Departamento de Teoría de la Señal y Comunicaciones

Grado en Ingeniería en Tecnologías de Telecomunicación

TRABAJO FIN DE GRADO

Estudio experimental de costes para aprendizaje estadístico con etiquetas parciales

Autor: Pablo Moreno Muñoz

Director: Dr. Jesús Cid Sueiro

Leganés, Junio 2014

Título: Estudio experimental de costes para aprendizaje estadístico con etiquetas parciales
Autor: Pablo Moreno Muñoz
Director: Dr. Jesús Cid Sueiro

EL TRIBUNAL

Presidente:

Vocal:

Secretario:

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día ..... de ..... de ... en ....., en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de:

VOCAL

SECRETARIO PRESIDENTE

Agradecimientos

Antes que nadie, debo y quiero agradecer a Dr. Jesús Cid Sueiro, mi director del presente Trabajo Fin de Grado, su total dedicación conmigo desde el primer día, Martes tras Martes, todas y cada una de las tutorías que me ha dedicado y el tiempo que las mismas implican. En segundo lugar, por lo que merecen y por ser una pequeña gran parte de este trabajo, agradezco a mis padres Marisa y Luis todo lo que me han dado estos años, pero sobretodo, haberme hecho vivir desde bien pequeño el mundo de la ciencia y la ingeniería al que ellos pertenecen. Mi admiración por ello y gracias por guiarme hasta aquí.

Continúo con mis amigos, fuente de apoyo incondicional, en los que estos años siempre me he apoyado. Pablo e Isa, los que nunca fallásteis y que habeis compartido conmigo la alegría que hemos podido encontrar en todo momento y cualquier lugar, sois vosotros los que hicisteis mi camino universitario feliz como pocos. No me olvido de ti, Juanjo, que sabes lo gran amigo que eres y lo que hemos sufrido juntos, aquí terminamos y allí seguiremos, nuestros caminos no se separan todavía.

Por último y no por ello menos que agradecer, todos los que habéis estado presentes en mi vida todo este tiempo, no en la universidad, pero si fuera de ella. Carlos, eres grande amigo, y más lo es el apoyo que siempre me das para seguir adelante; Javi, que qué decir, desde hace años estamos así, siempre al pie del cañón, y Jaime, que agradezco lo gran amigo que eres y has sido conmigo todo este tiempo. Sigo, porque no puedo olvidarme de Jimena, con quién comparto desde hace innumerables años algo que vale más que el oro, la confianza, gracias por estar ahí; también Pilar, por mis primeros días de ingeniería, donde fue todo paciencia y cariño sin doblegarse por nada. Y a los que ya no seguís cerca pero si lo estuvisteis tanto tiempo, también sois importantes aquí.

Gracias a todos vosotros.

Resumen

En este trabajo se aborda el estudio del comportamiento de conjuntos de datos dotados tanto de un etiquetado supervisado como de etiquetas parciales. Dicho estudio se realiza mediante el análisis de costes para la función *NLL* o *verosimilitud logarítmica negativa*. Se realizan además procesos de optimización con dos algoritmos clásicos, *steepest descent* y *Newton's method*. La optimización se realiza sobre la función *NLL*, empleándose observaciones generadas artificialmente. La creación de los datos y su etiquetado, tanto supervisado como parcial, viene acompañada de un modelo probabilístico ofrecido por la *regresión logística*. Se trata por tanto de una evaluación de algoritmos de entrenamiento para un problema de clasificación basado en etiquetas parciales. Todas las implementaciones se llevan a cabo empleando técnicas de aprendizaje máquina y siguiendo metodologías del aprendizaje. La evaluación y las conclusiones sobre los resultados ofrecen las diferencias entre ambos tipos de etiquetas y las pérdidas que se producen con los dos modelos.

Palabras clave: clasificador, regresión logística, optimización, etiquetas parciales, aprendizaje máquina.

Abstract

This work deals with the study of different types of data clusters. These ones are classified with supervised labels and partial labels. An analysis is performed about costs over the function *NLL* or *negative log-likelihood*. Also classic optimization methods are applied, for example: *steepest descent* and *Newton's method*. All the optimization process is done using artificially generated observations. The creation of data and its labeling, both partial as supervised, is followed by a probabilistic model originated with *logistic regression*. Each implementation is made with machine learning techniques and using also learning methodology. Evaluation of results and conclusions provide differences between both types of labeling and losses that are produced by the two models.

Keywords: classification, logistic regression, optimization, partial labels, machine learning.

Índice general

Índice de figuras

Capítulo 1

Introducción

En este capítulo se explican las principales motivaciones y objetivos que tienen este trabajo y su realización. Se hace un breve recorrido por los problemas y soluciones teóricas más comunes a fin de poder introducir al lector en los aspectos significativos que afectan al presente estudio.

1.1. Contexto socioeconómico y tecnológico

Hoy en día podemos estar seguros de que nuestro futuro inmediato va camino de convertirse en la era del **big data**. La inmensidad de datos que en la última década están siendo generados obliga a replantearse la dimensión que soluciones y problemas podrán tener en los próximos años. Al hablar de tantos *bits* que compondrán los miles de *terabytes* y *petabytes* del futuro, es fácil olvidar la importancia que tanto cantidad de datos como valor de cada uno de ellos tienen. Es precisamente con esto con lo que el **aprendizaje máquina** trabaja, aportando una capacidad de análisis sin precedentes que nos permita sintetizar, descubrir, recopilar y obtener información de valor incalculable. Es en el ámbito del aprendizaje máquina donde este trabajo y el estudio experimental encuentran su lugar. Definir el aprendizaje sin embargo no es tarea fácil, aunque podemos entenderlo como el conjunto de métodos y algoritmos capaces de encontrar de manera automática patrones dentro de conjuntos de datos, y además, emplearlos con el fin de hacer predicciones futuras sobre los mismos.

Podemos en esta definición por tanto, encontrar ya indicios de la importancia que esta disciplina tiene para la sociedad. La capacidad de predecir comportamientos futuros, establecer patrones de actuación o conocer los factores que tomarán mayor probabi-

lidad, suponen un activo incalculablemente valioso para cualquier organización. Aunque por supuesto, no todo tiene por orientación el dinero; la posibilidad por ejemplo de aplicaciones en la biomedicina para la detección de dolencias, irregularidades o tumores en los tejidos internos puede ser clave en la prevención de enfermedades graves.

Una vez introducidos a las posibilidades y aplicaciones que el análisis de datos y más concretamente, el aprendizaje máquina, nos puede proporcionar; hacemos hincapié en los escenarios, tipos de aprendizaje y entrenamientos comunes que podemos encontrar en problemas similares a los que este trabajo pretende resolver.

1.1.1. Aprendizaje máquina

El aprendizaje máquina surge a partir de variedad de problemas que no pueden ser resueltos mediante los procesos estadísticos habituales que se darían si se dispusiera del conjunto completo de la información estadística necesaria. Se busca disponer sin embargo de conjuntos de datos *etiquetados*, de manera que, como más adelante se detalla, cada uno de los datos u observaciones lleva asociado un valor de una variable concreta, que facilitará a los estimadores y clasificadores lograr su fin. Este valor o *etiqueta* proporciona a los algoritmos su utilidad principal, y cuanto mayor sea la cantidad de observaciones etiquetadas, más se aproximarán los resultados obtenidos al caso óptimo.

Dentro del aprendizaje máquina encontramos dos clases principales: **aprendizaje supervisado** y **aprendizaje no supervisado**. En cuanto al primero de ellos, el caso **supervisado**, nos encontramos con un conjunto de entrenamiento conformado por observaciones etiquetadas formando N pares de datos y etiquetas. A partir de ellos el objetivo es realizar un aprendizaje estadístico mapeando el conjunto del que se dispone. En segundo lugar, el caso **no supervisado**, se caracteriza por un conjunto de entrenamiento formado únicamente por datos no etiquetados, de manera que se intenta encontrar patrones, similitudes o aspectos de importancia a partir de las observaciones *n-dimensionales*.

Este trabajo, y el estudio experimental, por ser acerca de conjuntos de datos etiquetados *total* y *parcialmente*, se engloba dentro del aprendizaje máquina supervisado.

1.1.2. Escenarios y clasificación

Con el fin de comprender mejor el escenario sobre el cual se construye este estudio, detallamos previamente un escenario típico y sencillo de clasificación que también sirvió al autor de motivación en la realización.

Figura 1.1: Flor de iris. *Iris versicolor*. [1]

En problemas de clasificación se suelen usar bases de datos para investigadores muy variadas [1], y en la mayoría de las mismas es corriente encontrar conjuntos de datos de entrenamiento clásicos para ganar experiencia con propuestas de reconocimiento de dígitos, imágenes o figuras. En este caso, la Figura 1.1 nos muestra la flor de iris, una variedad floral que dispone de muchos tipos distintos diferenciados por mínimos detalles. En este caso, por cada observación, cada una de las variables n-dimensionales tomaría el valor correspondiente a altura, anchura de los pétalos, longitud del tallo, entre otros, según se disponga. Atendiendo también a la clasificación que se detalla anteriormente sobre el aprendizaje máquina, cada observación incluirá una etiqueta indicando la variedad o tipo de flor a la que correspondería. Con esto y teniendo presente los algoritmos disponibles para problemas de clasificación, se podrá buscar, analizar y clasificar las distintas observaciones, y qué factores se muestran más relevantes para cada tipo de flor. Partiendo de este caso sencillo, el autor pudo desarrollar mejor sus objetivos y motivaciones para la realización del presente trabajo.

1.1.3. Clases de etiquetado

Respecto al etiquetado de las observaciones, hablamos de **aprendizaje supervisado**, sin embargo, el aspecto de las etiquetas se

muestra variado y presenta diferentes casos y situaciones. Una clasificación coherente del etiquetado es hacerla según rangos de *supervisión*, atendiendo al número, nivel y relación de las etiquetas con las observaciones.

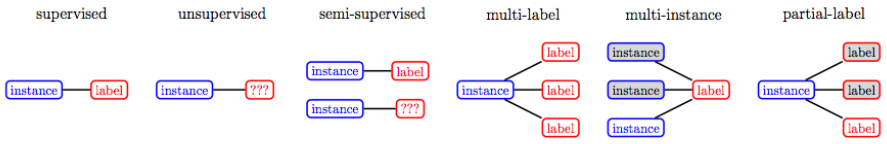

Figura 1.2: Distintos tipos de etiquetado para las observaciones [2]

En la Figura 1.2 observamos seis tipos diferentes de etiquetado: *supervisado*, *no supervisado*, *semi-supervisado*, *multi-etiqueta*, *múltiples observaciones* y *etiquetas parciales*. Definiríamos cada una de ellos de la siguiente forma:

- **supervisado**. Se da una etiqueta por cada una de las observaciones.
- **no supervisado**. No hay etiqueta para ninguna observación.
- **semi-supervisado**. Hay etiqueta solamente para algunas observaciones, para otras no.
- **multi-etiqueta**. Una observación puede tener más de una etiqueta.
- **múltiples observaciones**. Hay una misma etiqueta para más de una observación.
- **etiquetas parciales**. Para cada observación puede haber varias etiquetas, sin embargo, solamente una de ellas es correcta.

1.2. El problema del etiquetado parcial

El etiquetado parcial supone un inconveniente importante, el hecho de que cada observación presente etiquetas '*incorrectas*' causa una ambigüedad a lo largo del conjunto de entrenamiento. Esto se suele producir en muchos escenarios de clasificación, cuando por ejemplo en imágenes sabemos que se representan dos clases de entre otras posibles, y no podemos asegurar cuál es cuál; tendríamos por

tanto para cada observación dentro de la imagen dos etiquetas, una correcta y otra incorrecta.

Por lo tanto resolver esta ambigüedad supone un reto para los clasificadores, dado que la presencia de etiquetas '*incorrectas*' causa pérdidas y aumento de los errores. Se tiene de tal manera en consideración este aspecto, y se buscan soluciones adaptativas a este tipo de etiquetado que presenten pocas pérdidas, trabajen suficientemente bien y que, aumentando el orden del número de observaciones, pueda suplir adecuadamente los efectos de las etiquetas parciales.

1.3. Objetivo del trabajo

El objetivo de este trabajo es realizar un estudio de los costes y pérdidas en aprendizaje máquina con etiquetas parciales. Se busca obtener resultados que demuestren el buen funcionamiento de algoritmos de optimización para funciones de coste de un clasificador probabilístico. Dicho clasificador se construye sobre el modelo de **regresión logística**, que aún llamándose regresor, responde a un modelo de clasificador.

Se utilizarán entonces, dos algoritmos de optimización comunes, de los que se espera respondan adecuadamente tanto frente a un etiquetado supervisado como para un etiquetado parcial después. Dichos algoritmos son *steepest descent* y *Newton's method*. Añadir además, que dichos objetivos se llevarán a cabo siempre trabajando en un modelo de regresión logística multiclase, con observaciones n-dimensionales.

Capítulo 2

Revisión del estado del arte

2.1. El etiquetado parcial en la literatura

Existe un amplio número de artículos y literatura acerca de los problemas que se dan junto con el etiquetado. Aunque en este trabajo se habla siempre de etiquetado parcial; la verdad es que existe un gran número de términos que definen problemas parecidos y adyacentes al que aquí se trata. Un ejemplo de dichos términos que existen para denominar el etiquetado parcial son: etiquetado débil, etiquetado con ruido y aprendizaje multi-etiqueta. Todos ellos son tratados en profundidad, siempre con el objetivo de estudiar el rendimiento que algoritmos y aprendizaje máquina tienen con esta clase de etiquetas.

Se pueden encontrar numerosos escenarios donde la resolución de problemas de etiquetado parcial sean de gran ayuda. Sin embargo, es típicamente en el ámbito multimedia (imágen, vídeo, etc), donde se suelen encontrar más cómodas este tipo de soluciones. Aunque pueda parecer profundamente teórico e incluso abstracto, todo el ámbito del etiquetado parcial va de la mano, como complemento fundamental a cualquier tipo de trabajo sobre reconocimiento de imagen. Resulta por lo tanto, una ayuda más a reconocimiento de caracteres, formas y personas en imágenes que están evolucionando positivamente durante los últimos años.

Como se comenta, el etiquetado parcial no es todo *blanco* o *negro*, un conjunto de datos no se encuentra ligado a un etiquetado únicamente de un tipo u otro, sino que existen combinaciones muy variadas. Cada problema es un mundo, y el número de casos de etiquetado estudiado también lo es.

Un caso singular, muy relacionado con lo que se pretende tratar en este estudio, es el análisis de algoritmos para aprendizaje supervisado [3], cuando las observaciones presentan distintos grados

de pertenencia a diferentes clases. Esto sería de la forma en que $\mathbf{x}_i$ tuviera etiquetas que, en vez de representarse como cero o uno, pudieran tener porcentajes representativos de la clase. Se plantearía una observación x_1 con un etiquetado tipo: $y_{c=1} = 0{,}8$, $y_{c=2} = 0{,}5$ e $y_{c=3} = 0{,}15$.

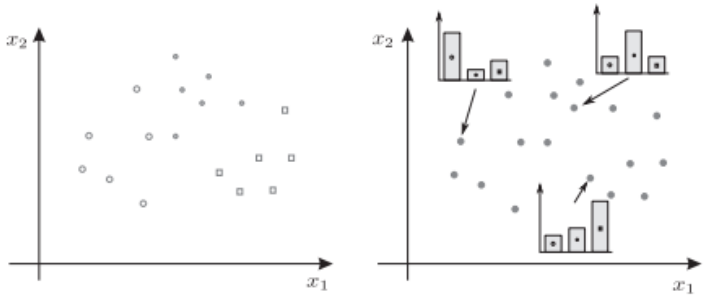

Figura 2.1: Observaciones con grados de pertenencia a 3 clases [2]

Es fundamental reconocer la utilidad de algoritmos de aprendizaje con este tipo de etiquetado, ya que permite ampliar mucho más los escenarios donde pueden encontrarse aplicaciones verdaderamente útiles. Ejemplo de ello, son clasificadores que trabajen sobre aspectos genéticos de la humanidad, donde no puede determinarse el origen concreto de individuos en el ámbito de la medicina (antepasados con enfermedades genéticas). De esta forma, teniendo un modelo porcentual de los antepasados, podrían obtenerse conclusiones acerca de los riesgos que se pudieran prevenir.

$$\mathbf{y} = \begin{pmatrix} y_{c=1} = 0{,}8 & y_{c=2} = 0{,}5 & y_{c=3} = 0{,}15 \end{pmatrix} \tag{2.1}$$

Otro caso relacionado, es el tratado como *WELL* o *WEak Label Learning* [4], es decir etiquetado débil. Esta clase de etiquetas tiene un origen distinto al que trata el etiquetado parcial en este trabajo. Ahora las etiquetas provienen de un modelo de etiquetado multi-etiqueta, en el que, como se explica en la introducción, cada observación o instancia tiene más de una etiqueta. Lo que quiere decir que pertenece a varias clases al mismo tiempo.

La posibilidad de disponer de un conjunto de datos dotado de un etiquetado multi-etiqueta resulta bastante lejana; por lo cual se adoptan soluciones para situaciones más comunes y realistas, como son las etiquetas débiles. Estas implican, que una observación perteneciente a n clases, disponga de un número menor que n etiquetas. Se implementan algoritmos de clasificación que sean capaces de trabajar adecuadamente con una falta notable de información.

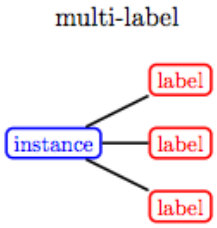

Figura 2.2: Etiquetado multi-etiqueta ó *multi-label*

Figura 2.3: Ejemplo de etiquetado débil

Con la Figura 2.3 se comprende fácilmente la ausencia de etiquetas. Las tres imágenes pueden pertenecer a las tres clases distintas (cada una de tres tenistas: **Dementieva**, **Kournikova** y **Sharapova**) y mientras que en la primera de ellas, el etiquetado es completo, en las otras dos fotografías faltaría una etiqueta en cada una.

