CAPÍTULO 9

CLASIFICACIÓN Y RECONOCIMIENTO DE PATRONES

María T. GARCÍA-ORDÁS[1], Rocío ALAIZ-RODRÍGUEZ[1], Enrique ALEGRE[1]

[1] Departamento de Ingeniería Eléctrica y de Sistemas y Automática, Universidad de León, León, España

En este capítulo se presentan las ideas básicas de la etapa de clasificación en un sistema de reconocimiento de patrones. Comienza el capítulo recordando los fundamentos del aprendizaje a partir de ejemplos para, posteriormente, hacer una revisión de las métricas y métodos más habituales de evaluación del rendimiento de un clasificador. El capítulo continúa mostrando el ciclo completo de diseño de un clasificador y finalmente, se describen, a modo de ilustración, tres modelos de aprendizaje correspondientes a los enfoques de clasificación supervisada, regresión y clasificación no supervisada.

9.1 Introducción

Los avances tecnológicos de las últimas décadas han hecho posible automatizar muchas tareas que previamente requerían una cantidad significativa de tiempo de trabajo manual repetitivo, mejorando la velocidad y reduciendo errores en esas tareas esencialmente mecánicas.

Actualmente, la tecnología nos permite disponer, almacenar y procesar gran número de datos. Igualmente, los investigadores han desarrollado nuevos modelos y algoritmos para trabajar con estos datos. Todo ello, junto con el creciente interés comercial e industrial por obtener información y conocimiento de los datos recopilados, ha permitido la automatización de tareas que no son meramente mecánicas sino que requieren inteligencia en mayor o menor grado.

Algunas de estas tareas, como el reconocimiento de rostros, no requiere esfuerzo alguno para un ser humano. Este proceso lo realizamos a diario, reconociendo a familiares o conocidos, independientemente de diversos factores que lo pueden dificultar, como son la postura, la iluminación

o el estilo de peinado, entre otros. Lo hacemos de forma inconsciente y nos resulta imposible explicar cómo lo llevamos a cabo. De ahí, la dificultad para escribir un programa para que se realice automáticamente.

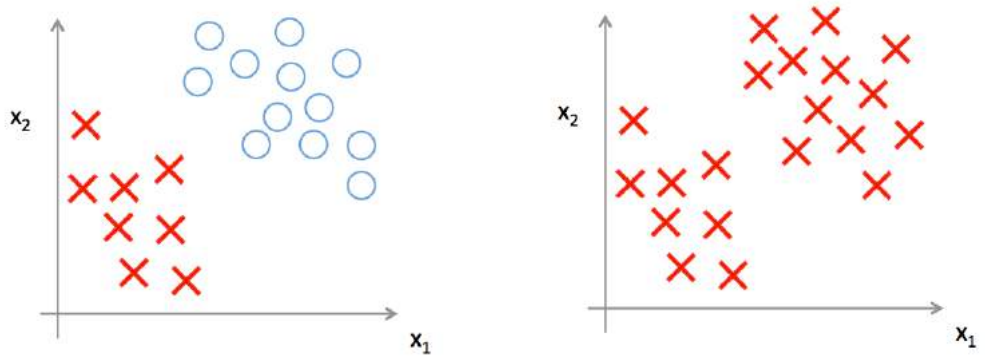
Hay otras tareas, sin embargo, que a las personas les resultan mucho más complejas e incluso difícilmente llevan a cabo con éxito. Esto se debe, en parte, al gran volumen de datos que hay que manejar y a la presencia de patrones poco evidente, no lineales muchas veces, que se deben analizar simultáneamente.

Las técnicas de aprendizaje automático y minería de datos han conseguido grandes avances en esta dirección, haciendo posible que los sistemas inteligentes sean una parte, a veces muy importante, del modelo de negocio de muchas empresas.

9.2 Fundamentos del Reconocimiento de Patrones

El reconocimiento de patrones consiste básicamente en asignar etiquetas a objetos indicando a qué clase pertenecen. Estos objetos deben ser representados por un conjunto de medidas, a las que nos referiremos como atributos o características. La tarea de reconocimiento implica necesariamente un proceso de aprendizaje a partir de un conjunto de objetos (conjunto de datos). Los fundamentos de este campo se establecieron ya en los años sesenta y setenta, con excelentes libros (Duda y Hart, 1973; Fukunaga, 1972), que dieron forma a este campo de investigación.

Hay dos grandes tipos de problemas de reconocimiento de patrones: supervisado y no supervisado. En el aprendizaje supervisado se dispone de un conjunto de datos junto con sus etiquetas, indicando a qué clase pertenecen, para cada dato o ejemplo. El objetivo que se persigue en este caso es conseguir un modelo (clasificador) que pueda etiquetar automáticamente nuevos datos que no se hayan empleado en el ajuste del modelo de clasificación. Este proceso es conocido como entrenamiento del clasificador o creación del modelo. Si las etiquetas no tienen un valor discreto, sino que toman valores continuos, hablaremos, entonces, de regresión. El reconocimiento óptico de caracteres sería un ejemplo de clasificación supervisada, mientras que la estimación de la edad de una persona a partir del rostro constituiría un problema de regresión.



(a) (b)

Figura 9.1 Ejemplo de clasificación supervisada (a) y no supervisada (b). Los datos se representan por dos características x_1 y x_2.

En las técnicas de aprendizaje no supervisado sólo se dispone de los datos sin las etiquetas de clase. Tratan de modelar los datos en sí mismos, lo que se traduce normalmente en descubrir grupos (clustering), es decir, determinar si hay ejemplos similares en el conjunto de datos y qué características hacen que estos datos sean similares dentro del grupo y diferentes del resto.

En la figura 9.1 (a) se puede ver un ejemplo de clasificación supervisada. Conocemos la clase a la que pertenecen los datos que se encuentran en el espacio de representación (clase roja y clase azul). En la figura 9.1 (b), sin embargo, no se conoce la clase a la que pertenecen los datos disponibles y buscamos técnicas que los agrupen en función de sus características.

9.2.1 Concepto de clase, características y conjunto de datos

En un problema de reconocimiento de patrones, cada muestra se identifica por un conjunto de características representada por un vector n-dimensional $\boldsymbol{x} = [x_1,...,x_n]$. Estas características pueden ser cuantitativas o cualitativas.

En el caso más extendido de aprendizaje supervisado, tendremos varias clases predefinidas, de forma que un objeto pertenece a una y sólo a una de estas clases. Cada una de las clases tiene objetos similares que son diferentes de los de las otras clases. Así, por ejemplo, en un contexto de autentificación biométrica a través de la escritura, tendríamos un problema de aprendizaje supervisado con dos clases: una determinada letra corresponde o no corresponde al usuario cuya identidad debemos verificar.

Asumiremos que hay L posibles clases, mutuamente excluyentes, etiquetadas como $u_0,...,u_{L-1}$, y organizadas en un conjunto $U_L = \{u_0,...,u_{L-1}\}$.

