Taller de Introducción al Aprendizaje de Máquina: Regresión Lineal

Julián D. Arias Londoño, Juan Felipe Pérez
Departamento de Ingeniería de Sistemas
Universidad de Antioquia, Medellín, Colombia
jdarias@udea.edu.co

February 17, 2017

1 Marco teórico

Como hemos visto hasta ahora el aprendizaje de máquina se interesa por el desarrollo de modelos que permiten, a partir de información del pasado, tomar decisiones futuras. Los modelos ajustados o entrenados a partir del aprendizaje nos permiten hacer predicciones sobre una variable a partir de otro conjunto de variables. Suponga que Ud es contratado/da para diseñar el sistema de información de la unidad de oncología de un hospital. El sistema permite, entre muchas otras cosas, crear historias clínicas digitales de los pacientes y e ingresar información de un conjunto de variables del paciente, como edad, peso, estatura, si se es fumador o no, si está en riesgo genético, y todas las demás que los médicos especialistas hayan definido durante el análisis de requisitos. Adicionalmente el sistema de información debe permitir almacenar y visualizar material radiológico (por ejemplo una mamografía). Asumamos que su sistema incluye un componente que permite procesar la imagen y establecer si la imagen presenta o no un tumor y permite calcular las dimensiones del mismo. Hasta este punto el sistema de información cumple con todas las expectativas de la unidad hospitalaria que contrató el servicio. Sin embargo, después de uno o dos años de puesto en funcionamiento el sistema, los médicos han podido recoger información adicional, por ejemplo cuales de los pacientes a los que se les detecto un tumor, les fue determinado como maligno o benigno e incluso si después de definir un tratamiento (ej: radioterapia o quimioterapia), el paciente mejoró, o por el contrario, tuvo un deceso y su correspondiente fecha de muerte.

Pasados dos años es posible que el sistema tenga en su base de datos con históricas clínicas de 1000 0 1500 pacientes que han pasado por la unidad. Teniendo en cuenta que para cada uno se cuenta con un conjunto de variables definidas inicialmente, se podría pensar en definir un modelo y un criterio de aprendizaje de tal manera que el sistema pueda determinar, únicamente con

las variables definidas inicialmente y las obtenidas de la radiografía, si es más probable que el tumor sea maligno o benigno, e incluso en el caso en que sea maligno intentar predecir el tiempo que le queda de vida al paciente.

Un sistema de información que además pudiese proveer dicha información sería de gran utilidad para determinar políticas de priorización de los pacientes en los sistemas de salud.

Desde el punto de vista del aprendizaje, el problema que se ha planteado tiene dos componentes. Predecir si el tumor es benigno o maligno es un problema en el que la variable a predecir es categórica, por lo tanto corresponde a un problema de clasificación. Por otro lado, intentar predecir el tiempo que le queda de vida al paciente es una variable que puede tomar valores reales y por lo tanto es un problema de regresión. Como en ambos casos se cuenta con etiquetas (los valores de maligno o benigno y de tiempo de vida de los 1500 pacientes estudiados en los últimos 2 años), entonces ambos problemas se pueden catalogar como de aprendizaje supervisado.

Es claro que existe un grado de incertidumbre en la toma de ambas decisiones. Si representamos el conjunto de variables del paciente i como $\mathbf{x}_i = \{x_1, x_2, ..., x_d\}$ y la etiqueta como y_i (es decir una muestra del conjunto de entrenamiento $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$), y teniendo en cuenta que existe un grado de incertidumbre en la toma de la decisión, cualquiera de las dos predicciones mencionadas pasará por establecer la probabilidad del valor de y_i, dado el vector $\mathbf{x}_i$, es decir $p(y_i|\mathbf{x}_i)$. Hasta el momento hemos visto que la forma más simple de resolver un problema de regresión es a través de un modelo de regresión múltiple, con un polinomio de grado p. Para ajustar un modelo de regresión de ese tipo podemos tomar dos caminos. Plantear el problema como una ecuación lineal y resolver el problema inverso, o minimizar la función de error cuadrático medio a través de algún algoritmo de optimización, por ejemplo un algoritmo de gradiente descendente. En el primer caso el problema consiste en plantear el sistema lineal dado por $\mathbf{y} = \mathbf{X}\mathbf{w}$, donde $\mathbf{y}$ es un vector columna que contiene todas las etiquetas del conjunto de entrenamiento, $\mathbf{w}$ es el vector que contiene todos los coeficientes del polinomio y $\mathbf{X}$ es una matriz de la forma:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1M} \\ x_{21} & x_{22} & \cdots & x_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{M1} & \cdots & x_{NM} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{bmatrix}$$