Ambos casos comentados, se encuentran ampliamente relacionados con el tema tratado; ahora bien, dentro del puro ámbito del etiquetado parcial, existen también muchos otros aspectos. Se proponen numerosas soluciones que asumen los problemas de la misma forma que se intenta plantear en este trabajo. Dichas soluciones teóricas se tratan en el siguiente punto.

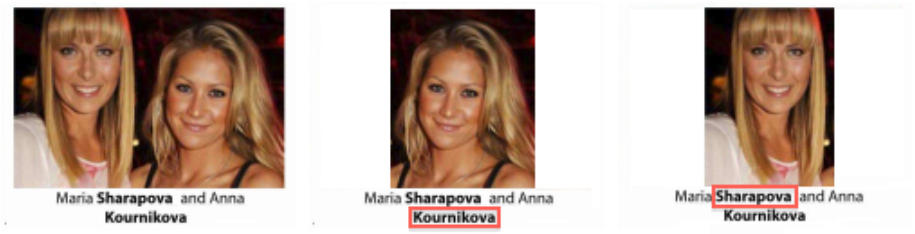

Figura 2.4: Ejemplo de etiquetado parcial

En la Figura 2.4 se pretende de nuevo clarificar el etiquetado parcial con un ejemplo sencillo. En este caso, vemos que en la imagen más a la izquierda, en la que figuran dos tenistas está clasificada

por dos nombres o clases. Al separar y decidir que tenista es cada una, tenemos que cada una de las imágenes central y derecha están clasificadas con dos etiquetas. Estas etiquetas serán parciales, dado que una es verdadera y la otra no.

Nota: Tanto la Figura 2.3 como 2.4 están basadas en la *Figure 1* explicativa de [2].

2.2. Soluciones teóricas

Conseguir un buen comportamiento de algoritmos de aprendizaje máquina para datos con etiquetado parcial, implica un gran paso de cara a mejorar el funcionamiento de los clasificadores. Se comenta ampliamente en la literatura las características que presentan estas etiquetas, señalándolas como un punto intermedio entre etiquetados supervisados muy estrictos y datos directamente sin etiquetar. No se trata tanto de las estructuras de datos que se presentan con el etiquetado, que son muy variadas, sino la forma de actuar en presencia de ambigüedades con las etiquetas.

Cuando se habla de la ambigüedad y de como resolverla, siempre se hacer referencia a clasificadores que sean capaces de actuar en presencia de etiquetas erróneas por así decirlo. Estos clasificadores se sustentan siempre sobre modelos probabilísticos, y estos son precisamente los que ofrecen soluciones.

Los modelos probabilísticos son capaces de proporcionar herramientas efectivas que permitan suplir los efectos de etiquetados de este tipo. Se proponen modelos de aprendizaje basados en optimización, minimizando funciones cuadráticas convexas [5]. En estos modelos, se pretende además, conseguir aproximaciones de aprendizaje que incorporen la información de las etiquetas parciales al escenario convencional. Dicha incorporación se produce ampliando el escenario de aprendizaje sobre el cual trabaja el clasificador.

Otras opciones tratadas, son por ejemplo, ligar como se ha dicho los clasificadores a un modelo probabilístico de interés, y una vez sobre él, maximizar la verosimiltud del conjunto de etiquetas para cada observación [6]. Esto se encuentra muy en línea con las soluciones tratadas alrededor de este trabajo, y sobre las cuales se ha basado.

Sin embargo, se buscan siempre análisis razonables de las pérdidas producidas para minimizarlas al máximo. Se ha demostrado que se pueden conseguir clasificadores no específicos que generalicen adecuadamente con datos no visibles [2] y que puedan diferenciar entre el conjunto de etiquetas. Se proponen estos clasificadores, unidos

a aprendizajes con funciones de pérdidas convexas que se puedan minimizar apropiadamente. Las perdidas producidas por observaciones etiquetadas parcialmente se conocen como *CLPL* ó *pérdidas convexas para etiquetas parciales*.

Precisamente son las pérdidas las que se pretenden calibrar en otra serie de artículos. Un caso de ellos es el que discute la obtención de un modelo efectivo que sepa estimar las pérdidas reales producidas por el etiquetado parcial [7]. Estas implican generalizar bastante el concepto de pérdidas, asociandolas con funciones de etiquetado ambigüo. Se tratan de establecer por tanto, una serie de condiciones suficientes y necesarias a cumplir por las funciones de pérdidas, de forma que si lo hicieran, se podría construir un modelo de pérdidas reales para etiquetado parcial a partir de otro modelo convencional.

El trabajo y el estudio realizado, provienen de este último artículo como propuesta a experimentos con modelos de regresión logística para etiquetado parcial y el análisis llevado a cabo sobre las pérdidas producidas para todos los casos tratados.

Capítulo 3

Desarrollo del proyecto

3.1. Introducción

A fin de facilitar la comprensión del proceso de realización del presente estudio, se hace una descripción rápida de las etapas y objetivos logrados. En primer lugar, previamente a la implementación de los algoritmos, se estudiaron las posibilidades de *steepest descent* y *Newton's method* para la función de coste *NLL* en el caso binario. Una vez decidido que con ambos algoritmos se trabajaría y analizarían los costes, se procede a su adaptación al caso multinomial o multiclase; calculando y determinando los operadores matemáticos necesarios para la función de coste. A continuación y siendo el paso previo a implementar mediante *MATLAB* los algoritmos, se decide crear un conjunto de datos artificiales, que asegurara al autor la obtención de resultados coherentes y facilidad en el análisis de los mismos.

Con el conjunto de datos creado, habiendo diseñado unas condiciones y estructuras adecuadas para los mismos; se comienza por trabajar con ambos algoritmos para el caso del etiquetado completo. Se sigue además una **metodología del aprendizaje** cuidadosa que evite arrojar cualquier duda de error sobre los experimentos. De manera similar, una vez alcanzados los objetivos propuestos en la etapa anterior, se adaptan los datos a un etiquetado diferente, en este caso el etiquetado parcial, y de nuevo se vuelve a trabajar con ellos para ambos algoritmos. Por último, con los resultados de los experimentos ya obtenidos, se realiza un análisis acerca de los objetivos propuestos y si los resultados coinciden con lo esperado. Se comparan los costes tanto para **etiquetado supervisado** como **etiquetado parcial**, y se detallan las conclusiones que dotan a este trabajo de su razón de ser y coherencia en el desarrollo.

3.2. Regresión logística

La **regresión logística** es llamada así, debido a su similitud con la regresión lineal, sin embargo se trata de una forma de **clasificación** y no de regresión.

Un clasificador mediante regresión logística puede por lo tanto aproximarse a un regresor lineal usado en estimación. Similarmente a un estimador *MLE* o estimador de *máxima verosimilitud* podemos hacer la siguiente equiparación entre un regresor lineal y la regresión logística.

- **Estimador *MLE***. Atendiendo a una observación x y deseándose estimar el parámetro w, se realiza la maximización de la probabilidad *a priori* o *verosimilitud* respecto del valor de w, de la siguiente forma:

$$\hat{W}_{MLE} = \arg\max_{w} p(x|w) \tag{3.1}$$

- **Regresión logística**. Siendo ahora el conjunto de los datos más amplio, y con etiquetas de por medio, de la forma $\{x_i, y_i\}$, corresponde a x_i el vector de valores de las observaciones y a y_i la categoría a la que pertenecen dichos valores. Las observaciones se suponen **i.i.d** (independientes e idéndicamente distribuidas). Se estima el parámetro w así:

$$\begin{aligned}
\hat{W}_{ML} &= \arg\max_{w} p(x_1, y_1, \ldots, x_N, y_N|w) \\
&= \arg\max_{w} \prod_i p(x_i, y_i|w) \\
&= \arg\max_{w} \prod_i p(x_i, |w) p(y_i|x_i, w) \\
&= \arg\max_{w} \Big[\sum_i \ln p(y_i|x_i, w) + \sum_i \ln p(x_i) \Big] \\
&= \arg\min_{w} \big[- \sum_i \ln p(y_i|x_i, w) \big] \qquad (3.2)
\end{aligned}$$

Cabe destacar en la ecuación (3.2) que, dado x_i no dependiente del coste w sobre el cual se maximiza, términos y funciones dependientes únicamente de x_i no pueden ser maximizados y por tanto se eliminan de la ecuación. Además, se pasa de maximizar la función logarítmica a minimizar la versión negativa de la misma.

Una vez vista la correlación y similitud entre ambos casos, introducimos el objetivo de la regresión logística con la función de la que dependerán el resto de los algoritmos que en este trabajo se detallan. Añadir que dicha función, conocida como *NLL* ó **verosimilitud logarítmica negativa**, procede del desarrollo de la ecuación (3.2), de la forma:

$$\arg\min_{w} \big[-\sum_{i} \ln p(y_i|x_i, w) \big] = \arg\min_{w} NLL(w) \tag{3.3}$$

Por lo que:

$$NLL(w) = -\sum_{i} \ln p(y_i|x_i, w) \tag{3.4}$$

El modelo logístico

Al implementar algoritmos de entrenamiento de un clasificador basado en regresión logística, se construye todo el estudio sobre un clasificador probabilístico que procede del siguiente modelo binario:

$$p(y|\mathbf{x}, \mathbf{w}) = Ber(y|\mu(\mathbf{x})) \tag{3.5}$$

En el cual, Ber, corresponde a la **distribución de Bernoulli** [8], cuya densidad de probabilidad puede definirse con la siguiente función:

$$f(x) = p^x(1-p)^{1-x} \tag{3.6}$$

con $x = \{0, 1\}$.
Correspondiendo $\mu(\mathbf{x})$ en la ecuación (3.5) a la función *sigmoide*.

$$\mu(\mathbf{x}) = sigm(\mathbf{w}^T\mathbf{x}) \tag{3.7}$$

Conocida también como función **logística**, se define así:

$$sigm(\eta) = \frac{1}{1+\exp(-\eta)} = \frac{e^\eta}{e^\eta + 1} \tag{3.8}$$

NLL aplicada al modelo logístico

Es como hemos dicho, la verosimilitud logarítmica negativa, de la que dependerán todos los procesos de optimización de costes en adelante. Por ello, es importante dejar definida dicha función dependiente de w de la siguiente forma para el modelo presentado: [9]

$$NLL(w) = -\sum_{i=1}^{N} \left[y_i \log \mu_i + (1 - \mu_i) \log(1 - \mu_i) \right] \tag{3.9}$$

Siendo en la ecuación (3.9) $y_i \in \{0, 1\}$, que indica el valor de la etiqueta para cada uno de los valores de las observaciones i-ésimas. Sobre la función *NLL* calcularemos los operadores *gradiente* y *hessiano*, que serán la base principal de los algoritmos de optimización, tanto de *steepest descent* como de *Newton's method*, que determinaremos más adelante.

3.3. Algoritmos de optimización

Dado que buscamos optimizar una función de coste $f(w)$, que en nuestro caso se trata de la función *NLL* anteriormente definida; hemos empleado en este estudio dos algoritmos de optimización. Dichos algoritmos son clásicos, y están basados en métodos derivativos que ayudan a obtener la dirección o valores de coste que más rápido minimizan la función sobre la que trabajan.

3.3.1. *Steepest descent*

El método de optimización *steepest descent* o *descenso de máxima pendiente* es posiblemente el algoritmo más simple en la optimización de funciones sin restricciones [10]. Consiste en el cálculo iterativo, mediante el operador gradiente, de los distintos valores de las pendientes pertenecientes al plano tangente en el punto de la función que se esté considerando. De esta manera, y mediante un factor η, se recalcula constantemente el coste de la función hasta hallar su valor mínimo posible.

$$w_{k+1} = w_k - \eta_k g_k \tag{3.10}$$

En la ecuación (3.10) se representa la operación que se realiza por cada cálculo iterativo, de forma que, el subíndice k indica la iteración, w el coste de la función a optimizar, η el *tamaño de salto* o *tasa de aprendizaje* y g el gradiente de dicha función como a continuación se indica:

$$g_k = \nabla f(w_k) \tag{3.11}$$

Aunque *steepest descent* pueda parecer sencillo, su principal dificultad reside en la determinación de la tasa de aprendizaje η adecuada. La elección de una tasa errónea puede provocar que el algoritmo

se vea incapaz de converger a un valor w mínimo para valores demasiado grandes de η, o que por contra, sea excesivamente lento, calculando muchas más iteraciones de las que necesitaría para valores excesivamente pequeños. Existen numerosas maneras de adoptar soluciones a este problema, desde un η adaptativo, que cambie su valor por cada iteración según haya sido el valor del coste, a un η constante durante toda la optimización.

$$\eta_k = \frac{\eta_0}{1+\alpha k} \;\rightarrow\; \eta_0 = 1,\; \eta_1 = \frac{1}{1+\alpha}, \ldots \tag{3.12}$$

Ejemplo de tasa de aprendizaje adaptativa, y dependiente de una constante a seleccionar $\alpha << 0$.

3.3.2. *Newton's method*

El *método de Newton* es a diferencia de *steepest descent* un algoritmo de optimización de segundo orden [10]. Es al igual que el anterior, un algoritmo iterativo, y emplea tanto el operador **gradiente** como el **hessiano** para buscar el parámetro w que minimice la función dada. El algoritmo puede describir su funcionamiento de la siguiente manera:

- Inicializar w_0
- **para** $k = 1, 2, 3 \ldots$ hasta **convergencia**
  - Evaluar $g_k = \nabla f(w_k)$
  - Evaluar $H_k = \nabla^2 f(w_k)$
  - $H_k d_k = -g_k$ para $d_k \rightarrow d_k = -H_k^{-1} g_k$
  - **búsqueda lineal** de η_k a lo largo de la direccion d_k
  - $w_{k+1} = w_k + \eta_k d_k$

Se puede observar la similitud entre ambos métodos, y de nuevo encontramos η_k como tasa de aprendizaje a ser determinada. En este caso se detalla que se realice una búsqueda lineal de la misma. Tanto el operador gradiente $g_k = \nabla f(w_k)$ como el hessiano $H_k = \nabla^2 f(w_k)$ deben ser calculados específicamente para la función de coste, previamente a la implementación. Otro aspecto destacable del algoritmo, es que se muestra más rápido habitualmente que *steepest descent*, dado que el número de iteraciones necesarias para poder converger es menor, sin embargo, por cada iteración es computacionalmente más costosa, ya que se debe realizar al menos una inversión de la *matriz hessiana*.

3.4. Regresión logística multiclase

Dado que se busca un estudio de costes para el etiquetado parcial, es razonable tener en cuenta un escenario realista y acorde a las estructuras de datos que pueden obtenerse en diferentes entornos de la sociedad. Por ello se adopta el caso multiclase o *multinomial* para el modelo de regresión logística, aunque ello implique adaptar algoritmos, funciones de coste y estructura de los datos a un escenario n-dimensional.

Se vió en el caso binario de la regresión logística de qué modelo probabilístico provenía el clasificador. En este caso el modelo es de la forma:

$$p(y = c|\mathbf{x}, \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T \mathbf{x})}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^T \mathbf{x})} \tag{3.13}$$

Es muy importante resaltar de la ecuación (3.13) la notación empleada, donde $y = c$ indica la clase a la que pertenece la observación $\mathbf{x}$, que al ser n-dimensional se representa como un vector de longitud n. $\mathbf{W}$ es por otra parte, la matriz que aglutina los parámetros deseados a calcular, uno por cada clase, de forma que $\mathbf{w}_c$ es el coeficiente, también n-dimensional, correspondiente a la clase bajo la que se está evaluando dicha función de probabilidad.

$$\mathbf{W} = \begin{pmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_{C-1} & \mathbf{w}_C \end{pmatrix} \tag{3.14}$$

Esta notación se usa a lo largo de todo el trabajo, y la probabilidad condicionada será clave en el diseño de la estructura de datos artificial sobre la que se sustentará todo el estudio. Más adelante esto se detallará cuidadosamente. Sin embargo, la ecuación (3.13) sólo representa el caso en que $y = c$, por lo tanto, el modelo probabilístico completo se define como:

$$p(\mathbf{y}|\mathbf{x}, \mathbf{W}) = \prod_{C} \mu_c^{y_c} \tag{3.15}$$

Donde μ_c en la ecuación (3.15) representa la probabilidad condicionada de la ecuación (3.13), tal que:

$$\mu_c = p(y = c|\mathbf{x}, \mathbf{W}) \tag{3.16}$$

Cabe destacar de la ecuación (3.15) un detalle más acerca de la notación, en este caso referente a las etiquetas y_c. Dichas etiquetas se distribuyen a lo largo del vector $\mathbf{y}$, correspondiendo cada una de las posiciones a la clase a la cuál puede pertenecer la observación. Se tiene entonces que:

$$\mathbf{y} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.17}$$

Correspondiendo $y_c = 1$ a la clase c que se indica mediante la posición en el vector. Atención además a que el vector $\mathbf{y}$ va asociado a una sola observación, es decir, un conjunto completo de observaciones llevará consigo otro conjunto completo de vectores etiqueta, conformando una matriz $\mathbf{Y}$. Por aclarar aún más esta notación se puede representar de la siguiente manera también:

$$\mathbf{y} = \begin{pmatrix} y_0 = 0 \\ y_1 = 0 \\ y_2 = 0 \\ \vdots \\ y_{c-1} = 0 \\ y_c = 1 \\ y_{c+1} = 0 \\ \vdots \\ y_C = 0 \end{pmatrix} \tag{3.18}$$

Junto con el modelo probabilístico se encuentra de nuevo la *verosimilitud logarítmica negativa* o *NLL* para nuestro planteamiento n-dimensional.