Dispondremos, pues, de un conjunto de muestras etiquetadas, $S = \{(\boldsymbol{x}^k, d^k), k = 1,...,N$ donde $\boldsymbol{x}^k$ es un objeto del conjunto de datos y $d^k \in U_L$ representa la etiqueta de dicha muestra. Si la clasificación fuera no supervisada no se dispondría de dichas etiquetas.

9.2.2 La clasificación

El clasificador es una función f_θ que establece la relación entre los datos de entrada $\boldsymbol{x}$ y las etiquetas d para dichos datos.

En algunos casos se trabaja con clasificadores que realizan la clasificación en dos etapas: calculan una decisión blanda y, a partir de la cual se toma la decisión dura final $\hat{d}$. Las salidas blandas vienen dadas por $y^k = f_\theta(\boldsymbol{x}^k)$ donde f_θ es una función con parámetros θ.

El algoritmo de entrenamiento ajusta los parámetros θ del clasificador de tal forma que se minimice una determinada función de error o función de coste.

9.2.3 Evaluación del clasificador

Aunque la función básica del clasificador es clara (discriminación entre dos o más clases mutuamente excluyentes), no ocurre lo mismo con la forma de evaluar las prestaciones de la clasificación realizada. Esto dificulta, por un lado, la decisión sobre cómo diseñar el clasificador (estrategia de aprendizaje, ajuste fino de los parámetros) y por otro, la comparación entre distintos clasificadores.

El resultado de la clasificación se puede resumir en una matriz de confusión con columnas y filas correspondientes a la clase pronosticada y a la real, respectivamente. Para un problema de clasificación binario (dos clases, etiquetadas como u_0, u_1) esta matriz tiene la estructura reflejada en la Figura 9.2 donde:

- TN (True Negative): Ejemplos de la clase 0 correctamente clasificados
- FN (False Negative): Ejemplos de la clase 1 clasificados incorrectamente.
- TP (True Positive): Ejemplos de la clase 1 clasificados correctamente
- FP (False Positive): Ejemplos de la clase 0 clasificados incorrectamente

		Predicción	
		u_1	u_0
Clase Real	u_1	TP	FN
	u_0	FP	TN

Figura 9.2 Matriz de confusión para un caso binario.

El número total de ejemplos positivos es $P = FN + TP$ y el número total de ejemplos negativos es $N = FP + TN$.

True Positive Rate (También llamado "hit rate", "*recall*", "Sensitivity" o "TP Rate").	TP/P	Proporción de muestras positivas que han sido correctamente clasificadas.
False Positive Rate (También llamado "False Alarm Rate" o "FP Rate").	FP/N	Proporción de muestras negativas que han sido erróneamente clasificadas como positivas.
False Negative Rate (También llamado "FN Rate").	FN/P	Proporción de muestras positivas que han sido erróneamente clasificadas como negativas (1 – TP Rate).
True Negative Rate (También llamado "Specificity" o "TN Rate").	TN/N	Proporción de muestras negativas que han sido correctamente clasificadas.
Precision (También llamado "Positive Predictive Value").	TP/(TP + FP)	Proporción de muestras clasificadas como positivas, que realmente son positivas.

F1 Score	2x Precision x Recall /(Precision + Recall)	Métrica que combina Precision y Recall.
Error Rate	(FP + FN)/(P + N)	Proporción de muestras mal clasificadas.

Figura 9.3 Algunas métricas de rendimiento para la evaluación de un clasificador.

9.2.4 Métricas de rendimiento

Con la información de la matriz de confusión se definen y emplean, diferentes métricas, en muchos casos dependiendo del campo de aplicación, para evaluar el rendimiento del clasificador. De todas ellas, la más popular ha sido la tasa de error global. Otras opciones son las tasas de error condicionadas a cada clase (conocidas en el campo del diagnóstico médico como sensibilidad y especificidad) o la utilización de tasas de riesgo. En el contexto de recuperación de información es popular la evaluación en términos de precisión y recuperación.

En las aplicaciones reales, no se consiguen clasificadores perfectos (aquel que tiene FN=0, FP=0, TN=N y TP=P). La figura 9.3 resume las métricas más empleadas para evaluarlo.

9.2.5 Técnicas de evaluación

A la hora de diseñar un clasificador disponemos de un conjunto discreto de muestras. Si utilizamos ese conjunto completo para entrenar el clasificador y posteriormente llevamos a cabo las pruebas con ese mismo conjunto de datos, tendremos una buena tasa de acierto ya que estamos probando con los mismos datos con los que hemos entrenado el clasificador. El principal problema será que habremos obtenido un modelo que probablemente no sea capaz de generalizar para datos nuevos, otros que no se hayan empleado en el entrenamiento, es decir, este modelo sólo clasificaría bien los datos que ya conoce. Este problema se conoce como sobreajuste (*overfitting*), figura 9.4.

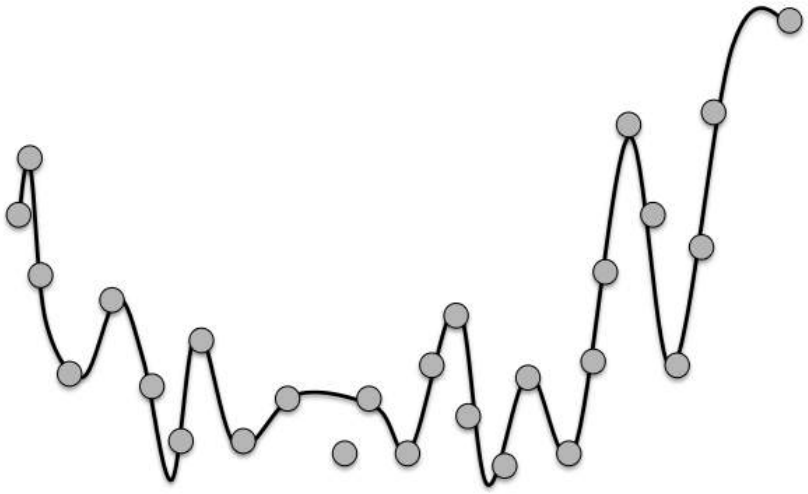


Figura 9.4 Ejemplo de *overfitting*. El modelo se ajusta a los datos pero no es capaz de generalizar para datos de entrada desconocidos.

Para que no se produzca este sobreajuste, podemos seguir diferentes estrategias: División del conjunto en dos particiones (*training* y *test*), validación cruzada con *k* particiones (*k*-fold cross validation) o Bootstrap.

9.2.5.1 División del conjunto en training y test

Cuando se divide el conjunto de datos en dos subconjuntos, uno de ellos (conjunto de training) se utiliza para entrenar, es decir, para construir el modelo, y el otro, el conjunto de prueba, para llevar a cabo la evaluación. Se suele optar por reservar 1/3 de los datos para hacer el test y los otros 2/3 para entrenar el clasificador, figura 9.5 .