conocida como la Matriz extendida porque una de sus columnas corresponde a un vector de 1's el cual multiplicará al coeficiente w_0 o *bias* dentro de la función polinomial. Cada fila de la matriz contiene una muestra y cada columna es una de las variables o características que se están usando para predecir el valor de y. El problema consiste en encontrar el vector $\mathbf{w}$ a partir de $\mathbf{X}$ y $\mathbf{y}$; de manera simple podemos establecer que:

$$\mathbf{w} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{1}$$

Esta aproximación aunque simple tiene problemas debido a la necesidad de invertir la matriz $\left(\mathbf{X}^T\mathbf{X}\right)$, cuando se presentan niveles de ruido altos en los datos, es posible que la matriz presente problemas de singularidad, lo cual dificulta el problema.
Por otro lado, si decidimos solucionar el problema a través de la minimización de la función de error cuadrático medio, es necesario entonces determinar la regla de actualización de los parámetros. Usando un algoritmo básico por gradiente descendente tenemos que la regla de actualización estará dada por:

$$w_j(t+1) = w_j(t) - \eta\frac{\partial E(\mathbf{w})}{\partial w_j}$$

donde η es la tasa de aprendizaje. La derivada del error puede ser fácilmente calculada como:

$$\begin{aligned}\frac{\partial E(\mathbf{w})}{\partial w_j} &= \frac{1}{2N}\sum_{i=1}^{N}\frac{\partial}{\partial w_j}\left(f(\mathbf{x}_i,\mathbf{w}) - y_i\right)^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(f(v_i,\mathbf{w}) - y_i\right)\frac{\partial}{\partial w_j}f(\mathbf{x}_i,\mathbf{w}) \\ &= \frac{1}{N}\sum_{i=1}^{N}\left(f(\mathbf{x}_i,\mathbf{w}) - y_i\right)x_{ij}\end{aligned}$$

Con lo cual queda determinada por completo la regla de actualización de los parámetros dentro del algoritmo.

De una forma similar podemos pensar el problema de clasificación como el problema de encontrar un función polinomial f que pueda dividir los conjuntos de datos de las diferentes clases (ver Fig. 1).
Si se pudiese encontrar la función dada por la línea negra en la figura 1, la cual sería de la forma $f(\mathbf{x}) = w_1x_1 + w_2x_2 + w_0$, se podría utilizar como función de clasificación, tal que cualquier muestra evaluada en la función obtendrá como resultado un valor positivo si se ubica a un lado y un valor negativo si se ubica al otro (los valores ubicados justo en la función obtendrían un valor de 0). Teniendo en cuenta que los valores de las etiquetas para el problema de clasificación (las variables a predecir y_i), solo pueden tomar dos valores $0, 1$, entonces una forma simple de usar la función f como clasificador sería asignar las muestras a la clase 1 cuando al ser evaluadas en la función f obtengan una valor positivo y asignar 0 cuando suceda lo contrario. Eso implicaría definir una función de clasificación completa dada por:

$$g(\mathbf{x}) = \{1 \text{ if } f(\mathbf{x}) \geq 0$$

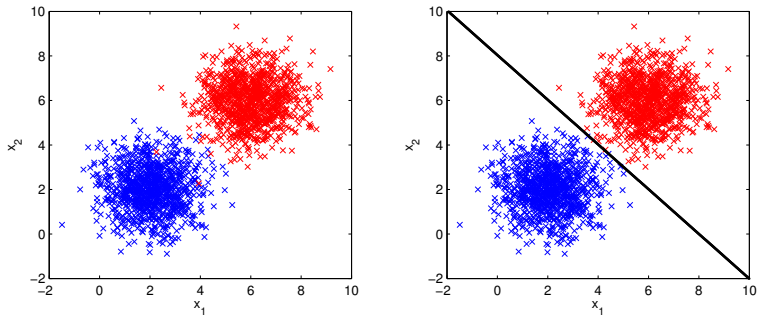

Figure 1: Función de clasificación lineal

El problema con la función g es que es discontinua y no puede ser usada como criterio de optimización para ningún algoritmo basado en gradiente. Un alternativa es utilizar alguna función que tenga una comportamiento asintótico similar pero que sea continua y derivable, por lo que podemos usar la función sigmoide dada por:

$$g(u) = \frac{\exp(u)}{1 + \exp(u)}$$

El método que utiliza la función sigmoidal para encontrar una frontera de separación polinomial se conoce como Regresión Logística. La función objetivo (criterio de entrenamiento) está dada por:

$$J(\mathbf{w}) = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} -\mathbf{y_i} \log(\mathbf{g}(\mathbf{f}(\mathbf{x}))) - (\mathbf{1} - \mathbf{y_i}) \log(\mathbf{1} - \mathbf{g}(\mathbf{f}(\mathbf{x})))$$

Si se analiza con detenimiento, la función criterio J minimiza el error de clasificación. Es necesario tener en cuenta que dicha función está definida para y_i que toman valores $0, 1$. La ventaja del método de regresión logística es que la función para la actualización de los pesos $\mathbf{w}$ es muy similar a la función para la regresión lineal. La derivada de J está dada por:

$$\frac{\partial J(\mathbf{w})}{\partial w_j} = \frac{1}{N} \sum_{i=1}^{N} \left(g(f(\mathbf{x}_i, \mathbf{w})) - y_i\right) x_{ij}$$

La única diferencia es la inclusión de la función sigmoidal g.
De manera alternativa podemos usar un modelo de clasificación basado en funciones discriminates Gausianas, en cuyo caso lo que debemos hacer es estimar una función de la forma:

$$\mathcal{N}(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} \left|\Sigma\right|^{1/2}} exp\left[-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1} (\mathbf{x} - \mu)\right]$$

para cada una de las clases.

2 Ejercicios

1. Adjunto a este taller encontrará los archivos: *Main.m, normalizar.m, potenciaPolinomio.m, regresionLogistica.m, regresionMultiple.m y sigmoide.m.* El archivo *Main.m* es el script principal, desde el cual se ejecutan todas las instrucciones para realizar el taller. Una vez corra el Script principal se le solicitará ingresar el tipo de problema que desea resolver (para regresión múltiple ingrese 1 ó ingrese 2 para regresión logística). Analice con cuidado el script y comprenda como esta construido. Cada opción genera un conjunto de muestras las cuales son divididas en un subconjunto para entrenar el modelo y un subconjunto para evaluarlo.

2. Primero describa como esta construida la base de datos utilizada en el problema de regresión; para esto puede graficar los datos, o analizar la función que se usa para generarlos. Explique cual es la función polinomial con la cual están construidos los datos en matlab. Diga cuantas muestras de entrenamiento y de validación se usan para resolver el problema, también escriba cuantas características tienen las muestras.

   R/:

   Para poder resolver el problema de regresión debe completar el archivo *regresionMultiple.m* el cual tiene indicado las lineas donde se debe implementar la ecuación de actualización de los pesos del polinomio de acuerdo con el algoritmo de gradiente descendente:

$$w_j(t+1) = w_j(t) - \eta \frac{\partial E(\mathbf{w})}{\partial w_j}$$

   Una vez haya implementado la regla de actualización, ejecute varias veces el proceso de entrenamiento y evaluación cambiando los parámetros *grado y taza de aprendizaje* y complete la tabla con los valores del error cuadrático medio (ECM) obtenidos:

Taza de Aprendizaje	Grado del Polinomio	Error Cuadrático Medio (ECM)
0,1	1	
	2	
	3	
	4	
	5	
0,001	1	
	2	
	3	
	4	
	5	
0,00001	1	
	2	
	3	
	4	
	5	

Table 1: Tabla ejercicio 2

Responda las siguientes preguntas:

(a) En muchos casos el resultado del experimento toma el valor NaN, ¿Cuál cree que es la causa del problema?