3.4.1. *NLL* para el caso multinomial

Siguiendo con la notación descrita anteriormente, la verosimilitud logarítmica se puede representar como:

$$
\begin{aligned}
l(\mathbf{W}) &= \log \prod_{i=1}^{N} \prod_{c=1}^{C} \mu_{ic}^{y_{ic}} \\
&= \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log \mu_{ic} \\
&= \sum_{i=1}^{N} \Big[\big(\sum_{c=1}^{C} y_{ic} \mathbf{w}_c^T \mathbf{x}_i \big) - \log \big(\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^T \mathbf{x}_i) \big) \Big] \quad (3.19)
\end{aligned}
$$

Por facilitar la comprensión de la ecuación (3.19), se observa en primer lugar que el término $\mu_i c$ corresponde al mismo descrito en la ecuación (3.16), de manera que describe la probabilidad condicionada correspondiente a una única observación para la clase asignada. Vemos que la ecuación tiene también dos *recorridos* distintos, uno por cada operador sumatorio; los índices N y C son respectivamente el número total de observaciones n-dimensionales y el número de clases que hay para los datos. De nuevo tanto los vectores $\mathbf{w}_c$ y $\mathbf{x}_i$ como la etiqueta y_{ic} vuelven a tener exactamente el mismo significado que el que se describió con el modelo probabilistico junto a la ecuación (3.13).

Sin embargo, la ecuación (3.19) describe únicamente la verosimilitud logarítmica multinomial, por ello y dado que el fin de este apartado es la función *NLL*, obtenemos la versión negativa de la anterior ecuación, resultando que:

$$NLL_{multi-class} = f(\mathbf{W}) = -l(\mathbf{W}) \quad (3.20)$$

Por lo tanto, la **función de coste** o función sobre la cual deseamos aplicar los algoritmos de optimización y a la cual se pretendía llegar es la siguiente:

$$f(\mathbf{W}) = -\sum_{i=1}^{N} \Big[\big(\sum_{c=1}^{C} y_{ic} \mathbf{w}_c^T \mathbf{x}_i \big) - \log \big(\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^T \mathbf{x}_i) \big) \Big] \quad (3.21)$$

Será la ecuación (3.21) sobre la cual se calculan los operadores matemáticos necesarios para los dos algoritmos. Únicamente añadir sobre la misma, que el coste de dicha función es en realidad una matriz de parámetros $\mathbf{W}$; por lo que al optimizar no se buscará unicamente el factor w que más minimice, sino todo el conjunto de parámetros w correspondientes a las clases, que se aglutinan en dicha matriz. Lo que conlleva a lo siguiente:

$$\mathbf{W} = \begin{pmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_{C-1} & \mathbf{w}_C \end{pmatrix} \tag{3.22}$$

Siendo C el número de clases y la agrupación por tanto de C vectores de parámetros n-dimensionales por columnas en la matriz $\mathbf{W}$.

$$\mathbf{W} = \begin{pmatrix} w_1^0 & \dots & w_C^0 \\ \vdots & \ddots & \vdots \\ w_C^n & \dots & w_C^n \end{pmatrix} \tag{3.23}$$

Nótese como detalle tanto para la ecuación (3.22) como para la ecuación (3.23), que el número de clases responde a un conjunto numérico $[1, C]$, mientras que el de dimensiones lo hace a uno del tipo $[0, n]$. Esto es debido a que para las n dimensiones consideramos un factor de **sesgo** que obliga a incluir una dimensión más. Esto se detallará más adelante y también la importancia que tiene sobre el problema a resolver.

3.4.2. Operadores matemáticos

A continuación se detallan los operadores matemáticos calculados para la función $f(\mathbf{W})$ o NLL multinomial. Se detallan matemáticamente junto con sus expresiones, sin embargo en la implementación se explicará de nuevo como fueron calculados.

Operador *gradiente*: $\nabla f(\mathbf{W})$

$$\nabla f(\mathbf{W}) = \sum_{i=1}^{N} (\mu_i - y_i) \otimes \mathbf{x}_i \tag{3.24}$$

Donde el vector $\mathbf{x}_i$ es de nuevo la observación i con sus dimensiones, mientras que μ_i e y_i corresponden a lo siguiente:

$$y_i = \big(\mathbb{I}(y_i = 1), \dots, \mathbb{I}(y_i = C - 1)\big) \tag{3.25}$$

$$\mu_i(\mathbf{W}) = \big[p(y_i = 1|\mathbf{x}_i, \mathbf{W}), \dots, p(y_i = C - 1|\mathbf{x}_i, \mathbf{W})\big] \tag{3.26}$$

Siendo $\mu_i(\mathbf{W})$ en la ecuación (3.26) el vector con todas las probabilidades condicionadas a la etiqueta de clase, por cada una de las observaciones. De nuevo N es el número de observaciones que presenta el conjunto de datos.

Por otro lado, el producto $\otimes$ es conocido como el *producto de Kronecker* y opera de la siguiente forma con matrices y vectores:

$$\{\mathbf{A} \otimes \mathbf{B}\} = \begin{pmatrix} a_{11}\mathbf{B} & \dots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \dots & a_{mn}\mathbf{B} \end{pmatrix} \tag{3.27}$$

Por lo que si $\mathbf{A}$ fuera $\{m \times n\}$ y $\mathbf{B}$ $\{p \times q\}$, el resultado de $\{\mathbf{A} \otimes \mathbf{B}\}$ sería una matriz de dimensiones $\{mp \times nq\}$. Esto será muy importante a la hora de implementar los algoritmos, ya que cualquier error en dimensionamiento u operación de los vectores y observaciones haría completamente inservible el estudio.

Operador *hessiano*: $\nabla^2 f(\mathbf{W})$

$$\nabla^2 f(\mathbf{W}) = \sum_{i=1}^{N} (\text{diag}(\mu_i) - \mu_i \mu_i^T) \otimes (\mathbf{x}_i \mathbf{x}_i^T) \tag{3.28}$$

Al igual que en la ecuación (3.11), aquí se presentan los mismos vectores que para el caso del gradiente, sin embargo queda por detallar la abreviatura $\text{diag}(\mu_i)$ que corresponde a una diagonalización sencilla como la del siguiente ejemplo:

- Si tenemos un vector $\mathbf{v}$ tal que

$$\mathbf{v} = \begin{pmatrix} 1 \\ 3 \\ 5 \\ 7 \end{pmatrix}$$

  entonces $\text{diag}(v)$ tendra la forma

$$\text{diag}(\mathbf{v}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 7 \end{pmatrix}$$

- Respondiendo por lo tanto a la diagonalización del vector $\mathbf{v}$ de longitud $\{n\}$ en una matriz de dimensiones $\{n \times n\}$.

3.5. Datos artificiales

Antes de hacer un recorrido por el desarrollo de los datos artificiales, es conveniente saber las razones de porqué este estudio no empleó datos y observaciones de problemas reales. Un inconveniente habitual en esta clase de problemas es determinar en qué grado la solución es satisfactoria. Haber utilizado datos reales provenientes de una base de datos habitual, a disposición de estudiantes e investigadores, hubiera conllevado un sobreesfuerzo a la hora de determinar modelos probabilísticos sobre el conjunto a utilizar. Por ello, y dado que este trabajo se centra en un estudio experimental de análisis de costes, y no en una solución directamente aplicada a un problema concreto, se creyó muy conveniente el realizar un diseño e implementación de un conjunto de datos artificiales. De la misma manera, el crear el autor mismo los datos implica bastante maniobrabilidad y evita riesgos de caer en sobreajuste con los algoritmos. Ahora bien, los datos se crean a partir del modelo probabilístico determinado mediante geometrías, de forma que se busca y estudian los costes mediante los puntos de máxima probabilidad.

Es decir, se trata de crear un modelo a partir de unos parámetros w, para después ocultarlos y emplear los algoritmos de optimización para obtener los mismos a partir del modelo probabilístico tanto para etiquetado supervisado como parcial.

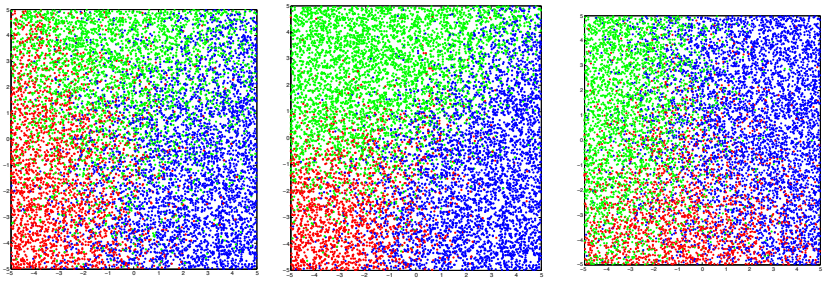

Figura 3.1: Representación de las observaciones etiquetadas. Tres geometrías

El proceso de creación de los datos tiene también su propia metología a seguir, y su índice de pasos es el siguiente:

- Elección de un conjunto de parámetros w.
- Generación aleatoria de observaciones $x_1, x_2, \ldots, x_N$
- Implementación del modelo probabilístico para cualquier observación x_i o parámetro w

- Cálculo de la probabilidad de cada observación x_i de pertenecer a una clase $c = 1, 2, \ldots, C$ asociada a los parámetros w.
- Por cada observación x_i, con las probabilidades calculadas, generar el etiquetado y_i que responda a la distribución probabilística.

3.5.1. Geometrías

Para la elección del conjunto de parámetros w que conforman la matriz **W** se usan una serie de geometrías, que en el caso de este estudio son tres. Cada una de las tres geometrías consta de 3 regiones diferentes y separadas entre ellas dentro de un escenario bidimensional xy en el que: $x \in [0, 1]$ e $y \in [0, 1]$ también.

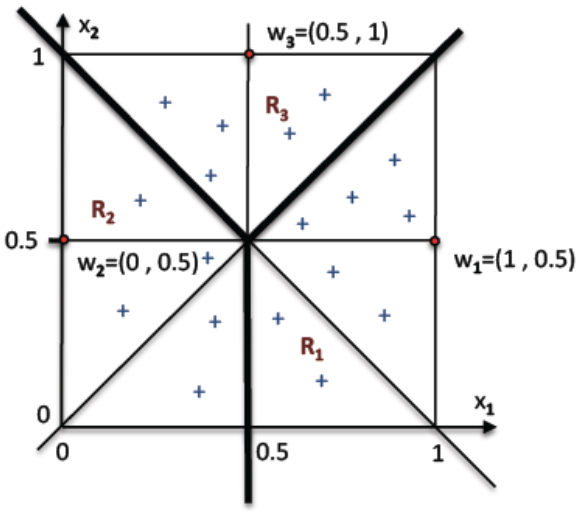

Figura 3.2: Primera geometría

En la Figura 3.2 se observa que a partir de tres rectas con un punto de corte común en el centro del escenario, se determinan las regiones **R1**, **R2** y **R3**. Cada una de estas regiones corresponde a cada una de las clases que luego se emplean en el etiquetado. Aunque no lo parezca, solamente ya al tomar esta geometría se están determinando tres hechos importantísimos acerca de los experimentos, estos son:

- Elegir un escenario bidimensional implica que el modelo multinomial de regresión logística trabajará para una dimensión *dos*, *tres* si contamos el **sesgo**.

- Al dividir dicho escenario en tres regiones, se está determinando que el número de clases será *tres*, por lo que $C = 3$ y los vectores etiqueta $\mathbf{y}_i$ tendrán longitud tres también.

- Todas las observaciones que se generen tendrán que pertenecer al entorno de las regiones, por lo cual xy serán también: $x \in [0, 1]$ e $y \in [0, 1]$.

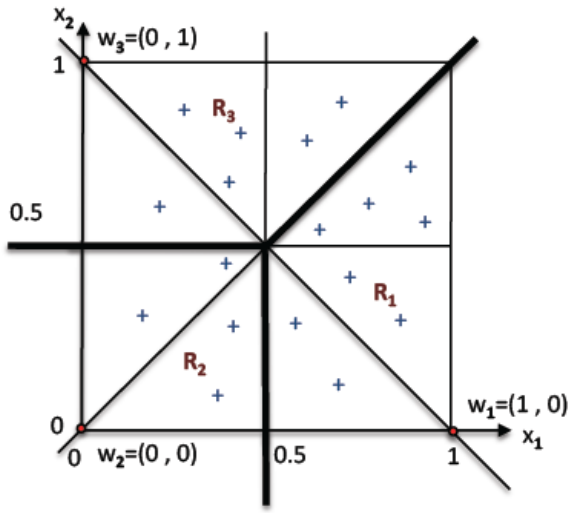

Figura 3.3: Segunda geometría

Si se observan la Figura 3.2 y la Figura 3.3 se puede observar que no todas las regiones son iguales de tamaño. Esto implicará distribuciones de probabilidad diferentes entre las regiones que sean distintas, aunque claro, siempre dependiendo del modelo probabilístico de la ecuación (3.13) como se explica.

Posiblemente, la clave de las tres geometrías sea el conjunto de parámetros w, dado que, tal y como se observa en la Figura 3.4, representan los puntos de **máxima probabilidad** en cada una de las regiones, siendo equidistantes al resto. Presentan una geometría triangular, y cada pareja de parámetros, equidista de la frontera entre las regiones que representan. Estos parámetros, en forma de conjunto o matriz $\mathbf{W}$ son los que minimizan la función $f(\mathbf{W})$ dentro del entorno determinado. Hay otros costes fuera de las regiones determinadas que minimizarán más, pero no interesan dado que se salen de la geometría diseñada; esto se trata más adelante.

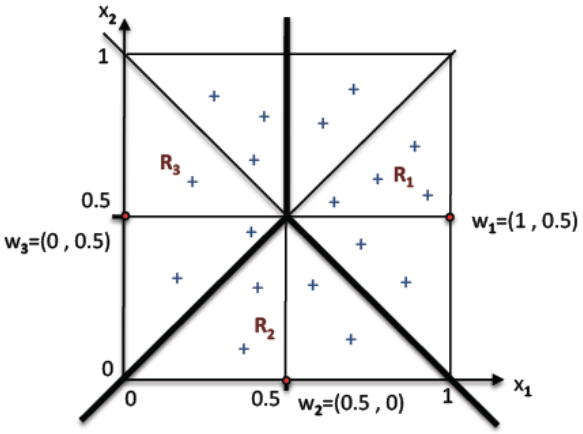

Figura 3.4: Tercera geometría

3.5.2. Cálculo de los parámetros *w*

Como podemos ver en las Figuras 3.2, 3.3 y 3.4, se indican los parámetros w_i junto con sus coordenadas en el plano de datos. Sin embargo, como ya se ha comentado, dichos costes tienen tres y no dos coordenadas, siendo la tercera el **sesgo**. Su cálculo es muy importante, dado que será el que determina la posición de las fronteras entre las regiones; dicho cálculo se consigue con la resolución del sistema de ecuaciones:

$$\begin{aligned} (\mathbf{w}_1 - \mathbf{w}_2)^T \mathbf{x} &= 0 \\ (\mathbf{w}_2 - \mathbf{w}_3)^T \mathbf{x} &= 0 \\ (\mathbf{w}_3 - \mathbf{w}_1)^T \mathbf{x} &= 0 \end{aligned} \tag{3.29}$$

Mediante el sistema de ecuaciones (3.29) se obtiene el sesgo para cualquiera de las tres geometrías. Con lo cual para este trabajo, los coeficientes o costes con los que se trabaja y por tanto aquellos que serán los puntos de máxima probabilidad y que minimizan al máximo la función de coste son:

Geometría 1:

$$\mathbf{w}_1 = \begin{pmatrix} 1 \\ 1 \\ 0{,}5 \end{pmatrix} \mathbf{w}_2 = \begin{pmatrix} 0 \\ 0 \\ 0{,}5 \end{pmatrix} \mathbf{w}_3 = \begin{pmatrix} 1 \\ 0{,}5 \\ 1 \end{pmatrix} \tag{3.30}$$

Geometría 2:

$$\mathbf{w}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \mathbf{w}_2 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \mathbf{w}_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \tag{3.31}$$

Geometría 3:

$$\mathbf{w}_1 = \begin{pmatrix} 1 \\ 1 \\ 0{,}5 \end{pmatrix} \mathbf{w}_2 = \begin{pmatrix} 0 \\ 0{,}5 \\ 0 \end{pmatrix} \mathbf{w}_3 = \begin{pmatrix} 0 \\ 0 \\ 0{,}5 \end{pmatrix} \tag{3.32}$$

Precisamente serán estos parámetros los cuales servirán para la evaluación de las pérdidas resultantes con los algoritmos de optimización. Serán por así decirlo, la referencia en todo momento de hacia donde deben tender los parámetros recalculados. Es importante también caer en la cuenta que estos son únicamente dependientes de las geometrías diseñadas, por lo que, serán iguales tanto para etiquetado supervisado como etiquetado parcial. Lo que quiere decir, que el proceso de evaluación será el mismo tanto para uno como para otro.

3.5.3. Generación de los datos

En el proceso de creación de los datos intervienen dos factores. Por una parte, la generación de las observaciones, que implica que haya una aletoriedad establecida; y por la otra, el cálculo de las probabilidades de dichas observaciones, que serán importantísimas en el proceso de etiquetado. La generación de las observaciones es efectivamente aleatoria, esto se consigue mediante la útil herramienta que dispone *MATLAB* para proporcionar estas soluciones. Usamos por tanto el siguiente comando:

$$\mathbf{x}_{obs} = rand(n_{obs}, 2);$$

Que nos genera un número determinado n_{obs} de observaciones de dos dimensiones cada una. El hecho fundamental del comando *rand* es que los valores que genera siempre estarán dentro del conjunto $[0, 1]$ para las dimensiones que se le propongan. Por lo tanto, cumpliendo la condición que se necesitaba sobre las geometrías establecidas.

En segundo lugar, una vez se tienen las observaciones e implentada la función de probabilidad de la ecuación (3.13), que ahora, sabiendo que hay tres clases, toma la forma de:

$$p(y = c|\mathbf{x}_i, \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T \mathbf{x}_i)}{\exp(\mathbf{w}_1^T \mathbf{x}_i) + \exp(\mathbf{w}_2^T \mathbf{x}_i) + \exp(\mathbf{w}_3^T \mathbf{x}_i)} \quad (3.33)$$

Se calculan las probabilidades de cada una de las observaciones para cada una de las clases, estableciendo una matriz $\mathbf{P}$ de dimensiones $\{N \times C\}$ siendo $C = 3$, que resultaría como la del ejemplo a continuación.

$$\mathbf{P} = \begin{pmatrix} 0{,}6920 & 0{,}0861 & 0{,}2219 \\ 0{,}6682 & 0{,}0604 & 0{,}2713 \\ \vdots & \vdots & \vdots \\ 0{,}1725 & 0{,}2395 & 0{,}5879 \end{pmatrix} \quad (3.34)$$

Cada una de las columnas de $\mathbf{P}$ en la matriz (3.34) corresponde a las probabilidades de pertenecer a la clase determinada, siendo la primera columna para la clase uno, la segunda para la dos, y la tercera para la tres. Esta matriz es la que emplea el proceso de etiquetado para generar los vectores etiqueta.