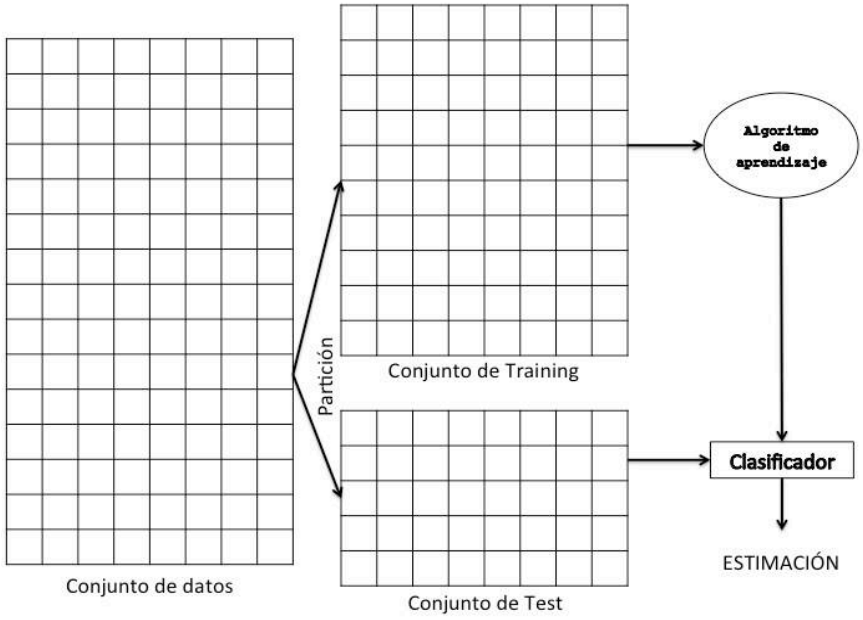


Figura 9.5 División del conjunto de datos en Training y Test.

9.2.5.2. Validación cruzada con *k* particiones

La validación cruzada con *k* particiones, llamada en inglés k-fold cross validation, es una alternativa al enfoque "División en conjuntos de *training* y *test*". El procedimiento es el siguiente: el conjunto de datos formado por N muestras, es dividido en *k* partes iguales. El valor de *k* suele ser un número pequeño, del orden de 5 o 10 y en el caso de que las partes no pudieran ser exactamente iguales, simplemente se dejaría alguna partición con un dato menos que el resto.

Una vez hechas las divisiones, se llevan a cabo *k* iteraciones. En cada una de ellas, una de las *k* partes en las que se ha dividido el conjunto de datos se usa como conjunto de test y las otras *k*-1 partes se usan como conjunto de training, figura 9.6.

Finalmente, los valores predichos para los datos de test (teniendo en cuenta las *k* iteraciones, será un total de N datos) se contrastan con las etiquetas reales de los mismos y se evalúa el modelo con las métricas seleccionadas.

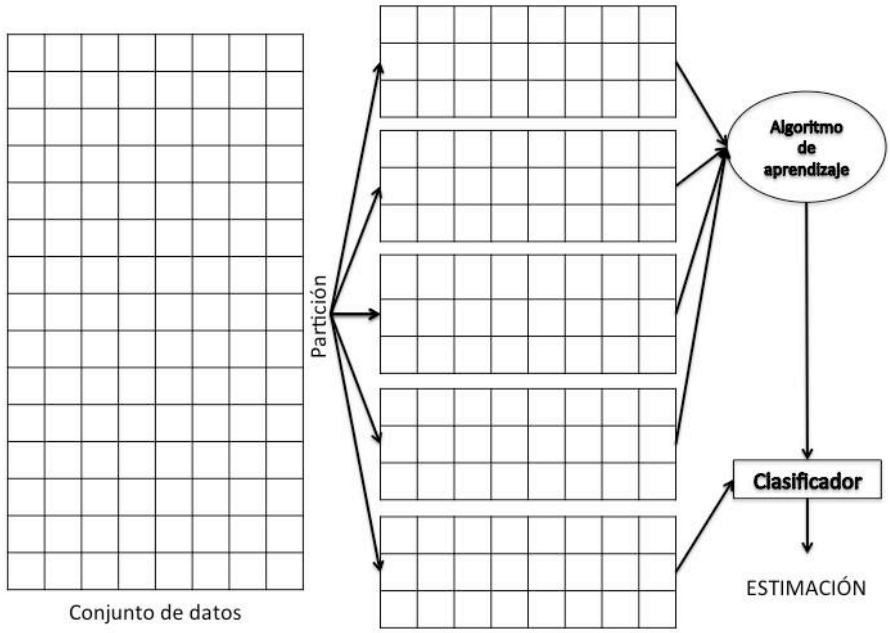


Figura 9.6. Validación cruzada k-Fold con k = 5. Ejemplo para una iteración cualquiera.

Existe un caso particular, en el que el valor del parámetro *k* coincide con el número de datos N. Se le conoce como "leave-one-out", ya que en cada una de las N iteraciones se utiliza un solo dato como conjunto de test y el resto, N-1, como conjunto de training.

9.2.5.3. Bootstrap

El conjunto de training en este caso se crea realizando N extracciones aleatorias con repetición sobre el conjunto de datos total, siendo N el número total de datos. De esta manera, tenemos un conjunto de training que tiene el mismo número de elementos que el conjunto de datos original

(algunos de ellos repetidos). El conjunto de test estará formado por todos aquellos datos que no hayan sido extraídos a la hora de crear el conjunto de training, figura 9.7.

El tamaño del conjunto de test será $0.368N$ que se obtiene simplemente calculando la probabilidad de que un ejemplo no salga en una extracción $(1-\frac{1}{N})$ y multiplicando este número por las veces que se realiza la extracción N.
Más formalmente tenemos:

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^n=\lim_{n\to\infty}\left(1+\frac{1}{-n}\right)^n=\lim_{n\to\infty}\left(1+\frac{1}{-n}\right)^{n\frac{-1}{-1}}=\lim_{n\to\infty}\left(1+\frac{1}{-n}\right)^{-n^{-1}}=e^{-1}=\frac{1}{e}=0.368 \qquad (9.1)$$

Este proceso se repite un número prefijado de veces B y después se actúa como en el caso de validación cruzada, promediando las estimaciones de las métricas obtenidas para cada conjunto de test.

