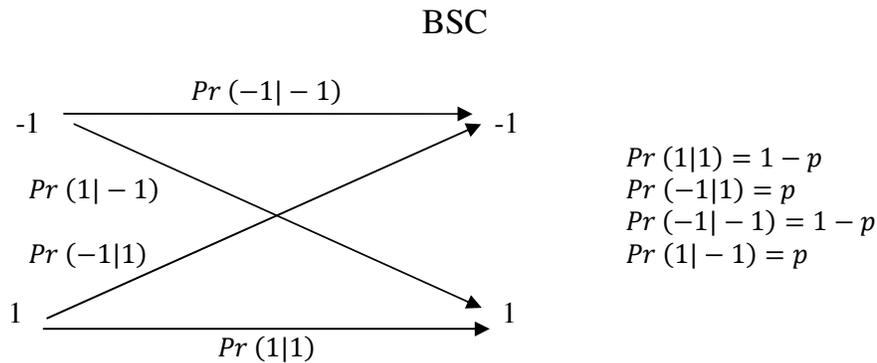
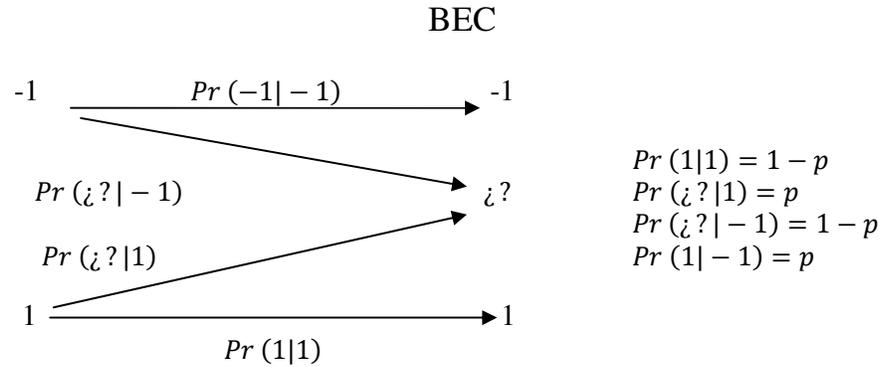


Figura 63. Canal binario simétrico.

De este modo, además de las probabilidades de error de los decisores, habría que considerar los errores producidos durante la transmisión con probabilidad  $p$ , abriendo un nuevo escenario al tratamiento digital de la información recibida.

También podría considerarse otra forma de modelar el canal que contempla la posibilidad de que la información no alcance su destino, un canal binario de borrado (BEC, “Binary Erasure Channel”), cuyo esquema se muestra en la Figura 64. En este caso, la información procedente de cada nodo sensor, podría llegar o desvanecerse por el trayecto con probabilidad  $p$ , generando confusión con las abstenciones. Como complemento, podría considerarse incluso una combinación de ambos modelos de canal.



**Figura 64.** Canal binario simétrico con borrado.

### 7.3.3 Mejoras del método de estimación

Dado que los contenidos referentes a la estimación agrupados en la Sección 5.4 no se tratan más que de una primera aproximación al problema y los resultados obtenidos no son excesivamente buenos, podrían considerarse otros enfoques para resolverlo.

El principal problema es que la forma de las trayectorias y la distribución de los sensores sesgan las estimaciones hacia el centro, por lo que cualquier método que pueda combatir este efecto podría propiciar un menor error cuadrático medio. Por ejemplo se podrían usar de filtros de partículas. Típicamente se usa para estimar el estado de un sistema que cambia a lo largo del tiempo y de modo sencillo e introductorio, podemos decir que se trata un método de Monte Carlo secuencial, donde la idea básica es determinar los puntos del espacio donde muestrear a partir de un estado inicial y una distribución de probabilidad que se presupone para el movimiento. Posteriormente se ponderan las muestras en función de la información que tenemos del estado previo, es decir, se asigna cada peso en función del estado anterior y por tanto de la estimación previa [74].