Con el fin de dejar aún más claro esto:

$$\mathbf{P} = \begin{pmatrix} p_{1,i=1} & p_{2,i=1} & p_{3,i=1} \\ p_{1,i=2} & p_{2,i=2} & p_{3,i=2} \\ \vdots & \vdots & \vdots \\ p_{1,i=N} & p_{2,i=N} & p_{3,i=N} \end{pmatrix} \quad (3.35)$$

En la que:

$$\begin{aligned} p_{1,i} &= p(y = 1|\mathbf{x}_i, \mathbf{W}) \\ p_{2,i} &= p(y = 2|\mathbf{x}_i, \mathbf{W}) \\ p_{3,i} &= p(y = 3|\mathbf{x}_i, \mathbf{W}) \end{aligned} \quad (3.36)$$

Añadir además que cada uno de los datos incluidos en la matriz $\mathbf{P}$ del ejemplo (3.34), han sido tomados de los valores reales obtenidos en la implementación con MATLAB. Con lo que se puede observar que dicho grado de trabajo con decimales será de cuatro cifras.

3.5.4. Proceso de etiquetado

Tal y como se lleva hablando durante todo el presente trabajo, hay dos tipos de etiquetado, y por tanto dos procesos distintos de generación de etiquetas. Sin embargo en el caso de este estudio, el

etiquetado parcial vendrá en consecuencia al etiquetado supervisado, por lo que, el parcial se explicará brevemente en la parte de los experimentos en el Capítulo 4. Se habla aquí entonces del etiquetado supervisado, con una única etiqueta por cada una de las observaciones, que constan de tres probabilidades de pertenecer a cada una de las clases, como se organiza en la matriz **P** obtenida en el apartado anterior.

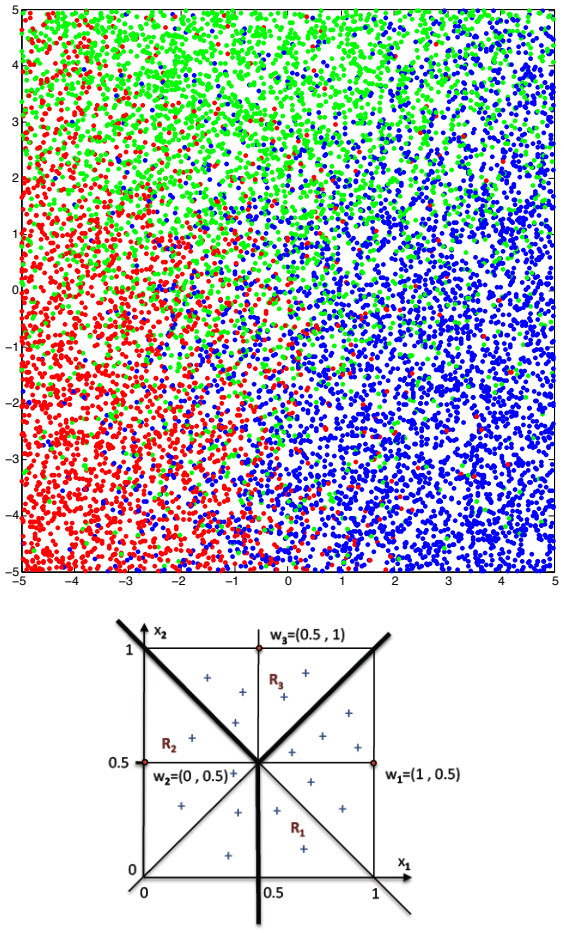

Figura 3.5: Etiquetado supervisado en la primera geometría

Para obtener la matriz de etiquetas **Y** deseada a partir de la matriz de probabilidades **P**, se generan aleatoriamente valores entre cero y uno, de manera que según su valor, se le asigna una etiqueta de clase de esta forma:

- **para** i=1,2,3 ... hasta número de observaciones N

- Se toma $\mathbf{p}_i = \begin{pmatrix} p_{clase1} & p_{clase2} & p_{clase3} \end{pmatrix}$ en la cual:

$$p_{clase1} + p_{clase2} + p_{clase3} = 1$$

- Se genera un número aleatorio r mediante el comando de MATLAB:

$$r = rand(1);$$

- Se determina a partir de r y $\mathbf{p}_i$ a que clase pertenece la observación $\mathbf{x}_i$, siguiendo el siguiente proceso:
  - **if** $r < p_{clase1} \rightarrow y_i = clase1$
  - **elseif** $r < p_{clase2} \rightarrow y_i = clase2$
  - **else** $y_i = clase3$
  - **end**

- **Nota**: Aunque a $\mathbf{x}_i$ se le asigne una etiqueta de clase, esto no quiere decir que tenga que pertenecer obligatoriamente a la región correspondiente a dicha clase. Por esto, se determinarán zonas de mayor y menor probabilidad de clases. Esto se puede ver facilmente en las Figuras 3.5, 3.6 y 3.7.

Un hecho curioso que se da en este proceso de etiquetado, es que mediante representaciones gráficas reales del etiquetado es muy difícil determinar regiones que presenten mayor o menor concentración de etiquetas. Esto es debido a que el área del plano geométrico en la que trabajamos en realmente pequeña, sin embargo, aunque no podamos reconocer ningún patrón a simple vista, los algoritmos si son capaces de hacerlo y trabajar adecuadamente como se demuestra en el siguiente capítulo.

En las tres Figuras 3.5, 3.6 y 3.7, se observa que los datos etiquetados representados se encuentran entre $[-5, 5]$, esto quiere decir que la superficie en la cual se distinguen las tres regiones para cada una de las geometrías es cien veces mayor que sobre la que trabajan los algoritmos de optimización. Esto nos facilita a nosotros percibir las regiones con algo de ayuda, pero como ya se ha dicho, los algoritmos son capaces de trabajar con densidades de etiquetas mucho más adecuadas. Sorprende incluso la precisión con la que logran aproximarse a los parámetros óptimos. Dichos algoritmos trabajarán para un escenario igual al que se representa en la Figura 3.8.

Añadir por último, que para las cuatro Figuras 3.5, 3.6, 3.7 y 3.8 se ha fijado $N = 10000$, es decir, *diez mil* observaciones etiquetadas.

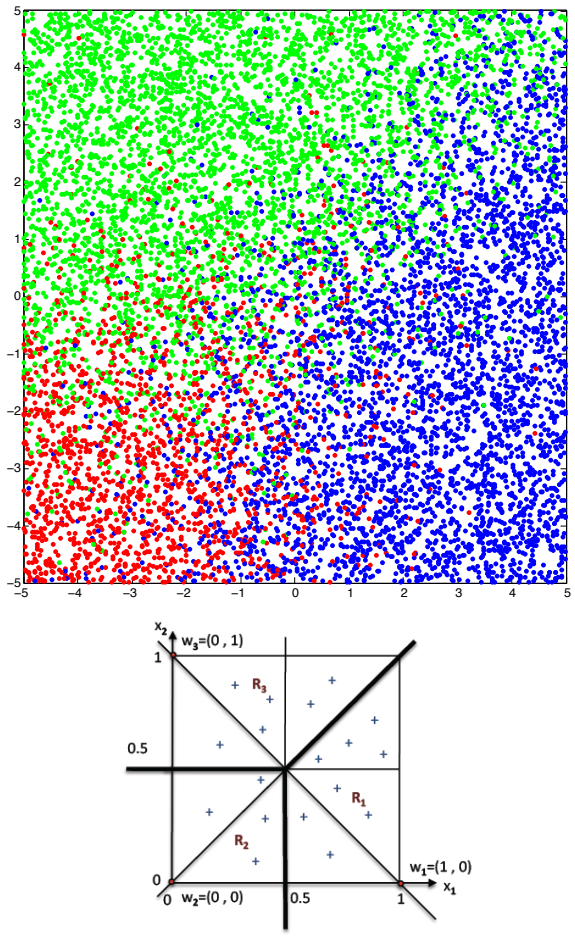

Figura 3.6: Etiquetado supervisado en la segunda geometría

3.6. Metodología experimental

Para llevar a cabo los experimentos deseados sobre la estructura de datos creada, es necesario seguir una serie de pasos y ser cuidadoso con ciertos detalles a lo largo de la implementación. Dichos pasos se determinan a lo largo del trabajo con el conjunto de los **datos**, de los **algoritmos** y de los **parámetros**.

3.6.1. Metodología del aprendizaje

El **aprendizaje máquina** requiere seguir una metología muy determinada a la hora de implementar cualquier tipo de experimento, solución o algoritmo concreto. Esto se debe a la importancia de tener garantías sobre los resultados obtenidos, dado que trabajar con da-

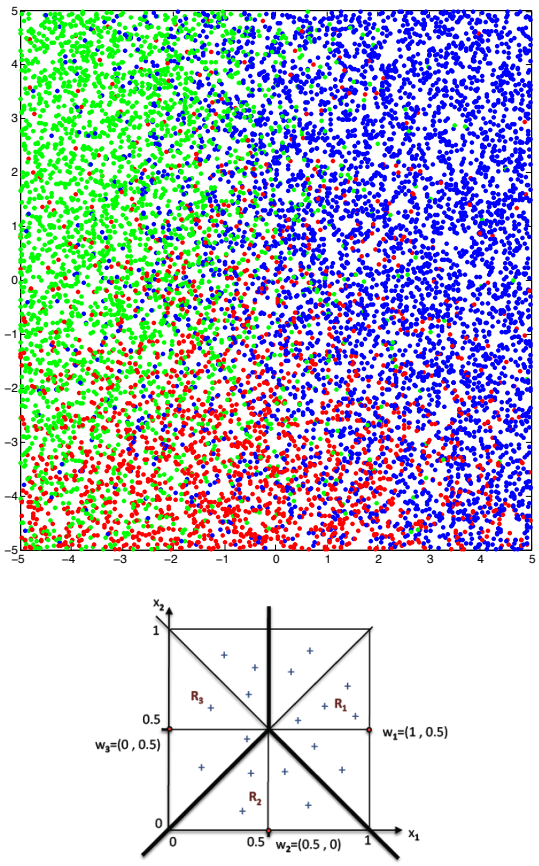

Figura 3.7: Etiquetado supervisado en la tercera geometría

tos y modelos probabilísticos, implica que exista una facilidad muy grande de caer en errores dificílmente detectables. Por ello como se indica, se trabaja con la **metodología del aprendizaje**, que se conforma por tres etapas en serie: *entrenamiento*, *validación* y *test*. La subdivisión en estas etapas se lleva a cabo a partir del conjunto de los datos disponibles, lo que conlleva a dividir dicho conjunto en tres subconjuntos, cada uno de ellos con el mismo origen pero diferente destino, dado que la labor de cada etapa será diferente que la del resto.

Se observa en la Figura 3.9 que a partir del conjunto de datos total se reserva la mitad de ellos para *test*, mientras que la otra mitad se guarda también mitad y mitad para *entrenamiento* y *validación*. A continuación se detalla las funciones y propósitos con cada uno de los subconjuntos.

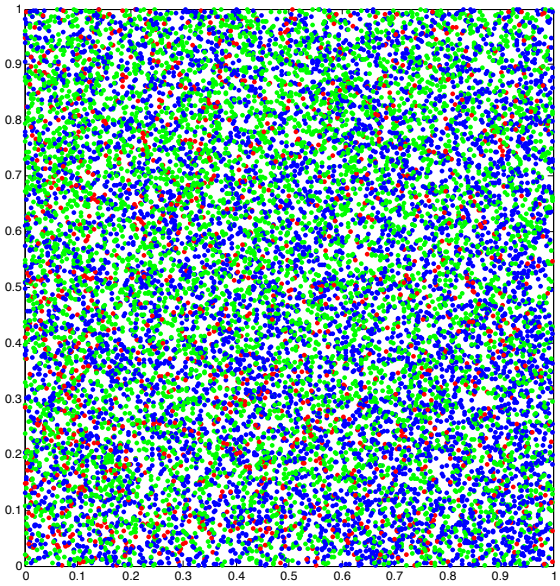

Figura 3.8: Distribución real del etiquetado para $[0, 1]$. Primera geometría

Figura 3.9: Datos de *entrenamiento*,*validación* y *test*

Datos entrenamiento

Los datos de entrenamiento suponen la primera etapa del proceso de implementación, y son por tanto, los primeros datos y observaciones con los que se tiene contacto. El algoritmo se entrena con ellos, y se comprueba el funcionamiento correcto y deseado, se ajusta el número de iteraciones con las que se trabaja o las que sean suficientes para converger al resultado esperado. Estos ajustes son sobre los *parámetros*, de ahí que se hable de estimación paramétrica.

Datos validación

Seguidamente después del proceso de entrenamiento, se toma la otra parte reservada al proceso de validación. Con los *parámetros* ya fijados, se prueban de nuevo los algoritmos, esta vez con el objetivo de ajustar los *hiperparámetros*, que en nuestro problema concreto, sería por ejemplo la **tasa de aprendizaje**. Se busca por tanto encontrar los *hiperparámetros* óptimos, que son parte fundamental en la implementación.

Datos test

Por último, una vez tanto *parámetros* como *hiperparámetros* fijados, procedemos a obtener los **W** deseados a partir del conjunto de datos de validación. Con estos costes obtenidos, se prueban y evalúan los algoritmos con los datos de test. A partir de los resultados que esta evaluación arroje, se estudiarán las pérdidas que los costes traen consigo, esto llevará a la obtención de conclusiones.

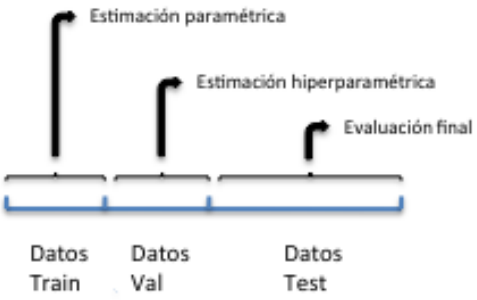

Figura 3.10: Estimación *paramétrica*, *hiperparamétrica* y evaluación final

3.6.2. Elección de parámetros

Tanto parámetros como hiperparámetros se eligen durante el entrenamiento y validación. Ambos se eligen a raíz de la obtención de las menores pérdidas por parte de los algoritmos. Su ajuste es siempre difícil, dado que en este caso, número de iteraciones con tasa de aprendizaje se encuentran muy relacionadas. Se trata por tanto de encontrar un equilibrio adecuado que permita converger a una tasa de error mínima sobre las capacidades que se le están dando al algoritmo para trabajar. Normalmente, los parámetros escogidos durante el entrenamiento proporcionan pérdidas menores que las que

luego son obtenidas durante la evaluación de los algoritmos con los datos de *test*.

3.6.3. Evaluación de los algoritmos

Evaluar los parámetros sobre el conjunto datos de *test* implica siempre atenerse a resultados diferentes que con el conjunto de entrenamiento y validación. Se aplican las mismas condiciones para la evaluación y se realiza un cálculo de las pérdidas para los mejores parámetros e hiperparámetros escogidos en la etapa anterior. Dichas pérdidas se analizan, y se descubre si entran dentro de lo esperado y adecuado. Si esto resulta así, se puede dar por satisfactorio el funcionamiento del algoritmo bajo los parámetros proporcionados, si no lo fuera, se deberá volver al proceso de entrenamiento con el otro conjunto, para de nuevo escoger tasas de aprendizaje y número de iteraciones.

3.7. Etiquetado supervisado

El etiquetado supervisado es el primero con el cual se van a enfrentar ambos algoritmos, es por tanto, el que mejores condiciones de aprendizaje va a presentar en este estudio. Atendiendo a (3.18), se detalla como se representan dichas etiquetas en este trabajo. Sin embargo, (3.18) supone únicamente una etiqueta para una observación de muchas que habrá en el conjunto. Entonces, ¿cómo se organiza dicho etiquetado para tantas observaciones? Tal y como a continuación se muestra:

$$\mathbf{y}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \mathbf{y}_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \dots \mathbf{y}_{N-1} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \mathbf{y}_N = \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (3.37)$$

Cada etiqueta $\mathbf{y}_i$ pertenece a una observación del conjunto $[0,N]$ con N observaciones, y cada una de ellas pertenece a una **única** clase, lo cual no implica que dos o más no puedan pertenecer a la

misma. Cada una de ellas se organiza como un vector, y se aglomeran en forma de matriz como a continuación vemos:

$$\mathbf{Y}_{supervisado} = \begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{N-1} & \mathbf{y}_N \end{pmatrix} \tag{3.38}$$

Cada vector o etiqueta $\mathbf{y}_i$ corresponde a una columna de la matriz $\mathbf{Y}$, la cual tendrá por número de fijas el número de clases, y por número de columnas el de etiquetas. Siendo sus dimensiones $\{C \times N\}$.

$$\mathbf{Y}_{supervisado} = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \tag{3.39}$$

Habrá por lo cual en el etiquetado supervisado, un único elemento distinto de cero por cada columna, es decir habrá un elemento unidad por cada una de las observaciones. La matriz (3.39) es la que sale resultante tras el proceso de etiquetado en la creación de los datos artificiales. Tal cual se muestra, es como se trabaja con ella.