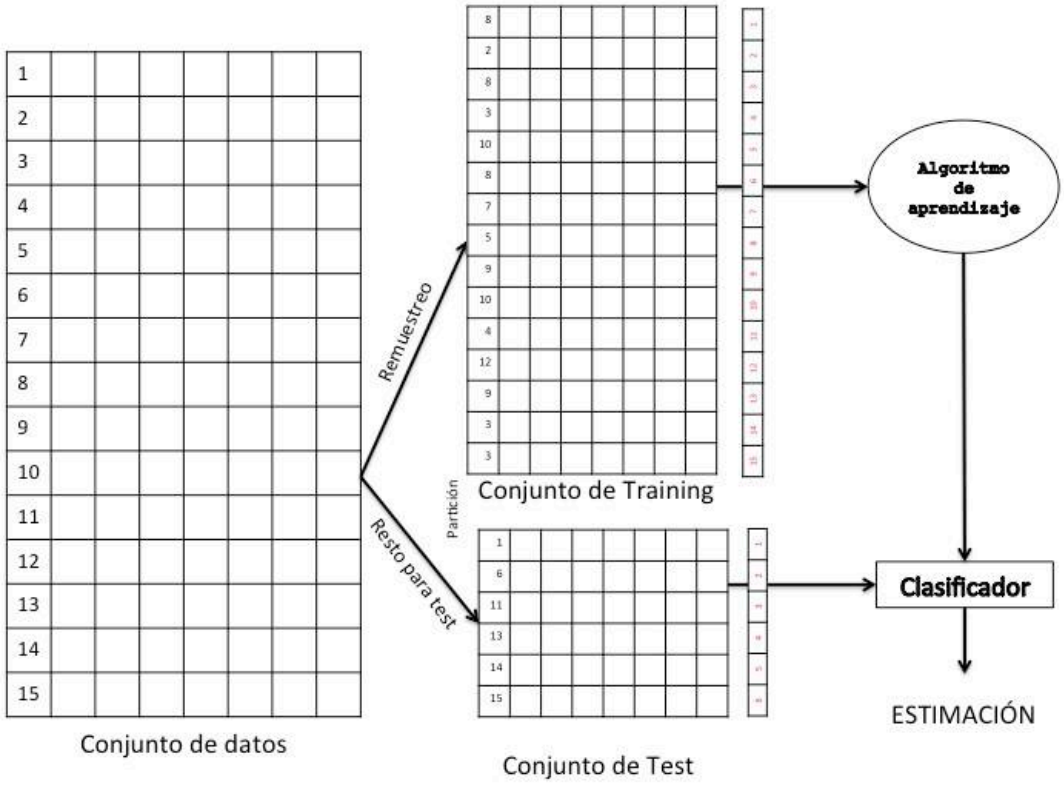


Figura 9.7 Bootstrap.

9.3 Ciclo de diseño de un clasificador

Cuando se afronta un problema de reconocimiento de patrones, hemos de considerar una serie de etapas para diseñar el clasificador.

En primer lugar, el conjunto de datos con el que se va a realizar el aprendizaje debe ser representativo de las condiciones en las que va a operar el clasificador. Si aún no disponemos de ese conjunto de datos, es necesario definir las características que puedan ser relevantes para el problema y diseñar un experimento para tomar las medidas necesarias. Si es posible, se incluirán también

características que pueden no parecer relevantes en esta etapa pero que, en combinación con otras, pudieran aportar información. Las limitaciones en la recolección de datos generalmente vienen dadas por la capacidad de financiación del proyecto, la disponibilidad de tiempo, la posibilidad de acceder a las muestras, etc.

En ocasiones, las medidas realizadas, como por ejemplo las imágenes tomadas por una cámara, deben ser procesadas para obtener, a partir de ellas, características relevantes para el sistema de reconocimiento de patrones. Esto puede incluir operaciones de filtrado, normalización, segmentación de imágenes, u obtención de descriptores de las imágenes que sean, por ejemplo, invariantes a la rotación, a la escala, etc.

Los descriptores o características (*features*) obtenidos normalmente no son todos igual de relevantes. Unos pueden serlo únicamente en relación con otros y otros pueden ser irrelevantes constituyendo sólo ruido para el sistema de reconocimiento en cuestión. De ahí que aplicar técnicas de selección de características o crear nuevas características a partir de las ya existentes, puede mejorar la calidad de la descripción.

La selección del modelo de clasificación, su entrenamiento (*training*) y la evaluación del mismo (*test*) constituye el núcleo central del desarrollo de un sistema de reconocimiento de patrones. Como se ilustra en la figura 9.8 con las líneas punteadas, el ciclo del diseño de un clasificador se puede cerrar en diferentes puntos. Podemos decidir usar el mismo modelo de clasificación con diferentes parámetros, cambiar el modelo de clasificación o bien, emplear diferentes técnicas de selección/extracción de características.

