

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS  
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



*Trabajo Fin de Grado*

**APLICACIÓN DE TÉCNICAS DE  
CLASIFICACIÓN SUPERVISADA AL  
ANÁLISIS DE LA ACTIVIDAD DE REDES DE  
COMUNICACIONES INALÁMBRICAS**  
(Application of supervised classification  
techniques to the analysis of the activity in  
wireless communications networks)

Para acceder al Título de

***Graduado en  
Ingeniería de Tecnologías de Telecomunicación***

Autor: Marta Gómez Goitia

Julio - 2016



**GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE  
TELECOMUNICACIÓN**

**CALIFICACIÓN DEL TRABAJO FIN DE GRADO**

**Realizado por: Marta Gómez Goitia**

**Director del TFG: Jesús Pérez Arriaga**

**Título: “Aplicación de técnicas de clasificación supervisada al análisis  
de la actividad de redes de comunicaciones inalámbricas”**

**Title: “Application of supervised classification techniques to the  
analysis of the activity in Wireless communications networks“**

**Presentado a examen el día:**

para acceder al Título de

**GRADUADO EN INGENIERÍA DE TECNOLOGÍAS DE  
TELECOMUNICACIÓN**

Composición del Tribunal:

Presidente (Apellidos, Nombre):

Secretario (Apellidos, Nombre):

Vocal (Apellidos, Nombre):

Este Tribunal ha resuelto otorgar la calificación de: .....

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFG  
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Grado Nº  
(a asignar por Secretaría)



## **Agradecimientos**

No podría haber realizado este trabajo sin toda la información, formación, apoyo y ayuda que he recibido.

Tengo que agradecer especialmente su enorme y valiosa supervisión a Jesús Pérez, que me ofreció la oportunidad de trabajar en este proyecto tan interesante. Me siento muy agradecida por todo lo que he aprendido de él, por toda su ayuda, sus correcciones y sus valiosas opiniones; dedicándome todo el tiempo que he necesitado sin ningún límite ni problema.

A todos los profesores que me han formado a lo largo de estos cuatro años de vida académica. Todos ellos han creado la base necesaria para que yo pudiera realizar este trabajo de fin de grado.

A todos mis compañeros de clase, los que han recorrido conmigo parte o todo el camino durante la carrera, por haber compartido conmigo tantos momentos, algunos más complicados, pero la mayoría buenos.

A mi familia, mis padres y hermana, por su apoyo incondicional en todos los aspectos y por creer en mí y animarme continuamente.



# ÍNDICE GENERAL

1. <b>Introducción</b> .....	13
2. <b>Descripción del problema</b> .....	15
2.1. Red de comunicaciones.....	15
2.2. Red de sensores.....	15
2.3. Modelo canal radio.....	17
2.3.1. Caracterización de las pérdidas del trayecto.....	18
2.3.2. Caracterización del desvanecimiento por sombra.....	19
2.3.3. Caracterización del desvanecimiento multicamino.....	19
3. <b>Formulación como problema de clasificación supervisada</b> .....	21
3.1. Modelo logistic regression.....	23
3.2. Estimación de Máxima Verosimilitud.....	26
3.3. Algoritmo Gradient Ascent.....	27
3.4. Algoritmo Stochastic Gradient Ascent.....	28
4. <b>Resultados y conclusiones</b> .....	31
4.1. Entorno de propagación.....	31
4.2. Caracterización canal entre usuarios y sensores.....	31
4.2.1. Las pérdidas del trayecto.....	32
4.2.2.El desvanecimiento por sombra.....	32
4.2.3.El desvanecimiento multicamino.....	32
4.3. Modelado de la actividad de los usuarios.....	33
4.4. Descripción de las simulaciones.....	34
4.5. Resultados sobre canal invariante / Algoritmo Bloque.....	36
4.5.1. Efecto del exponente de pérdidas, $\gamma$ , sobre la precisión.....	36
4.5.2. Efecto del número de sensores sobre la precisión.....	37
4.5.3. Efecto del número de muestras de energía, $N_s$ , sobre la precisión.....	38
4.5.4. Efecto de la potencia de transmisión de los usuarios, $P$ , sobre la precisión..	39
4.5.5. Efecto del número de datos de entrenamiento, $n$ , sobre la precisión.....	40
4.5.6. Efecto del modelado de la actividad de los usuarios sobre la precisión.....	41
4.6. Resultados sobre canal variante / Algoritmo Adaptativo.....	43
5. <b>Líneas futuras</b> .....	49
6. <b>Apéndices</b> .....	51
6.1. Apéndice 1: Demostración de la expresión del gradiente de $q_n(\mathbf{W})$ .....	51
6.2. Apéndice 2: Probabilidades de estado en el límite.....	52
7. <b>Referencias</b> .....	55



# ÍNDICE FIGURAS

Figura 1.1. Conversión del problema de comunicaciones en uno de clasificación supervisada.....	13
Figura 2.1. El centro de fusión recibe los niveles de energía de los sensores y la clase a la que éstos pertenecen.....	16
Figura 2.2. Entorno de propagación constituido por usuarios y sensores, existiendo un canal entre cada par usuario-sensor.....	16
Figura 2.3. Las tres escalas de variación del canal radio en función de la distancia.....	18
Figura 2.4. Evolución de la respuesta del canal $\nu$ , a lo largo del tiempo, entre usuarios y sensores.....	19
Figura 3.1. Con el conjunto de datos de entrenamiento, $D$ , a través del centro de fusión que construye el vector de energía, se hace la predicción sobre los nuevos datos sin clasificar.....	21
Figura 3.2. Realizaciones de los vectores de energía para una SNR=6dB.....	22
Figura 3.3. Realizaciones de los vectores de energía para una SNR=-6.5 dB.....	22
Figura 3.4. Realizaciones de los vectores de energía para una SNR=20 dB.....	22
Figura 3.5. Fronteras de decisión para realizaciones de los vectores de energía con una SNR=16 dB.....	25
Figura 3.6. Fronteras de decisión para realizaciones de los vectores de energía con una SNR=-6.5 dB.....	25
Figura 4.1. Entorno de propagación del problema con los usuarios situados aleatoriamente.....	31
Figura 4.2. Evolución con el tiempo de tres canales con diferentes $f_d$ .....	33
Figura 4.3. Probabilidad de estado en el límite de los estados del usuario en función de su probabilidad de pasar de estar activo a inactivo.....	34
Figura 4.4. Acierto en la identificación de la actividad de los usuarios en función del exponente de pérdidas, $\gamma$ .....	36
Figura 4.5. Acierto en la identificación de la actividad de los usuarios en función del número de sensores en cada eje del entorno de propagación.....	37
Figura 4.6. Acierto en la identificación de la actividad de los usuarios en función del número de muestras en los detectores de energía, $N_s$ .....	39
Figura 4.7. Acierto en la identificación de la actividad de los usuarios en función de la potencia con que transmiten, $P$ .....	40
Figura 4.8. Acierto en la identificación de la actividad de los usuarios en función del número de datos de entrenamiento usados, $N$ .....	41
Figura 4.9. Acierto en la identificación de la actividad de los usuarios en función de la probabilidad de un usuario de pasar de estar activo a inactivo.....	42
Figura 4.10. Relación entre el parámetro $\alpha$ de la constante de aprendizaje, $f_d$ y la precisión.....	43
Figura 4.11. Comparación algoritmos GA y SGA para una única realización de un canal invariante con el tiempo.....	44

Figura 4.12. Comparación algoritmos GA y SGA para un promedio de 200 realizaciones de canales invariantes con el tiempo.....	45
Figura 4.13. Acierto en la identificación de los usuarios a lo largo de las iteraciones...	46
Figura 4.14. Acierto en la identificación de los usuarios a lo largo de las iteraciones en función de la probabilidad de que un usuario pase de estar activo a inactivo.....	46
Figura 4.15. Acierto en la identificación de los usuarios a lo largo de las iteraciones en función del desplazamiento Doppler normalizado.....	47

## **Palabras clave**

Clasificación supervisada, Aprendizaje máquina, red de comunicaciones, sensor, Logistic Regression, algoritmo bloque, algoritmo adaptativo, detección de actividad, GA, SGA, estimador ML

## **Keywords**

Supervised classification, machine learning, communications network, sensor, Logistic Regression, batch algorithm, online algorithm, activity detection, GA, SGA, ML estimator



# 1. INTRODUCCIÓN

En este Trabajo de Fin de Grado se va a presentar un sistema capaz de analizar la actividad de redes de comunicaciones inalámbricas [1], que permite obtener información útil sobre ésta. Este sistema está compuesto por un conjunto de sensores, distribuidos en el entorno de propagación de la red.

En este trabajo, el problema de monitorizar la red de comunicaciones se plantea como un problema de aprendizaje máquina (o machine learning). En concreto, se plantea como un problema de clasificación supervisada [2], en el que las observaciones son los datos adquiridos por la red de sensores ( $X$ ) y las clases ( $Y$ ) son los distintos estados en que se puede encontrar la red de comunicaciones. Por tanto, el problema de monitorización de la red se convierte en un problema de clasificación supervisada, o reconocimiento de patrones.

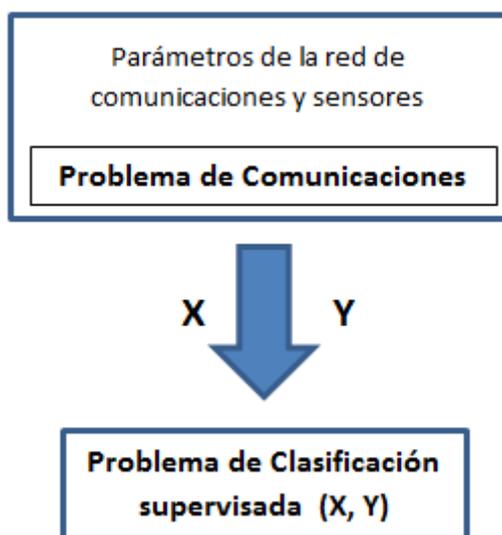


Figura 1.1. Conversión del problema de comunicaciones en uno de clasificación supervisada.

Más concretamente, este trabajo tratará sobre la identificación de la actividad de los usuarios en la red de comunicaciones, es decir, en saber si algún usuario de la red de comunicaciones está transmitiendo, y, si es el caso, cuál de ellos lo hace.

Este conocimiento puede ser muy útil en los contextos de Radio Cognitiva (CR) o redes heterogéneas (HetNets).

El concepto de CR en los sistemas de comunicaciones ha emergido en la última década para mitigar los problemas de escasez del limitado espectro radioeléctrico, mejorando la utilización del espectro [3]. Centrándose en el acceso oportunista al espectro [4][5][6], los dispositivos CR tienen que sensar el espectro radioeléctrico, en el que tienen licencia los usuarios primarios de la red de comunicaciones, y consecuentemente, utilizar las oportunidades del espectro libre para transmitir. Sin embargo, el problema a tratar, no sólo sensa si el canal está libre o no en una determinada banda de frecuencia, como generalmente ocurre en todos los problemas de CR [7], si no que da información sobre qué usuario de la red primaria está transmitiendo.

De esta forma, conociendo qué usuario está transmitiendo y con qué potencia lo hace, se puede averiguar cómo son los canales entre ese usuario y todos los sensores. Así, si el canal entre el usuario y un sensor es muy bajo, suponiendo que el canal es recíproco, el sensor o dispositivo CR puede transmitir a la vez que el usuario de la red primaria. Asegurando, siempre, que lo hace con suficiente baja potencia para no interferir nunca la transmisión del usuario primario.

En el contexto de redes heterogéneas [8], el conocimiento del estado de la red de comunicaciones es una información crucial. El estado de la red es un concepto amplio que incluye el conocimiento de la actividad de los usuarios en cada momento, así como su descripción estadística. Esto permite crear los llamados mapas de interferencia [9] y mapas espectrales [10].

## 2. DESCRIPCIÓN DEL PROBLEMA

### 2.1. Red de comunicaciones

Se considera una red de comunicaciones, consistente en una serie de  $M$  usuarios, de los cuales solo uno puede estar activo transmitiendo. De esta forma la red puede estar en  $M+1$  estados, el estado en el que ninguno está transmitiendo y los  $M$  estados en los que uno de los usuarios lo está. Se considera  $\mathbf{s}$  como el vector, de tamaño  $M+1$ , que define el estado de la red y que tiene todos sus elementos a cero excepto un 1, la posición del 1 indica que usuario está transmitiendo, teniendo en cuenta que si el 1 está en la primera posición del vector significa que no hay usuarios activos. Una forma alternativa de describir el estado de la red es mediante el número entero  $y$ , que indica también el estado de la red.

Es decir,  $\mathbf{s} = [s_0 \ s_1 \ \dots \ s_M]$ , para  $s_m = \begin{cases} 1, \text{ usuario } m \text{ activo} \\ 0, \text{ otro caso} \end{cases}$  e  $y = m + 1$ , con el usuario  $m$  activo.

### 2.2. Red sensores

Se considera un conjunto de  $K$  sensores, los cuales realizan un sensado cooperativo en tiempo real donde cada uno estima el nivel de energía que recibe, y lo reporta a un centro de fusión, que bien puede ser uno de los sensores o un centro aparte que procesa todos los datos recibidos.

Para estimar el nivel de energía, los sensores, adquieren  $N_s$  muestras durante un periodo de tiempo  $\tau$ . La muestra  $i$ -ésima de la señal tomada por el sensor  $k$  se denota  $Z_k(i)$ ,

$$Z_k(i) = \sum_{m=1}^M s_m h_{m,k} T_m(i) + N_k(i),$$

donde  $s_m$  indica el estado del usuario  $m$ ,  $h_{m,k}$  denota la ganancia del canal entre el usuario  $m$  y el sensor  $k$ ,  $T_m(i)$  es la señal transmitida por el usuario  $m$ , y  $N_k(i)$  es el ruido térmico en el sensor  $k$ . La potencia media de transmisión del usuario  $m$  será  $\rho_m = \sum_{i=1}^{N_s} E[|T_m(i)|^2] / \tau$  y la densidad espectral de potencia de ruido es denotada como  $\eta = E[|N_k(i)|^2]$ . El detector de energía del sensor  $k$  estima el nivel de energía normalizado por la densidad espectral de potencia de ruido, el cual se denota  $x_k$ , a partir de las muestras de la señal como

$$x_k = \frac{2}{\eta} \sum_{i=1}^{N_s} |Z_k(i)|^2.$$

Todos los sensores una vez han estimado su nivel de energía, lo envían a un centro de fusión que construye el “vector de energía” (Ver figura 2.1),

$$\mathbf{x} = (x_1, x_2, \dots, x_K)^T$$

donde la componente k-ésima de vector indica el nivel de energía estimado por el sensor k-ésimo.

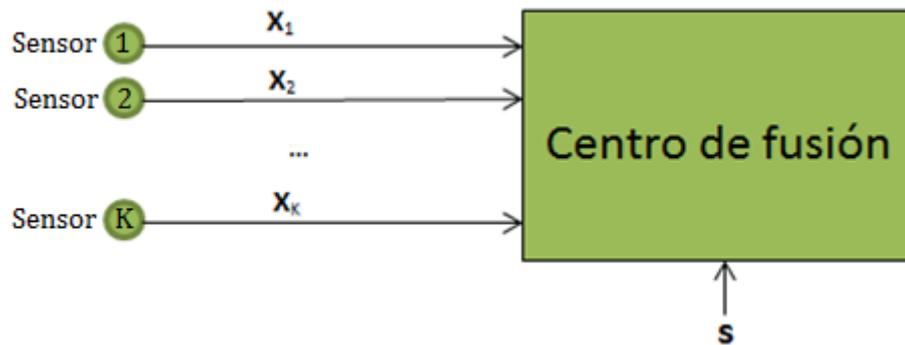


Figura 2.1. El centro de fusión recibe los niveles de energía de los sensores y la clase a la que éstos pertenecen.