3.8. Etiquetado parcial

Al igual que en el caso del etiquetado supervisado, el etiquetado parcial comienza desde la creación de los vectores $\mathbf{y}_i$ o etiquetas. En esta versión de las etiquetas hay que observar que ya no se tiene únicamente un sólo elemento distinto de cero por cada vector, sino que hay más de uno. De hecho, solamente uno de ellos representará la verdadera pertenencia a la clase de la observación, y en el ejemplo que a continuación se ve, dicho elemento se resalta en *negrita*.

$$\mathbf{y}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \mathbf{1} \\ 0 \\ \vdots \\ 1 \end{pmatrix} \mathbf{y}_1 = \begin{pmatrix} 0 \\ \mathbf{1} \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \cdots \mathbf{y}_{N-1} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ \mathbf{1} \end{pmatrix} \mathbf{y}_N = \begin{pmatrix} \mathbf{1} \\ 0 \\ 1 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (3.40)$$

Y de nuevo, la matriz $\mathbf{Y}$ se representará con la unión de los vectores $\mathbf{y}_i$, ocupando cada uno una columna.

$$\mathbf{Y}_{parcial} = \begin{pmatrix} \mathbf{y}_0 & \mathbf{y}_1 & \cdots & \mathbf{y}_{N-1} & \mathbf{y}_N \end{pmatrix} \quad (3.41)$$

Por lo que finalmente la matriz $\mathbf{Y}$ resulta tener un aspecto similar con el caso del etiquetado supervisado. Esto es muy importante, dado que la diferencia fundamental entre los experimentos que se realizan es la que presentan $\mathbf{Y}_{supervisado}$ e $\mathbf{Y}_{parcial}$. Lo que implica que los algoritmos que se utilizan deberán responder adecuadamente ante observaciones etiquetadas con ambas matrices.

$$\mathbf{Y}_{parcial} = \begin{pmatrix} 0 & 0 & \dots & 1 & \mathbf{1} \\ 0 & \mathbf{1} & \dots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 1 & 0 & \dots & 0 & 0 \\ \mathbf{1} & 1 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 1 & 0 & \dots & \mathbf{1} & 0 \end{pmatrix} \quad (3.42)$$

En los ejemplos (3.39) y (3.42) se destacan los elementos representativos de la verdadera clase de las observaciones únicamente por facilitar la comprensión de la arquitectura planteada. En la aplicación e implementación real no se destacará la verdadera etiqueta de la observación, dado que todas las etiquetas tendrán mismo grado de importancia. Serán los algoritmos los que consiguen resolver dicha ambigüedad.

Capítulo 4

Experimentos

4.1. Introducción

Una vez que la implementación del trabajo se ha llevado a cabo, es el momento de recabar toda la información obtenida y realizar un análisis detallado de los resultados. Se han realizado numerosos experimentos con las implementaciones, tanto para las tres geometrías propuestas, como para los dos tipos de etiquetado y los dos algoritmos de optimización.

Se tiene siempre presente en los resultados, que se trabaja con datos artificiales, de forma que las soluciones siempre se verán acotadas a valores óptimos determinados por el autor. Esto es, muy probablemente, un factor que implica aún mas exigencia sobre las soluciones obtenidas, que si se hubieran empleado datos reales por ejemplo. Se clasifica este capítulo en pequeñas partes que ayuden a la comprensión; cada resultado irá acompañado por una detallada explicación acerca de qué parte del estudio se está tratando.

Todos los resultados experimentales provienen de la herramienta matemática MATLAB, sobre la que, como se ha dicho, se hizo la implementación. Una cosa que esto implica, es la capacidad de cálculo que se ha podido alcanzar en este trabajo, dado que siempre se ha visto limitada por lo que MATLAB era capaz de proporcionar. Las implementaciones se han tenido que limitar tanto en número de iteraciones como número de observaciones, poniendo un límite máximo de alrededor de *veinte mil*, nunca superado. Hay que tener en cuenta el número de operaciones que implican para la herramienta la iterativa inversión de matrices, verdaderamente costosas para *Newton's method* por ejemplo.

Sin embargo, los rangos de observaciones han sido suficientes para la obtención de conclusiones satisfactorias como se deseó desde el inicio del trabajo.

4.2. Resultados experimentales

En cuanto a los experimentos realizados, se pretende buscar resultados adecuados que permitan *caracterizar* el funcionamiento de ambos algoritmos de optimización. Para llevar esto a cabo se han tenido en cuenta una larga lista de factores. Es importante tener en cuenta que el trabajo con datos de este tipo es en ocasiones complicado, y aunque se desee y tengan más o menos acotados los resultados que se obtendrán, siempre se presenta una cierta variabilidad en los resultados, debido a la naturaleza probabilística del problema.

4.2.1. Objetivos de los algoritmos

Cada uno de los dos algoritmos presenta sus propias características, y como se explica en el capítulo anterior, *Newton method* es siempre algo mejor que *steepest descent*, dado que el segundo es algo menos elaborado que el primero. Sin embargo, el segundo se muestra mucho más ágil a la hora del entrenamiento con los datos, esto se detallará junto con la presentación de los resultados.

Objetivo con *steepest descent*

Como ya se dice, *steepest descent* conlleva una cantidad menor de cálculos que el otro algoritmo. Se pretende observar el funcionamiento del método, primero con datos dotados de etiquetado supervisado que aseguren un correcto funcionamiento y probabilidades de error razonables; y segundo, con datos que presenten etiquetas parciales de manera completa (todas las observaciones muestran ambigüedad con su etiquetado). Una vez observado el comportamiento del algoritmo para los casos, y las geometrías diseñadas, se comparan los resultados obtenidos que proporcionen un análisis satisfactorio.

Objetivo con *Newton's method*

Mientras que con *steepest descent* se pueden realizar entrenamientos muy diversos y con cantidades de datos bastante grandes, *Newton's method* no permite eso. Sin embargo su capacidad de optimización es bastante más potente y hay que observar de nuevo su funcionamiento tanto con etiquetado supervisado como parcial, y detallar que trabaja adecuadamente para ambos casos.

4.2.2. Resultados

Para la obtención de los resultados se ha seguido la metodología del aprendizaje descrita. Ha consistido en entrenamiento con las observaciones generadas artificialmente, obteniendo los mejores parámetros que proporcionaran la menor *tasa de error* posible.

Las figuras que se incluyen en este apartado, que sirven como exposición de los resultados obtenidos, muestran las tasas de error obtenidas frente al número de observaciones con las que el algoritmo ha trabajado.

Tasa de error

La tasa de error pretende proporcionar una medida fiable de cómo esta trabajando el algoritmo, para obtenerla se sigue el siguiente proceso, que al mismo tiempo es precisamente el seguido como metodología en la obtención de los resultados.

- **Generación** de observaciones, junto con el cálculo de probabilidades y el etiquetado de los datos.
- **Entrenamiento** con observaciones etiquetadas, se determinan iteraciones y tasa de aprendizaje junto con el proceso de validación.
- **Obtención** de parámetros $\mathbf{W}$ que minimizan la función de coste para distintos valores de tasa de aprendizaje. Se escoge una tasa de aprendizaje η_{opt}.
- **Elección** de un $\mathbf{W}_{opt}$ que provoque la mínima cantidad de errores comparando su etiquetado sobre el conjunto de validación con el etiquetado original.
- **Evaluación** del algoritmo sobre el conjunto de test, únicamente etiquetando con $\mathbf{W}_{opt}$ desde el cálculo de probabilidades. Se compara dicho etiquetado con el original para el mismo conjunto de test. Se obtiene una tasa de error.

Cabe destacar dos diferencias entre los etiquetados. En el original, para cada observación, se presentan unas probabilidades de pertenencia a una clase, y a partir de la aletoriedad de un número generado, se asigna una etiqueta de clase. En cambio, para el etiquetado resultante se escoge directamente la clase más probable que resulta a partir del proceso de optimización.

Por lo tanto la tasa de error se calcula como:

$$t_{error} = \frac{n_{errores}}{n_{obs}} \tag{4.1}$$

En la que $n_{errores}$ indica la cantidad de etiquetas erróneas entre ambos etiquetados, y n_{obs} el número de observaciones empleadas para optimizar. Se ha considerado que tasa de error muy cercanas a 0,5 implican que el algoritmo no estaba trabajando adecuadamente, es decir, no estaba siendo capaz de clasificar las observaciones ni discernir entre las 3 clases.

Resultados con *steepest descent*

Un hecho que se pudo comprobar desde el principio del proceso de entrenamiento con *steepest descent*, fue obtener pérdidas muy altas para el conjunto de datos etiquetado, en el que las observaciones $\mathbf{x}$ y los parámetros $\mathbf{W}$ se encuentran dentro de $[0, 1]$. Verdaderamente, con tasas de errores, cercanas a 0,5, quiere decir que el algoritmo no solamente no trabaja adecuadamente, sino que el escenario no le está permitiendo clasificar ni discernir las observaciones entre las tres clases. Esto se explica y resuelve así:

Los parámetros W y las fronteras de decisión

Como se ha comentado ya, los parámetros $\mathbf{W}$ son el origen del cálculo del etiquetado sobre el que trabaja el algoritmo de optimización. Si dichos parámetros, como se muestra en la Figura 4.1, se encuentran muy juntos o por contra, no lo suficientemente separados, se producirá un etiquetado *difícil* sobre el que optimizar.

Esto se debe a que los parámetros, con su posición, determinan fronteras de decisión en el escenario. Todas las observaciones que se encuentren en el espacio no determinado entre las tres fronteras de la Figura 4.1, serán dotadas de un etiquetado posiblemente erróneo que no indicará al optimizador la verdadera clase o región a la que pertenece.

Se vió en el capítulo anterior, como una ampliación del escenario [0,1] producía etiquetados mucho más fáciles de trabajar. Sin embargo, si se quiere una solución efectiva que no afecte al escenario de las observaciones, se pueden recalcular los parámetros $\mathbf{W}$ de forma que las fronteras de decisión se junten, reduciendo el espacio de indeterminación.

$$\mathbf{W}' = k\mathbf{W} \tag{4.2}$$

La Figura 4.2 muestra como esto sucede. El etiquetado resultante será más o menos adecuado según sea la constante k de la

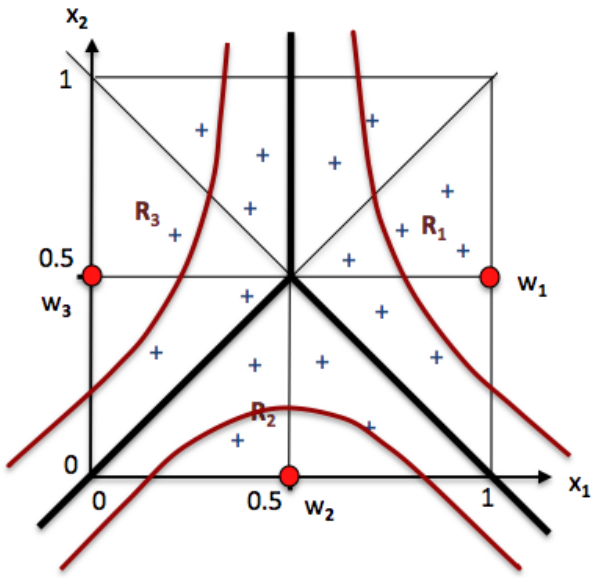

Figura 4.1: Escenario [0,1] con las fronteras de decisión dependientes de los parámetros **W**.

.

ecuación (4.2). Cuanto mayor sea la constante, menor será el espacio entre fronteras de decisión, el etiquetado mejor, y por tanto menores tasas de error resultantes del proceso de optimización. Para los experimentos con *steepest descent* se trabaja con una constante $k = 8$ que se ha considerado suficiente para que el algoritmo trabaje adecuadamente, presentándose tasas de error en los entrenamientos satisfactorias.

Por último acerca del recálculo de los parámetros, se comprueba experimentalmente que a partir de valores de $k > 10$, las tasas de error no varían ni mejoran, por lo cuál es una de las razones por las que se eligió el valor indicado.

Las geometrías y sus resultados

Para la obtención de resultados se trabajó con un valor máximo de *veinte mil* observaciones. Tal y como se explica en la **metodología del aprendizaje**, se genera un conjunto total de *cuarenta mil* observaciones, de las cuales, la mitad se dedican a entrenamiento y validación. Una vez realizado el proceso de entrenamiento, se emplea la otra mitad en la obtención de las tasas de error para el conjunto de *test*.

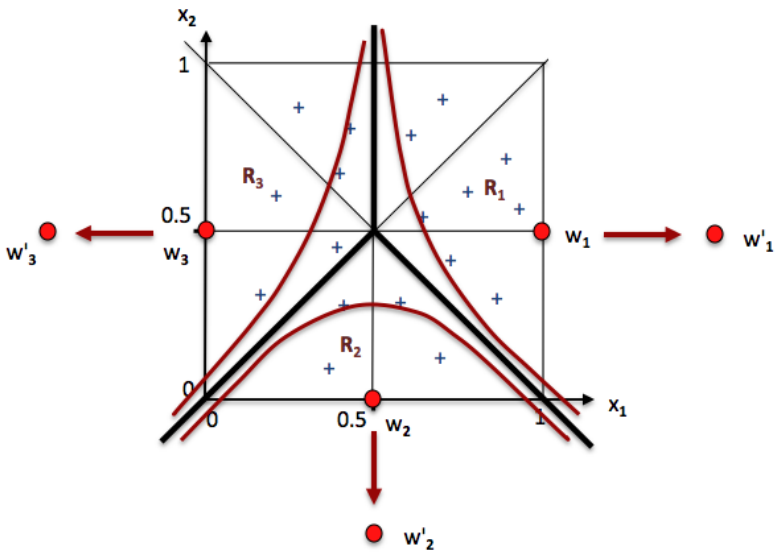

Figura 4.2: Parámetros $\mathbf{W}'$ alejados implican fronteras de decisión más juntas.

.

Cada una de las geometrías presenta características diferentes, aunque para todos ellas se determinó la constante k detallada. Dadas las diferencias, cada una de ellas implica un proceso de entrenamiento diferente, en el cual número de iteraciones y tasa de aprendizaje (parámetros e hiperparámetros del aprendizaje) serán distintas.

- **Geometría 1**: Se eligieron los siguientes parámetros e hiperparámetros:

  Tasa de aprendizaje:

  $$\eta_{k(1)} = 0{,}00002$$

  Iteraciones:

  $$n_{iteraciones(1)} = 50$$

- **Geometría 2**: Se eligieron los siguientes:

  Tasa de aprendizaje:

  $$\eta_{k(2)} = 0{,}00002$$

  Iteraciones:

  $$n_{iteraciones(2)} = 110$$

- **Geometría 3**: Se eligieron los siguientes:
  Tasa de aprendizaje:

$$\eta_{k(3)} = 0{,}00003$$

  Iteraciones:

$$n_{iteraciones(3)} = 100$$

Cada uno de ellos supuso para su caso concreto, la obtención de las tasas de error menores durante el entrenamiento. Una razón fundamental para la elección de tasas de aprendizaje tan pequeñas es el de valores del gradiente muy grandes para la función $f(\mathbf{W})$ en un escenario muy pequeño, lo que provoca grandes errores.

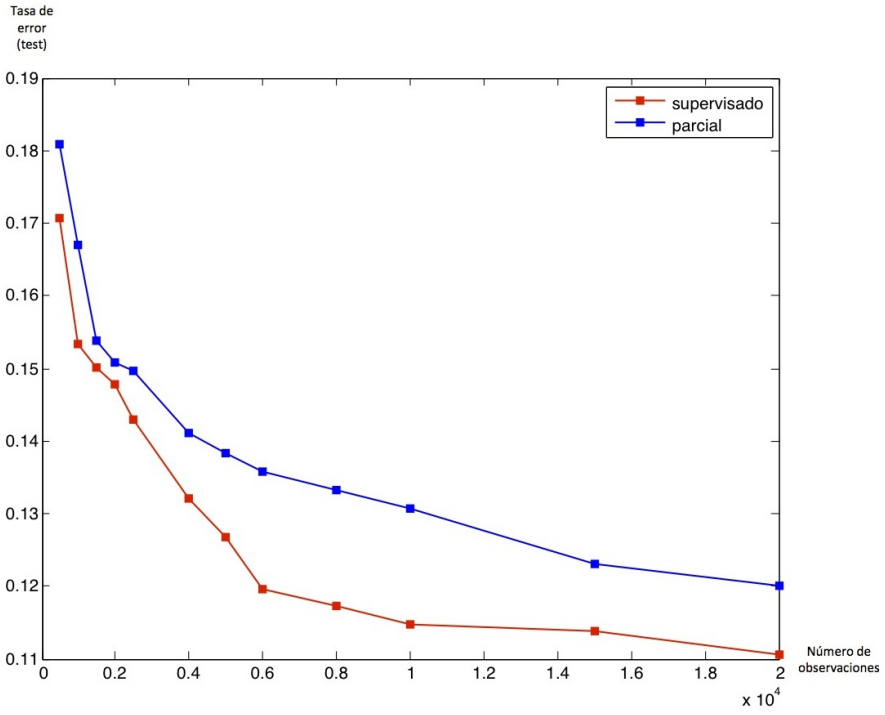

Figura 4.3: Tasas de error frente a número de observaciones. Etiquetados *supervisado* y *parcial*. Steepest descent. Primera geometría

.

En las gráficas incluidas, se observa en el primer caso represetado en la Figura 4.3, dos curvas formadas por doce puntos de medida según el número de observaciones. Estos doce puntos serán iguales para las tres gráficas de *steepest descent* y son: { 500, 1.000, 1.500, 2.000, 2.500, 4.000, 5.000, 6.000, 8.000, 10.000, 15.000 y 20.000 observaciones } . Se ha considerado un límite inferior de 500 ó 1.000 observaciones, entendiendo que un número menor, no implica relevancia

en los resultados obtenidos, y un límite superior de 20.000 observaciones, considerando una cantidad suficientemente grande como para proporcionar al algoritmo un máximo de información con la que trabajar.

En la Figura 4.3 se representan además, tasas de error para los dos tipos de etiquetado. Para supervisado se da una tasa de error máxima de $t_{error_{max}} = 0{,}1706$ y para parcial $t_{error_{max}} = 0{,}1809$. Mientras que las mínimas son $t_{error_{min}} = 0{,}1105$ para supervisado, $t_{error_{min}} = 0{,}1201$ para parcial.Se observan dos curvas que aproximadamente se muestran paralelas, manteniendo unos márgenes aceptables en cuanto a tasas de error.

