

## **PRAC6: Discriminantes Lineales**

Asignatura: CLP: Clasificación de Patrones.  
Optativa de 2º Ciclo  
ETSETB  
UPC

Profesores:  
Margarita Cabrera  
Josep Vidal  
UPC-TSC-D5

Abril-2007

1	Objetivos de la práctica 6 .....	2
2	Programas facilitados .....	2
3	Trabajo Previo al Laboratorio.....	3
4	Trabajo en el Laboratorio. ....	3
5	Anexo 1 .....	5
6	Anexo 2: Base de Datos Titanic .....	9

## 1 Objetivos de la práctica 6

En esta práctica se prueba la clasificación tanto de datos gaussianos como de la base de datos ZIP, mediante los algoritmos adaptativos:

- Fixed Increment Single Simple Perceptron (Transparencia 19 y 21) (Perceptron en el resto de este documento)
- MMSE-LMS (Transparencia 27 y 36) (MMSE en el resto de este documento)

El resultado de clasificación se compara con el ldc.

Para la base de datos gaussiana también se testea el clasificador sv: (Subrutina svc).

Mediante este tipo de algoritmos se “aprende” a partir de la base de datos una función discriminante que clasifica dos clases:

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0 = g(\mathbf{y}) = \mathbf{a}^T \mathbf{y} \begin{matrix} > \omega_1 \\ < \omega_2 \end{matrix} 0$$

En definitiva se obtendrá el valor del vector  $\mathbf{a}$ , actualizado a lo largo de un número determinado de iteraciones, que minimizan iterativamente una función objetivo  $\mathbf{J}$ .

Mediante el perceptron se minimiza la función  $J_p(\mathbf{a}) = \sum_{\mathbf{y} \in E} (-\mathbf{a}^T \mathbf{y})$  y mediante el

MMSE se minimiza la función  $J_s(\mathbf{a}) = \sum_{i=1}^N (\mathbf{a}^T \mathbf{y}_i - b_i)^2 = \|\mathbf{Y}\mathbf{a} - \mathbf{b}\|^2 = (\mathbf{Y}\mathbf{a} - \mathbf{b})^T (\mathbf{Y}\mathbf{a} - \mathbf{b})$ .

## 2 Programas facilitados

### Prac6\_gauss.m

- Genera datos gaussianos binarios: 2 clases
- Clasifica mediante criterio ldc
- Implementa el alg. Perceptron, con la posibilidad de ver gráficamente la evolución de la frontera de decisión a lo largo de todas las iteraciones.

### Prac6\_leeZIP.m

- Lee base de datos ZIP en el dominio transformado (DCT). De entrada los vectores son de 64 componentes.
- Se eligen únicamente dos clases de las 10 correspondientes a los 10 dígitos.
- Clasifica mediante criterio ldc.

Con ambos programas las variables de interés para probar los dos algoritmos adaptativos son:

N\_datos= Número total de vectores  
K\_dim= dimensión de los vectores.  
Signal= Matriz de N\_datosxk\_dim y contiene todos los vectores.  
Nlab=Vector de N\_datosx1 contiene la clase correspondiente a cada vector  
Lablist=Vector de etiquetas 2x1

### Prac6\_adapta.m

- Implementa el algoritmo adaptativo “Perceptron”
- Se calcula el error de clasificación para este algoritmo.
- Implementa el algoritmo adaptativo LMS (INCOMPLETO).
- Se calcula el error de clasificación para este algoritmo.
- Dibuja la evolución de la función objetivo: J para ambos algoritmos
- Dibuja el mapa de señal y las soluciones obtenidas para los vectores.

## 3 Trabajo Previo al Laboratorio.

- Observe el código de la subrutina Prac6\_adapta.m disponible en el anexo.
- Especialmente fíjese en el algoritmo adaptativo PERCEPTRON.
- En la parte del algoritmo MMSE, implemente en matlab los pasos (1), (2) y (3) según el algoritmo explicado en la clase correspondiente a este tema.
- Para el paso (1) programe  $Mu = \frac{1}{\text{suma}(\text{autovalores}(Y^T Y))}$ . Matriz YY en el programa.

