Regresión Lineal Múltiple y Logística

García Prado, Sergio
sergio@garciparedes.me

Fernández Angulo, Óscar
oscar.fernandez.angulo@alumnos.uva.es

10 de mayo de 2017

Resumen

En este documento se realiza una descripción acerca de la Regresión Lineal Múltiple *y la* Regresión Logística *desde el punto de vista del ámbito de la* Inteligencia Artificial*. Además, se han realizado implementaciones de dichas técnicas en el lenguaje* Octave*(*MatLab*) para después utilizarlas en la realización de varios experimentos de comparación y cotas de tasa de error con los conjuntos de datos* Housing *[UCIa] y* Wine*[UCIb]*

1. Introducción

1.1. Regresión Lineal Múltiple

La *regresión lineal* es un modelo matemático usado para aproximar la relación de dependencia entre una variable dependiente Y, las variables independientes X_i y un término aleatorio ϵ. Este modelo se describe en la ecuación (1) donde las variables β_i representan los pesos que ajustan los valores de las variables X_i para aproximar su suma total al valor deseado ($= Y$).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon \tag{1}$$

Por tanto, en este modelo de aprendizaje, se trata de aprender el valor de dichos pesos a partir de un conjunto de datos de entrenamiento. La estrategia que se ha utilizado en este caso ha sido el ajuste por mínimos cuadrados, el cual trata de minimizar la diferencia global ajustando dichos pesos. Para dicha tarea se ha utilizado la implementación en *Octave* que se muestra en la figura 1.

```
function w = regresion_lineal_K(x, y)
  A = zeros(size(x)(2), size(x)(2));
  B = zeros(size(x)(2), 1);
  for i=1:size(x)(2)
    for j=1:size(x)(2)
      A(i,j)=sum(x(:,i).*x(:,j));
    end
    B(i,1)=sum(y.*x(:,i));
  end
  w=inv(A)*B;
end
```

Figura 1: *Octave: /src/regresion_lineal_k.m*

1.2. Regresión Logística

La estrategia de *Regresión Logística* se corresponde con una transformación de la lineal descrita en la sección anterior. El nombre de dicha técnica viene dado por la función utilizada para la transformación (función logística), que se describe en la ecuación (2). Mediante esta transformación se consigue que los resultados de la regressión (variable Y) cambien su dominio al rango $[0, 1]$.

$$f(x) = \frac{1}{1 + \mathrm{e}^{-k(x - x_0)}} \tag{2}$$

El planteaminto de esta estrategia en el campo de la *Inteligencia Artifical* es diferente respecto de la anterior. En este caso en lugar de tratar de aproximar la regresión al valor deseado, se utiliza para conocer el grado de similitud de los datos de entrada respecto de una determinada clase. Por tanto, funciona como un clasificador de carácter binario si se fija el valor de destino cuando la regresión cruza el umbral 0.5.

Para extender dicho funcionamiento a clasificadores de varias clases se utiliza una estrategia de clasificación por pares, lo cual conlleva la utilización de $\frac{k(k-1)}{2}$ clasficadores cuando existen k clases distintas de destino. El valor que se asigna es el de la clase que más veces haya resultado ganadora y se toma como un fallo los casos de empate.

El algoritmo de aprendizaje de pesos utilizado (al igual que en el caso anterior) se describe mediante su implementación en el lenguaje *Octave* en la figura 2. Dicha implementación se refiere al ajuste por mínimos cuadrados al igual que en el caso anterior. En este caso se pueden realizar varias iteraciones para el ajuste, pero se ha comprobado que para el conjunto de datos utilizada una iteración ofrece buenos resultados.

```
function w = regresion_logistica_K(x, y)
  w = zeros( size(x,2),1);
  nu = 1 ./ (1 .+ e.^-(x * w));
  s = (nu .* (1 .- nu));
  w = inv(x' .* s' * x)*x'*(s'*x*w + y - nu);
end
```

Figura 2: *Octave: /src/regresion_ logistica_ k.m*

2. Evaluación de resultados a partir de distintas cotas de error relativo para Regresión Lineal Múltiple

En esta sección se realiza un experimento para estudiar la evolución de la tasa de error conforme aumenta la cota de error relativo fijada para admitir una instancia como clasificada de manera correcta. El algoritmo utilizado es el de Regresión Lineal Múltiple mediante el aprendizaje por mínimos cuadrados descrito anteriormente.

El conjunto de datos que se ha utilizado para dichos experimentos es **Housing Data Set** [UCIa], que está formado por **506 instancias**. Dichas instancias contienen **13 atributos** de carácter númerico. La **clase de destino es de carácter numérico** (por lo que es una regresión y no una clasificación). En cuanto a la metodología experimental, se ha seguido una extrategia de **HoldOut** con particionamiento de los datos de manera que $\frac{2}{3}$ son utilizados en la fase de entrenamiento y $\frac{1}{3}$ en la de test.

Los resultados obtenidos tras los experimentos se muestran de manera tabular en la tabla 1 mientras que se puede apreciar la evolución conforme aumenta el rango de la cota de error de manera gráfica en la figura 3.