Los niveles de energía recibidos por los sensores siguen una distribución Chi-Cuadrado no centrada con  $2N_s$  grados de libertad y un parámetro de centralidad,

$$\zeta_k = \frac{2\tau}{\eta} \sum_{m=1}^M s_m g_{m,k} \rho_m, \quad (1)$$

donde  $g_{m,k}$  es la ganancia de potencia del canal entre el usuario  $m$  y el sensor  $k$  (Ver figura 2.2), de tal manera que  $g_{m,k} = |h_{m,k}|^2$ . [7]

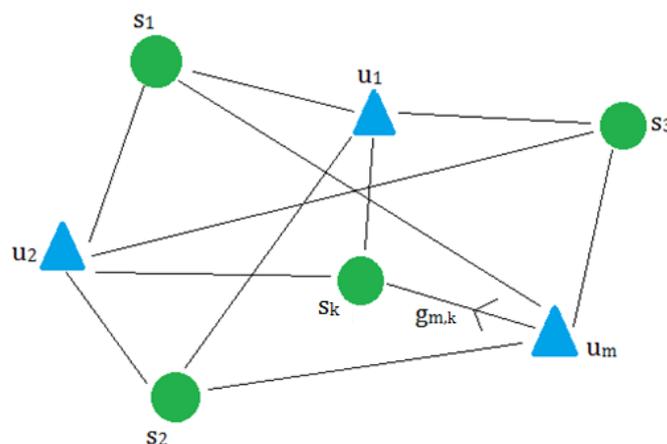


Figura 2.2. Entorno de propagación constituido por usuarios y sensores, existiendo un canal entre cada par usuario-sensor.

Si el número de muestras  $N_s$  es suficientemente grande, la distribución del nivel de energía  $x_k$  puede aproximarse por una distribución gaussiana de media  $\mu_k$  y varianza  $\sigma_k^2$ , [7]

$$x_k \sim N(\mu_k, \sigma_k^2),$$

donde,

$$\mu_k = 2N_s + \zeta_k,$$

$$\sigma_k^2 = 2(2N_s + 2\zeta_k) = 4N_s + 4\zeta_k.$$

Por lo tanto, el vector de energía  $\mathbf{x}$  sigue una distribución gaussiana multivariable:

$$\mathbf{x} \sim N(\boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$$

Con el vector de medias y la matriz de covarianza siguientes:

$$\boldsymbol{\mu}_x = (\mu_1, \dots, \mu_K)^T,$$

$$\boldsymbol{\Sigma}_x = \text{diag}(\sigma_1^2, \dots, \sigma_K^2),$$

donde  $\text{diag}(\sigma_1^2, \dots, \sigma_K^2)$ , denota la matriz diagonal cuyos elementos diagonales son  $(\sigma_1^2, \dots, \sigma_K^2)$ .

## 2.3. Modelo canal radio

Para la caracterización del canal radio se han desarrollado un gran número de aproximaciones. Entre las más comunes se encuentran las técnicas de trazado de rayos, los modelos analíticos basados en medidas empíricas o los modelos estadísticos. [11]

En su caracterización se distinguen tres escalas distintas de variación del canal con el tiempo (Ver figura 2.3): las pérdidas del trayecto, el desvanecimiento por sombra y el desvanecimiento multicamino.

- Las pérdidas del trayecto (Path Loss), para una frecuencia dada, se deben exclusivamente a la distancia entre el usuario y el sensor.
- El desvanecimiento por sombra (shadowing), se debe al bloqueo, por parte de algunos obstáculos, del camino que recorre la señal entre el usuario y el sensor.
- El desvanecimiento multicamino (fast fading), se debe al movimiento normalmente del usuario o de los elementos del entorno de propagación, produciendo reflexiones o dispersiones de la señal, llegando múltiples componentes de la señal transmitida a los sensores.

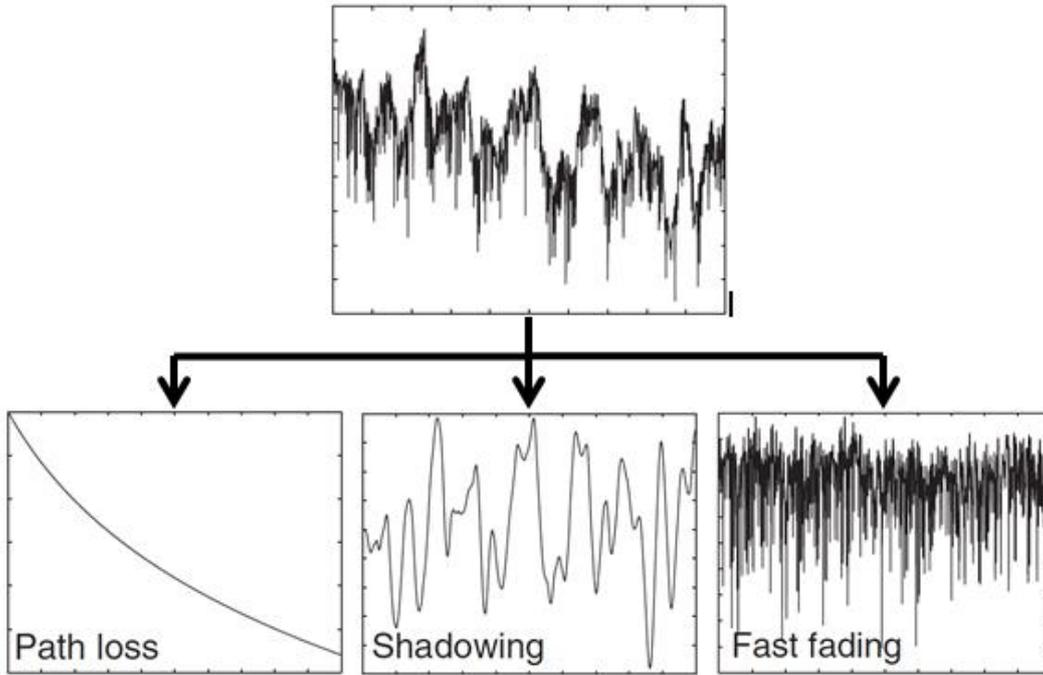


Figura 2.3. Las tres escalas de variación del canal radio en función de la distancia.

De esta forma la ganancia potencia del canal entre el usuario  $m$  y el sensor  $k$  viene dada por:

$$g_{m,k} = PL_{m,k} \psi_{m,k} \upsilon_{m,k} \quad (2)$$

Donde  $PL_{m,k}$  es la componente de las pérdidas del trayecto entre el usuario  $m$  y el sensor  $k$ ,  $\psi_{m,k}$  es la componente del desvanecimiento por sombra y  $\upsilon_{m,k}$  es la componente de desvanecimiento por multicamino. [11] [12]

### 2.3.1. Caracterización de las pérdidas del trayecto

En el problema que se está abordando, para caracterizar las pérdidas del trayecto se usará un modelo exponencial simplificado:

$$PL = \frac{P_r}{P_t} = K \left[ \frac{d_0}{d} \right]^\gamma.$$

En este modelo,  $K$  es una constante adimensional que depende de las características de las antenas y de la atenuación media del canal,  $d_0$  es una distancia de referencia para las antenas en campo lejano,  $\gamma$  es el exponente de pérdidas, y  $d$  la distancia del trayecto, para una potencia de transmisión  $P_t$ . Los valores de  $K$ ,  $d_0$  y  $\gamma$  se pueden obtener para aproximar tanto un modelo analítico como uno empírico. Cuando se simplifica el modelo para aproximar medidas empíricas el valor de  $K$  se suele fijar como:

$$K = \left( \frac{\lambda}{4\pi d_0} \right)^2,$$

Esta suposición es respaldada por datos empíricos. El valor de  $\gamma$  depende del entorno de propagación [11], como indica en la tabla 2.1.

Entorno	Rango $\gamma$
Macroceldas urbanas	3.7 – 6.5
Microceldas urbanas	2.7 – 3.5
Edificio de oficinas (misma planta)	1.6 – 3.5
Edificio de oficinas (varias plantas)	2 – 6
Almacén	1.8 – 2.2
Fábrica	1.6 – 3.3
Casa	3

Tabla 2.1. Rango del exponente de pérdidas en función del entorno de propagación.

De esta forma las pérdidas del trayecto resultan:

$$PL = \left( \frac{\lambda}{4\pi d_0} \right)^2 \left[ \frac{d}{d_0} \right]^{-\gamma} \quad (3)$$

### 2.3.2. Caracterización del desvanecimiento por sombra

En la caracterización del desvanecimiento por sombra hay que tener en cuenta que la variación ocurre sobre distancias comparables a anchos de edificios o colinas, y por tanto, su variabilidad con el tiempo será del orden de los segundos o decenas de segundos. [11][12]

### 2.3.3. Caracterización del desvanecimiento multicamino

Para caracterizar, por su parte, el desvanecimiento multicamino (Figura 2.4), las estadísticas de desvanecimiento proporcionan una probabilidad a la cual la señal está por encima o por debajo de un determinado nivel. Sin embargo, no dicen nada sobre la rapidez con la que varía la señal entre dos niveles de señal. Estos efectos son caracterizados por las estadísticas de desvanecimiento de segundo orden.

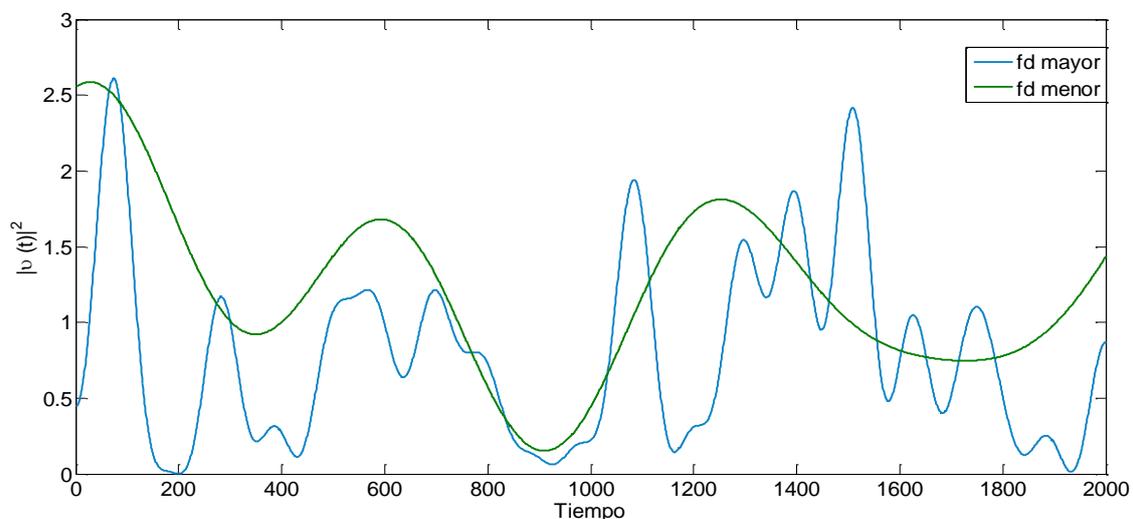


Figura 2.4. Evolución de la respuesta del canal  $v$ , a lo largo del tiempo, entre usuarios y sensores.

Uno de esos efectos, es el efecto Doppler. El efecto Doppler resulta en un cambio en la frecuencia aparente de la onda recibida, proporcional a la componente de la velocidad del usuario o del entorno de propagación. Si el movimiento se produce en la dirección de la onda, el canal existente varía más rápidamente, y por tanto la frecuencia aparente de la onda aumenta, sin embargo, si el movimiento es en sentido contrario, el canal varía más lentamente, así que la frecuencia aparente de la onda disminuye.

Cuando ocurre propagación multicamino, las ondas llegan de distintas direcciones, cada una de las cuales tiene su propia frecuencia Doppler asociada. De forma que el ancho de banda de la señal recibida está ensanchado respecto al de la señal transmitida. Este es el fenómeno de dispersión Doppler. [11][12]

Otra manera de ver el efecto de la dispersión Doppler es en el dominio del tiempo. A través del cual se puede definir el tiempo de coherencia  $T_c$  de un canal como el tiempo sobre el cual el canal se puede asumir constante. Este tiempo de coherencia es inversamente proporcional al desplazamiento Doppler del canal.

$$T_c \propto \frac{1}{f_d}$$

Dicho parámetro  $T_c$ , aporta una visión de la variabilidad con el tiempo de este tipo de desvanecimientos, siendo normalmente del orden de las decenas de milisegundos.

### 3. FORMULACIÓN COMO PROBLEMA DE CLASIFICACIÓN SUPERVISADA

La clasificación supervisada es la tarea de machine learning de obtener una función que proporciona la clase,  $y$ , a la que pertenece cada dato  $x$ :  $y(x)$ .

La forma de la función  $y(x)$  es determinada durante la fase de entrenamiento, también conocida como fase de aprendizaje, sobre la base de datos de entrenamiento  $D = \{\mathbf{X}, \mathbf{Y}\}$ . Dicha base de datos de entrenamiento está compuesta por los vectores de características o vectores de energía,  $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ , donde  $\mathbf{x}_n \in \mathbb{R}^{K \times 1}$ , y sus etiquetas correspondientes  $\mathbf{Y} = \{y_n\}_{n=1}^N$ , donde  $y_n \in \{1, 2, \dots, C\}$ , siendo  $C$  el número de clases posibles de los datos, los posibles estados de la red. Una vez el modelo está entrenado, la función puede determinar la clase de nuevos datos. Esta fase se conoce como fase de decisión.

Como ya se comentó, los vectores de características o vectores de energía,  $\mathbf{X}$ , del conjunto de datos,  $D$ , siguen una distribución gaussiana si el número de muestras  $N_s$  es suficientemente grande. Dichos vectores de energía se construyen en el centro de fusión con los niveles de energía enviados por cada sensor (Ver figura 3.1). Por su parte, el usuario activo en cada dato viene dado por la clase  $y$ , o por el vector  $\mathbf{s}$ , con el estado de la red.

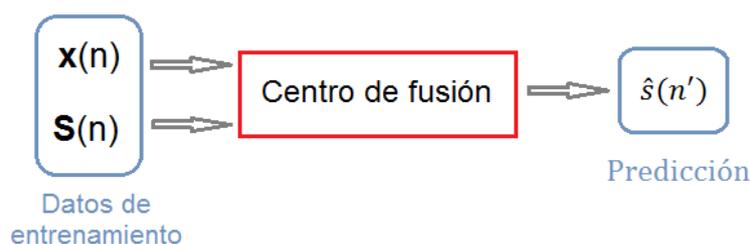


Figura 3.1. Con el conjunto de datos de entrenamiento,  $D$ , a través del centro de fusión que construye el vector de energía, se hace la predicción sobre los nuevos datos sin clasificar.

Además dichos vectores de energía,  $\mathbf{X}$ , están normalizados y escalados por un factor  $1/N_s$ .

En un caso concreto en el que el número de sensores fuera  $K=2$ , de forma que los vectores de energía se puedan dibujar en el plano, y el número de usuarios fuera  $M=3$ , de forma que hubiera  $C=4$  posibles clases. Teniendo  $N=1000$  realizaciones de los vectores de energía, con una potencia media de todos los usuarios,  $P= 0.2$  W,  $N_s=500$  muestras en los detectores de energía de todos los sensores, y una densidad espectral de potencia de ruido dependiente de la SNR.