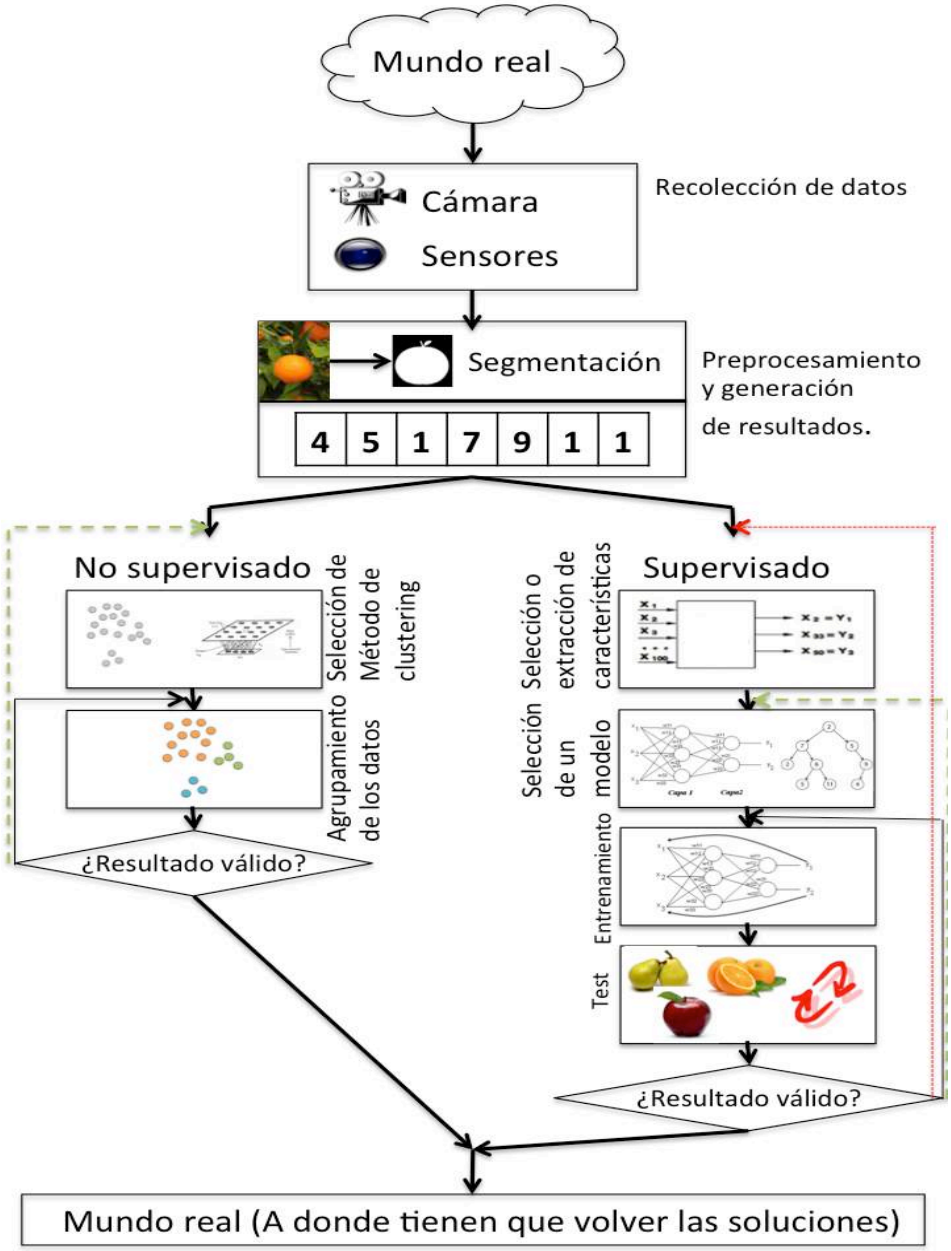


Figura 9.8 Esquema del ciclo de diseño de un clasificador.

9.4 Algoritmos de clasificación

Dado que el reconocimiento de patrones se enfrenta con problemas reales, se han desarrollado, y continúan desarrollándose, una plétora de algoritmos de aprendizaje, con sus respectivas limitaciones y fortalezas: Árboles de Decisión, Redes Neuronales, Máquinas de Vectores Soporte, Redes Bayesianas, etc. En esta sección presentamos algunos métodos básicos de clasificación, describiendo, a modo de ilustración: (a) una técnica sencilla de clasificación no supervisada (el algoritmo k-means), (b) un modelo estadístico simple de regresión lineal y (c) un modelo lineal de clasificación (regresión logística). Aspectos más avanzados, como el estudio profundo de las diferentes máquinas de clasificación, versiones avanzadas de las mismas y diferentes técnicas de optimización, quedan fuera

del alcance de este capítulo (para más información, Bishop, 2006; Duda y col. 2000; Mitchell, 1998; Pajares y de la Cruz, 2010).

9.4.1 El algoritmo K-means

K-means es un método de clasificación no supervisada muy popular para el análisis de agrupamientos o *clusters*. El término fue empleado por primera vez por MacQueen (1967) aunque el algoritmo fue propuesto por Lloyd (1957). El objetivo del algoritmo es organizar los datos en grupos (*clusters*), de tal manera que los miembros de cada grupo o *cluster* sean similares entre sí en cuanto a que sus características estén más próximas y diferentes a los de otro *cluster*.

El procedimiento consiste en dividir *N* observaciones en *k clusters*, de manera que cada observación pertenece al *cluster* más cercano. Esto da lugar a una partición del espacio de datos en celdas de Voronoi (Du y col. 2005).

Para este algoritmo, es el usuario el que decide a priori el número de grupos (*k*) en los que se quiere agrupar los datos. A continuación se explican los pasos a seguir con un ejemplo concreto (la figura 9.9 incluye un resumen del algoritmo) :

- Paso 1: Se inicializan aleatoriamente los *k* centros. En este caso, $k = 3$. Están representados en la figura 9.10 en color azul, rojo y verde.

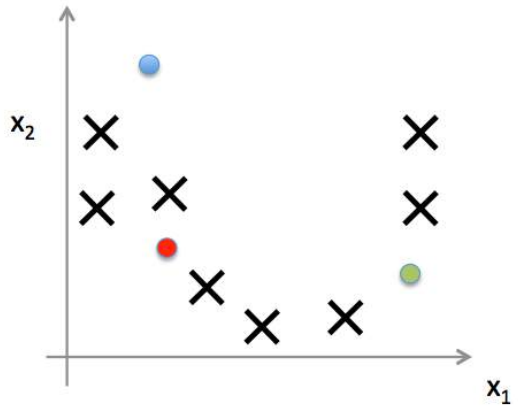


Figura 9.10 Paso 1: Inicialización aleatoria de los centros.

- Paso 2: Se asignan los datos a esos centros en función de las distancias, figura 9.11. El color de los datos representa el centro al cual ha sido asignado en función de su proximidad.

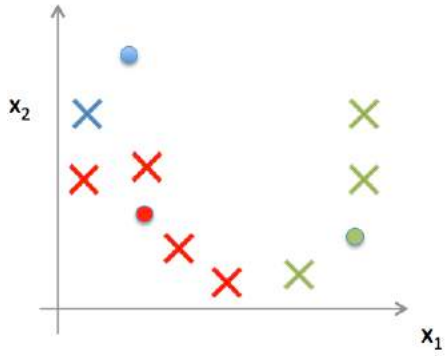


Figura 9.11 Paso 2: Asignación de puntos a centros en función de las distancias.

- Paso 3: Se reubica el centro teniendo en cuenta los puntos que se le han asignado, calculándolo como la media de todos los elementos asignados a su grupo, figura 9.12.

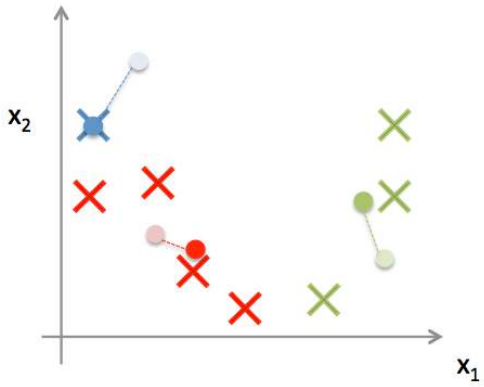


Figura 9.12 Paso 3: El centro se reubica pasando a ser el centroide de todos los puntos que tenía asignados.

- Paso 4: Se repite la asignación de puntos y la reubicación de los centros, es decir, los pasos 2 y 3 hasta que los centros no cambien de posición en dos iteraciones consecutivas.

Entrada: *k*, conjunto de N datos sin etiquetar: $\{x^1, x^2, ..., x^N\}$

1. Inicializar aleatoriamente los k centros dentro del espacio representado por los objetos.
2. Asignar cada muestra al centro más cercano.
3. Cuando todas las muestras hayan sido asignadas al centro correspondiente, recalcular las posiciones de los centros de manera que se conviertan en el centroide de las muestras que tienen asignadas dichos centros.
4. Repetir los pasos 2 y 3 hasta que los centros ya no se muevan. Esto produce una separación de los datos en *k* grupos de manera que muestras más parecidas entre sí pertenezcan al mismo grupo.

Figura 9.9 Algoritmo K-means.