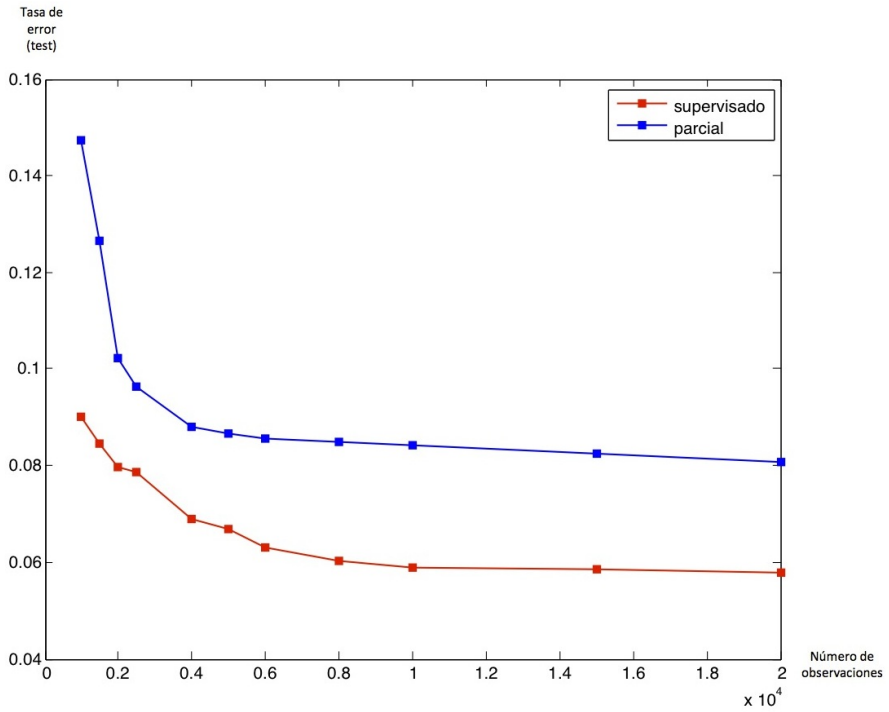

Figura 4.4: Tasas de error frente a número de observaciones. Etiquetados *supervisado* y *parcial*. Steepest descent. Segunda geometría

.

También en la Figura 4.4 se pueden ver curvas algo más parejas entre sí que en el caso anterior. Para esta segunda geometría se observa que a medida que se incrementa el número de observaciones, tanto el etiquetado supervisado como el parcial no reducen mucho más sus tasas de error. Esto puede ser debido a que en ambos casos se está alcanzando el mínimo posible para las condiciones dadas al algoritmo de optimización. Se dan unas tasas de error máximas y

mínimas de: $t_{error_{max}} = 0{,}0902$ y $t_{error_{min}} = 0{,}0578$ para supervisado y para parcial $t_{error_{max}} = 0{,}1475$ y $t_{error_{min}} = 0{,}0863$.

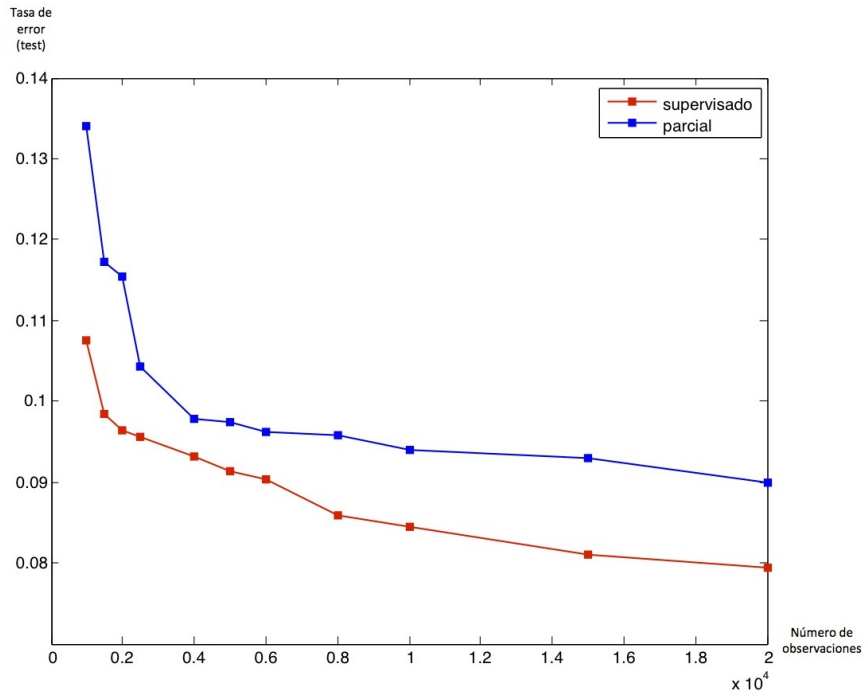

Figura 4.5: Tasas de error frente a número de observaciones. Etiquetados *supervisado* y *parcial*. Steepest descent. Tercera geometría

.

En cuanto a la Figura 4.5, similarmente se pueden determinar dos curvas de etiquetado parcial muy parecidas en prestaciones. Se establece un margen de diferencia entre etiquetado supervisado y parcial aproximadamente entre 0,01 y 0,005 en la tasa de error. Al igual que para los dos casos anteriores, se demuestra un funcionamiento satisfactorio del algoritmo con etiquetas parciales. Las tasas que se dan para este caso son: $t_{error_{max}} = 0{,}1075$ y $t_{error_{min}} = 0{,}0794$ para supervisado y para parcial $t_{error_{max}} = 0{,}1340$ y $t_{error_{min}} = 0{,}0899$.

Un hecho señalable sobre las Figuras 4.3, 4.4 y 4.5, es su utilidad para determinar las diferencias en cuanto a número de observaciones entre los etiquetados. Esto significa, que trazando líneas paralelas al eje de abcisas, a partir del punto de corte con ambas curvas se puede determinar el número de observaciones de más que necesitará el etiquetado parcial para alcanzar la misma tasa de error que el supervisado.

Convergencia a los parámetros W y W′

Una de las claves a entender en este trabajo, es que curiosamente, aunque los datos se valgan de los puntos de máxima probabilidad para establecer el etiquetado, los algoritmos, y concretamente el modelo probabilíistico no fuerza a que se pueda optimizar exclusivamente a estos puntos. Esto quiere decir que, si el conjunto de vectores $\mathbf{W}$ conformado según las ecuaciones (3.22) y (3.23) como:

$$\mathbf{W} = \begin{pmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \mathbf{w}_3 \end{pmatrix} \tag{4.3}$$

Se considera como el conjunto de parámetros óptimos o de máxima probabilidad de pertenencia a una clase concreta, se puede plantear sobre el modelo de probabilidad definido en la ecuación (3.33) lo siguiente:

- Definiendo p y p' según las ecuaciones (4.4) y (4.5) a continuación.

$$p = p(y = 1|\mathbf{x}_i, \mathbf{W}) = \frac{\exp(\mathbf{w}_1^T \mathbf{x}_i)}{\exp(\mathbf{w}_1^T \mathbf{x}_i) + \exp(\mathbf{w}_2^T \mathbf{x}_i) + \exp(\mathbf{w}_3^T \mathbf{x}_i)} \tag{4.4}$$

$$\begin{aligned}
p' &= p(y = 1|\mathbf{x}_i, (\mathbf{W} + \Delta)) \\
&= \frac{\exp\left((\mathbf{w}_1 + \Delta)^T \mathbf{x}_i\right)}{\exp\left((\mathbf{w}_1 + \Delta)^T \mathbf{x}_i\right) + \exp\left((\mathbf{w}_2 + \Delta)^T \mathbf{x}_i\right) + \exp\left((\mathbf{w}_1 + \Delta)^T \mathbf{x}_i\right)} \\
&= \frac{\exp(\mathbf{w}_1^T \mathbf{x}_i)\exp(\Delta^T \mathbf{x}_i)}{\exp(\mathbf{w}_1^T \mathbf{x}_i)\exp(\Delta^T \mathbf{x}_i) + \exp(\mathbf{w}_2^T \mathbf{x}_i)\exp(\Delta^T \mathbf{x}_i) + \exp(\mathbf{w}_3^T \mathbf{x}_i)\exp(\Delta^T \mathbf{x}_i)} \\
&= \frac{\exp(\mathbf{w}_1^T \mathbf{x}_i)\exp(\Delta^T \mathbf{x}_i)}{\Big(\exp(\mathbf{w}_1^T \mathbf{x}_i) + \exp(\mathbf{w}_2^T \mathbf{x}_i) + \exp(\mathbf{w}_3^T \mathbf{x}_i)\Big)\exp(\Delta^T \mathbf{x}_i)} \\
&= \frac{\exp(\mathbf{w}_1^T \mathbf{x}_i)}{\exp(\mathbf{w}_1^T \mathbf{x}_i) + \exp(\mathbf{w}_2^T \mathbf{x}_i) + \exp(\mathbf{w}_3^T \mathbf{x}_i)} \\
&= p
\end{aligned} \tag{4.5}$$

- Se tiene que $p = p'$, de forma que generalizando a todas las clases c se tiene:

$$p(y = c|\mathbf{x}_i, \mathbf{W}) = p(y = c|\mathbf{x}_i, (\mathbf{W} + \Delta)) \tag{4.6}$$

Esto explica que puede haber más parámetros $\mathbf{w}_c$ de la clase $(\mathbf{w}_c + \Delta)$ que conlleven a probabilidades p adecuadas que busca

el optimizador. Tiene mucha importancia, dado que graficamente, aunque los parámetros se encuentren distantes de los óptimos, se puede estar clasificando satisfactoriamente.

Resultados con *Newton's method*

El objetivo con *Newton's method* era al igual que con *steepest descent*, poder obtener unos resultados satisfactorios del entrenamiento con datos etiquetados de diferente forma. Desde el primer momento que se entrena con este algoritmo, el cual necesita ya de por sí un número menor de iteraciones e información para poder converger al valor deseado, se detecta que tratar de emplear las mismas cantidades de observaciones que en el caso anterior no iba a ser posible.

Trabajar con más de *cinco mil* observaciones implicaba demasiado tiempo al computador con el que se han realizado los experimentos. Por ello se establecieron unos límites en el número observaciones desde *tres mil* hasta *quinientas* observaciones.

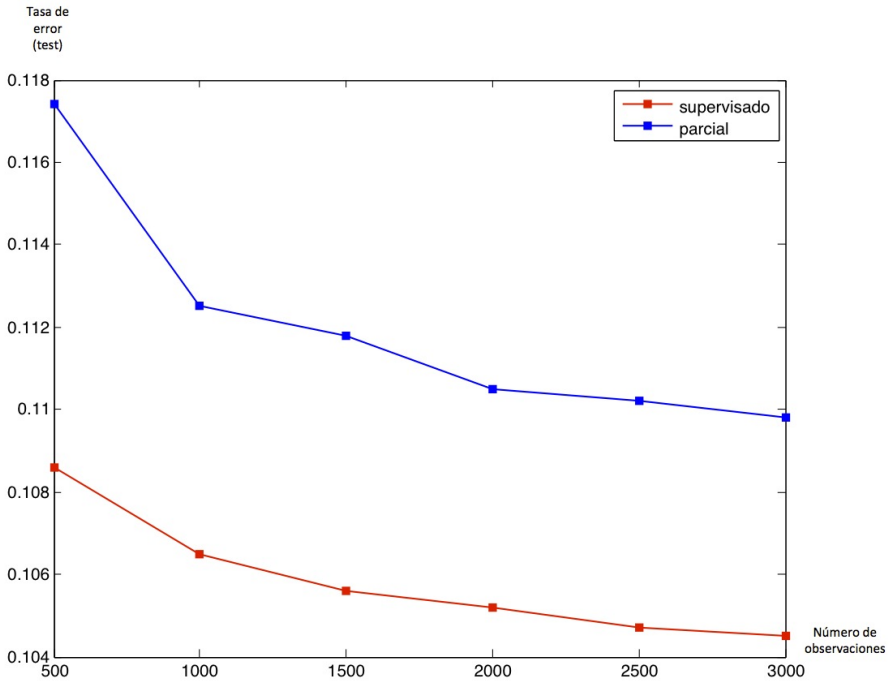

Figura 4.6: Tasas de error frente a número de observaciones. Etiquetados *supervisado* y *parcial*. Newton's method. Primera geometría

.

En la Figura 4.6 se observan unos valores para las tasas de error

que oscilan entre $t_{error_{max}} = 0{,}108$ y $t_{error_{min}} = 0{,}1053$ para el etiquetado supervisado y entre $t_{error_{max}} = 0{,}1174$ y $t_{error_{min}} = 0{,}1098$ para el etiquetado parcial. Al igual que para *steepest descent* se determina un comportamiento adecuado para el funcionamiento del algoritmo, que de nuevo es capaz de resolver la ambigüedad.

Puede además, tratarse de comparar las tasas de error de las Figuras 4.6, 4.7 y 4.8 con las anteriores 4.3, 4.4 y 4.5. Se observaría que para el mismo número de observaciones, las tasas de error son parecidas. Entonces, ¿cómo se asegura que *Newton's method* es más potente o mejor que *steepest descent*? Esto es debido a que el número de iteraciones empleadas es menor en el segundo algoritmo que en el primero. Es decir, *Newton's method* está logrando con el mismo número de observaciones e información las mismas tasas de errores en la mitad o menos iteraciones. Eso sí, cada una de ellas es bastante más lenta e implica un número mayor de operaciones. Esto se puede ve muy bien reflejado en la Figura 4.9 a continuación en el capítulo, donde ambos algoritmos trabajan en las mismas condiciones.

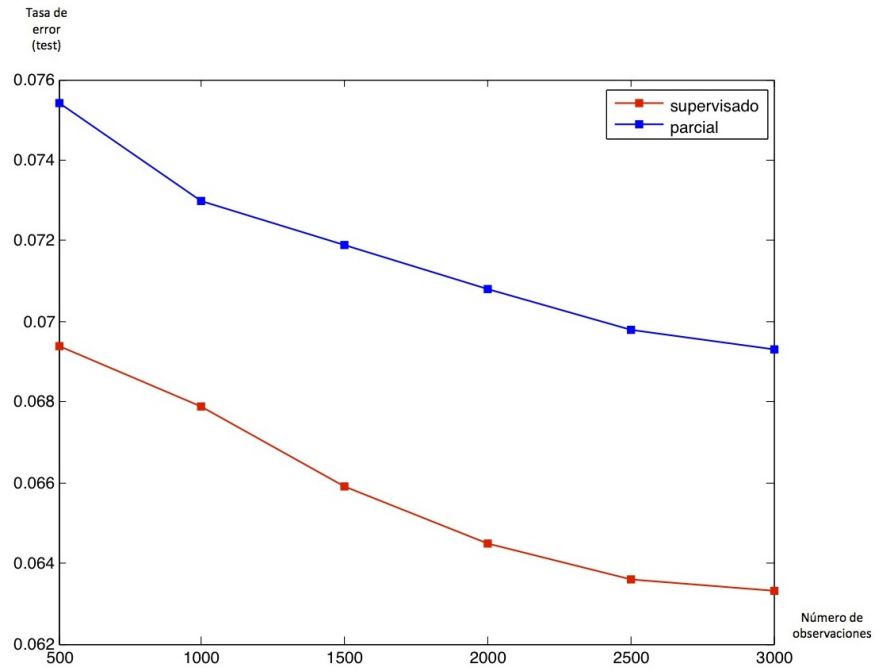

Figura 4.7: Tasas de error frente a número de observaciones. Etiquetados *supervisado* y *parcial*. Newton's method. Segunda geometría

.

Con la Figura 4.7, se muestra por último las dos curvas obtenidas.

Dichas curvas al igual que para la Figura 4.8, se basan en cinco puntos bajo los cuales se han obtenido las tasas de error. Estos son: {500, 750, 1.000, 2.000 y 3.000 observaciones} .

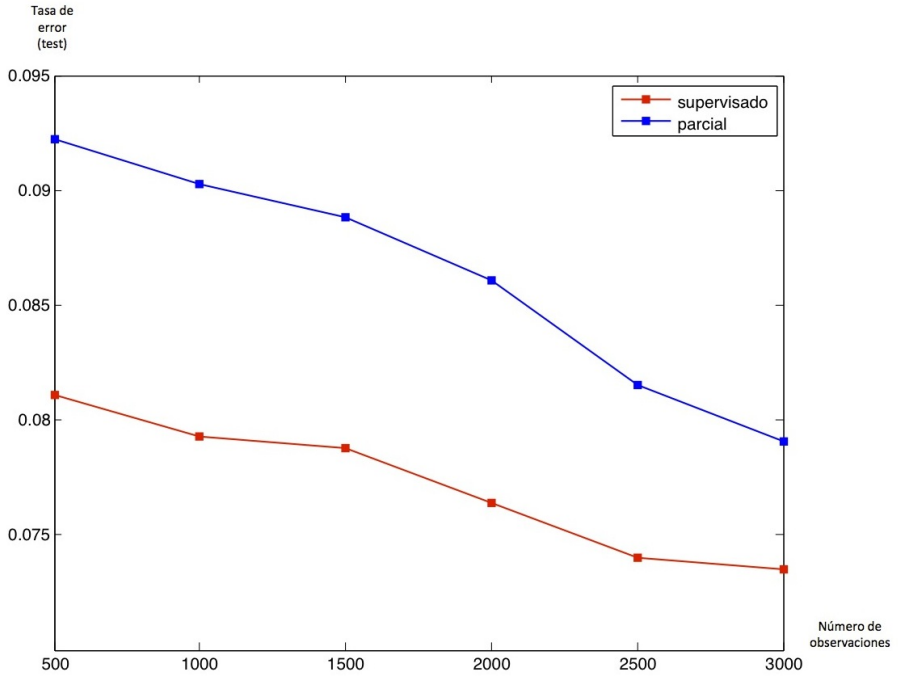

Figura 4.8: Tasas de error frente a número de observaciones. Etiquetados *supervisado* y *parcial*. Newton's method. Tercera geometría

.

Al igual que para las dos anteriores, la Figura 4.8 indica un funcionamiento adecuado del algoritmo para ambos casos. De nuevo se determinan unas diferencias en la tasa de error que rondan entre 0,01 y 0,05; que además decrece a medida que se dispone de más observaciones o muestras.