SNR la cual se define como  $SNR=P/\eta$ , es decir, sin tener en cuenta los canales existentes entre usuarios y sensores.

Analizando distintos valores de la SNR, se obtendrían las siguientes realizaciones de los vectores de energía (Figuras 3.2, 3.3, 3.4), en las que se observan los clusters de las cuatro posibles clases de los datos.

Con una SNR de 6 dB:

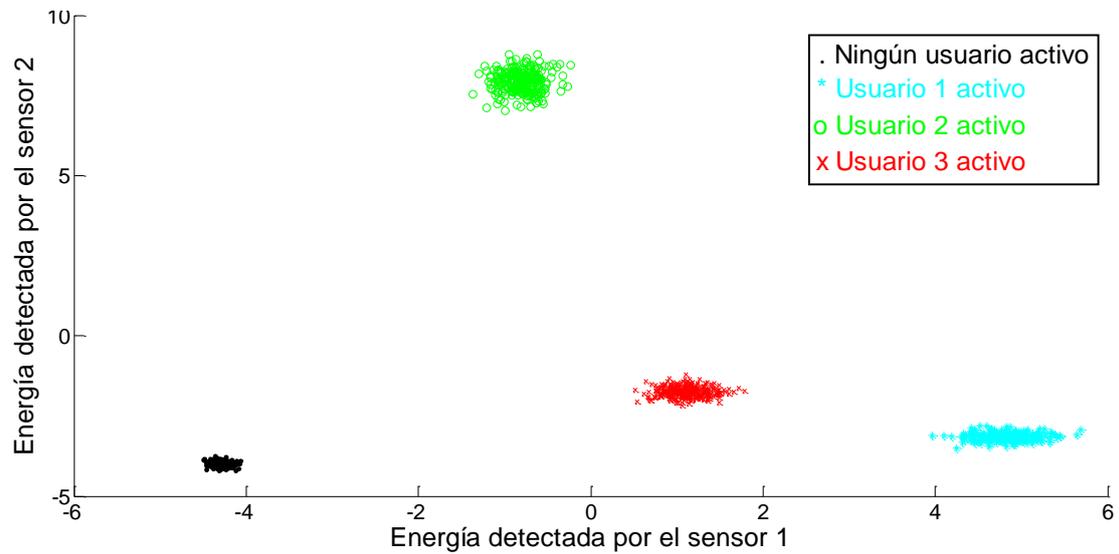


Figura 3.2. Realizaciones de los vectores de energía para una SNR=6 dB.

En un peor caso, para una SNR de -6.5 dB:

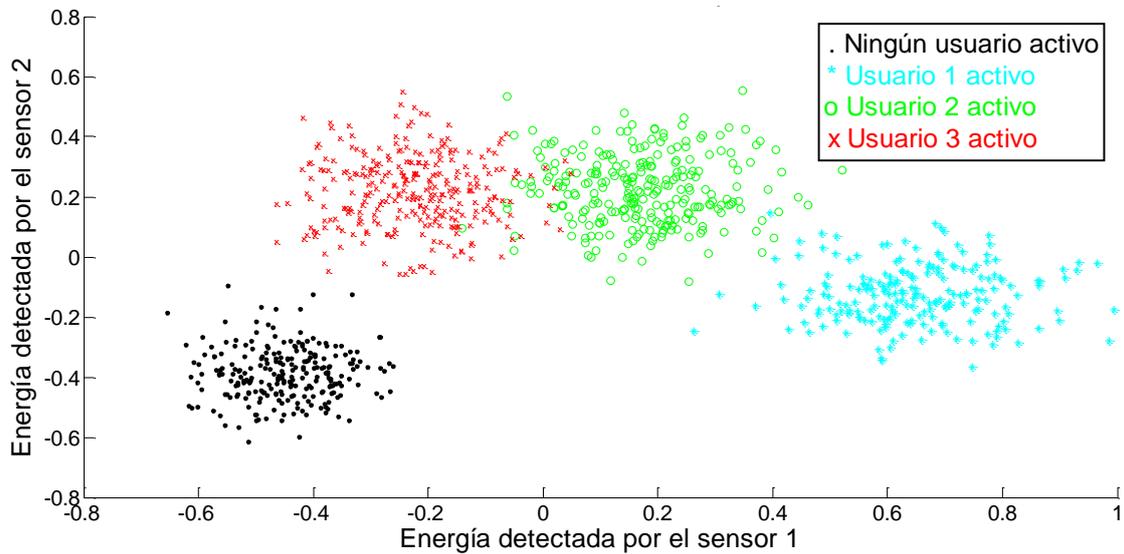


Figura 3.3. Realizaciones de los vectores de energía para una SNR=-6.5 dB.

Y en otro caso con una SNR=20 dB:

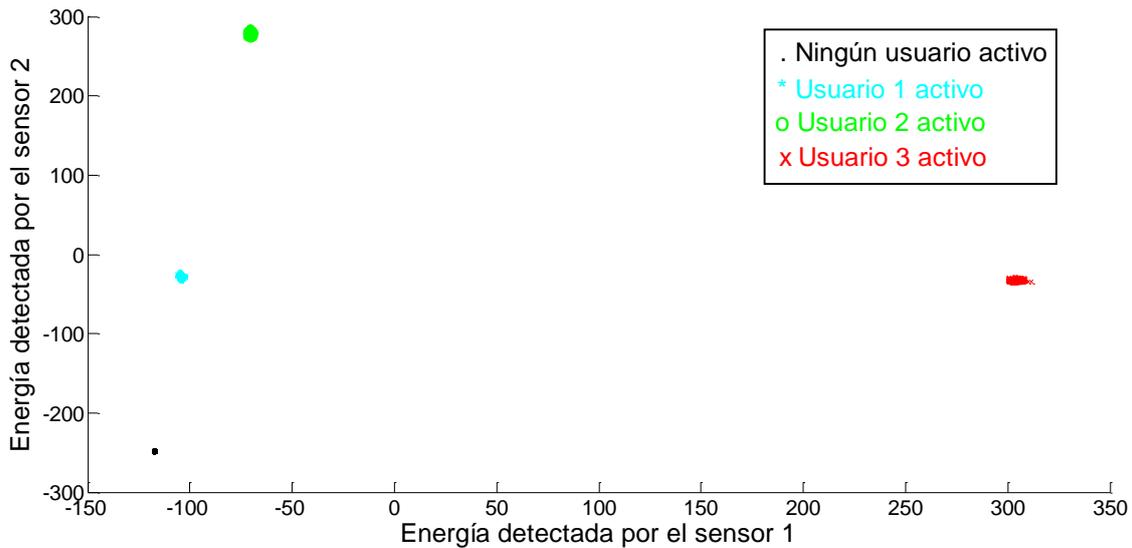


Figura 3.4. Realizaciones de los vectores de energía para una SNR=20 dB.

Comparación en la cual, se puede apreciar como la media y la varianza, gaussianas, de los vectores de energía,  $\mathbf{X}$ , se ven afectadas por la densidad espectral de potencia de ruido, debido al parámetro de centralidad,  $\zeta_k$  (1). A más ruido, menor  $\zeta_k$ , y por tanto las medias estarán más juntas, dando lugar a que las clases se superpongan independientemente de que la varianza también sea menor (Ver figura 3.3). Por el contrario, a menos ruido, mayor  $\zeta_k$  y consecuentemente, las medias serán mayores, al igual que las varianzas. Sin embargo, escalando, un menor ruido da lugar a mayores separaciones entre las distintas clases de datos con clusters menos dispersos (Ver figura 3.4).

### 3.1. Modelo logistic regression

Logistic regression (LR) es un modelo probabilístico discriminativo. La probabilidad a posteriori de una clase,  $C_k$ , es dada por una transformación softmax, o exponencial normalizada, de una función lineal de  $x$ . [13]

$$p(y_n = c | \mathbf{x}_n, \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T \mathbf{x}_n)}{\sum_{s=1}^C \exp(\mathbf{w}_s^T \mathbf{x}_n)} \quad (4)$$

En el cual la búsqueda de los parámetros del modelo lineal se realiza con la maximización de la función de verosimilitud, definida a través de la distribución condicional del problema  $p(C_k | x)$ , que representa una forma de entrenamiento discriminativo.

En el caso del modelo probabilístico discriminativo LR, la probabilidad a posteriori viene dada por una transformación softmax de las funciones lineales de las variables características del problema.

Esas variables características del problema son los vectores de energía,  $\mathbf{X}$ , y sus clases,  $\mathbf{Y}$ , anteriormente definidos y los parámetros  $\mathbf{W}$ . Estos parámetros  $\mathbf{W}$ , forman una matriz

$\mathbf{W} \in \mathbb{R}^{K \times C}$ , en la cual cada columna está formada por un vector característico de cada clase  $\mathbf{w}_c$ , como se muestra a continuación:

$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2 \dots \mathbf{w}_C], \quad \mathbf{w}_c \in \mathbb{R}^{K \times 1}$$

Parámetros, además los cuales, se obtienen en la fase de entrenamiento del problema a partir del conjunto de datos, D. Con los cuales, en la fase de decisión, se obtiene la clase de los nuevos datos sin etiquetar, siendo la función  $y(\mathbf{x})$ :

$$\hat{y} = y(\mathbf{x}) = \operatorname{argmax}_c p(c|\mathbf{x}, \mathbf{W})$$

La frontera de decisión entre dos clases: c y d, será el conjunto de vectores,  $\mathbf{x}$ , para los cuales:

$$p(c|\mathbf{x}, \mathbf{W}) = p(d|\mathbf{x}, \mathbf{W}) \quad (5)$$

Considerando (4), (5) se reduce a:

$$\exp(\mathbf{w}_c^T \mathbf{x}) = \exp(\mathbf{w}_d^T \mathbf{x})$$

Como la función exponencial es monótona, se tiene que la frontera de decisión entre ambas clases es:

$$\mathbf{w}_c^T \mathbf{x} = \mathbf{w}_d^T \mathbf{x} \Rightarrow (\mathbf{w}_c - \mathbf{w}_d)^T \mathbf{x} = 0$$

De dicha frontera, se puede destacar que el clasificador obtenido es lineal, y que la frontera de decisión será un hiperplano de dimensión  $K-1$  en un espacio  $\mathbb{R}^K$ , de forma que si  $K=2$ , la frontera de decisión será una recta en un espacio de dimensión  $\mathbb{R}^2$ .

En el caso particular de  $K=2$  sensores y  $M=3$  usuarios, como el que se usó para mostrar las realizaciones de algunos datos del problema, con una SNR de 16 dB, y aplicando uno de los algoritmos que se explicarán más adelante, se obtuvieron los siguientes parámetros  $\mathbf{W}$ :

$$\mathbf{W} = \begin{pmatrix} 0.2903 & 21.3706 & -10.1576 & -11.5032 \\ -6.1118 & -10.5224 & 15.4057 & 1.2284 \end{pmatrix}$$

A partir de los cuales, se obtuvieron las siguientes fronteras de decisión para una realización del problema, en la cual se tuvo un porcentaje de acierto en la fase de decisión de 77.7167 %:

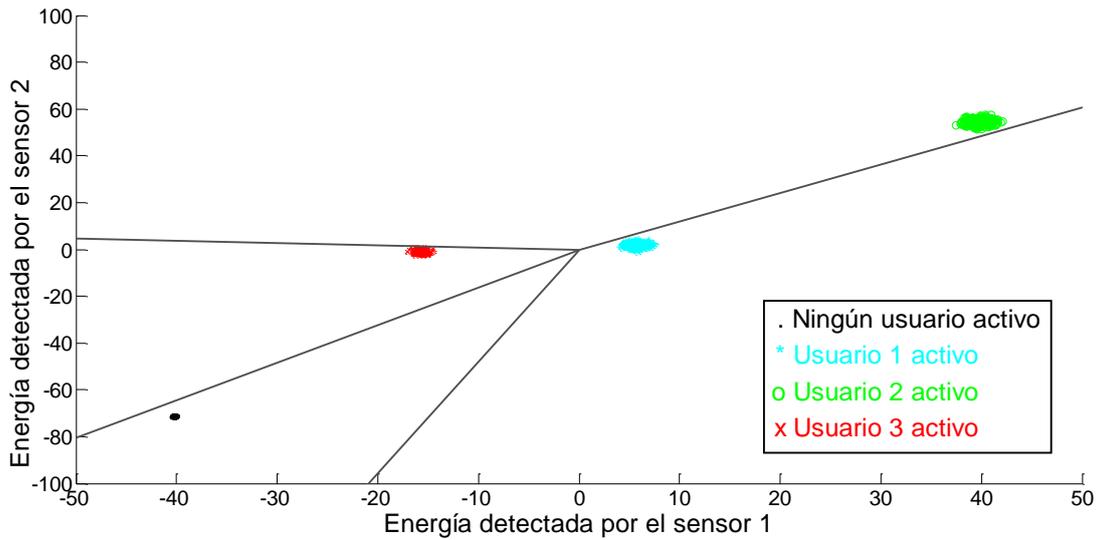


Figura 3.5. Fronteras de decisión para realizaciones de los vectores de energía con una SNR=16dB.

En la figura 3.5 se ve como a pesar de que los clusters de los datos no están muy dispersos, el porcentaje de acierto no es del todo bueno. Esto se debe a que existen dos clases de datos que son colineales. De forma que su frontera de decisión no separa bien esas dos clases, y por tanto, no será extraño que haya una equivocación en la clasificación entre datos de ellas.

Por otra parte, en el caso particular de  $K=2$  y  $M=2$ , de forma que  $C=3$  clases, con una SNR mucho más baja que en el caso particular anterior de  $-6.5$  dB, se obtuvo, de nuevo con uno de los algoritmos que se explicarán más adelante, los siguientes parámetros  $\mathbf{W}$ :

$$\mathbf{W} = \begin{pmatrix} -269.8313 & -270.2721 & 540.1034 \\ -54.8921 & -34.4430 & 89.3350 \end{pmatrix}$$

Parámetros con los cuales se obtiene un porcentaje de aciertos de 87.5833 %, y unas fronteras de decisión como las siguientes:

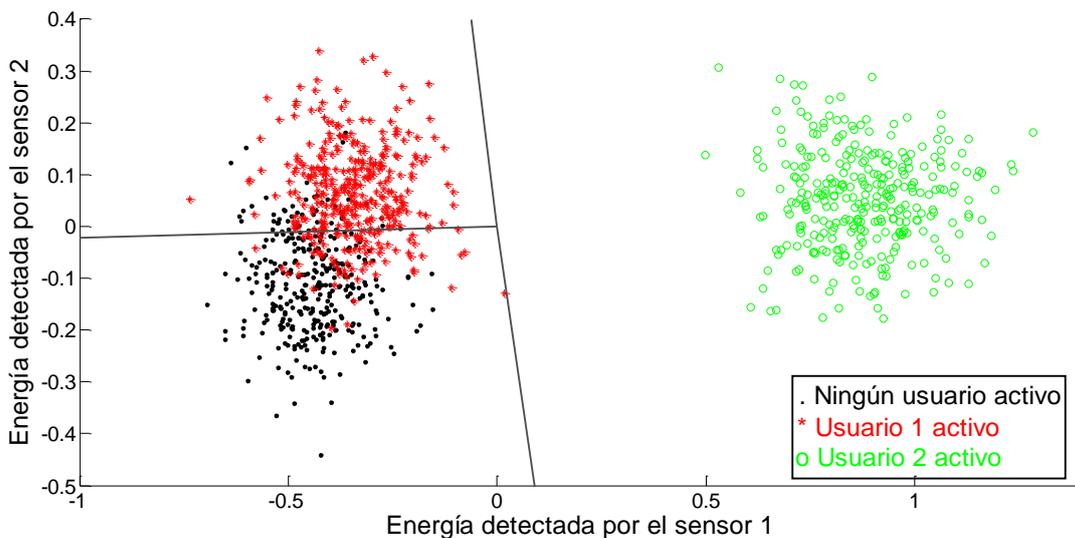


Figura 3.6. Fronteras de decisión para realizaciones de los vectores de energía con una SNR=-6.5dB.