Ejemplo 9.1. *Se quiere dividir un conjunto de individuos según su peso y altura en dos grupos para intentar determinar el sexo de cada uno de ellos (siendo la variable x_1 el peso y x_2 la altura del*

individuo). Utilizar KMeans con k=2 para determinar los clusters más apropiados para el siguiente conjunto de datos:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
x1	40	44	55	59	55	45	45	65	68	66	65	111	80	74	69	74	87	79	95	87	63	60	76	73	70
x2	150	148	154	162	165	148	163	160	171	179	170	195	180	172	179	181	173	168	180	192	165	158	181	177	155

Los dos centroides se han inicializado aleatoriamente a los siguientes valores:

	x1	x2
$C^0 1=$	40	150
$C^0 2=$	80	180

Solución:

Primera iteración

El primer paso consiste en calcular la distancia de cada uno de los elementos de nuestro conjunto de datos a los *clusters* para determinar a cuál de ellos pertenecen (al más cercano):

Centroide más cercano		Vector de datos x1	Vector de datos x2	Distancia euclidea a C1	Distancia euclidea a C2
1	1	**40**	**150**	0,0	50,0
1	2	44	148	4,5	48,2
1	3	55	154	15,5	36,1
1	4	59	162	22,5	27,7
1	5	55	165	21,2	29,2
1	6	45	148	5,4	47,4
1	7	45	163	13,9	38,9
2	8	65	160	26,9	25,0
2	9	68	171	35,0	15,0
2	10	66	179	38,9	14,0
2	11	65	170	32,0	18,0
2	12	111	195	84,1	34,4
2	13	**80**	**180**	50,0	0,0
2	14	74	172	40,5	10,0
2	15	69	179	41,0	11,0
2	16	74	181	46,0	6,1
2	17	87	173	52,3	9,9
2	18	79	168	43,0	12,0
2	19	95	180	62,6	15,0
2	20	87	192	63,0	13,9
2	21	63	165	27,5	22,7
1	22	60	158	21,5	29,7
2	23	76	181	47,5	4,1
2	24	73	177	42,6	7,6
2	25	70	155	30,4	26,9

A continuación, se actualizan los nuevos centroides teniendo en cuenta todos los elementos que han sido asignados a cada *cluster*. Así pues, los elementos que pertenecen al *cluster* 1 son:

	1	2	3	4	5	6	7	22
x1	**40**	44	55	59	55	45	45	60
x2	**150**	148	154	162	165	148	163	158

El nuevo centroide será la media de cada coordenada:

	x1	x2
$C^1 1=$	50,38	156

Los elementos que pertenecen al cluster 2 son:

	8	9	10	11	12	13	14	15	16	17	18	19	20	21	23	24	25
x1	65	68	66	65	111	**80**	74	69	74	87	79	95	87	63	76	73	70
x2	160	171	179	170	195	**180**	172	179	181	173	168	180	192	165	181	177	155

Por lo que el nuevo centroide $C^1 2$ será:

	x1	x2
$C^1 2=$	76,59	175,2

Como los centroides se han desplazado se continua con la siguiente iteración:

Segunda iteración

El primer paso es actualizar las distancias de cada elemento a cada centroide para determinar a que cluster pertenecen:

Centroide más cercano		Vector de datos x1	x2	Distancia euclidea a C1	Distancia euclidea a C2
1	1	**40**	**150**	12,0	44,4
1	2	44	148	10,2	42,4
1	3	55	154	5,0	30,2
1	4	59	162	10,5	22,0
1	5	55	165	10,1	23,9
1	6	45	148	9,6	41,7
1	7	45	163	8,8	33,9
1	8	65	160	15,2	19,1
2	9	68	171	23,1	9,5
2	10	66	179	27,8	11,3
2	11	65	170	20,2	12,7
2	12	111	195	72,1	39,7
2	13	**80**	**180**	38,1	5,9
2	14	74	172	28,5	4,1
2	15	69	179	29,6	8,5
2	16	74	181	34,4	6,4
2	17	87	173	40,4	10,6
2	18	79	168	31,0	7,6
2	19	95	180	50,7	19,0
2	20	87	192	51,4	19,8

1	21	63	165	15,5	17,0
1	22	60	158	9,8	23,9
2	23	76	181	35,8	5,9
2	24	73	177	30,9	4,0
1	25	70	155	19,7	21,2

A continuación, se actualizan los nuevos centroides teniendo en cuenta todos los elementos que han sido asignados a cada *cluster*. Así pues, los elementos que pertenecen al nuevo *cluster* 1 son:

	1	2	3	4	5	6	7	8	21	22	25
x1	**40**	44	55	59	55	45	45	65	63	60	70
x2	**150**	148	154	162	165	148	163	160	165	158	155

El nuevo centroide será la media de cada coordenada:

	x1	x2
$C^2 1=$	54,6	157,09

Los elementos que pertenecen al *cluster* 2 son:

	9	10	11	12	13	14	15	16	17	18	19	20	23	24
x1	68	66	65	111	80	74	69	74	87	79	95	87	76	73
x2	171	179	170	195	180	172	179	181	173	168	180	192	181	177

Por lo que el nuevo centroide $C^2 2$ será:

	x1	x2
$C^2 2=$	78,86	178,43

Como los centroides se han desplazado se continua con la tercera iteración:

Tercera Iteración

Al igual que en las iteraciones anteriores, el primer peso es actualizar las distancias y los miembros de cada cluster:

Centroide más cercano		Vector de datos x1	x2	Distancia euclidea a C1	Distancia euclidea a C2
1	1	**40**	**150**	16,3	48,1
1	2	44	148	14,0	46,3
1	3	55	154	3,1	34,1
1	4	59	162	6,6	25,8
1	5	55	165	7,9	27,4

1	6	45	148	13,2	45,5
1	7	45	163	11,3	37,2
1	8	65	160	10,8	23,1
2	9	68	171	19,3	13,2
2	10	66	179	24,7	12,9
2	11	65	170	16,6	16,2
2	12	111	195	67,9	36,2
2	13	**80**	**180**	34,2	1,9
2	14	74	172	24,4	8,1
2	15	69	179	26,2	9,9
2	16	74	181	30,8	5,5
2	17	87	173	36,1	9,8
2	18	79	168	26,7	10,4
2	19	95	180	46,4	16,2
2	20	87	192	47,6	15,8
1	21	63	165	11,5	20,8
1	22	60	158	5,4	27,8
2	23	76	181	32,1	3,8
2	24	73	177	27,1	6,0
1	25	70	155	15,5	25,0

Una vez hecho esto, se calculan los nuevos centroides. Para $C^3 1$:

	1	2	3	4	5	6	7	8	21	22	25
x1	40	44	55	59	55	45	45	65	63	60	70
x2	150	148	154	162	165	148	163	160	165	158	155