### 7.3.4 Consideración de otros escenarios

Otra posible línea futura sería la consideración de un escenario alternativo. Concretamente, un escenario donde la limitación que obliga a realizar el proceso de detección y estimación de modo distribuido sea distinta al caso estudiado (transmisión de un bit por sensor). Por ejemplo, podría plantearse que la limitación afectase al número de sensores con capacidad de transmitir hasta el centro de fusión, haciendo necesario el uso de “clusters” generados entorno a los nodos transmisores. La decisión en este caso podría tomarse de acuerdo a la información que disponen los sensores que forman cada “cluster” y ser transmitida al centro de fusión, que en este nuevo escenario dispondría de un menor número de fuentes de conocimiento pero que a priori serían más fiables.



# Apéndice A

## Presupuesto del proyecto

Este apéndice está dedicado a la elaboración del presupuesto asociado a la realización del presente proyecto. Para hacer la estimación de este importe económico se tiene en cuenta tanto el coste de los medios materiales como el coste de los recursos humanos empleados para su realización.

### A.1 Coste de los medios materiales

En la Tabla 24 se hace un desglose del coste unitario del material utilizado para realizar el proyecto. Al total del coste de material utilizado hay que añadir el coste del emplazamiento de trabajo donde se ha llevado a cabo la realización del proyecto. El lugar de trabajo está debidamente acondicionado mediante alumbrado, calefacción, aire acondicionado, servicio de limpieza, mesas, sillas, tomas de corriente, y red de datos de banda ancha. Su coste en alquiler se estima en 750€ por mes. Se ha estimado una duración de trabajo del proyecto en 10 meses, por lo que el coste en **concepto de alquiler de emplazamiento puede estimarse en 7.500€**.

Concepto	Coste unitario
Ordenador personal	1.200€
Licencia de sistema operativo Windows XP Professional	180€
Licencia MATLAB	2.400€
Paquete ofimático Microsoft Office 2003 Professional	180€
Impresora + Tóner + Papel	170€
Gastos de oficina	130€
Encuadernación de los tomos	180€
<b>Total material</b>	<b>4.440€</b>

**Tabla 24.** Coste unitario del material utilizado en la realización del proyecto.

Sumando el coste total del material más el alquiler del lugar de trabajo se obtiene que: **el coste total de medios materiales es de 11.940€**.

## A.2 Coste del personal

Según el Colegio Oficial de Ingenieros de Telecomunicación (COIT), el sueldo medio de un Ingeniero de Telecomunicación es de 30.000€ brutos anuales. Sin embargo, en este caso como referencia podemos escoger un sueldo inicial para un trabajador en fase de realización del proyecto de fin de carrera o recién titulado, que se establece en 18.000€ brutos anuales (incluido beneficios sociales). En dichas condiciones, el salario ronda los 1.500€ brutos al mes, valor que se tomará como referencia. Por lo tanto, el coste del proyecto **por coste de personal asciende a un total de 15.000€.**

## A.3 Coste de la dirección

Para la realización este proyecto de ingeniería, han participado una persona junto con su director de proyecto, que ha realizado una labor de apoyo y consultoría.

El salario del director del proyecto se estima, de forma general, como un 7% de la suma del coste de material más el coste de la mano de obra hasta un límite de 30.000€ anuales. Por tanto el **coste de la dirección del proyecto a lo largo de la elaboración de este estudio es de 1.885,8€.**

## A.4 Coste total del proyecto

El importe económico final del proyecto se calcula como la suma del coste de los medios materiales más el coste del personal y dirección del proyecto. Realizando este cálculo se obtiene que el **coste total del proyecto asciende a 28.825,8€.**