En la figura 3.6, se ve como al disminuir la SNR, debido al aumento del ruido, la dispersión de los datos aumenta llegando a solaparse dos clases. Este hecho da lugar a errores, que hacen disminuir el porcentaje de acierto. Sin embargo, éste no cae tanto como en el caso de la figura 3.5, ya que no hay clases alineadas con el centro del eje, sino que hay bastante ángulo de diferencia entre las medias de las tres clases.

Para la realización de la figura 3.6 en concreto, se ha analizado la matriz de confusión,  $\mathbf{M}$ . En la que sus componentes  $\mathbf{M}_{i,j}$  representan, el número de veces que se ha etiquetado un dato como clase  $i$  cuando realmente pertenece a la clase  $j$ .

Sabiendo que se han etiquetado 6000 datos sin clasificar, se obtiene la siguiente matriz  $\mathbf{M}$ :

$$\mathbf{M} = \begin{pmatrix} 1751 & 495 & 3 \\ 247 & 1520 & 0 \\ 0 & 0 & 1984 \end{pmatrix}$$

Es decir, en esta realización, en 495 de los 6000 datos, se ha clasificado el dato como clase 1 siendo realmente de la clase 2, 3 datos se han clasificado como clase 1 siendo de la 3, y otros 247 datos se han clasificado como clase 2 perteneciendo a la clase 1. Esas 745 etiquetas erróneas son las que dan ese 12.4167 % de errores en la fase de decisión. Ocurrendo este fenómeno debido a que dichos datos caen al otro lado de la frontera de decisión de su clase con la de su etiqueta errónea.

## 3.2. Estimación de Máxima Verosimilitud

Para estimar los parámetros  $\mathbf{W}$  a partir de los datos de entrenamiento, se emplea el estimador de Máxima Verosimilitud (ML)[14]. Siendo la función de verosimilitud:

$$L(\mathbf{W}) = p(\mathbf{X}, \mathbf{Y}|\mathbf{W}) = p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{X}|\mathbf{W}) = p(\mathbf{Y}|\mathbf{X}, \mathbf{W})p(\mathbf{X})$$

donde se ha tenido en cuenta que los vectores  $\mathbf{X}$  son independientes de los parámetros del modelo. Y, en la cual, para calcular las probabilidades a posteriori correspondientes, se usarán las probabilidades condicionadas.

El estimador ML para los parámetros del modelo  $\mathbf{W}$  es:

$$\hat{\mathbf{W}}_{ML} = \operatorname{argmax}_{\mathbf{W}} L(\mathbf{W}) = \operatorname{argmax}_{\mathbf{W}} p(\mathbf{X}, \mathbf{Y}|\mathbf{W})$$

El logaritmo de la verosimilitud es:

$$Q(\mathbf{W}) = \log p(\mathbf{X}, \mathbf{Y}|\mathbf{W}) = \log p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) + \log p(\mathbf{X})$$

Y, teniendo en cuenta que la función logaritmo es monótona creciente, y que los vectores  $\mathbf{X}$  también son independientes entre ellos, de forma que,

$$p(\mathbf{Y}|\mathbf{X}, \mathbf{W}) = \prod_{n=1}^N p(y_n|\mathbf{x}_n, \mathbf{W}),$$

$$\begin{aligned}\hat{\mathbf{W}}_{\text{ML}} &= \operatorname{argmax}_{\mathbf{W}} Q(\mathbf{W}) = \operatorname{argmax}_{\mathbf{W}} \left[ \sum_{n=1}^N (\log p(y_n | \mathbf{x}_n, \mathbf{W}) + \log p(\mathbf{x}_n)) \right] \\ &= \operatorname{argmax}_{\mathbf{W}} \sum_{n=1}^N \log p(y_n | \mathbf{x}_n, \mathbf{W}) = \operatorname{argmax}_{\mathbf{W}} \sum_{n=1}^N q_n(\mathbf{W}),\end{aligned}$$

Si se denominan los términos logarítmicos del sumatorio como  $q_n(\mathbf{W})$  y se desarrollan dichos términos a partir de la probabilidad a posteriori definida por el modelo (4) para las variables características del problema, se obtiene:

$$q_n(\mathbf{W}) = \log p(y_n | \mathbf{x}_n, \mathbf{W}) = \mathbf{w}_{y_n}^T \mathbf{x}_n - \log \left( \sum_{s=1}^C \exp(\mathbf{w}_s^T \mathbf{x}_n) \right).$$

### 3.3. Algoritmo Gradient Ascent

El algoritmo Gradient Ascent (GA) es un algoritmo de optimización de primer orden, el cual para buscar el máximo local de una función diferenciable utiliza el ascenso por gradiente. Para ello, empieza con una estimación inicial de la solución  $\mathbf{W}^{(0)}$  y calcula el gradiente de la función en el punto, repitiendo el proceso con pasos en la dirección positiva del gradiente hasta encontrar el máximo local. [13]

La fórmula del algoritmo GA para calcular el parámetro  $\mathbf{W}$  resulta:

$$\mathbf{W}^{(i)} = \mathbf{W}^{(i-1)} + \varphi(\mathbf{W}^{(i-1)}) * \nabla_{\mathbf{W}} Q(\mathbf{W}^{(i-1)}), \quad i = 1, 2, \dots \quad (6)$$

Siendo  $\varphi(\mathbf{W})$  un parámetro de diseño del modelo, la constante de aprendizaje, que se debe ajustar para obtener el mejor funcionamiento del algoritmo en función de los datos.  $\varphi(\mathbf{W})$ , en este problema, se obtiene de acuerdo al método Quasi-Newton BFGS. [15]

Además, al algoritmo GA es un algoritmo bloque (batch). Los algoritmos bloque son algoritmos iterativos que se caracterizan por procesar en cada iteración todo el conjunto de datos  $D$ .

En este problema, al usar el modelo LR, se puede demostrar que la función objetivo,  $Q(\mathbf{W})$ , es convexa [13], por lo que siempre se encontrará el máximo de dicha función, siempre y cuando, el valor de la constante de aprendizaje del algoritmo sea lo suficientemente pequeña. Cabe destacar que este hecho sólo se cumple cuando se elige el modelo LR, una de las principales razones por las cuales se ha elegido LR como modelo para este problema.

Además,  $\nabla_{\mathbf{W}} Q(\mathbf{W}) = [\nabla_{\mathbf{w}_1} Q(\mathbf{W}), \nabla_{\mathbf{w}_2} Q(\mathbf{W}) \dots \nabla_{\mathbf{w}_c} Q(\mathbf{W})]$ , es la matriz del gradiente de la función objetivo del problema  $Q(\mathbf{W})$ ,

$$\nabla_{\mathbf{W}} Q(\mathbf{W}) = \nabla_{\mathbf{W}} \log p(\mathbf{Y} | \mathbf{X}, \mathbf{W}) = \nabla_{\mathbf{W}} \sum_{n=1}^N q_n(\mathbf{W})$$

Siendo el gradiente de  $q_n(\mathbf{W})$ , según se demuestra en el Apéndice 1:

$$\nabla_{\mathbf{w}_c} q_n(\mathbf{W}) = \frac{1}{p(y_n | \mathbf{x}_n, \mathbf{W})} * \nabla_{\mathbf{w}_c} p(y_n | \mathbf{x}_n, \mathbf{W}),$$

$$\nabla_{\mathbf{w}_c} p(y_n | \mathbf{x}_n, \mathbf{W}) = \mathbf{x}_n p(c | \mathbf{x}_n, \mathbf{W}) [I(y_n = c) - p(y_n | \mathbf{x}_n, \mathbf{W})].$$

Otro hecho a destacar de la fórmula del algoritmo GA es su criterio de parada, es decir, hasta cuando hay que seguir ejecutando el algoritmo. En este caso el criterio de parada será el llegar al máximo:

$$|Q(\mathbf{w}^{(i)}) - Q(\mathbf{w}^{(i-1)})| < \varepsilon.$$

### 3.4. Algoritmo Stochastic Gradient Ascent

El algoritmo batch GA tiene ciertas limitaciones. Una muy importante es que se debe disponer del conjunto de datos de entrenamiento,  $D$ , completo en memoria. Esto es debido a que en cada iteración del algoritmo se hace uso de todos ellos. Otra limitación, es que el modelo (4) es el mismo para todos los datos, es decir, todos los datos han sido obtenidos bajo las mismas condiciones.

Estas limitaciones, son la razón por las que GA no funciona cuando no se tienen todos los datos en memoria o cuando el modelo no es el mismo para todos los datos. El hecho de tener todos los datos en memoria implica procesar grandes conjuntos de datos, suponiendo un gran coste computacional, que no siempre se puede asumir. Por otra parte, el hecho de que el modelo tenga que ser igual para todos los datos implica que, para este problema por ejemplo, no se pueda usar GA en entornos con canales variantes con el tiempo. Hecho bastante habitual en redes de comunicaciones inalámbricas.

Ambas limitaciones se pueden solventar usando el algoritmo Stochastic Gradient Ascent (SGA), el cual admite los datos de entrenamiento según lleguen secuencialmente y que el modelo (4) para cada uno de ellos sea diferente.

De esta forma, el algoritmo SGA es un algoritmo adaptativo (online), los cuales a diferencia de los algoritmos bloque se caracterizan por procesar un dato o conjunto de datos en cada iteración del algoritmo.

Así, la fórmula del algoritmo SGA para la obtención de los parámetros  $\mathbf{W}$  del modelo es:

$$\mathbf{W}^{(n)} = \mathbf{W}^{(n-1)} + \varphi_n * \nabla_{\mathbf{W}} q_n(\mathbf{W}^{(n-1)}), \quad n = 1, \dots, N. \quad (7)$$

Parámetros los cuales se inicializan  $\mathbf{W}^{(0)}$  y en cada paso del algoritmo se actualizan con un único par de datos de entrenamiento cada vez. Cada una de estas actualizaciones, en las que  $n$  es el índice a las observaciones, se realiza a partir del cálculo del gradiente para dichos datos y del ajuste de la constante de aprendizaje,  $\varphi_n$ .

El cálculo de gradiente de  $q_n(\mathbf{W})$  se realiza según lo demostrado en el Apéndice 1.

Por su parte, el ajuste de la constante de aprendizaje,  $\varphi_n$ , requiere un compromiso entre la velocidad de convergencia de los parámetros y la precisión en su ajuste. Para ello, se ha fijado la siguiente fórmula para la constante de aprendizaje:

$$\varphi_n = \frac{1}{n^\alpha}, \quad 0 \leq \alpha \leq 1$$

De forma que en las primeras iteraciones  $n$  es pequeña y por lo tanto la contribución de los primeros datos es relativamente grande para acercarse más rápido a un valor aproximado de convergencia de  $\mathbf{W}$ . Y que a medida que se van realizando más iteraciones esa contribución va disminuyendo para realizar el ajuste más fino y preciso de  $\mathbf{W}$ . Ya que si el valor de  $\varphi_n$  es muy grande se da mucha importancia al dato  $(\mathbf{x}_n, y_n)$  frente a los anteriores, pero si por el contrario  $\varphi_n$  es muy pequeño, apenas se da importancia frente a los anteriores. Sin embargo, al igual que pasa en el algoritmo GA, se debe ajustar el parámetro de diseño del modelo,  $\alpha$ , para obtener el mejor funcionamiento del algoritmo posible en función de los datos.



## 4. RESULTADOS Y CONCLUSIONES

### 4.1. Entorno de propagación

Se considera como caso particular una red de comunicaciones compuesta por  $M$  usuarios ubicados aleatoriamente sobre un espacio bidimensional.

En el entorno de la red de comunicaciones, hay una red de  $K$  sensores colocados formando una rejilla cuadrada, con una distancia mínima entre ellos de  $d_{\min}$ .

La figura 4.1 muestra un ejemplo de un entorno de propagación con  $M=3$  usuarios y  $K=4 \times 4=16$  sensores separados  $d_{\min}=500$  m.

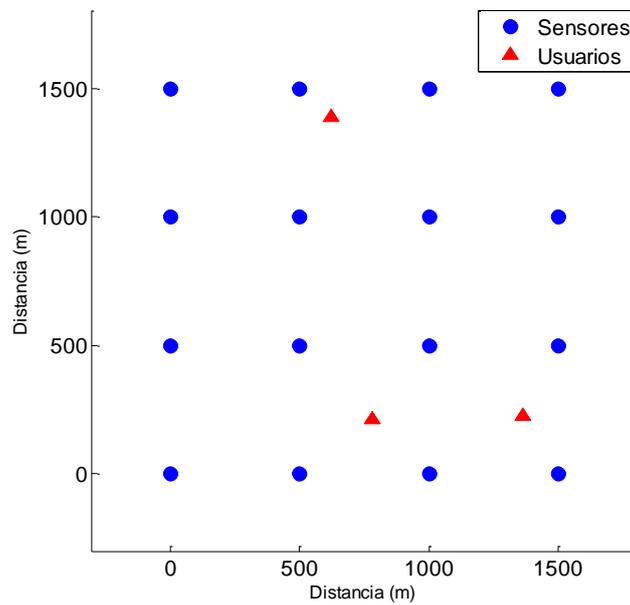


Figura 4.1. Entorno de propagación del problema con los usuarios situados aleatoriamente.

### 4.2. Caracterización canal entre usuarios y sensores

El canal entre cada pareja usuario-sensor se caracteriza como en la sección 2.3 (2).

$$g_{m,k} = PL_{m,k} \psi_{m,k} \upsilon_{m,k}$$

### 4.2.1. Las pérdidas del trayecto

Para la caracterización de las pérdidas del trayecto, se considerará una frecuencia central de  $f=1$  GHz, es decir, la longitud de onda será de  $\lambda=0.3$ m, y una distancia de referencia  $d_0=100$  m. Por tanto, la constante K será:

$$K = \left(\frac{\lambda}{4\pi d_0}\right)^2 = \left(\frac{0.3}{4\pi 100}\right)^2 \cong 5.7 \cdot 10^{-8}$$

Se considerará también un exponente de pérdidas  $\gamma$ , por lo que el Path Loss o las pérdidas del trayecto (3) entre cada pareja usuario-sensor será:

$$PL = 5.7 \cdot 10^{-8} \left[\frac{d}{100}\right]^{-\gamma},$$

Donde d es la distancia entre ellos.

### 4.2.2. El desvanecimiento por sombra

La captura de datos en nuestro problema se produce en un intervalo de tiempo muy breve, lo suficiente para que sea menor que el orden en el que varía el shadowing. Es por eso, que, durante esa captura, el shadowing se puede suponer constante, y de valor la unidad.

### 4.2.3.El desvanecimiento multicamino

Además de las pérdidas del trayecto, se considerarán las pérdidas de pequeña escala o multicamino.

Se supone que el canal sigue una distribución Rayleigh, por tanto,  $v$  sigue una distribución exponencial con media:

$$E[v] = 2\sigma^2 = 1$$

Es decir,

$$v \sim \exp(1)$$

Con una función densidad de probabilidad:

$$p_v(z) = e^{-z}, \quad z \geq 0$$

De esta forma la ganancia en potencia total del canal incluyendo los dos efectos significativos para este caso es:

$$g_{m,k} = PL_{m,k} u_{m,k}$$

En la resolución del problema se van a diferenciar dos tipos de canales. Canales invariantes y canales variantes con el tiempo.