El nuevo centroide será la media de cada coordenada:

	x1	x2
$C^3 1$=	54,6	157,09

Para $C^3 2$:

	9	10	11	12	13	14	15	16	17	18	19	20	23	24
x1	68	66	65	111	**80**	74	69	74	87	79	95	87	76	73
x2	171	179	170	195	**180**	172	179	181	173	168	180	192	181	177

Por lo que el nuevo centroide $C^3 2$ será:

	x1	x2
$C^3 2$=	78,86	178,43

Como los centroides de la iteración 2 y 3 son iguales, el algoritmo se detiene y ya tenemos los clusters creados correctamente dividiendo el conjunto de muestras en función del sexo del individuo:

$C^2 1$=	54,64	157,09	$C^3 1$=	54,64	157,09
$C^2 2$=	78,86	178,43	$C^3 2$=	78,86	178,43

9.4.2 Regresión Lineal

Si disponemos de un conjunto de datos $S = \{(\boldsymbol{x}^k, y^k), k = 1,...,N\}$, donde tratamos de predecir un valor numérico continuo, $y^k \in R$, podemos emplear un modelo de regresión lineal. El objetivo que se persigue es encontrar la función h que mejor establezca la relación entre las variables independientes (descriptores) y la variable dependiente (etiqueta numérica). En el caso de la regresión lineal, la función h que buscamos se expresa como una combinación lineal de los descriptores con pesos θ y será de la forma:

$$h_\theta(\boldsymbol{x}) = \theta_0 + \theta_1 x_1 + ... + \theta_m x_m \tag{9.2}$$

El ajuste de los parámetros θ_i ha de hacerse con el conjunto de datos disponibles S, de manera que $h_\theta(\boldsymbol{x}^k)$ sea lo más próximo a y^k. Para calcular el error existente entre los datos reales y los datos estimados por nuestra función, definimos una función de coste, el error cuadrático medio:

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}(h_\theta(\boldsymbol{x}) - y)^2 \tag{9.3}$$

Donde *N* se corresponde con el número de datos.

Debemos escoger los parámetros θ que minimizan la función de coste $J(\theta)$. Para ello, podemos comenzar inicializando aleatoriamente los parámetros y después, emplear un algoritmo de búsqueda, como puede ser el algoritmo de descenso de gradiente, que sucesivamente cambie el valor de θ de forma que el valor $J(\theta)$ sea cada vez menor y converja al valor de los parámetros que minimiza la función de coste.

El algoritmo de descenso por gradiente comienza un con valor arbitrario para los parámetros, que se actualizan de la forma

$$\theta_j = \theta_j - \alpha \frac{d}{d\theta_j} J(\theta) \tag{9.4}$$

donde α es la tasa de aprendizaje, que indica cuánto cambian los parámetros y el signo menos indica que nos movemos en la dirección opuesta al gradiente (ya que la dirección del gradiente es la de máximo incremento de la función). Si elegimos un valor de α muy pequeño, la función tardará mucho en converger. En cambio, si escogemos un valor muy grande puede oscilar y no llegar a converger en el mínimo global esperado.

Ilustraremos este proceso, con un problema de regresión con dos parámetros donde la función de regresión lineal toma la forma $h_\theta(x) = \theta_0 + \theta_1 x$.

La figura 9.16 muestra un ejemplo de la función de coste $J(\theta)$ donde podemos observar que es una función convexa con un mínimo global.

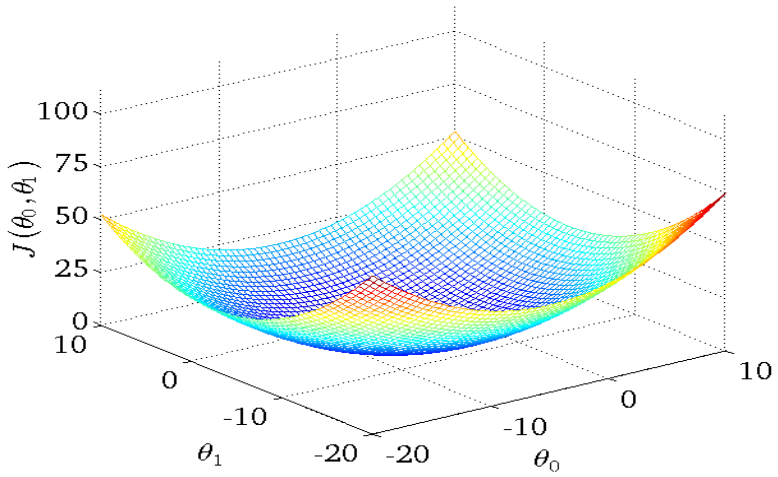


Figura 9.16 Representación de la función de coste para un problema de regresión lineal con dos parámetros.

Particularizando la ecuación 9.4 para un caso con dos parámetros, nos queda

$$\frac{d}{d\theta_j}J(\theta_0,\theta_1)=\frac{d}{d\theta_j}\frac{1}{2N}\sum_{i=1}^{N}(h_\theta(x)-y)^2=\frac{d}{d\theta_j}\frac{1}{2N}\sum_{i=1}^{N}(\theta_0+\theta_1x_1-y)^2 \tag{9.5}$$

Si derivamos ahora con relación a ambos parámetros, tenemos:
Para θ_0 :

$$\frac{d}{d\theta_0}J(\theta_0,\theta_1)=\frac{1}{N}\sum_{i=1}^{N}(h_\theta(x)-y) \tag{9.6}$$

Para θ_1 :

$$\frac{d}{d\theta_0}J(\theta_0,\theta_1)=\frac{1}{N}\sum_{i=1}^{N}(h_\theta(x)-y)x \tag{9.7}$$

Por tanto, el algoritmo de descenso de gradiente aplicado al problema de regresión lineal, se resume como:

$$\theta_0=\theta_0-\alpha\frac{1}{N}\sum_{i=1}^{N}(h_\theta(x)-y) \tag{9.8}$$
$$\theta_j=\theta_j-\alpha\frac{1}{N}\sum_{i=1}^{N}(h_\theta(x)-y)x$$

La figura 9.17 muestra cómo sería la actualización del parámetro θ_1 con el algoritmo de descenso de gradiente.

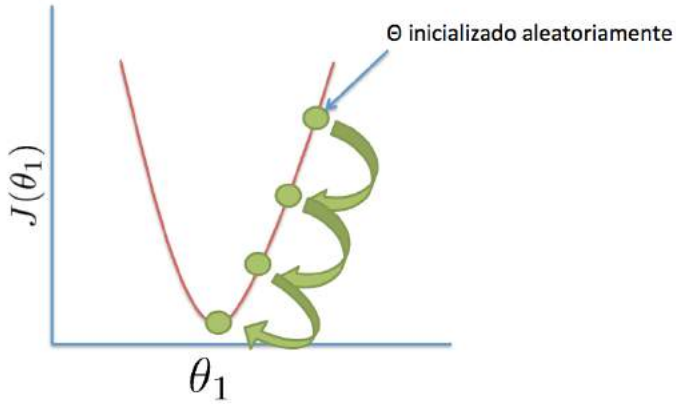


Figura 9.17 Descenso por gradiente.