R/:

(b) ¿Cual es el número de coeficientes W que se obtienen del algoritmo al ingresar un polinomio de grado 4, y por qué?

R/:

(c) ¿Cuál es el orden en el cual le fueron ingresadas las variables al algoritmo de entrenamiento cuando el grado del polinomio es 4?
(A) $X1X2X1^2X2^2X1^3X2^3X1^4X2^4$
(B) $X1X1^2X1^3X1^4X2X2^2X2^3X2^4$
(C) $X1^2X1^3X1^4X1X2^2X2^3X2^4X2$
(D) $X1^4X1^3X1^2X1X2^4X2^3X2^2X2$

3. Repita el experimento anterior descomentando las lineas de normalización de los datos, y complete la tabla:

Taza de Aprendizaje	**Grado del Polinomio**	**Error Cuadrático Medio (ECM)**
0,1	1	
	2	
	3	
	4	
	5	
0,001	1	
	2	
	3	
	4	
	5	
0,00001	1	
	2	
	3	
	4	
	5	

Table 2: Tabla ejercicio 3

Responda las siguientes preguntas:

(a) ¿Qué proceso hace la normalización sobre los datos? Consulte porqué es necesaria la normalización en el modelo de regresión logística y cuáles son los tipos de normalización más comunes. ¿Cuál de ellos se aplicó en el laboratorio?
R/:

(b) Compare el resultado con el punto anterior donde no se usó la normalización, ¿Cuál es el grado que mejor representa el conjunto de muestras y por qué?

R/:

(c) Escriba la función polinomial que el modelo entrenado genera para predecir los datos.

R/:

(d) *Realice una gráfica del error cuadrático medio vs iteraciones para la mejor configuración de la tabla anterior. Interprete el resultado obtenido.

4. Describa como está construida la base de datos para el problema de clasificación, es decir, ¿qué forma tienen los datos cuando se grafican en el espacio de características? Mencione el número de muestras usadas para entrenar el sistema y para evaluarlo. ¿Cuántas características tienen las muestras?

   R/:

   Para poder resolver el problema de clasificación, debe completar el archivo *regresionLogistica.m* en las lineas indicadas para implementar la regla de actualización de los pesos para el caso de la regresión logística. Una vez haya implementado la regla de actualización, ejecute varias veces el proceso de entrenamiento y evaluación y complete la siguiente tabla:

Taza de Aprendizaje	Grado del Polinomio	Eficiencia
0,1	1	
	2	
	3	**0.64**
	4	
	5	
0,001	1	
	2	
	3	
	4	
	5	
0,00001	1	
	2	
	3	
	4	
	5	

Table 3: Tabla ejercicio 4

5. Repita el experimento anterior descomentando las lineas de normalización de los datos, y complete la siguiente tabla:

Taza de Aprendizaje	Grado del Polinomio	Eficiencia
0,1	1	
	2	
	3	
	4	
	5	
0,001	1	
	2	
	3	
	4	
	5	
0,00001	1	
	2	
	3	
	4	
	5	

Table 4: Tabla ejercicio 5

(a) ¿La normalización tiene un efecto similar al problema de regresión? Explique.
R/:

(b) Describa cual es la función polinomial de la mejor frontera encontrada.
R/:

(c) *Realice una gráfica de la frontera de separación generada por el modelo, muestre la frontera alcanzada durante cada iteración del algoritmo de entrenamiento.

(d) Realice una comparación entre el modelo de regresión logística y el modelo clasificación basado en funciones discriminates Gausianas. Para esto use la función de Matlab `classify`. Esta función recibe como parámetros básicos las muestras a clasificar, las muestras de entrenamiento y la clase a la que pertenece cada muestra de entrenamiento. También es posible modificar si la frontera se desea lineal o cuadrática (use varias). La función retorna un vector con la clase a la que pertenece cada muestra a clasificar. Estime el error de clasificación y compárelo con el mejor resultado obtenido con el modelo de regresión logística, ¿cuál obtiene mejor resultado
R/:

Nota: Los puntos con "*" no se deben entregar, deben realizarlos para mejorar la comprensión de las técnicas analizadas.