Los canales variantes lo harán conforme al Modelo de Jake [16][11], con una correlación temporal determinada por  $f_d$ . Esta variación, además seguirá el modelo block-fading, es decir, el canal no cambiará durante la adquisición de las  $N_s$  muestras por parte de los sensores (tiempo  $\tau$ ).

$$\tau < T_c \propto \frac{1}{f_d}$$

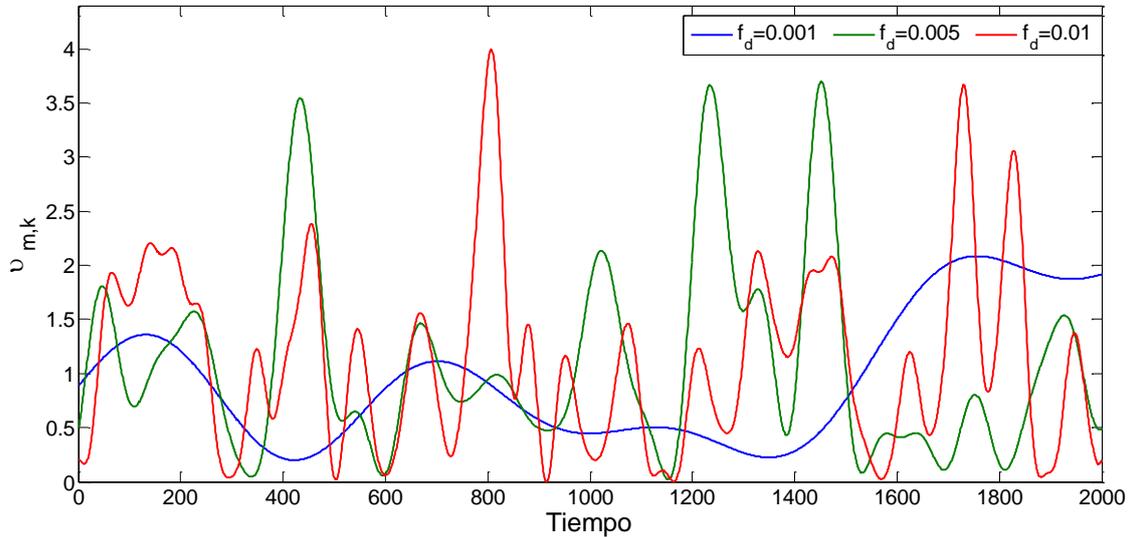


Figura 4.2. Evolución con el tiempo de tres canales con diferente  $f_d$ .

En la figura 4.9 se puede ver la evolución de tres canales con diferente grado de correlación temporal o distinto desplazamiento Doppler normalizado,  $f_d$ . Observándose como el canal de menor desplazamiento Doppler, es el que más lentamente varía y el que más correlación temporal tiene.

### 4.3. Modelado de la actividad de los usuarios

La secuencia de estados de la red se modela a través de una cadena de Markov. La transición de los estados de la red, en dicha secuencia, se suele representar mediante una matriz de transición, en la cual sus elementos,  $T_{i,j}$ , representan la probabilidad de pasar del estado, o clase,  $i$  al  $j$  de la red. En este caso, la matriz de transiciones de la red para el caso particular de un problema con cuatro clases ( $M=3$  usuarios), es:

$$T = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ pr & 1 - pr & 0 & 0 \\ pr & 0 & 1 - pr & 0 \\ pr & 0 & 0 & 1 - pr \end{pmatrix}$$

donde,  $pr$  es la probabilidad de un usuario de pasar de estar activo a inactivo. Siendo esa probabilidad igual para todos los usuarios.

Si se analizan los posibles valores que puede tomar  $pr$ , entre 0 y 1, y como afecta ese cambio al valor a las probabilidades de estado en el límite (Apéndice 2), se obtiene la siguiente gráfica (Figura 4.3):

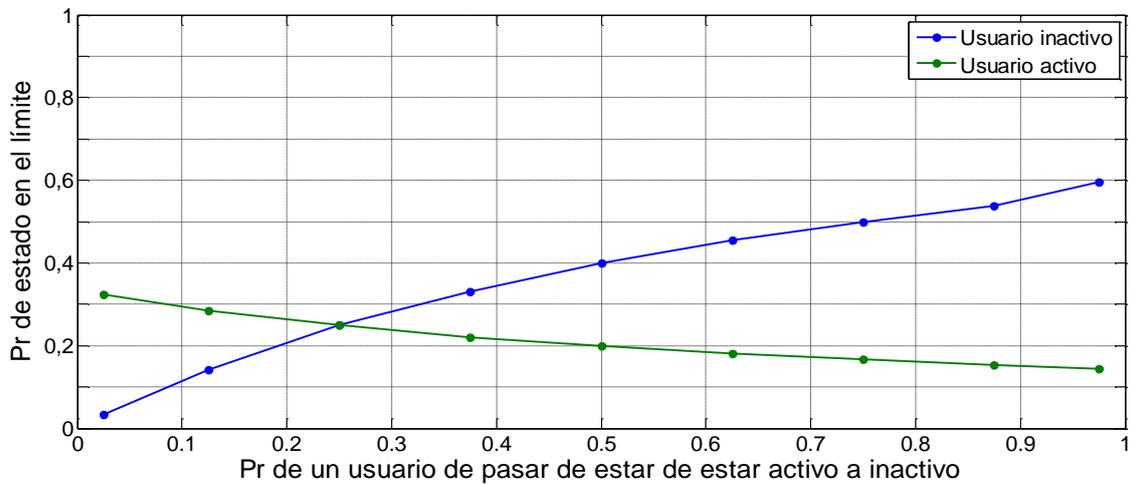


Figura 4.3. Probabilidad de estado en el límite de los estados del usuario en función de su probabilidad de pasar de estar activo a inactivo.

En la cual se observa que, si  $pr$  tiende a cero, la probabilidad en el límite del estado inactivo tiende también a cero y la del estado activo a  $1/3$ , existiendo mucha correlación temporal entre los estados de la red. Por el contrario, si  $pr$  tiende a 1, la probabilidad del inactivo tiende a  $4/7$  y la del activo a  $1/7$ , existiendo menor correlación temporal.

Ese grado de correlación temporal de la secuencia de estados, va a tener una gran influencia sobre  $\mathbf{Y}$  o  $\mathbf{S}$ , indistintamente. Ya que si la correlación temporal es muy fuerte, la probabilidad del estado inactivo es muy baja, y los periodos de transmisión de los usuarios serán largos, trasladándose a  $\mathbf{Y}$  o  $\mathbf{S}$  en pocos cambios de clase. En el caso contrario de que la correlación sea débil, la probabilidad del estado activo será baja, y los periodos de transmisión de los usuarios serán cortos, produciéndose muchos cambios de clase en  $\mathbf{Y}$  o  $\mathbf{S}$ .

## 4.4. Descripción de las simulaciones

El análisis de los resultados obtenidos para este problema, se realizan con una serie de simulaciones de Montecarlo. En cada una de esas simulaciones, se van a tener varios parámetros aleatorios, que van a cambiar de una simulación a otra:

- Los estados de la red representados bien mediante  $\mathbf{S} = \{\mathbf{s}_n\}_{n=1}^N$  o bien mediante  $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1}^N$ , los cuales en cada simulación se van a obtener a partir de las  $N$  realizaciones de la matriz de transiciones
- La posición de los usuarios, van a ser una realización diferente del entorno de propagación en cada simulación. Además, al variar las posiciones, también variarán las distancias entre las parejas usuario-sensor, cambiando el PL con ellas

- Las componentes de fast fading del canal,  $\mathbf{v} = \{v_{m,k}\}_{m=1:M, k=1:K}$ , las cuales en cada simulación van a ser una realización de los canales
- Los vectores de energía,  $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ , los cuales en cada simulación van a ser una realización calculada a partir de su distribución Chi-cuadrado, y van a incluir a los tres procesos aleatorios anteriores

Para realizar las simulaciones, se supondrán una series de valores para los parámetros del problema, los cuales se usarán en todas la simulaciones salvo que se diga lo contrario.

- Parámetros de la red de comunicaciones:
  - Número de usuarios:  $M=3$
  - Probabilidad de los usuarios de pasar de estar activos a inactivos (T):  $pr=0.25$
  - Potencia de transmisión de todos los usuarios:  $P=0.05 \text{ W}$
- Parámetros de la red de sensores:
  - Número de sensores  $K=16$
  - Distancia mínima entre los sensores en una rejilla rectangular como se muestra en la figura 4.1:  $d_{\min}=500\text{m}$
  - Número de muestras de los detectores de energía para todos los sensores:  $N_s=500$
- Parámetros del modelo de canal:
  - Exponente de pérdidas del Path Loss:  $\gamma=3.7$
  - Densidad espectral de ruido:  $\eta=1e-9 \text{ W/Hz}$
- Parámetros de las simulaciones
  - Número de datos de entrenamiento:  $N=1000$
  - Número de simulaciones de Montecarlo:  $N_{\text{sim}}=200$

Con estos parámetros, en cada una de las simulaciones de Montecarlo se van a seguir los siguientes pasos:

1. Se genera el conjunto de vectores  $\mathbf{X}$  y sus etiquetas  $\mathbf{Y}$ , con todo lo explicado en los capítulos 2 y 3 y los parámetros del problema anteriormente definidos.
2. Fase de entrenamiento: Se le pasan los datos al algoritmo correspondiente, (6) o (7), que entrenará los parámetros  $\mathbf{W}$ .
3. Fase de decisión: con los  $\mathbf{W}$  obtenidos, el clasificador proporciona una etiqueta a unos nuevos datos sin clasificar, obtenidos de la misma forma que los de entrenamiento.

Sin embargo, para probar la calidad del clasificador en las simulaciones, los datos de test no son exactamente datos sin clasificar. Dichos datos también se generan junto a su etiqueta  $\mathbf{Y}$ , la diferencia es que su etiqueta no se utiliza en el proceso de decisión, sólo en el procesado de los resultados. Proceso en el cual, las etiquetas reales obtenidas se comparan con las etiquetas proporcionadas por el clasificador. De la comparación de dichas etiquetas, se obtienen los errores que se han producido en la fase de decisión. Errores los cuales se cuentan, y se calcula el porcentaje que ese número de fallos supone sobre el total

de los datos de test. Valor que también nos proporciona el porcentaje de aciertos que se ha obtenido con el clasificador, el cual a partir de ahora se denominará precisión.

Precisión, la cual, normalmente se calcula sobre el conjunto de  $N_{sim}$  simulaciones. Ya que una realización específica puede ser buena, mala, regular... Es decir, los resultados de una sola realización no representan a los del algoritmo. Es por eso, que los resultados y la precisión se promedian sobre el conjunto de las  $N_{sim}$  simulaciones, donde se tendrán realizaciones de todos los tipos, representando, así, verdaderamente al algoritmo.

## 4.5. Resultados sobre canal invariante / Algoritmo Bloque

Un canal invariante con el tiempo es aquel en el que durante la captura del conjunto de datos de entrenamiento el canal es constante, es decir, el tiempo de captura de los  $N$  datos de entrenamiento es menor que el tiempo de coherencia del canal.

Los problemas los cuales muestran un canal invariante se resuelven mediante algoritmos bloque (batch), en este caso mediante el algoritmo GA.

### 4.5.1. Efecto del exponente de pérdidas, $\gamma$ , sobre la precisión

El primer efecto que se va a analizar va a ser el efecto que tiene el exponente de pérdidas,  $\gamma$ , de la componente PL (3) del canal, sobre la precisión. Dicho parámetro se va a analizar en un rango de 2 a 4, ya que en ese rango se encuentran los valores típicos de la mayoría de los entornos (tabla 2.1).

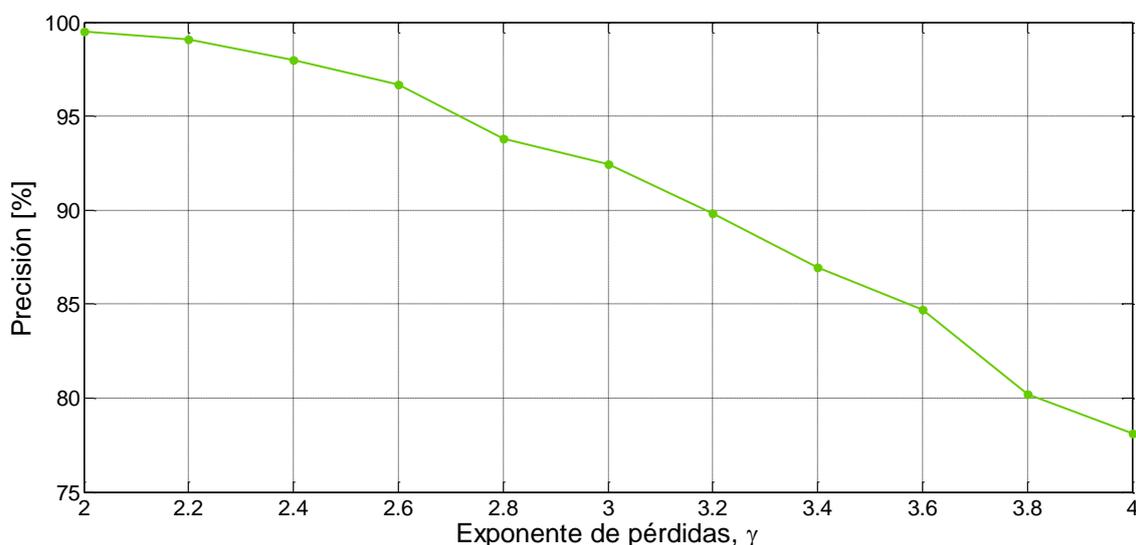


Figura 4.4. Acierto en la identificación de la actividad de los usuarios en función del exponente de pérdidas,  $\gamma$ .

En la gráfica (Figura 4.4) se puede comprobar como para exponentes bajos, la precisión es mayor que para exponentes altos, con mejor PL. Es decir, la precisión al decidir será mejor sobre entornos como una única planta de un edificio de oficinas o un almacén que sobre

entornos de microceldas o de macroceldas urbanas. Esto es debido a que, un canal con un  $\gamma$  bajo tiene mejor componente PL que uno con un  $\gamma$  alto. Al tener una mejor componente de PL, la ganancia en potencia del canal también será mejor, aumentando el parámetro de centralidad,  $\zeta_k$  (1). Así, las medias de los datos  $\mathbf{X}$  serán mayores, estando más separadas entre ellas. De forma que, aunque la varianza de los datos también sea mayor, los clusters de los datos no llegan a solaparse. Hecho que si ocurrirá cuando  $\gamma$  sea alto, ya la componente de PL, y la ganancia en potencia del canal serán peores, siendo  $\zeta_k$  pequeña. Y por tanto, las medias de los datos  $\mathbf{X}$  estarán más juntas llegando a solaparse los clusters en algunas realizaciones, aunque la varianza de los datos sea baja. Esto ocurre, porque aunque las varianzas sean menores con menor  $\zeta_k$ , escalando los datos, los clusters son menos dispersos cuando es mayor, debido a la mayor distancia entre las medias.

Dicha disminución del valor de la precisión con el aumento del exponente implica que para entornos con  $\gamma=2$ , el número de aciertos en la fase de decisión será un 20 % más que para entornos con  $\gamma=4$ , siendo esa disminución más o menos lineal con  $\gamma$ , lo cual nos da una idea de cuánto afecta el exponente de pérdidas a la precisión del algoritmo.

