
Table of Contents

Segundo Proyecto Análisis Aplicado	1
Parámetros de Entrada y Salida	1
Paths que necesitaremos a lo largo del programa:	2
Automatizar creación del archivo *.mod	2
Automatizar creación del archivo *.nl	2
Encontrar las Betas	2
Comparar con el test y los resultados originales	3

Segundo Proyecto Análisis Aplicado

```
% Dado cualquier base de datos estandarizada del formato:
%
% Identificador      VarIndep      VarDep1      VarDep2 ...
%
% 984727             1             .9867        -2.5446 ...
%
% 984728             1             .9313         0.2342 ...
%
% 984729             0             .3354        -3.4521 ...
%
% este programa realiza la regresión logística regularizada penalizando
% a la función objetivo con la norma 2 de las betas multiplicada por una
% constante
%
% Es decir la función objetivo queda como sigue:
%
% min -sum(yi*(xi'Beta))+ sum(log(1+exp(xl'Beta))) + gamma/2*norm(Beta)^2
%
% Podemos correr este ejemplo para probar la función:
%
% [x1,porc]=regL(5,.5,.1, 30,0,'/bin/AMPL/');
%
function [betas, porc]=regL (gamma,corte,tamTest, nvar,reor,pathAmpl);
```

Parámetros de Entrada y Salida

```
% [ENTRADA] gamma = valor para regularizar
%
% [ENTRADA] corte = el valor de probabilidad para el cual aceptaremos entre (0,1)
%
% [ENTRADA] tamTrain = porcentaje que queremos que sea de test
%
% [ENTRADA] nvar = el numero de variables independientes
%
% [ENTRADA] reor = para sacar la muestra aleatoriamente (0=no aleat,1=aleat)
%
% [ENTRADA] pathAmpl = path de los programas de ampl
%
```

```
% [SALIDA] x1 = valor de las betas que toma el modelo
%
% [SALIDA] porc = porcentaje de aciertos que tuvo el modelo
%
```

Paths que necesitaremos a lo largo del programa:

```
%Path de ampl:
    path( path,pathAmpl)
%Path de nuestro trabajo
    work=pwd
```

```
work =
```

```
/Users/SavrGG/Dropbox/Computadoras/001_MATLAB/001_AnalisisAplicado/006_Pr
```

Automatizar creación del archivo *.mod

```
fil2 = fopen( 'LIST_large2', 'w' );
[train1,test1,ntrain1,ntest1] = wdbcData('Segundo',nvar,tamTest,reur);
fprintf('el tamaño del conjunto de entrenamiento es : %i \n',ntrain1)
fprintf('el tamaño del conjunto de test es : %i \n\n',ntest1)
estDatos(train1,'trainReor1.mod',gamma);
fprintf('Se ha creado el archivo trainReor1.mod en la dirección: \n %s \n\n',pwd);
fprintf(fil2,'trainReor1 \n');
fclose(fil2);
```

```
el tamaño del conjunto de entrenamiento es : 512
el tamaño del conjunto de test es : 57
```

```
Se ha creado el archivo trainReor1.mod en la dirección:
```

```
/Users/SavrGG/Dropbox/Computadoras/001_MATLAB/001_AnalisisAplicado/006_Pr
```

Automatizar creación del archivo *.nl

```
writePerl(work);
perl('rCUTE');
fprintf('Se ha creado el archivo trainReor1.nl en la dirección: \n %s \n\n',pwd);
use_ampl_stub trainReor1.nl;
```

```
Se ha creado el archivo trainReor1.nl en la dirección:
```

```
/Users/SavrGG/Dropbox/Computadoras/001_MATLAB/001_AnalisisAplicado/006_Pr
```

Encontrar las Betas

```
betas=Newton1pruebas ('amplpnt','amplstub', 'trainReor1', 50)
```

```
% Valor de la función y gradiente en el óptimo:
```

```
betas =  
  
    2.0260  
   -4.8345  
    0.2017  
   -3.3105  
    0.1829  
    1.1873  
   -6.6502  
    4.9073  
    3.3270  
   -0.7268  
    1.4684  
    4.0385  
   -1.3002  
   -2.2868  
    5.1602  
    0.7778  
    2.7905  
   -4.0470  
    4.7538  
   -0.9940  
   -6.5278  
    5.3618  
    3.4039  
    4.9252  
    7.5667  
   -0.1536  
   -2.0334  
    2.6261  
   -0.2926  
    2.1219  
    4.2440
```

Comparar con el test y los resultados originales

```
Orig=test1(:,1);  
resul=zeros(ntest1,1);  
test1(:,1)=ones(ntest1,1);  
suma=0;  
for i=1:ntest1  
    b=-betas'*test1(i,:)';  
    valor = 1/(1+exp(b));  
    if valor >= corte  
        resul(i)=1;  
    end
```

```
        if(Orig(i)==resul(i))
            suma=suma+1;
        end
    end

    porc=suma/ntest1;
    fprintf('El porcentaje de predicción es: %5.4f \n', porc)

        El porcentaje de predicción es: 1.0000

    end
```

Published with MATLAB® 7.12