## 4 Trabajo en el Laboratorio.

Programe los pasos (1), (2) y (3) en prac6\_adapta.m

### Parte 1

- Ejecute prac6\_gauss con SNR=5dB
- A continuación y sin borrar las variables ejecute prac6\_adapta.m
- Observe la evolución de la función objetivo: J para ambos algoritmos
- Observe el mapa de señal y las soluciones obtenidas para los vectores, compare con la solución obtenida para el ldc.
- Compare los errores de clasificación: ldc, Perceptron y MMSE
- Repita los anteriores pasos con SNR=0dB

### Parte 2

- Ejecute prac6\_leeZIP con dos clases (2 dígitos cualesquiera)
- A continuación y sin borrar las variables ejecute prac6\_adapta.m
- Observe la evolución de la función objetivo: J para ambos algoritmos
- Compare los errores de clasificación: ldc, Perceptron y MMSE

- Repita los anteriores pasos con  $N_{dim}=4$

### Parte 3

- A partir del fichero `prac6_gauss.m` añada como opción de clasificación la del “Support Vector Classifier”. Para ello dibuje los resultados en la misma figura donde se presentan los del clasificador ldc. Se sugieren los siguientes parámetros de llamada a la subrutina

```
[W_svc,Ind_sc]=svc(A1_gauss,'r',0.1);
```

```
[W_svc,Ind_sc]=svc(A1_gauss);
```

- Pruebe cada uno de los dos casos con  $SNR=5dB$  y con  $SNR=0dB$ , evalúe el error de clasificación y comente los resultados.

### Parte 4 (OPTATIVO)

En el caso de que tenga tiempo pruebe los algoritmos de `prac6_adapta.m` con otras base de datos binarias:

*Ejemplos:*

- Shearth (Práctica 0)
- SPAM (Práctica 2)
- Titanic (Ver Anexo 2 y fichero **prac6\_titanic.m** para obtener información de esta base de datos).

**COMENTE LOS RESULTADOS EN EL DOCUMENTO: PRAC6\_cla\*.doc**

## 5 Anexo 1

### Prac6\_adapta.m

```

%EJEMPLO DE BUSQUEDA DE VECTOR SEPARADOR;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% *****
%  ALGORITMO LMS - FUNCION PERCEPTRON
% *****
% *****
%  CAMBIO DE SIGNO CLASE 2
% *****
YY=[signal,ones(size(nlab))];
labsYY=nlab;
YY(find(nlab==lablist(2),:)-YY(find(nlab==lablist(2),:));

% *****
%  INICIO VECTOR
% *****
A_vector=ones(k_dim+1,1);
N_puntos=1000;
% *****
%  PERMUTACION DE LOS DATOS
% *****
Index_perm=randperm(N_datos);
YY=YY(Index_perm,:);
labsYY=labsYY(Index_perm);

%Funcion objetivo PERCEPTRON
J=length(find(YY*A_vector<0));
N_max=1000;
K_sample=0;    %Contador modulo N_datos
K_cont=0;     %cuenta el numero de iteraciones
K_ada=0;     %cuenta el numero de adaptaciones realmente realizadas
Mu=0.5;
Jevol=J;

while(and(J>0,K_cont<N_max))
    K_sample=mod(K_sample+1,N_datos);
    y_k=YY(K_sample+1,:);
    y_k=y_k(:);
    if A_vector'*y_k<0
        %Si la muestra correspondiente a K_sample+1 esta mal clasificada
        % realizamos una adaptacion
        A_vector=A_vector+Mu*y_k;
        K_ada=K_ada+1;
        %Actualizamos el valor de la funcion J
        J=length(find(YY*A_vector<0));
    end
    K_cont=K_cont+1;
    Jevol=[Jevol,J];
end

figure('name','Error J Evolution')
plot(Jevol,'b');

```

```

hold on

fprintf(1,' ITERACIONES REALIZADAS = %g \n', K_cont)
fprintf(1,' ADAPTACIONES REALIZADAS = %g \n', K_ada)

% *****
% MEDIDA PROBABILIDAD DE ERROR
% *****
labs2=zeros(size(nlab));
for i1=1:N_datos
    V_signal=[signal(i1,:) 1];
    if (V_signal*A_vector>0)
        labs2(i1)=lablist(1);
    else
        labs2(i1)=lablist(2);
    end
end

err_1=length(find(nlab==labs2));
err_1=(N_datos-err_1)/N_datos;
fprintf(1,' ERROR PERCEPTRON = %g \n', err_1)

% *****
% ALGORITMO LMS - FUNCION PERCEPTRON
% *****
% *****
% CAMBIO DE SIGNO CLASE 2
% *****
YY=[signal,ones(size(nlab))];
labsYY=nlab;
YY(find(nlab==lablist(2)),:)-YY(find(nlab==lablist(2)),:);

% *****
% INICIO VECTOR
% *****
A_lms=ones(k_dim+1,1);
N_puntos=1000;

% *****
% PERMUTACION DE LOS DATOS
% *****
Index_perm=randperm(N_datos);
YY=YY(Index_perm,:);
labsYY=labsYY(Index_perm);

% *****
%Funcion objetivo MMSE
% *****

Vector_b=ones(size(nlab));

J=1;
N_max=5000;
K_sample=0; %Contador modulo N_datos
K_cont=0; %cuenta el numero de iteraciones

% Inicialice la variable Mu (1)
% *****