Dicha gráfica, también da una visión de lo que ocurriría si en vez de variar el exponente de pérdidas, se variara la distancia entre usuarios y sensores, se produciría el mismo efecto en la fórmula del PL (3). Ya que, tanto si se aumenta el exponente de pérdidas, como si se aumenta la distancia, el PL empeora en ambos casos.

#### 4.5.2. Efecto del número de sensores sobre la precisión

El siguiente efecto a analizar va a ser el efecto que tiene el número de sensores existente,  $K$ , en el entorno de propagación, sobre la precisión. Dicho parámetro se va a variar entre dos y diez sensores en cada eje del entorno de propagación, o lo que es lo mismo, se va a variar la rejilla de sensores entre una de 2x2 a una de 10x10, manteniendo en todos los casos la distancia mínima entre sensores.

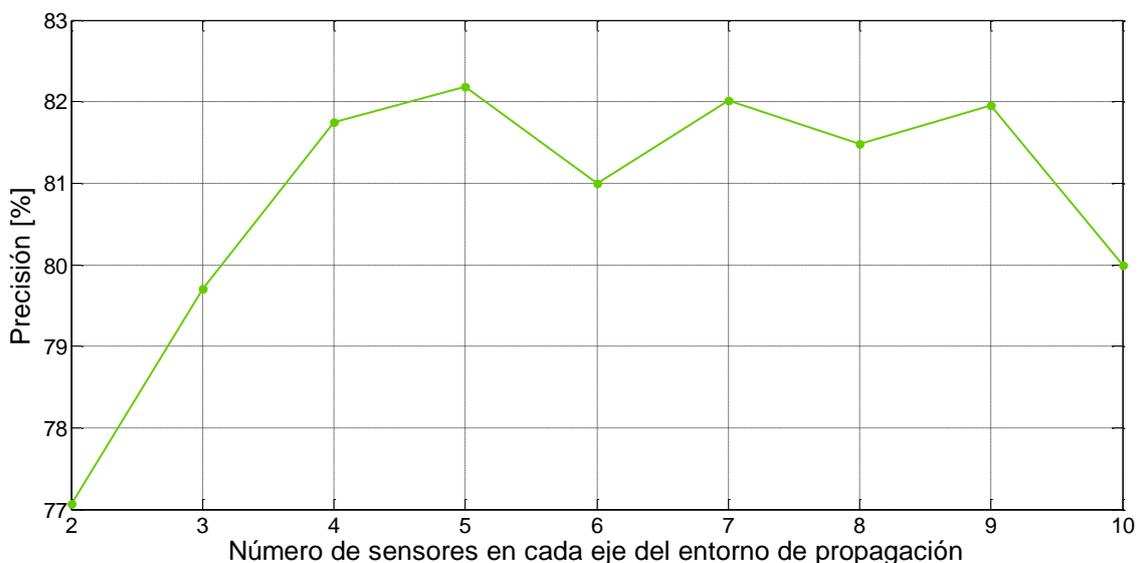


Figura 4.5. Acierto en la identificación de la actividad de los usuarios en función del número de sensores en cada eje del entorno de propagación.

En la gráfica (Figura 4.5) se puede observar como para pocos sensores se obtiene una menor precisión que para un mayor número de sensores. Esto es debido a que en rejillas pequeñas, los usuarios están más juntos y es más complicado diferenciar en  $\mathbf{X}$  cuál de ellos está transmitiendo. Mientras que en rejillas mayores, los usuarios ya tienen más espacio donde colocarse, por lo que los sensores podrán diferenciar mejor en  $\mathbf{X}$  cuál de ellos es el que está activo.

Una apreciación importante que se puede hacer de la gráfica, es que una vez llegado a un número de sensores lo suficientemente alto, la precisión se mantiene oscilando alrededor de un mismo valor aunque se siga subiendo el número de sensores. Esto ocurre debido a que a partir de los 16 sensores, los usuarios ya tienen suficiente espacio para colocarse en la rejilla de forma que los sensores detecten bien cuál de ellos está activo. Por lo que, analizando la gráfica, los mejores resultados en la fase de decisión se obtienen con una rejilla de 16 o 25 (4x4 o 5x5) sensores. Ya que con menos sensores la precisión es más baja. Y con más sensores, además de que no se obtiene una mejor precisión, la longitud de los  $\mathbf{X}$  será mayor, y con ello habrá más parámetros  $\mathbf{W}$  que estimar.

Por otra parte, cabe comentar que esa oscilación de la precisión a partir de los cuatro sensores en cada eje entre el 81% y 82 %, se debe en parte a lo comentado a cerca de la detección del usuario activo, pero también a la existencia de clases colineales. La existencia de esas clases en algunas de las realizaciones en las simulaciones, es la que no permite a la precisión subir de ese valor. Sin embargo, a pesar de esa oscilación se aprecia perfectamente el efecto que tiene el número de sensores en la precisión, siendo la variación entre el peor valor y éste una vez estabilizado de 4-5 %.

### **4.5.3. Efecto del número de muestras de energía, $N_s$ , sobre la precisión**

Otro efecto a analizar va a ser el que tiene el número de muestras de los detectores de energía,  $N_s$ , que forman cada componente del vector de energía, sobre la precisión. Dicho parámetro se va a variar entre las 100 y las 1000 muestras de los detectores, siendo el mismo número de muestras para todos los sensores.

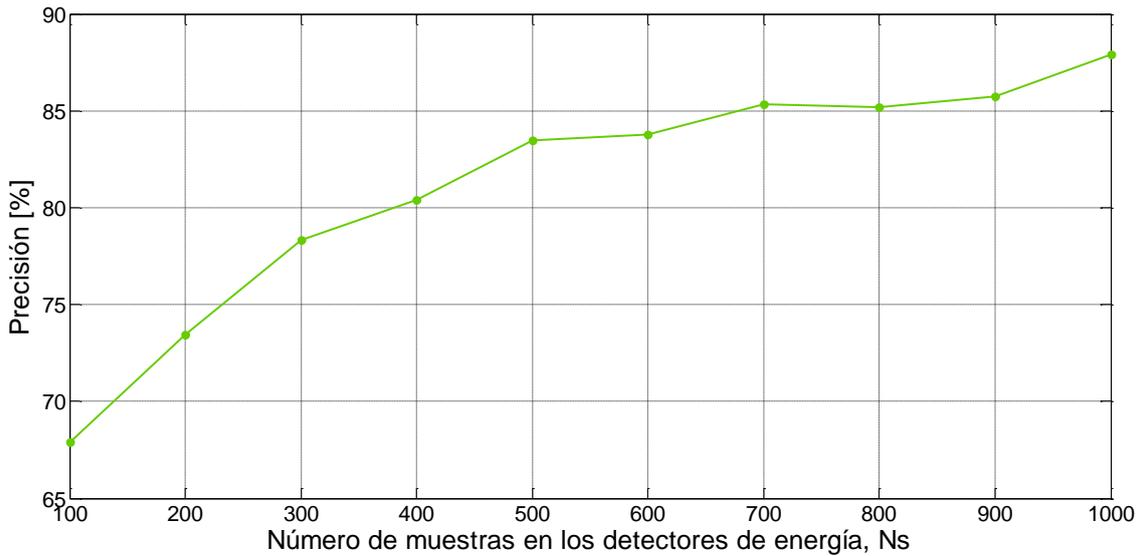


Figura 4.6. Acierto en la identificación de la actividad de los usuarios en función del número de muestras en los detectores de energía, Ns.

En la gráfica (Figura 4.6) se comprueba como cuantas más muestras del detector se utilicen para calcular posteriormente los vectores de energía,  $\mathbf{X}$ , mejor es la precisión en la fase de decisión. Esto es debido a que para valores de Ns pequeños, las medias y las varianzas de los  $\mathbf{X}$  también serán pequeñas, produciéndose solapamientos en los clusters de las distintas clases, debido a la cercanía de los valores de las medias. De la misma forma, para valores de Ns grandes, grandes serán las medias y las varianzas. Aunque como ya se comentó, aunque las varianzas sean mayores en el segundo caso, escalando son menores. Aumentando la precisión, en el segundo caso, al no producirse solapamientos entre clases.

Y como se aprecia en la gráfica, ese efecto es considerable llegando al variar la precisión casi en un 20 % desde las 100 muestras hasta las 1000, en los detectores de energía.

Sin embargo, a partir de las 600-700 muestras del detector, hay poca probabilidad de que los clusters se superpongan. Los errores se producen porque los clusters pueden estar alineados. Y aunque la precisión siga aumentando, el aumento es poco significativo.

#### **4.5.4. Efecto de la potencia de transmisión de los usuarios, P, sobre la precisión**

El siguiente efecto a analizar será el de la potencia con que transmiten los usuarios, suponiendo siempre, que todos los usuarios cuando están activos transmiten con la misma potencia, sobre la precisión. Esta vez, la potencia se variará entre 0.005 W y 0.25 W.

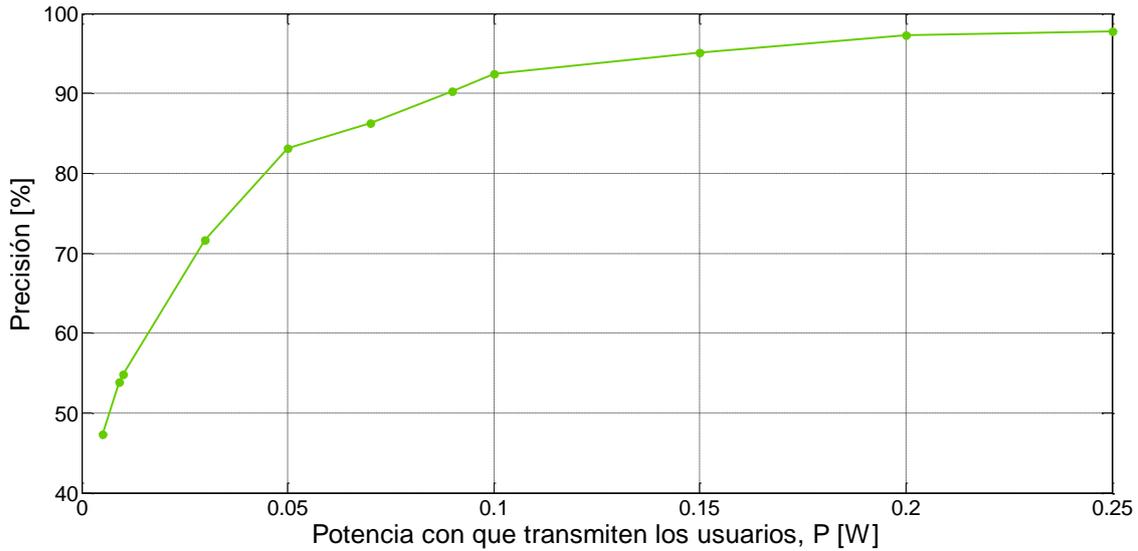


Figura 4.7. Acierto en la identificación de la actividad de los usuarios en función de la potencia con que transmiten, P.

En esta gráfica (Figura 4.7), se observa como con potencias de transmisión muy bajas la precisión no es buena, aunque al subir la potencia, la precisión crece de forma logarítmica. Esto es debido a que para potencias pequeñas, el parámetro de centralidad de los  $\mathbf{X}$ , es también pequeño, y por ello las medias y las varianzas de  $\mathbf{X}$  también serán pequeñas. Produciendo solapamientos entre clases. Hecho el cual deja de ocurrir al llegar a potencias más elevadas, donde el parámetro de centralidad ya no es tan pequeño. Teniendo suficiente peso en las medias y varianzas sobre  $N_s$ , aumentando su valor hasta llegar a separarlas de forma que ya no se produzcan solapamientos entre los clusters.

En el análisis de este efecto, se puede advertir como la precisión en la fase de decisión se ve afectada de manera muy significativa cuando los valores de potencia son muy bajos.

Pero, por otra parte, se puede observar que a partir de una potencia de transmisión de 0.1 W, donde la precisión alcanza ya aproximadamente el 92 %, no habría mucho beneficio en seguir subiendo la potencia de transmisión. Es por eso que una potencia de transmisión de 0.1 W sería un valor muy bueno para este entorno de propagación, en cuanto que aumentar la potencia de los usuarios sin mejoras significativas en la fase de decisión sería empeorar la productividad de la red, al tener que invertir más energía en alimentar a los usuarios.

#### 4.5.5. Efecto del número de datos de entrenamiento, $N$ , sobre la precisión

El siguiente efecto a analizar será el efecto del número de datos de entrenamiento usados para obtener  $\mathbf{W}$ , sobre la precisión. Se variará entre 100 y 2000, siempre asegurando que todas las clases están presentes.

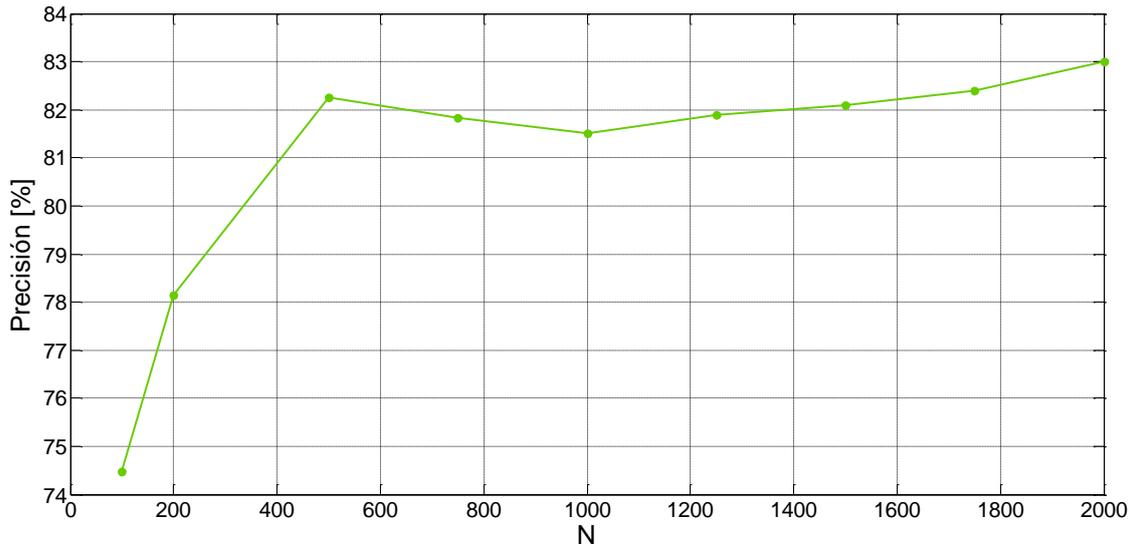


Figura 4.8. Acierto en la identificación de la actividad de los usuarios en función del número de datos de entrenamiento usados, N.

En la gráfica (Figura 4.8) se aprecia como cuantos más datos de entrenamiento haya mejor será la precisión en la fase de decisión. Esto es debido al estimador ML usado para la obtención de los parámetros  $\mathbf{W}$ . El estimador ML es asintóticamente insesgado y asintóticamente eficiente. Esto es, para un número grande de muestras, número de datos N en nuestro caso, el valor estimado va a ser el valor del parámetro buscado, con el mínimo error posible.

Sin embargo, ese aumento de la precisión se aprecia sobre todo por debajo de los 500 datos de entrenamiento que es cuando realmente se produce el verdadero aumento de la precisión, oscilando para un mayor número de datos en un valor prácticamente constante.

Es por eso, que al no obtener un beneficio muy grande de tener muchos datos de entrenamiento. Un valor óptimo para el número de datos de entrenamiento estaría por los N=500 datos. De esa forma, es seguro que todas las clases de datos van a estar perfectamente representadas en el conjunto, pero a la vez se disminuye el coste computacional que se tendría con un mayor número de datos.