9.4.3 Regresión Logística

Cualquier técnica de regresión puede adaptarse fácilmente para abordar problemas de clasificación supervisada (con un número discreto de clases). El truco consiste en llevar a cabo la regresión para cada clase asignando el valor uno para los ejemplos que pertenecen a la clase de interés y cero para el resto.

En este tipo de clasificación queremos que nuestra etiqueta de salida sea 0 o 1, por tanto, podemos acotar la función de regresión lineal entre esos valores. Para ello, vamos a emplear la función sigmoidal, que viene dada por:

$$g(z) = \frac{1}{1+e^{-z}} \tag{9.9}$$

y se representa de la forma mostrada en la figura 9.19:

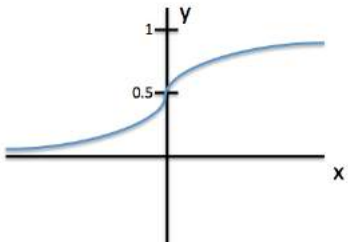


Figura 9.19 Ejemplo de funcionamiento del ajuste de los parámetros.

De esta manera, nuestra hipótesis quedaría acotada:

$$h_\theta = g(\theta_0 + \theta_1 x_1 + ... + \theta_m x_m) \tag{9.10}$$

Podemos decir que $h_\theta(\boldsymbol{x}^k)$ representa la probabilidad de que la salida sea uno, dado el dato $\boldsymbol{x}^k$, y parametrizado con θ.

No podemos utilizar la misma función de coste que hemos estado utilizando para los problemas de regresión, porque al haber acotado nuestra función, tendríamos una representación similar a la de la gráfica mostrada en la figura 9.20 izquierda.

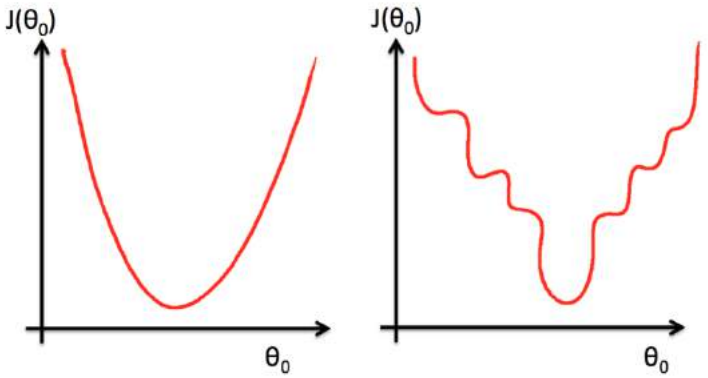


Figura 9.20 Diferencia entre la función coste de regresión lineal (izquierda) y regresión logística (derecha).

Si utilizáramos la misma función coste para ambos métodos, en el caso de la regresión logística, es probable que no consiguiéramos llegar a encontrar el mínimo global, por la existencia de muchos mínimos locales.

La función coste que emplearemos para regresión logística es la siguiente:

$$J(\theta) = -\frac{1}{N}[\sum_{i=1}^{N} d \log h_\theta(\boldsymbol{x}) + (1-d)\log(1-h_\theta(\boldsymbol{x}))] \tag{9.11}$$

La figura 9.21 muestra la contribución a la función de coste para: un ejemplo $\boldsymbol{x}^i$ que pertenece a la clase 1 (izquierda) y un ejemplo que pertenece a la clase 0 (derecha). Cuando la etiqueta es $d = 1$, el segundo sumando de la ecuación (9.11) queda anulado y si nuestra salida de la función se aproxima a uno (que es la etiqueta real), el error tiende a cero. En cambio, si se aproxima a cero, el error tiende a infinito.

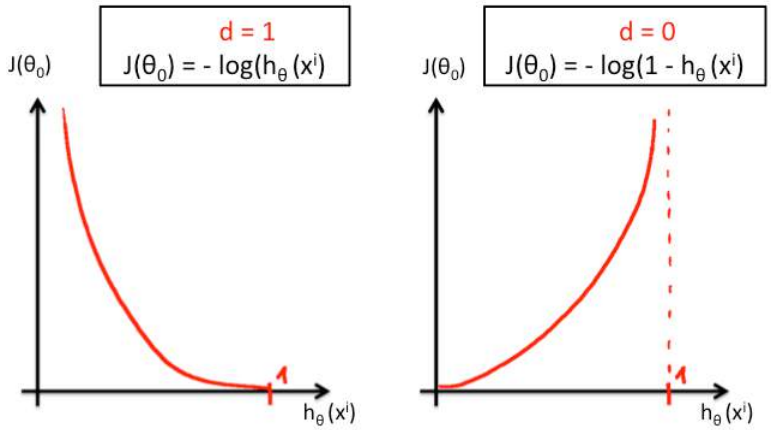


Figura 9.21 Función coste de la regresión logística.

Una vez ajustado el modelo de clasificación, éste nos devolverá, para los datos de *test*, un valor en el intervalo de cero a uno que indicará la probabilidad de pertenencia a la clase uno. Se aplicará, finalmente, un umbral para tomar la decisión final de pertenencia a la clase.

9.5 Bibliografía

Duda, R. O.; Hart, P.E. (1973) Pattern classification and scene analysis. John Wiley & Sons.

Fukunaga, K. (1972) Introduction to Statistical Pattern Recognition. Academic Press, Inc.

Bishop, C.M. (2006) Pattern Recognition and Machine Learning. Springer.

Duda, R. O.; Hart, P. E.; Stork, D. G. (2000) Pattern Classification, John Wiley & Sons.

Mitchell, T. (1998) Machine Learning, MIT Press.

Pajares, G. y de la Cruz, J.M. (Eds.) (2010). Aprendizaje Automático: un enfoque práctico. RA-MA, Madrid.

Du, Qiang; Wang, Desheng (2005), The Optimal Centroidal Voronoi Tessellations and the Gersho's Conjecture in the Three-Dimensional Space, *Computers and Mathematics with Applications (49)*: 1355–1373

Fix, E.; J.L. Hodges (1989) An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges. *International Statistical Review / Revue Internationale de Statistique 57* (3): 233–238

Viñuela Isasi, P.; Galván León, I.M. (2003). Redes de neuronas artificiales. Un enfoque práctico. Prentice Hall.

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory 28* 129–137.

MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1*. University of California Press. pp. 281–297.

Ng, A., Curso de "Machine Learning". Coursera. <https://www.coursera.org/learn/machine-learning/> [Consulta: abril 2015]

Yan. X; Gang Su. X. (2009). Linear Regression Analysis. *Theory and Computing.* World Scientific Pub Co.