Leganés a 27 de Enero de 2011  
El ingeniero proyectista

Fdo. José Luis Barbosa Moreno

# Referencias

- [1] Carlos Franco Beltran, “*Wireless Sensor Networks*”, Tendencias BIT 165. Octubre-Noviembre 2007.
- [2] Chee-Yee Chong y Srikanta P. Kumar, “*Sensor Networks: Evolution, Opportunities & Challenges*”. Proceedings of the IEEE vol. 91(8), Agosto 2003,
- [3] “*Proceedings of the Distributed Sensor Nets Workshop*”. Eds. Dept. Comput. Sci. Carnegie Mellon Univ. Pittsburgh, PA, 1978.
- [4] R. Rashid y G. Robertson, “*Accent: A communication oriented network operating system kernel*”. 8th Symp. Operating System Principles, 1981.
- [5] A. Oppenheim, R. Davis, y W. Dove C. Myers, “*Knowledge based speech analysis and enhancement*”, Proceedings of Speech and Signal Processing on Acoustics, San Diego, CA, 1984.
- [6] Johns Hopkin, “*The Cooperative Engagement Capability*”, APL Technical Digest 14, 1995.
- [7] Y. Carts-Powell, “*Unattended ground sensors stop and analyze the roses*”, 2000.
- [8] S. Kumar y D. Shepherd, “*SensIT: Sensor information technology for the warfighter*”. in Proc. of the 4th International Conference on Information Fusion, 2001.
- [9] D. A.Tibaduiza Burgos, L. C.Aparicio Zafra, L. M. Caro Ortiz y N. Chio Cho, “*Redes de Sensores Inalámbricos*”, II Congreso de Mecatrónica. Universidad Autónoma de Bucaramanga, Colombia, 2009.
- [10] Jose A. Vera, F. Losilla, P. J. Meseguer y M. J. Buendía, “*Redes de Sensores y Actuadores (WSAN) en domótica*”. Universidad Politécnica de Cartagena, 2007.
- [11] Jacob Fraden, “*Handbook of modern sensors*”, 4th ed.: Springer, 2010.
- [12] Mehrdad Babazadeh, “*Energy Management in Wireless Sensor Networks*”: Universität Bremen, 2010.
- [13] W. Su, Y. Sankarasubramaniam, E. Cayirci y I. F. Akyildiz, “*Wireless sensor networks: a survey*”. Computer Networks, 2002, vol. 38.
- [14] Niels Aakvaag y Jan-Erik Frey, “*Redes de Sensores Inalámbricos*”, Revista ABB, Febrero 2006.
- [15] Malik Tubaishat y Sanjay Madria, “*Sensor networks: an overview*”, IEEE POTENTIALS, Abril/Mayo 2003.
- [16] R. Govindan, J. Heidemann, S. Kumar. y D. Estrin, “*Next century challenges: Scalable coordination in sensor networks*”. Proceedings of the fifth annual ACM/IEEE International Conference on Mobile Computing and Networking, 1999.
- [17] Jamal N. Al-Karaki y Ahmed E. Kamal, “*Routing techniques in Wireless Sensor Networks: A survey*”, IEEE Wireless Communications, Ed., Diciembre 2004.
- [18] M. Younis y A. Ahmed Abbasi, “*A survey on clustering algorithms for wireless*