Para la experimentación con *Newton's method* se trabajó con las dos primeras geometrías, razón de ello el coste que representaba entrenar con tantas observaciones. Para el óptimo funcionamiento del algoritmo se determinan los siguientes parámetros e hiperparámetros:

- **Geometría 1**: Se eligieron los siguientes parámetros e hiperparámetros:

  Tasa de aprendizaje:

  $$\eta_{k(1)} = 0{,}18$$

Iteraciones:

$$n_{iteraciones(1)} = 35$$

- **Geometría 2**: Se eligieron los siguientes:

  Tasa de aprendizaje:

  $$\eta_{k(2)} = 0{,}175$$

  Iteraciones:

  $$n_{iteraciones(2)} = 35$$

- **Geometría 3**: Se eligieron los siguientes:

  Tasa de aprendizaje:

  $$\eta_{k(2)} = 0{,}18$$

  Iteraciones:

  $$n_{iteraciones(2)} = 35$$

El problema de la inversión de matrices

Comenta este trabajo las diferencias entre ambos algoritmos; Newton's method requiere de un número menor de iteraciones para poder converger, sin embargo la cantidad de operaciones que realiza es mucho mayor por cada iteración. Esto se debe en primer lugar al cálculo del operador *hessiano*.

$$\mathbf{H}_k = \nabla^2 f(\mathbf{W}) \tag{4.7}$$

Y en segundo lugar a la inversión de la *matriz hessiana* de manera que si la operación realizada es la siguiente:

$$\mathbf{W}_{k+1} = \mathbf{W}_k + \eta_k \mathbf{d}_k \tag{4.8}$$

Entonces el vector $\mathbf{d}_k$ se calcula de esta forma:

$$\mathbf{d}_k = -\mathbf{H}_k^{-1} \nabla f(\mathbf{W}) \tag{4.9}$$

Precisamente es la inversión de la matriz $\mathbf{H}_k$ la que implica que las iteraciones del algoritmo sean más lentas. En general la inversión de matrices es un aspecto complicado en el tratamiento de datos, dado que se trata de una de las operaciones matemáticas más costosas, por la propia naturaleza que requiere el cálculo a través de todos y cada uno de los elementos de la matriz. Esto, por desgracia, limita bastante el entrenamiento con grandes cantidades de datos, tanto por tiempo como por esfuerzo computacional del ordenador.

En la implementación se planteó la matriz *hessiana* $\mathbf{H}_k$ teniendo en cuenta el producto de *Kronecker* detallado en la ecuación (3.27). Esto implica que la matriz tiene la siguiente forma:

$$\mathbf{H}_k = \nabla^2 f(\theta_k) = \begin{pmatrix} \mathbf{H}_{k_{c,c}}^{1,1} & \mathbf{H}_{k_{c,c}}^{1,2} & \mathbf{H}_{k_{c,c}}^{1,3} \\ \mathbf{H}_{k_{c,c}}^{2,1} & \mathbf{H}_{k_{c,c}}^{2,2} & \mathbf{H}_{k_{c,c}}^{2,3} \\ \mathbf{H}_{k_{c,c}}^{3,1} & \mathbf{H}_{k_{c,c}}^{3,2} & \mathbf{H}_{k_{c,c}}^{3,3} \end{pmatrix}_{9\times 9}$$

Lo que significa que la matriz $\mathbf{H}_k$ está compuesta por 9 matrices $\mathbf{H}_{kc,c}$. Cada una de ellas proviene de la ecuación (3.28), y tienen dimensiones (3×3), respondiendo los índices c, c a la combinatoria de las tres clases que se realiza para el cálculo de cada una de ellas.

Por lo tanto se invertirá la matriz de dimensiones (9×9) tantas veces como iteraciones se indique al algoritmo, eso implicará una gran cantidad de operaciones, teniendo en cuenta que cada una de las matrices $\mathbf{H}_{kc,c}$ se debe calcular para todas y cada una de las observaciones. Esto quiere decir, que si se trabajara con *veinte mil* observaciones y *cincuenta* iteraciones; se estarían manejando *nueve* matrices que son la suma de *veinte mil* matrices cada una, calculadas iterativamente, y esto sería solamente una de las *cincuenta* iteraciones.

Rebajar la carga computacional

Aunque en este trabajo no se realiza, si es interesante contar soluciones que se pueden tomar en cálculos matriciales para ahorra el esfuerzo computacional que se realiza con ellas. Una de esas soluciones es por ejemplo sobre una matriz obtenida $\mathbf{A}_o$ que tuviera esta forma:

$$\mathbf{A}_o = \begin{pmatrix} {a_o}^{1,1} & {a_o}^{1,2} & {a_o}^{1,3} \\ {a_o}^{2,1} & {a_o}^{2,2} & {a_o}^{2,3} \\ {a_o}^{3,1} & {a_o}^{3,2} & {a_o}^{3,3} \end{pmatrix}_{9\times 9} \tag{4.10}$$

Se podría considerar, según la naturaleza del problema que se estuviera resolviendo, que la información más valiosa de la matriz se agrupa en torno a la diagonal principal: ${a_o}^{1,1}$, ${a_o}^{2,2}$ y ${a_o}^{3,3}$. Decidido esto, se podrían eliminar elementos lejanos a dicha diagonal. Provoca que el número de elementos distintos de cero sea menor, la función de evaluar los elementos sea algo más sencillo, y por tanto el esfuerzo de cálculo menor en operaciones como la inversión matricial.

El problema de esto, es que obviamente se pierde información en el grado que se eliminen elementos de la matriz. La clave es, que en matrices de tamaños descomunales, y muy difíciles de manejar, la

eliminación de estos elementos no afecta en tanta medida como lo haría en el ejemplo planteado. Dos maneras pueden ser las siguientes:

$$\mathbf{A}_o = \begin{pmatrix} {a_o}^{1,1} & {a_o}^{1,2} & 0 \\ {a_o}^{2,1} & {a_o}^{2,2} & {a_o}^{2,3} \\ 0 & {a_o}^{3,2} & {a_o}^{3,3} \end{pmatrix} \mathbf{A}_o = \begin{pmatrix} {a_o}^{1,1} & 0 & 0 \\ 0 & {a_o}^{2,2} & 0 \\ 0 & 0 & {a_o}^{3,3} \end{pmatrix} \quad (4.11)$$

4.2.3. *Steepest descent* frente a *Newton's method*

Durante todo el trabajo se ha hablado de las diferencias que existían entre ambos algortimos, y por ello, se hicieron una serie de experimentos que permitieran demostrarlo. Verdaderamente las diferencias entre ambos métodos son grandes, dado que uno es de primer orden y el otro de segundo orden. El avance por iteración de *Newton's method* es mucho mayor que *steepest descent*.

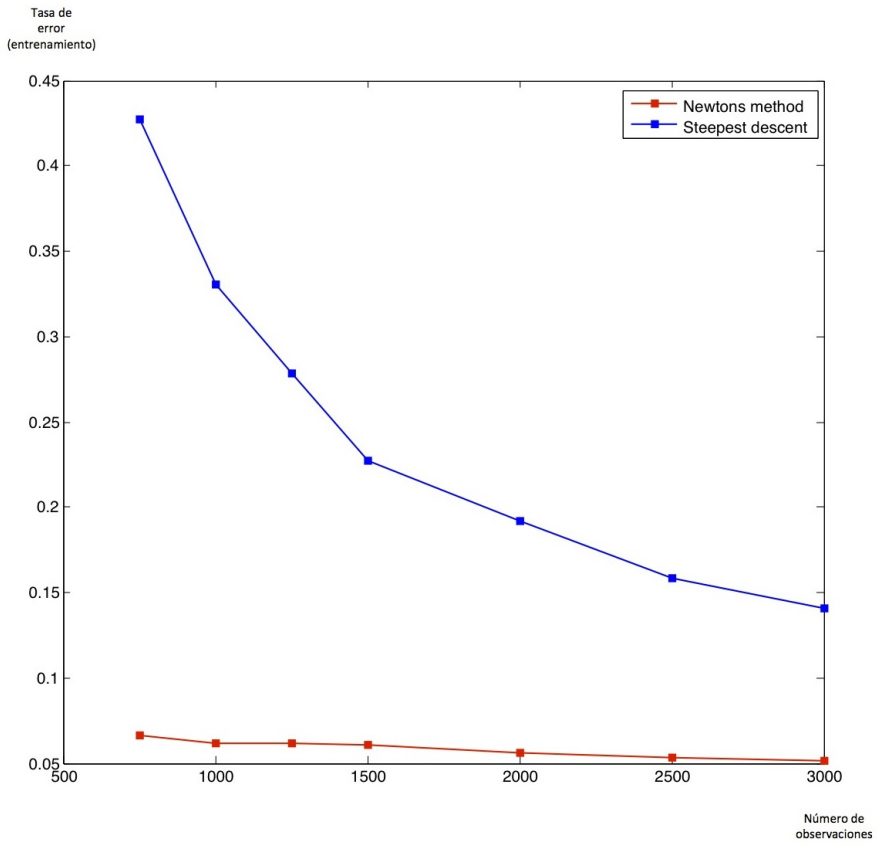

Figura 4.9: Comparación de los dos algoritmos de optimización. *Steepest descent* frente a *Newton's method* en igualdad de condiciones.

.

La Figura 4.9 representa las tasas de error obtenidas durante el entrenamiento con los dos métodos de optimización. Las condiciones en que ambos trabajan son exactamente las mismas, usando un conjunto de observaciones desde 500 a 3.000 observaciones, unas tasas de aprendizaje óptimas para ambos, y el mismo número de iteraciones. El etiquetado usado ha sido en este caso exclusivamente supervisado y se aplicó sobre la geometría dos.

Una de las claves sobre la figura, es el uso del número de iteraciones, usando en ambas *veinticinco* para cada una de las muestras de la gráfica tomadas, es decir para cada número de observaciones estudiado. Se observa que la curva de *steepest descent* desciende drásticamente a medida que gana información con el número de observaciones. Sin embargo, el descenso en la tasa de error para *Newton's method* es bastante menor, aunque parte de un valor muy bajo. Esto se produce, debido a que para 25 iteraciones y menos de 1.000 muestras u observaciones, *Newton's* está optimizando adecuadamente ($t_{error} < 0{,}1$), mientras que *steepest descent* no lo consigue ($t_{error} > 0{,}4$).

4.2.4. La importancia del etiquetado parcial

Se ha tratado de demostrar con los resultados expuestos que dos algoritmos de optimización, trabajando sobre un modelo probabilístico basado en regresión logística, son capaces de clasificar adecuadamente incluso en presencia de etiquetado parcial. Sin embargo, demostrado esto, faltarían otras cuestiones que resolver también. Una de ellas es, si en presencia de un etiquetado mixto, es decir, habiendo tanto etiquetas supervisadas como etiquetas parciales, merecería la pena incluir el etiquetado parcial o directamente quitarlo y usar el supervisado.

Esta parte trata de demostrar por lo tanto, que aunque se trate de etiquetado parcial, el algoritmo es capaz de discernir entre la ambigüedad del etiquetado y obtener información para optimizar mejor. Que siempre es mejor incluir las etiquetas parciales, dado que, aunque sea en menor medida que las supervisadas, provocan un descenso en las pérdidas.

La Figura 4.10 muestra este aspecto sobre el etiquetado parcial. La gráfica procede de los experimentos y pruebas realizados para la geometría uno con un conjunto total de datos dotado con 10.000 observaciones. Con dicho conjunto, se procedió gradualmente a ir etiquetando porcentualmente las observaciones con etiquetas parciales, desde un cero por ciento a un noventa por ciento. Con lo cual, por cada una de las proporciones de etiquetado supervisado

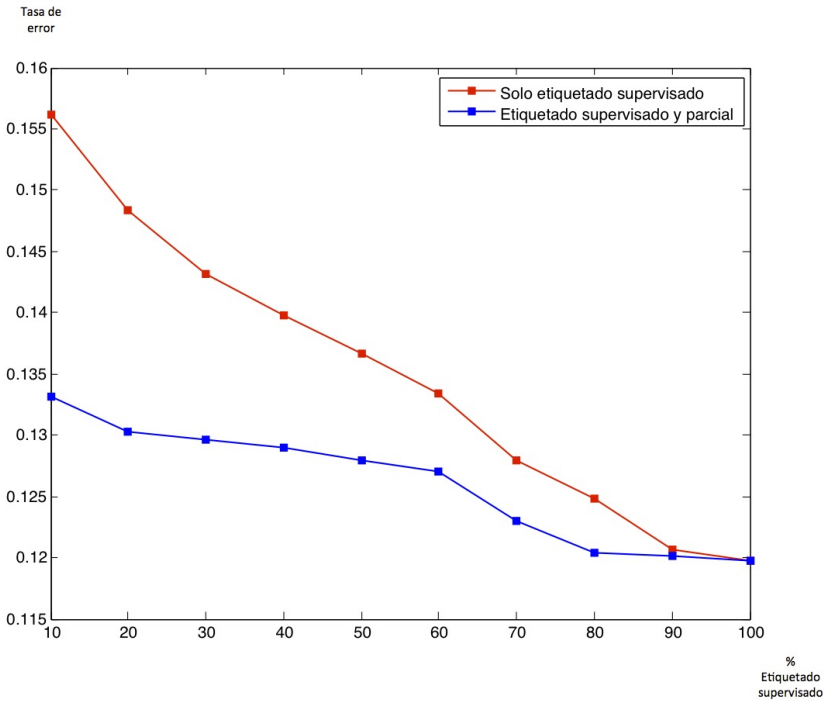

Figura 4.10: Tasas de error usando el etiquetado parcial y no usándolo. Etiquetado mixto (supervisado y parcial) con diferentes porcentajes de etiquetas supervisadas.

.

frente a parcial, se entrena primero sólo con las etiquetas supervisadas (menos de 10.000 cuando no hay el cien por ciento de etiquetado supervisado), para después entrenar con todo el conjunto mixto (las 10.000 muestras tanto de etiquetado parcial como supervisado).

Por lo tanto se observa adecuadamente lo propuesto, las etiquetas parciales aportan al proceso de optimización mucha información y son de gran ayuda. Ejemplo de ello, se ve para un diez por ciento de etiquetas supervisadas, usar sólo 1.000 observaciones con etiquetado supervisado implica una tasa de error de: ($t_{error} > 0{,}155$), mientras que usando las 10.000 observaciones disponibles, de las cuales 9.000 constan de etiquetas parciales, se obtiene una tasa de error: ($t_{error} < 0{,}135$).

4.2.5. Breves conclusiones de experimentos

Prestando atención a cada una de las gráficas representadas por las Figuras 4.3, 4.4, 4.5, 4.6, 4.7 y 4.8; se puede determinar que se han cumplido gran parte de los objetivos que se deseaban. En

todas y cada una de ellos se dan valores de tasas de error algo mayores, siempre razonables, para etiquetado parcial, lo cual es lógico y esperado. Aunque estos valores sean mayores, no quiere decir que no esté optimizando adecuadamente, ya que los valores son parejos. Siguiendo este razonamiento, el etiquetado parcial necesita de un número algo mayor de observaciones para poder llegar al mismo valor de la tasa de error que el etiquetado supervisado.

Esto significa, que no solamente el modelo probabilístico es adecuado para etiquetas parciales, sino que funciona igual de bien tanto para *Newton's method* como para *steepest descent*. Sacando a relucir, que con un número no significativamente mayor de observaciones, se pueden conseguir las mismas prestaciones que con un conjunto de datos estrictamente etiquetado con etiquetas supervisadas.

4.3. Ejemplos experimentales

Aunque en las secciones anteriores se han presentado las gráficas que arrojan razón a los objetivos propuestos, la verdad es que hubo mucho más trabajo por debajo, junto con las implementaciones. Previamente al cálculo de las tasas de error y el entrenamiento con conjuntos de datos de diferente número de observaciones, se trató de validar el correcto funcionamiento de los algoritmos implementados mediante experimentos y representaciones gráficas de los parámetros $\mathbf{W}$ iterativamente recalculados.

Aunque tal como se explica, los parámetros $\mathbf{W}$ no son los únicos a los cuales los algoritmos pueden optimizar adecuadamente, si en el caso dado lo hicieran, sería también señal de que se está logrando una implementación adecuada.

Representar gráficamente supuso en el inicio de la implementación una valiosa prueba de que se estaba yendo por el buen camino. El trato con un escenario tan pequeño, implicaba una terrible sensibilidad de los algoritmos a cambios minúsculos. Por ello, la representación es de gran ayuda para entender el efecto que variaciones en los parámetros e hiperparámetros tienen en el proceso de optimización.

La Figura 4.11 representa precisamente el proceso de optimización que siguen los parámetros recalculados. Cada círculo que compone las tres trazadas es una iteración. Se observa, que según se va acercando cada trayectoria a su destino (puntos cuadrados del mismo color), la diferencia entre iteraciones es lógicamente menor.

Steepest descent presenta en un escenario tan reducido una capacidad de variación algo menor por cada iteración que *Newton's method*. La concentración de una gran cantidad de círculos en el tramo

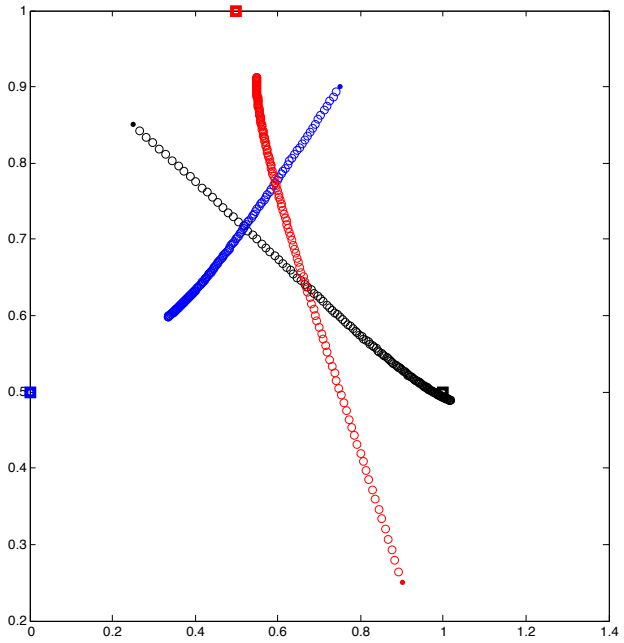

Figura 4.11: Representaciones gráficas del recálculo iterativo de los parámetros **W** para la función de coste. Steepest descent.