#### 4.5.6. Efecto del modelado de la actividad de los usuarios sobre la precisión

Otro análisis a realizar va a ser el efecto del modelo de actividad de los usuarios durante la adquisición de los datos de entrenamiento, sobre la precisión. Para ese análisis se va a variar la probabilidad de que estando un usuario activo, siga transmitiendo o pase a estar inactivo. Esta probabilidad de que el usuario pase a estar inactivo, se va a variar entre 0.025 y 0.975, probabilidades con las cuales que simularán las redes en las que los usuarios estén largos periodos transmitiendo o por el contrario redes en las que los usuarios transmitan muy cortos periodos de tiempo.



Figura 4.9. Acierto en la identificación de la actividad de los usuarios en función de la probabilidad de un usuario de pasar de estar activo a inactivo.

En la gráfica (Figura 4.9) se observa como la curva de precisión decrece con la probabilidad. Esto se debe a que cuando  $pr$  es bajo, la probabilidad en el límite del estado activo de los usuarios tiende a  $1/3$ . De esta forma los usuarios transmitirán largos periodos de tiempo, introduciendo mucha correlación temporal en los datos. Por lo tanto, las tres clases de uno de los usuarios activo, aparecerán con más frecuencia en **Y** o **S**, entrenándose mejor los parámetros **W** de esas clases, aumentando posteriormente la precisión en la decisión. Sin embargo, cuando esa probabilidad baja, la correlación temporal de los datos baja, los usuarios irán transmitiendo en periodos de tiempo menores cada vez, y las probabilidades de estado en el límite se irán igualando. Sin embargo, cuando esa  $pr$  sube mucho, los datos apenas tendrán correlación temporal, los usuarios transmitirán periodos muy cortos de tiempo, y la probabilidad en el límite del estado inactivo tenderá a  $4/7$  y la del estado activo a  $1/7$ . Apareciendo la clase de los usuarios inactivo mucho más en **Y** o **S**, que las otras clases. Pero en este caso, la precisión apenas aumenta, dado que se entrena muy bien una sola clase frente a las cuatro existentes, por el contrario que en el otro extremo de la curva.

Por otra parte, se comprueba el efecto de este parámetro sobre la precisión, existiendo una diferencia de hasta un 6 % entre los valores para la mejor y peor probabilidad en términos de precisión en la fase de decisión.

El hecho de esa diferencia, lleva a plantearse que la mejor opción podría llegar a ser que las probabilidades en el límite de los estados de la red fueran mayores para el estado activo. Sin embargo, este hecho provocaría que la precisión sobre la clase inactiva no fuera tan buena como las otras. Empeorando el clasificador, aunque la precisión total sea mejor. Por lo que la mejor opción sería que la probabilidad de estado en el límite sea igual para todas las clases.

## 4.6. Resultados sobre canal variante / Algoritmo Adaptativo

En canales variantes con el tiempo, los datos de entrenamiento pueden haber sido generados bajo diferentes realizaciones del canal. Esto ocurre si el tiempo de coherencia del canal,  $T_c$ , es menor que el tiempo de captura de los  $N$  datos de entrenamiento. No obstante, la rapidez con la que varíen dichos canales va a depender del desplazamiento Doppler.

Los datos de entrenamiento,  $\mathbf{X}$ , en este caso, son generados bajo diferentes canales. Lo que provoca que la distribución de los  $\mathbf{X}$  cambie con el tiempo. De forma que, los parámetros del modelo LR ( $\mathbf{W}$ ) también cambian con el tiempo. Por lo que el clasificador bloque no funcionará con estos datos  $\mathbf{X}$ . Siendo necesario utilizar un algoritmo de aprendizaje adaptativo como SGA.

Para las simulaciones de este capítulo se harán los siguientes cambios respecto a los parámetros del problema definidos en la sección 4.4.

- Parámetros de la red de comunicaciones:
  - Potencia de transmisión de todos los usuarios:  $P=0.075$  W
- Parámetros de las simulaciones
  - Número de datos de entrenamiento:  $N=2000$

En primer lugar, antes de simular cualquier resultado relativo al canal variante con el tiempo, es necesario, como ya se comentó en la sección del algoritmo SGA, ajustar el parámetro  $\alpha$  de la constante de aprendizaje. Para ello, se han realizado una serie de simulaciones de Montecarlo para los distintos valores que puede tomar  $\alpha$  dentro de su rango definido, entre cero y uno, para posibles desplazamientos Doppler.

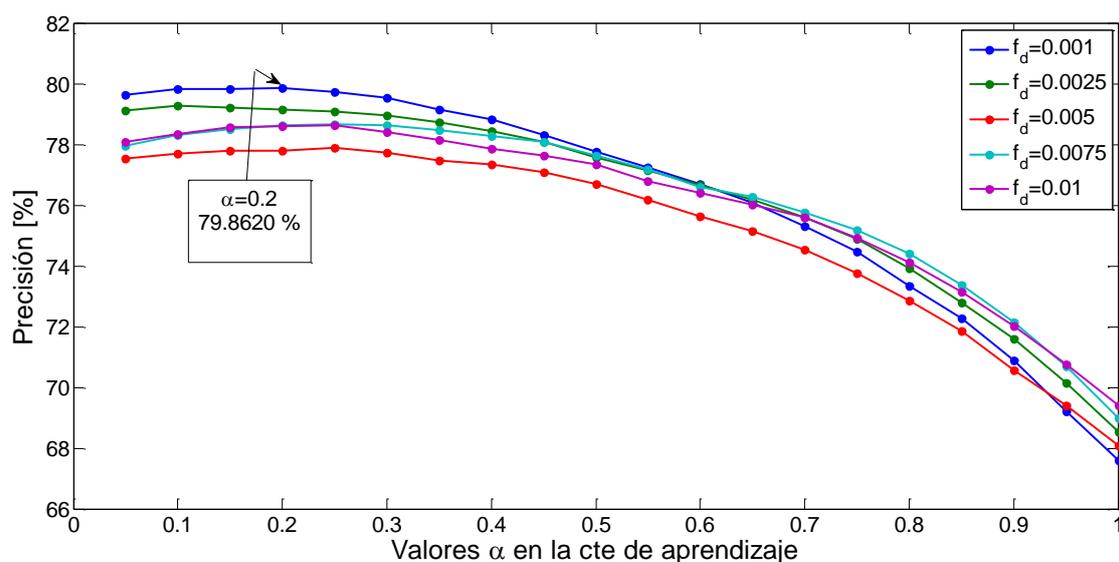


Figura 4.10. Relación entre el parámetro  $\alpha$  de la constante de aprendizaje,  $f_d$  y la precisión.

En dichas curvas (Figura 4.10) se aprecia como para valores bajos de  $\alpha$  la precisión para las curvas de menor desplazamiento Doppler es menor. Esto se debe a que al ser el desplazamiento Doppler menor, el canal varía más lentamente y por tanto es más fácil adaptar los parámetros  $\mathbf{W}$  a las variaciones del canal.

Por otra parte también se aprecia como para valores bajos del posible rango de  $\alpha$  se obtienen los mejores resultados, en este caso, es debido a que para esos valores es cuando se consigue el mejor compromiso entre la velocidad de convergencia del parámetro y su precisión en el ajuste. Ya que para valores de  $\alpha$  más altos, se obtiene una mayor velocidad de convergencia pero el ajuste es mucho más grueso, observándose un rizado mayor en las curvas de aprendizaje.

Por otra parte, también se observa como las curvas son bastante planas entorno al valor óptimo de  $\alpha$ , por lo que la precisión no es muy sensible al valor de  $\alpha$  siempre que éste no sea excesivamente grande. Pero sobre todo, lo que apenas incluye en la precisión es el desplazamiento Doppler, sólo hay alrededor de un 3% de diferencia en la precisión para los diferentes desplazamientos Doppler analizados en la gráfica. Esto muestra que el algoritmo adaptativo funciona bien en un rango razonablemente alto de valores de  $f_d$ , con el mismo valor de  $\alpha$ .

Por todo ello, e intentando conseguir la mejor precisión en simulaciones posteriores se va a trabajar salvo que se diga lo contrario con  $\alpha=0.2$ , y con un valor intermedio de desplazamiento de  $f_d=0.005$ .

En una primera simulación se va a analizar un caso de canal invariante mediante ambos algoritmos GA y SGA, para analizar las comparaciones entre ellos.

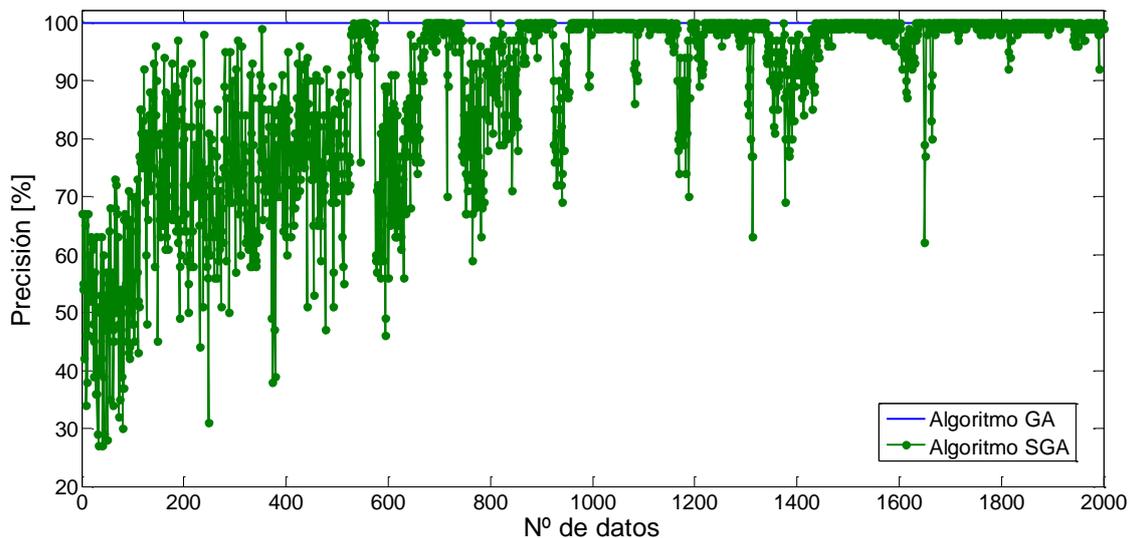


Figura 4.11. Comparación algoritmos GA y SGA para una única realización de un canal invariante con el tiempo.

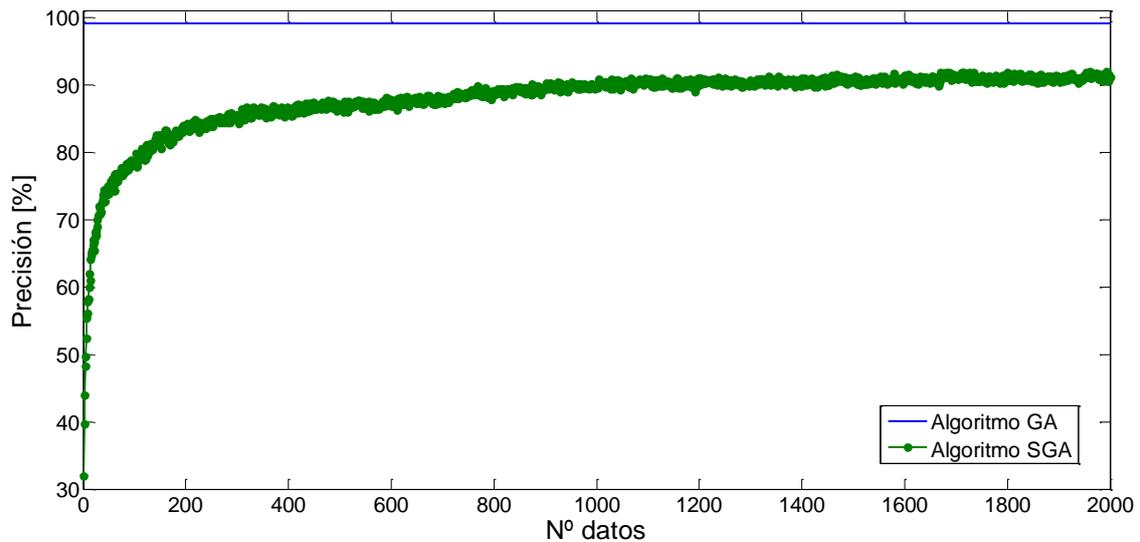


Figura 4.12. Comparación algoritmos GA y SGA para un promedio de 200 realizaciones de canales invariantes con el tiempo.

En la primera gráfica (Figura 4.11) se muestra la curva de aprendizaje obtenida de una realización del problema con canal invariante. En la segunda (Figura 4.12), se muestra la curva de aprendizaje para un promedio de 200 realizaciones de canales invariantes con el tiempo. Además, en ambas gráficas, se observa como la precisión conseguida con el algoritmo online (SGA), por muchos datos de entrenamiento que se tengan y muchas iteraciones que se realicen, nunca se va a igualar con el conseguido con el algoritmo batch (GA). Si bien, las gráficas muestran como la precisión crece con el número de observaciones, hasta un número en el cual ya ha llegado un valor óptimo del parámetro  $\mathbf{W}$  para el clasificador, y por tanto la precisión a partir de ese punto se mantiene estable. Es decir, la precisión conseguida con el algoritmo SGA siempre va a tender al porcentaje conseguido con el algoritmo GA, de modo que tras estabilizarse la precisión del SGA, su valor, aunque peor, se asemeje bastante al conseguido con el GA.

Si se realiza el promedio de las  $N_{sim}=200$  realizaciones del canal, pero en este caso, de canales variantes con el tiempo se obtiene la curva de aprendizaje de la figura 4.13.

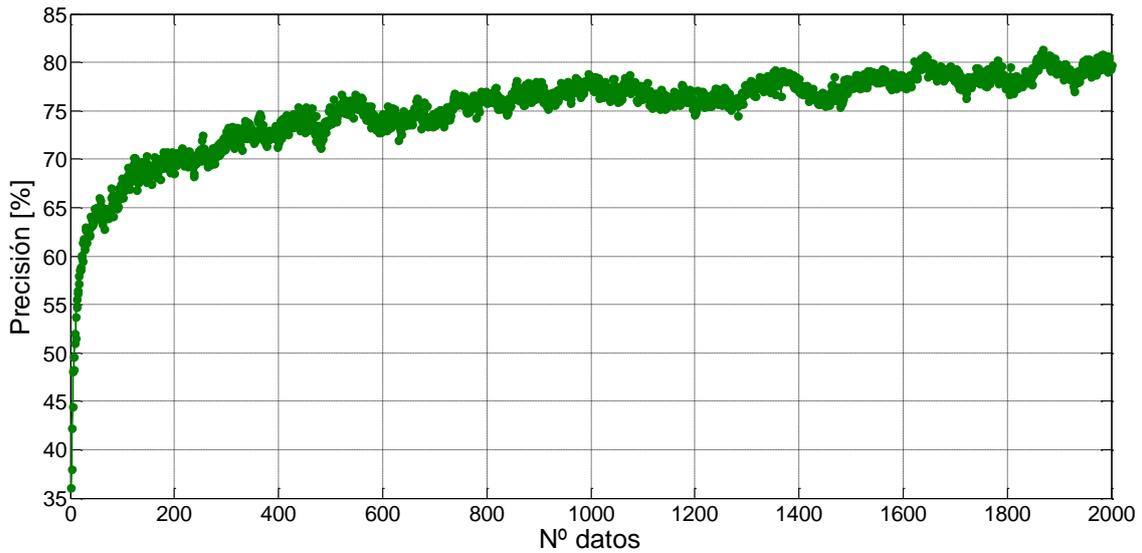


Figura 4.13. Acierto en la identificación de los usuarios a lo largo de las iteraciones.