- sensor networks*". Elsevier, Junio 2007.
- [19] Dragus Niculescu, "Positioning in Ad Hoc Sensor Networks", IEEE NETWORK. Ad Hoc Networking: Data communications and Topology Control, Julio/Agosto 2004, vol. 18 No.4.
- [20] John Stankovic, David Wagner y Adrian Peerig, "Security in Wireless Sensor Networks", Communications of the ACM, Junio 2004, vol. 47, No. 6.
- [21] Sanjeev R. Kulkarni, H. Vincent Poor y Joel B. Predd, "Distributed Learning in Wireless Sensor Networks", IEEE Signal Processing Magazine, Julio 2006, vol. 23, nº 4.
- [22] J. Lygeros, S. Manesis. y Th. Arampatzis, "A Survey of Applications of Wireless Sensors and Wireless Sensor Networks", Junio 2005.
- [23] S. Krishnamurthy, J. A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J.Hui B. Krogh y T. He, "An energy-efficient surveillance system using wireless sensor networks", Boston, MA, 2004.
- [24] A. Rytter, "Vibration based inspection of civil engineering structures", PhD dissertation, Dept. of Building Technology and Structural Eng. ed. Aalborg University, Denmark, 1993.
- [25] E. Flynn M. Todd, G. Park, C. Farrar y D. Mascareñas, "Wireless Sensor Technologies for Monitoring Civil Structures". Sound and Vibration, Abril 2008.
- [26] J. Polastre, R. Szewczyk, D. Culler, J. Anderson. y A. Mainwaring, "Wireless sensor networks for habitat monitoring", Proc. ACM Inter. Workshop on Wireless Sensor Networks and Applications Atlanta, GA, 2002.
- [27] K. Bridges y E. Biagioni, "The application of remote sensor technology to assist the recovery of rare and endangered species", Distributed Sensor Networks for the Int. Journ. Of High Performance Computing Applications, 2002, vol. 16 No. 3.
- [28] T. Brooke, R. Beckwith y J. Burrell, "Vineyard computing: sensor networks in agricultural production", 2004.
- [29] IEEE Instrumentation and Measurement Society , "IEEE Standard for A Smart Transducer Interface for Sensors and Actuators" (1451.7)., 2010.
- [30] Reich J. E., Zhao y F. Chu M., "Distributed Attention for Large Video Sensor Networks" London, UK, 2004.
- [31] T. Knot, "Issue 9". BP Frontiers Magazine, April 2004.
- [32] R. R. Muntz, M. Potkonjak y M. B. Srivastava, "Smart kindergarten: sensor based wireless networks for smart developmental problem-solving environments", Mobile Computing and Networking, 2001.
- [33] N. Bulusu, A. Savkin, S. Jha. y P. Pathirana, "Node Localization Using Mobile Robots in Disconnected Sensor Networks", IEEE Transactions on Mobile Computing, Junio 2004.
- [34] B. Fisher, S. Hollar, K.S.J. Pister y J. Perng, "Acceleration Sensing Glove (ASG)", ISWC International Symposium on Wearable Computers. San Francisco, CA, 1999.
- [35] H. Vincent Poor, "An Introduction to Signal Detection and Estimation". Ed. Springer, 1994.
- [36] Vojislav Kecman, "Learning and Soft Computing". Massachuchetts Institute of Technology, 2001.
- [37] Anthony D. Whalen. y Robert N. McDonough, "Detection of signals in noise". AT&T Bell Laboratories and Academic Press, 1995.

- [38] Steven M. Kay, "*Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*". Prentice Hall, 1998.
- [39] David Corrigan, "*An Introduction to PDF Estimation and Clustering*". Electrical and Electronic Engineering Dept., University of Dublin, Trinity College.
- [40] A.M. Legendre, "*Nouvelles Méthodes pour la Détermination des Orbites des Comètes*". Paris, 1805.
- [41] Juan M. Vilar Fernández, "*Modelos Estadísticos Aplicados*", Universidad de La Coruña, Junio 2006.
- [42] David C. Lay, K "*Algebra lineal y sus aplicaciones*" Prentice-Hall, 2001.
- [43] H.P. Graf, E. Cosatto, L. Bottow, I Dourdanovic, y V. Vapnik, "*Parallel support vector machines: The cascade SVM in Advances in Neural Information Processing Systems*": MIT press, 2005, vol. 17.
- [44] D.N.Hennessy y F. Provost, "*Scaling up: Distributed machine learning with cooperation*, Portland, Or, 1996.
- [45] Z. Zhang, H. Vishwanathan y T. Berger, "*The CEO problem*", IEEE Trans. Inform. theory, 1996, vol. 42(3).
- [46] L. Tong, P. K. Varshney y B. Chen, "*Channel are distributed deteccion in Wireles Senso Networks*", IEEE Signal Processing Mag, vol. 23(4), June 2006.
- [47] J. F. Veeravalli y V. V. Chamberland, "*Asymtotic results for decentraliced detection in power constrained wireless sensor networks*", IEEE J.Select. Areas Communication, vol. 22(6), 2004.
- [48] R.Neelamani, R.Baraniuk. y V.Delouille, "*Robust distributed estimation in sensor networks using the embedded polygons algorithm*". In Proc. Int. Symp. Information Processing Sensor Networks. Berkley, CA, Abril 2004.
- [49] Christopher J. C. Burges, "*A tutorial on Support Vector Machines for pattern recognition*". Data Mining and Knowledge Discovery 2, 1998.
- [50] Christopher J. C. Burges, Alexander J. y Smola Bernhard Scholkopf, "*Advances in kernel methods: support vector learning*". 1998.
- [51] Florian Markowitz, "*Classification by Nearest Shrunken Centroids and Support Vector Machines*", Max Planck Institute for Molecular Genetics. Practical Microarray Analysis, 2006.
- [52] Phillip H. Sherrod, "*DTREG, Predictive modeling Software*". Phillip H. Sherrod, 1991-2004.
- [53] Jiawei Han y Micheline Kamber, "*Data Mining: Concepts and Techniques*". University of Illinois.
- [54] Daryle Niedermayer, "*An Introduction to Bayesian Networks and their Contemporary Applications*". 1998.
- [55] Christopher K. I Williams y Carl E. Rasmussen, "*Gaussian Processes for Regression*".
- [56] Elias M. Stein y Rami Shakarchi, "*Real Analysis: Measure Theory, Integration, and Hilbert Spaces*": Princeton University Press, 2005.
- [57] A. Nedic y D. Bertsekas, "*Incremental subgradient methods nondifferentiable optimization*", MIT, Cambidge, MA, 1999.
- [58] S. Uryasev y P. M. Pardalos, "*Convergence rate of incremental subgradient algorithms*". Dordrecht, 2000.