.

final de las trayectorias, formando una gruesa línea curva, indica que hay un número alto de iteraciones concentradas, y que seguramente el algoritmo se encuentra cerca de converger. Una solución a tantas iteraciones juntas, que significan un desaprovechamiento de la capacidad del algoritmo, sería emplear una tasa de aprendizaje variable, de forma que a iteraciones muy parecidas, se aumentara la tasa, eso sí siempre evitando que el algoritmo fuera incapaz de converger al valor deseado.

Por otra parte, en la Figura 4.12 podemos observar en general, una cantidad menor de círculos o iteraciones, lo cual indica un menor número de iteraciones necesarias para converger. Esto nos indica que se trata de *Newton's method*. En general y aunque no lo refleje el ejemplo, este algoritmo tiene una capacidad mayor de variación entre las iteraciones.

Un asunto que falta por comentar, es el de la **inicialización** de los algoritmos, que generalmente es un asunto también escabroso. En un principio, por la comprobación de los algoritmos, se utilizaron

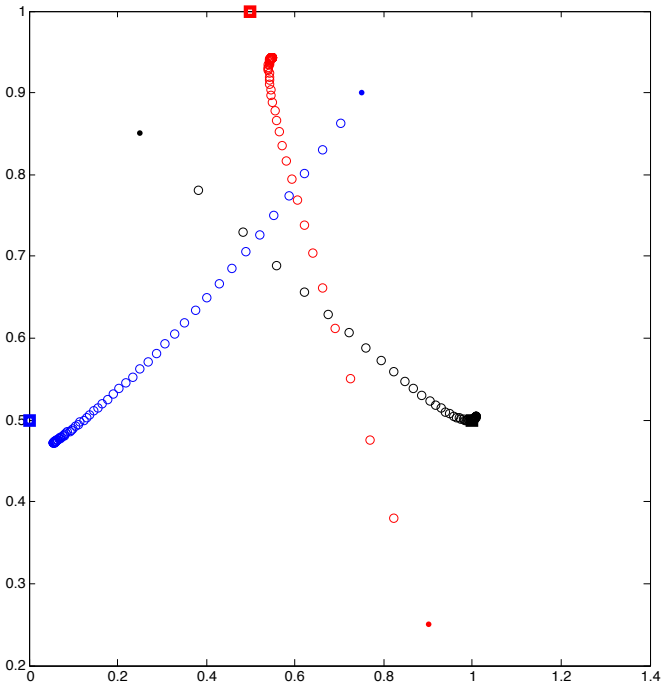

Figura 4.12: Representación gráfica del recálculo iterativo de los parámetros **W** en el proceso de optimización. Newton's method.

.

valores concretos para inicializar los parámetros a optimizar. Dichos valores, se forzaban a estar en una región difícil para el algoritmo, de manera que se pudiera comprobar que se desenvolvía bien.

En el caso de los experimentos realizados, posteriormente a las pruebas de la etapa de implementación, se usaron valores aleatorios, que se mantenían para cada geometría con los diferentes etiquetados, a fin de conservar unas condiciones similares con los conjuntos de etiquetas, y que la inicialización no fuera un hecho significativo que impidiera sacar conclusiones del trabajo realizado.

Por último, añadir que todos y cada uno de los procesos de experimentación y pruebas realizados en este capítulo, siguieron cuidadosamente la metodología del aprendizaje descrita en el capítulo anterior; respetando los métodos y etapas del entrenamiento con datos que permitan obtener resultados válidos que aseguren conclusiones acertadas.

Capítulo 5

Discusión sobre el estudio experimental

5.1. Breve discusión

Buscando una visión global de todo lo realizado, puede decirse que el trabajo se ve dividido en tres grandes etapas: establecimiento de objetivos a partir del modelo con los problemas de etiquetado, implementación de los algoritmos junto con la creación de los datos artificiales y por último experimentación con el etiquetado. Quizás el más importante de los procesos fue el de implementación; donde prueba y error fueron protagonistas. Se ha decidido no incluir el amplio código que acompaña este estudio, por su falta de importancia sobre las conclusiones que se desean obtener y por su extensión.

Experimentar nunca es tarea fácil, más cuando existen tantas razones que induzcan al error al estudio. Se ha tratado de explicar con la mayor claridad posible algunos de los aspectos que afectaron a la obtención de los resultados. Todos y cada uno de los descritos, han afectado en menor o mayor medida al proceso de realización, del que el autor cree, se ha logrado conseguir un pequeño triunfo sobre los objetivos que se plantearon al inicio del proceso.

Detrás de los experimentos ha habido dificultades de numerosos tipos, numerosas pruebas y errores que no indicaban nada bueno respecto a lo que se esperaba resultase.

La Figura 5.1 es ejemplo de ello, dado que se trata de la representación que se hizo en el entrenamiento con *Newton's method* para la segunda geometría. Se variaban los parámetros, tratando de ajustarlo a un número menor de tasa de error. Por ello, quiere decir que aunque las trayectorias puedan parecer lejanas o erróneas, en realidad podían presentar pérdidas menores que las otras. Se puede ob-

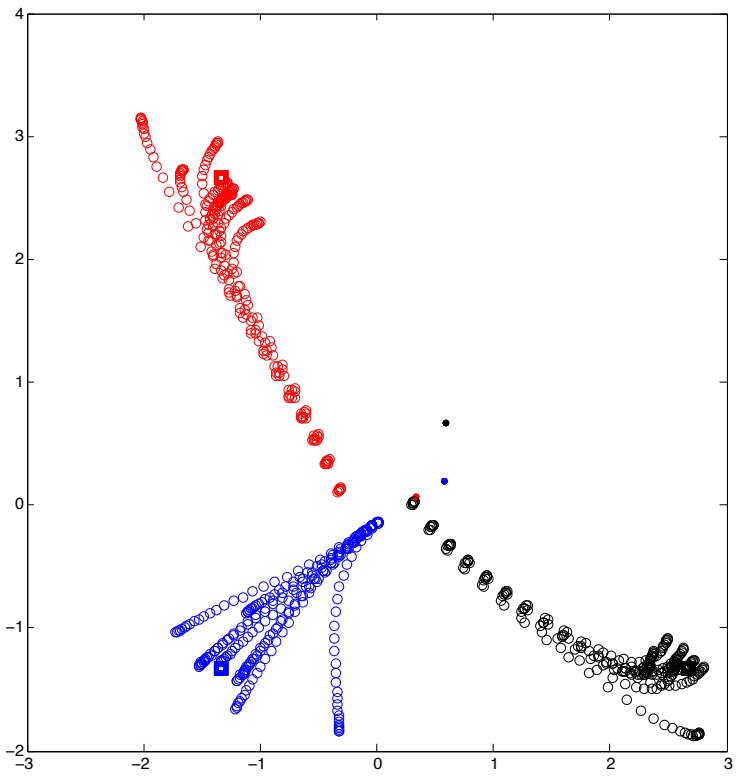

Figura 5.1: Ejemplo de numerosos intentos de optimización en los experimentos con *Newton's method*.

.

servar también, que unas convergen antes y otras después, resultado de las variaciones que se estaban realizando en cada optimización.

5.2. Sobre los datos artificiales

El uso de los datos artificiales y del escenario junto a las geometrías usadas ha resultado finalmente satisfactorio. Se normalizaron siempre los datos buscando siempre seguir *buenas prácticas* en lo referente al tratamiento con el conjunto de observaciones, proporcionando un medio a los algoritmos para trabajar de la manera más estable posible.

Capítulo 6

Conclusiones

6.1. Conclusión

La idea principal con la que iniciar las conclusiones sobre el trabajo debe ser, que el uso de etiquetas parciales permite clasificar. Se demuestra en primer lugar, que el modelo probabilístico del que procede la regresión logística, es adecuado para la optimización de funciones de coste a partir de datos con etiquetado supervisado. A partir de un proceso de minimización coherente, se concluye también, las diferencias existentes entre los algoritmos clásicos de *Newton's method* y *steepest descent*, dejando claras y subrayadas a lo largo del desarrollo del estudio, tanto sus características como diferencias significativas. Respecto al uso de dos casos diferentes dentro del ámbito del etiquetado, se ha determinado la efectividad de las etiquetas parciales. Cómo son capaces de proporcionar a los algoritmos la información necesaria para diferenciar entre las etiquetas correctas e incorrectas, y clasificar con garantías. Se han señalado las equivalencias entre etiquetado supervisado y parcial, las pérdidas en las que se incurre por el uso de uno y otro; obteniendo una demostración creíble y válida sobre su funcionamiento con tres planteamientos geométricos distintos dentro del escenario propuesto. Por último, y enriqueciendo un poco más las ideas obtenidas de la experimentación, se ha demostrado que el uso de etiquetas parciales es útil en conjuntos con etiquetado mixto, y que aunque sea sumar una serie de observaciones con etiquetado parcial a un conjunto de observaciones supervisadas, se gana en prestaciones para el clasificador, reduciendo las pérdidas.

Se considera por lo tanto, que se han logrado gran parte de los objetivos propuestos en el inicio, ganando incluso en aspectos que no se esperaba estudiar, y cómo los resultados han ofrecido credibilidad al modelo y las estructuras de datos que fueron creadas con ese

fin. Son finalmente, las etiquetas parciales las que protagonizan este Trabajo Fin de Grado, las que han permitido a lo largo de todo el desarrollo concluir decididamente en la importancia que tienen, que es posible tratar adecuadamente datos con ellas y clasificar mediante su uso.

6.2. Desarrollos futuros

Las líneas futuras, sobre las que se podría continuar el estudio, son en primer lugar las referentes al uso de datos reales en la implementación. Dichos datos podrían provenir de bases de datos para investigadores, donde es fácil encontrar conjuntos variados con etiquetados muy diferentes, y entre ellos presencia de etiquetas parciales. Por otra parte, se podrí ahondar aún más en la convergencia de algoritmos de optimización con el uso de etiquetado parcial, estudiando sus carácterísticas más allá de lo tratado en el trabajo. Además, otra línea a seguir, podría ser el uso de otros algoritmos optimizadores sobre el modelo planteado, determinando de nuevo, el comportamiento de los mismos para problemas de clasificación de este tipo.

Por último, es interesante para el autor, poder en un futuro, emplear el estudio experimental, para realizar entrenamientos de clasificación adaptados a datos del tipo audiovisual, imagen o vídeo, tal y como otros trabajos de investigación que sirvieron de motivación y guía lo hicieron.

Capítulo 7

Etapas del proyecto y presupuesto

7.1. Diagrama de Gantt

El desarrollo completo del trabajo tuvo en total una duración de *seis* meses. Aunque el trabajo se solicitó en el mes de Diciembre para su realización, no fue hasta comienzos del año 2014 cuando se procedió a trabajar en él.

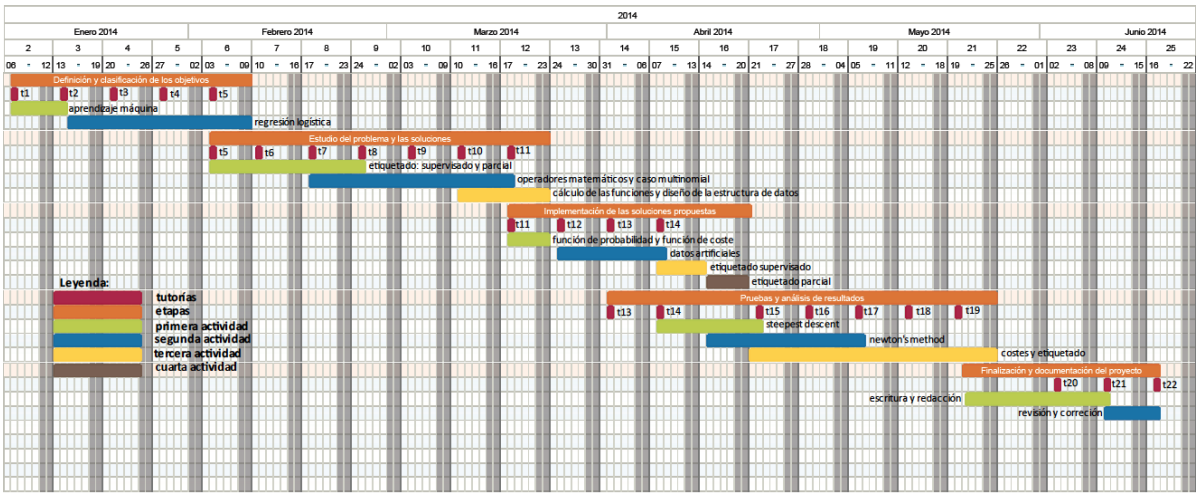

Figura 7.1: Diagrama de Gantt

Dicha duración se puede dividir en *cinco* etapas principales.

- Definición y clasificación de los objetivos.
- Estudio del problema y las soluciones.
- Implementación de las soluciones propuestas.

- Prueba y análisis de resultados.
- Finalización y documentación del proyecto.

Las etapas se planifican en forma de *Diagrama de Gantt*, en el cual cada una de ellas se encuentra desglosada en las tareas que le corresponden para la consecución de dicha etapa.

El *Diagrama de Gantt* de la Figura 7.1, es una de las herramientas más útiles en cuanto a planificación de tareas, y su uso se encuentra ampliamente extendido en numerosos programas y aplicaciones especializados en **gestión de proyectos**. Igualmente, ha sido utilizado, por ser aquel que se explicó cuidadosamente en la asignatura de *Proyectos, normativa y política de las telecomunicaciones* impartida durante el primer cuatrimestre de cuarto curso.

7.2. Presupuesto

El estudio presenta un presupuesto, partiendo de que su realización fue llevada a cabo por un graduado en Ingeniería de Tecnologías de Telecomunicación con una experiencia laboral menor a 5 años. Con la información facilitada por el *Colegio Oficial de Ingenieros de Telecomunicación* y la *Asociación Española de Ingenieros de Telecomunicación* en su reciente estudio [11], se pudo estimar satisfactoriamente el precio alcanzado por las horas trabajadas. Dicho precio fue de $11,48$ euros/hora.

Se incluye además que toda la documentación y artículos empleados tuvieron coste nulo. Esto es debido, a que al haberse desarrollado el trabajo en el ámbito de la Universidad Carlos III de Madrid, el acceso a dichos artículos era facilitado gratuitamente por la misma.

En cuanto a material de trabajo, sólo destacar que se usó la versión de estudiante de MATLAB perteneciente al año *dos mil doce* y que tiene un precio reducido frente a versiones profesionales.

7.3. Aspectos legales

Este trabajo, por tener un espíritu experimental, y querer reflejar una pequeña parte investigadora acerca del aprendizaje máquina, no se ve tan afectado por un marco regulador como podrían tener otros trabajos. En el caso de haberse utilizado bases de datos reales, se hubieran empleado bases de datos muy especializadas dedicadas a investigadores con total disposición de uso de la información proporcionada. Es un hecho que el día de mañana, la disponibilidad de datos e información sobre la sociedad facilitará obtener infinidad

Presupuesto	
Horas estimadas **(390 horas)** Precio ingeniero de tratamiento de datos:	**4477 €**
Versión MATLAB (R2012a) Precio: 69 €	69 €
Documentación y artículos. Precio: 0 €	0€
Computador de trabajo (2,4Ghz Intel Core i5) Precio:	650€
Papel y material (200 A4 + material) Precio:	10 €
Desplazamientos (16 veces_30km) Precio:	45 €
Total:	**5251 €**

Figura 7.2: Presupuesto Trabajo Fin de Grado. Primer semestre 2014

de soluciones y establecer patrones gracias al aprendizaje máquina. Sin embargo, no se verán afectados posiblemente los algoritmos y los métodos empleados en un aspecto legal, sino serán los datos los que serán cuidadosamente regulados y su privacidad.

En lo referente al estudio que aquí se presenta, se puede asegurar que no se ha infringido la privacidad de ningún conjunto de datos ni se ha dispuesto de observaciones con copyright o derechos de uso reservados. Esto se debe al uso que se ha hecho de datos generados artificialmente por el autor.

En caso de tener en cuenta aspectos reguladores, sean técnicos o legales, solamente podría verse afectada cualquier disciplina adyacente a este estudio en cuanto al origen de los datos empleados, y el fin que tuviera el uso de los mismos. Se tuvo en cuenta y no afectó a la realización de este estudio experimental en ningún ámbito.

Bibliografía

[1] UCI Machine Learning Repository. University of California Irvine.

[2] T. Cour; B. Sapp; B. Taskar. *Learning from partial labels*. Journal of Machine Learning Research 12 (2011) pp. 1501-1536, 2011.

[3] W. Waegeman; J. Verwaeren; B. Slabbnick; B. De Baets. *Supervised learning algorithms for multi-class classification problems with partial class memberships*. Fuzzy Sets and Systems 184 (2011) pp. 106-125, 2010.

[4] Yu-Yin Sun; Yin Zhang; Zhi-Hua Zhou. *Multi-label learning with weak label*. In Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10) pp. 593-598, 2010.

[5] N. Nguyen; R. Caruana. *Classification with partial labels*. In Proceedings of 14th ACM SIGKDD International conference on knowledge discovery and data mining pp.551-559, 2008.

[6] Liu; Thomas G. Dietterich. *A conditional multinomial mixture model for superset label learning*. In Advances in Neural Information Processing Systems pp. 557-565, 2012.

[7] J. Cid-Sueiro. *Proper losses for learning from partial labels*. In Advances in Neural Information Processing Systems pp. 1565-1573, 2012.

[8] David M. Diez; Christopher D. Barr; Mine Cetinkaya-Rundel. *OpenIntro Statistics*. Second Edition. 2013.

[9] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[10] R. Fletcher. *Practical methods of Optimization*. Vol. 1 *Unconstrained optimization*. Ed. John Wiley & Sons. 1980

[11] *El ingeniero de telecomunicación: Perfil socio-profesional*. Colegio Oficial de Ingenieros de Telecomunicación, 2013.