En esta gráfica, se puede apreciar con respecto a la gráfica con el promedio de los canales invariantes (Figura 4.12), como la precisión alcanzada es mejor en canales invariantes. Este hecho ocurre debido a que en canales variantes los parámetros  $\mathbf{W}$  deben adaptarse a esos cambios de canal para cada nuevo dato de entrenamiento,  $\mathbf{X}$ .

Para canales variantes con el tiempo, el análisis de los efectos de la variación de parámetros del problema antes realizados (secciones 4.5.1-4.5.5) no varía cualitativamente respecto al hecho para el algoritmo GA, por lo que no se repetirá. La única diferencia entre los efectos de la variación de dichos parámetro estará en el valor de la precisión final obtenida.

Sin embargo, si es interesante la evolución de las curvas de aprendizaje para una serie de valores de la probabilidad  $pr$ , es decir, ver el efecto que tiene el modelado de la actividad de los usuarios en dichas curvas.

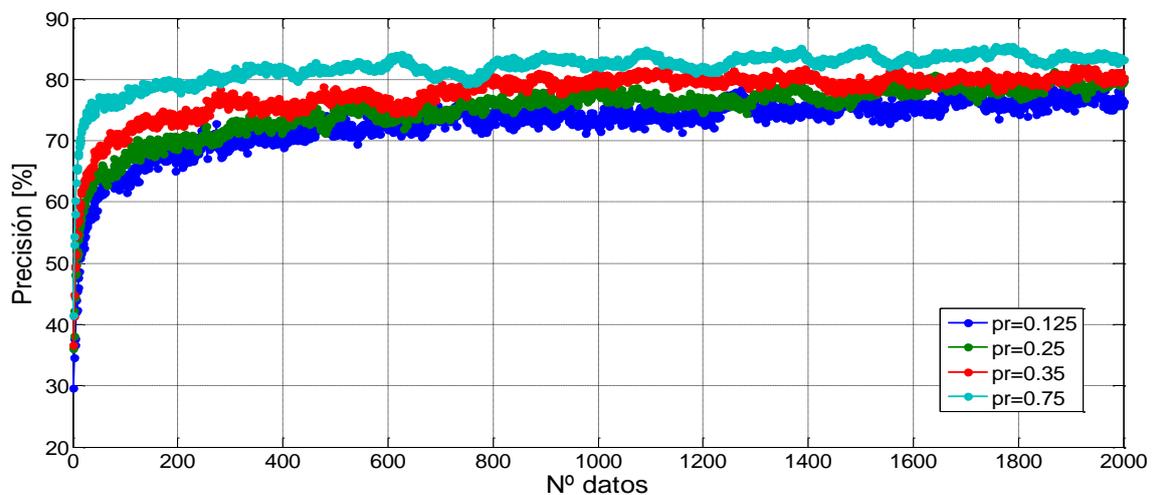


Figura 4.14. Acierto en la identificación de la actividad de los usuarios a lo largo de las iteraciones en función de la probabilidad de que un usuario pase de estar activo a inactivo.

De esta forma, en la gráfica (Figura 4.14), se obtiene que para valores de  $\rho_r$  altos, en la secuencia de datos de entrenamiento aparecen observaciones bajo las distintas clases rápidamente, por lo que la curva evoluciona más rápidamente hacia el valor final de la precisión.

Sin embargo, para valores de  $\rho_r$  bajos, es decir, secuencias de datos muy correladas en el tiempo y con periodos muy largos de transmisión, la curva evoluciona de forma más lenta. Ya que tardan más en aparecer todas las clases en la secuencia de entrenamiento, de forma que, les cuesta más a los parámetros  $\mathbf{W}$  entrenarse.

Otro análisis interesante, es la variación de las curvas de precisión en función del desplazamiento Doppler normalizado. Para ello, se va a obtener la curva para cada uno de los cinco valores de  $f_d$  que se analizaron previamente.

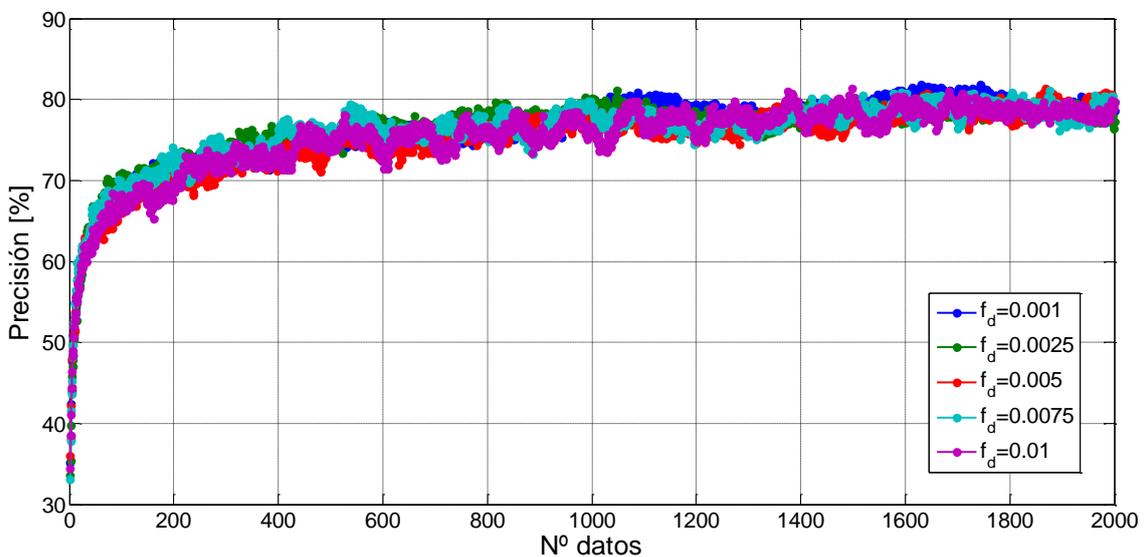


Figura 4.15. Acierto en la identificación de la actividad de los usuarios en función del desplazamiento Doppler máximo normalizado.

En la gráfica (Figura 4.15), se observa como la evolución de la precisión apenas depende del desplazamiento Doppler normalizado que tenga el canal, es decir, de la rapidez con que éste varíe. Evoluciona prácticamente igual la precisión obtenida de un canal que varía muy rápido que el de un canal que apenas varía. Esto es debido a que en cada iteración se adapta los parámetros  $\mathbf{W}$  a los datos que vienen en ese momento con un canal diferente a los anteriores. Por lo tanto, al variar de todas formas el canal, no hay muchas diferencias cuantitativas en la evolución de  $\mathbf{W}$  y en su precisión final. Simplemente, para canales con mayor desplazamiento, dentro de los mismos valores de precisión, la evolución tiene más rizado, que para canales con menor desplazamiento.



## **5. LÍNEAS FUTURAS**

### **5.1. Varios usuarios activos simultáneamente**

Una de las características de la red de comunicaciones analizada es que sólo un usuario puede estar transmitiendo en cada instante. En una línea futura, se podría extender la transformación del problema de comunicaciones al problema de clasificación supervisada, al caso en que cualquier usuario puede estar activo en cualquier instante de tiempo.

### **5.2. Modelo no lineal**

Además, en la resolución del problema de clasificación supervisada, se usó LR, un modelo lineal. Otra posible línea futura a este trabajo, sería la utilización de un modelo no lineal como base para los algoritmos GA y SGA. Disminuyendo, de esa forma, los errores producidos en la fase de decisión por clases colineales.



## 6. APÉNDICES

### 6.1. Apéndice 1: Demostración de la expresión del gradiente de $q_n(\mathbf{W})$

Se sabe que

$$q_n(\mathbf{W}) = \log p(y_n = c | \mathbf{x}_n, \mathbf{W})$$

Si se aplica la expresión del gradiente de un logaritmo:

$$\nabla_{\mathbf{w}_r} q_n(\mathbf{W}) = \frac{1}{p(c | \mathbf{x}_n, \mathbf{W})} \nabla_{\mathbf{w}_r} p(c | \mathbf{x}_n, \mathbf{W})$$

Sabiendo que,

$$p(c | \mathbf{x}_n, \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T \mathbf{x}_n)}{\sum_s \exp(\mathbf{w}_s^T \mathbf{x}_n)}$$

El gradiente, si  $r=c$ , es:

$$\begin{aligned} \nabla_{\mathbf{w}_r} p(c | \mathbf{x}_n, \mathbf{W}) &= \frac{\exp(\mathbf{w}_c^T \mathbf{x}_n) \mathbf{x}_n \sum_s \exp(\mathbf{w}_s^T \mathbf{x}_n) - \exp(\mathbf{w}_c^T \mathbf{x}_n) \exp(\mathbf{w}_r^T \mathbf{x}_n) \mathbf{x}_n}{(\sum_s \exp(\mathbf{w}_s^T \mathbf{x}_n))^2} \\ &= \mathbf{x}_n p(c | \mathbf{x}_n, \mathbf{W}) - \mathbf{x}_n p(c | \mathbf{x}_n, \mathbf{W}) p(r | \mathbf{x}_n, \mathbf{W}) \\ &= \mathbf{x}_n p(c | \mathbf{x}_n, \mathbf{W}) (1 - p(r | \mathbf{x}_n, \mathbf{W})) \end{aligned}$$

Sin embargo, si  $r \neq c$ , el gradiente es:

$$\nabla_{\mathbf{w}_r} p(c | \mathbf{x}_n, \mathbf{W}) = \frac{-\exp(\mathbf{w}_c^T \mathbf{x}_n) \exp(\mathbf{w}_r^T \mathbf{x}_n) \mathbf{x}_n}{(\sum_s \exp(\mathbf{w}_s^T \mathbf{x}_n))^2} = -p(c | \mathbf{x}_n, \mathbf{W}) p(r | \mathbf{x}_n, \mathbf{W}) \mathbf{x}_n$$

En general, ambos resultados se pueden agrupar en la siguiente expresión para el gradiente de  $q_n(\mathbf{W})$ :

$$\nabla_{\mathbf{w}_r} p(c | \mathbf{x}_n, \mathbf{W}) = \mathbf{x}_n p(r | \mathbf{x}_n, \mathbf{W}) [I(c = r) - p(c | \mathbf{x}_n, \mathbf{W})]$$

## 6.2. Apéndice 2: Probabilidades de estado en el límite

Si se parte de la matriz de transición para el caso concreto de  $M=3$  usuarios y  $C=4$  clases:

$$T = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ pr & 1 - pr & 0 & 0 \\ pr & 0 & 1 - pr & 0 \\ pr & 0 & 0 & 1 - pr \end{pmatrix}$$

Se sabe, además, que el vector de probabilidades de estado en el límite para esa matriz de transición  $T$  es:

$$\Pi = [\Pi_1 \Pi_2 \Pi_3 \Pi_4], \text{ con } \sum \Pi_i = 1$$

Vector, el cual, se calcula con la expresión:

$$\Pi = \Pi T$$

La cual si sustituimos, resulta:

$$[\Pi_1 \Pi_2 \Pi_3 \Pi_4] = [\Pi_1 \Pi_2 \Pi_3 \Pi_4] T$$

Expresión de la cual, operando, se obtiene el siguiente sistema de ecuaciones:

$$\Pi_1 = \frac{1}{4} \Pi_1 + pr \Pi_2 + pr \Pi_3 + pr \Pi_4$$

$$\Pi_2 = \frac{1}{4} \Pi_1 + (1 - pr) \Pi_2$$

$$\Pi_3 = \frac{1}{4} \Pi_1 + (1 - pr) \Pi_3$$

$$\Pi_4 = \frac{1}{4} \Pi_1 + (1 - pr) \Pi_4$$

Si para su resolución, sumamos las tres últimas ecuaciones:

$$1 - \Pi_1 = \frac{3}{4} \Pi_1 + (1 - pr)(1 - \Pi_1)$$

$$1 - \Pi_1 = \frac{3}{4} \Pi_1 + 1 - \Pi_1 - pr + pr \Pi_1$$

$$pr = \frac{3}{4} \Pi_1 + pr \Pi_1$$

Obteniendo,

$$\Pi_1 = \frac{pr}{pr + \frac{3}{4}} = \frac{4pr}{4pr + 3}$$

Resultado, el cual, si se sustituye en el resto de ecuaciones, resulta:

$$\Pi_2 = \frac{1}{4}\Pi_1 + \Pi_2 - pr\Pi_2$$

$$\Pi_2 = \frac{1}{4} \frac{\Pi_1}{pr}$$

$$\Pi_2 = \frac{1}{4pr + 3} = \Pi_3 = \Pi_4$$

Se comprueba, que las probabilidades obtenidas cumplen la condición,

$$\sum \Pi_i = \frac{4pr}{4pr + 3} + 3 \frac{1}{4pr + 3} = 1$$

Al obtenerse una única solución de las probabilidades de estado en el límite  $\Rightarrow$  Es una cadena de Markov ergódica



## 7.REFERENCIAS

- [1] A. Molisch, "Wireless communications", John Wiley & Sons, Ltd, 2005.
- [2] Duda, R. O., P. E. Hart, and D. G. Stork, "Pattern Classification", Second ed., Wiley, 2001.
- [3] E. Hossain and V. K. Bhargava, "Cognitive Wireless Communication Networks", Springer, 2007.
- [4] H. Huang, Z. Zhang, P. Cheng and P. Qiu, "Opportunistic spectrum access in cognitive radio system employing cooperative spectrum sensing", in Proc. of IEEE Vehicular Technology Conference, pp. 1-5, 26-29 April 2009.
- [5] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications", IEEE Commun. Surveys Tutorials, vol. 11, no. 1, pp. 116-130, First Quarter 2009.
- [6] C. Li and C. Li, "Opportunistic spectrum access in cognitive radio networks", IEEE International Joint Conference on Neutral Networks, pp. 3412-3412, 1-8 June 2008.
- [7] K. M. Thilina, K. W. Choi, N. Saquib and E. Hossain, "Machine Learning Techniques for Cooperative Spectrum Sensing in Cognitive Radio Network", IEEE Journal on Selected areas in Communications, Vol. 3, No. 11, November 2013.
- [8] X. Wang, X. Li and V. Leung, "Artificial Intelligence-Based Techniques for Emerging Heterogeneous Network: State of the Arts, Opportunities, and Challenges", IEEE Special Section on AI Enabled Networking, Vol. 3, August 2015.
- [9] X. Chu, D. López-Pérez, Y. Yang, F. Gunnarson (Eds.), "Heterogeneous Cellular Networks", Cambridge University Press, 2013.
- [10] G. Boccolini, G. Hernandez, B. Beferull-Lozano, "Wireless sensor network for spectrum cartography based on Kriging interpolation", IEEE Int. Symposium on Personal Indoor and Mobile Radio Communications (PIMRC12), 2012.
- [11] A. Goldsmith, "Wireless Communications", Stanford University, 2005.
- [12] S. R. Saunders and A. Aragón-Zavala, "Antennas and Propagation for Wireless Communication Systems", Wiley, 2007.
- [13] C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

[14] S. M. Kay, Fundamentals of statistical signal processing, vol. I, Estimation theory, Prentice Hall, 1993.

[15] Documentación Matlab, “Unconstrained Nonlinear Optimization Algorithms”, <http://es.mathworks.com/help/optim/ug/unconstrained-nonlinear-optimization-algorithms.html#brnpcye>

[16] W. C. Jakes, “Microwave Mobile Communications”, Wiley, 1974.