- [59] R. D. Nowak y M. G. Rabbat, "*Quantized incremental algorithms for distributed optimization*", IEEE JSpecial Areas Commun. vol. 23 no 4, 2006,
- [60] R. Nowak y M. G. Rabbat, "*Distributed optimization in sensor networks*", Proc. 3rd Int. Symp. Information Processing Sensor Networks. Berkley, CA, 2004.
- [61] S. R. Kulkarni, H. V. Poor y J. B. Predd, "*Regression in sensor networks: Training distributively with alternating projections*", Architectures and Implementations XV Proc. SPIE Conf. Advanced Signal Processing Algorithms, San Diego, CA, 2005.
- [62] S. R. Kulkarni H. V. Poor y J. B. Predd, "*Distributed kernel regression: An algorithm for training collaboratively*", In Proc.2006 IEEE Information Theory Workshop, Punta del Este, Uruguay, Marzo 2006.
- [63] M. I. Jordan, B. Sinopoli y X. Nguyen, "*A kernel-based learning approach to ad hoc sensor network localization*", ACM Trans. Sensor Networks, 2005.
- [64] C. J. Stone, "*Consistent nonparametric regression*", 1977, vol. 5(4).
- [65] S. R. Kulkarni H. V. Poor y J. B. Predd, "*Consistency in models for distributed learning under communication constraints*", IEEE Trans. Inform. Theory, 2006, vol. 52(1).
- [66] Constantine A. Ballanis, "*Antenna Theory: Analysis and Design*", John Wiley & Sons Inc., 1997.
- [67] E. S. Gopi, "*Mathematical Summary for Digital Signal Processing Applications with Matlab*". Springer, 2010.
- [68] Alan V. Oppenheim, Alan S. Willsky, y S. Hamid, *Signals and Systems*. Prentice Hall Signal Processing Series, 1997.
- [69] Milton Abramowitz y Irene A. Stegun, "*Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*", 1965.
- [70] Chih-Jen Lin y Chih-Chung Chang, "*LIBSVM: a library for support vector machines*". Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001. [Online].
- [71] C. Cortes y V.Vapnik, "*Support vector networks. Machine Learning*". 1995.
- [72] Chih-Chung Chang y Chih-Jen Lin Chih-Wei Hsu, "*A practical guide to Support Vector Classification*", Dept. of computer Science. National Taiwan University, Ed. Taiwan, 2003-2010.
- [73] L. K. Luo, H. Peng, Q. S. Zhang, and C. D. Lin, A "*Comparison of Strategies for Unbalance Sample Distribution in Support Vector Machine*", Xiamen University. Department of Automation, Ed. China.
- [74] Petar M. Djuric et al., "*Particle Filtering*". IEEE Signal Processing Magazine, Septiembre 2003.