	Regresión — Housing Dataset			
	Lineal 10 %	*Lineal 15 %*	*Lineal 20 %*	*Lineal 25 %*
Error HoldOut	57.396 %	42.604 %	26.627 %	18.343 %

Tabla 1: *Evolución de la tasa de error para la* Regresión Lineal Múltiple *sobre el conjunto de datos* Housing *conforme aumenta la cota máxima de error relativo*

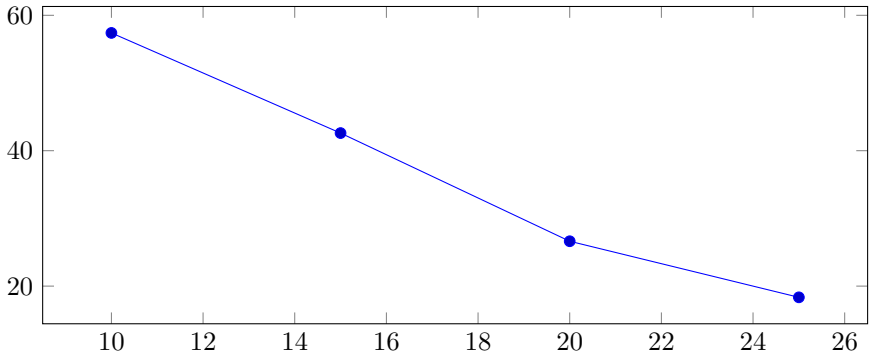

Figura 3: *Evolución de la tasa de error para la* Regresión Lineal Múltiple *sobre el conjunto de datos* Housing *conforme aumenta la cota máxima de error relativo*

3. Comparación de resultados entre Regresión Logística y Regresión Lineal Múltiple

En esta sección se realiza un experimento para estudiar la tasa de error obtenida mediante la estrategia de *Regresión Logística* para después compararla con los resultados obtenidos mediante la estrategia de *Regresión Lineal*. Dicha labor es posible cuando la clase de destino es numérica pero se puede discretizar. (En este caso es de carácter entero y acotada por lo que su discretización es trivial). Puesto que no es de tipo binario es necesario seguir la estrategía de clasificación por pares tal y como se ha descrito anteriormente.

El conjunto de datos que se ha utilizado para dichos experimentos es **Wine Data Set** [UCIb], formado por **178 instancias**. Dichas instancias contienen **12 atributos** de carácter númerico. La **clase de destino es de carácter numérico**. Sin embargo, en este caso es de tipo entero, por lo que puede ser vista como una variable categórica. En cuanto a la metodología experimental, se ha seguido una extrategia de **HoldOut** con particionamiento de los datos de manera que $\frac{2}{3}$ son utilizados en la fase de entrenamiento y $\frac{1}{3}$ en la de test.

Los resultados obtenidos tras los experimentos se muestran de manera tabular en la tabla 2. Además, se muestran de manera gráfica en la figura 4.

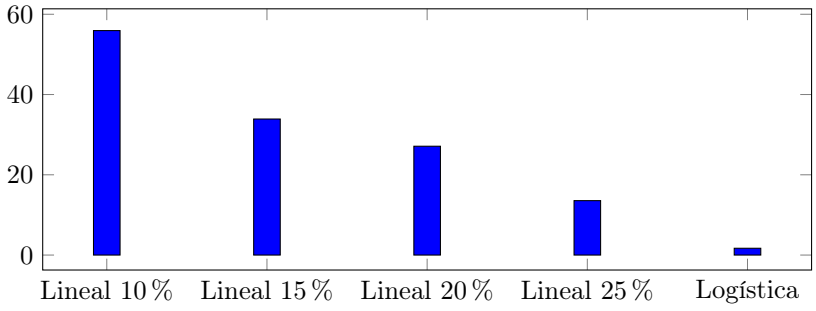

Figura 4: *Resultados obtenidos a nivel de tasa de error mediante* Regresión Lineal Múltiple *y* Regresión Logística *sobre el conjunto de datos* Wine

	Regresión — Wine Dataset				
	Lineal 10 %	*Lineal 15 %*	*Lineal 20 %*	*Lineal 25 %*	*Logística*
Error HoldOut	55.932 %	33.898 %	27.119 %	13.559 %	1.6949 %

Tabla 2: *Resultados obtenidos a nivel de tasa de error mediante* Regresión Lineal Múltiple *y* Regresión Logística *sobre el conjunto de datos* Wine

Tal y como se puede apreciar tras analizar los resultados, a través de la *Regresión Logística* se obtienen resultados mucho más óptimos. Sin embargo, para aplicar dicha técnica es necesario que la clase de destino esté discretizada. Además conlleva un mayor coste computacional derivado de la estrategia de clasificación por pares, que incrementa en un orden cuadrático los costes conforme aumentan el número de clases de destino.

Referencias

[CCAG17] Teodoro Calonge Cano and Carlos Javier Alonso GonzáLez. Técnicas de Aprendizaje Autómatico, 2016/17.

[GP17] Sergio García Prado. Regresión lineal, múltiple y logística. `https://github.com/garciparedes/machine-learning-regression`, 2017.

[UCIa] UCI Machine Learning Repository. Housing Data Set. `http://archive.ics.uci.edu/ml/datasets/Housing`.

[UCIb] UCI Machine Learning Repository. Wine Data Set. `http://archive.ics.uci.edu/ml/datasets/Wine`.