```

```

Zeta=0.0000001      %Cota minima de error
J_evol=J;

while(and(J>Zeta,K_cont<N_max))
    K_sample=mod(K_sample+1,N_datos);
    y_k=YY(K_sample+1,:);
    y_k=y_k(:);

    % Obtenga J=valor instantaneo del error
    % para la iteracion K_cont (2)
    % Adapte el Vector de pesos A_lms (3)

    J=abs(J);
    K_cont=K_cont+1;
    J_evol=[J_evol,J];
end

win=ones(1,50)/50;
J_evol=conv(J_evol,win);
plot(J_evol,'r');
hold off
grid
zoom on
title('Funcion perceptron - blue, MMSE red')

fprintf(1,' ITERACIONES REALIZADAS = %g \n', K_cont)

% *****
% MEDIDA PROBABILIDAD DE ERROR
% *****
labs2=zeros(size(nlab));
for i1=1:N_datos
    V_signal=[signal(i1,:), 1];
    if (V_signal*A_lms>0)
        labs2(i1)=lablist(1);
    else
        labs2(i1)=lablist(2);
    end
end
err_1=length(find(nlab==labs2));
err_1=(N_datos-err_1)/N_datos;
fprintf(1,' ERROR LMS = %g \n', err_1)

if (k_dim==2)

figure
hold on
Y=signal(find(nlab==lablist(1)),:);
plot(Y(:,1),Y(:,2),'+b');
Y=signal(find(nlab==lablist(2)),:);
plot(Y(:,1),Y(:,2),'xr');

X_eje=min(signal(:,1)):(max(signal(:,1))-min(signal(:,1)))/(N_puntos-1):max(signal(:,1));
Y_eje=-(A_vector(3)+A_vector(1))*X_eje/A_vector(2);
plot(X_eje,Y_eje,'b');

```



```
X_eje=min(signal(:,1)):(max(signal(:,1))-min(signal(:,1)))/(N_puntos-1):max(signal(:,1));
Y_eje=-(A_lms(3)+A_lms(1))*X_eje/A_lms(2);
plot(X_eje,Y_eje,'r');

hold off
grid
zoom on
title('Funcion perceptron - blue, MMSE red')

end
```

## 6 Anexo 2: Base de Datos Titanic

Esta base de datos es del link:

<http://www.cs.toronto.edu/~delve/data/datasets.html>

Y esta es la información que los autores facilitan sobre la misma:

```
NAME: Population at Risk and Death Rates for an Unusual Episode
TYPE: Complete record for all of population at risk
SIZE: 2201 observations, 4 variables

DESCRIPTIVE ABSTRACT:
For each person on board the fatal maiden voyage of the ocean liner
Titanic, this dataset records sex, age [adult/child], economic status
[first/second/third class, or crew] and whether or not that person
survived.

SOURCE:
"Report on the Loss of the `Titanic' (S.S.)" (1990), _British Board of
Trade Inquiry Report_ (reprint), Gloucester, UK: Allan Sutton
Publishing.

VARIABLE DESCRIPTIONS:
Column
  1  Class (4 = crew, 1 = first, 2 = second, 3 = third)
 10  Age  (1 = adult, 0 = child)
 19  Sex  (1 = male, 0 = female)
 28  Survived (1 = yes, 0 = no)

Values are aligned and delimited by blanks.  There are no missing
values.

SPECIAL NOTES:
There is not complete agreement among primary sources as to the exact
numbers on board, rescued, or lost.

STORY BEHIND THE DATA:
The sinking of the Titanic is a famous event, and new books are still
being published about it.  Many well-known facts--from the proportions
of first-class passengers to the "women and children first" policy,
and
the fact that that policy was not entirely successful in saving the
women and children in the third class--are reflected in the survival
rates for various classes of passenger.  These data were originally
collected by the British Board of Trade in their investigation of the
sinking.

PEDAGOGICAL NOTES:
These data make an interesting exercise if given to a class without
their context, which the students must attempt to discover.  The
instructor will probably want to answer questions from the class,
"Twenty Questions" style.

There is a similar set of data circulating without any detailed
explanation or compiler's name attached, under the same title, which
omits the crew (and does not agree with any of the primary sources
that
```

I was able to find.) Credit for the original idea goes to the originator of that exercise: my version is merely an attempt to provide a more complete context.

Additional information about these data can be found in the "Datasets and Stories" article "The `Unusual Episode' Data Revisited" in the Journal of Statistics Education (Dawson 1995). Send the message

```
send jse/v3n3/datasets.dawson
```

to the address [archive@jse.stat.ncsu.edu](mailto:archive@jse.stat.ncsu.edu)

SUBMITTED BY:

Robert J. MacG. Dawson

Department of Mathematics and Computing Science

Saint Mary's University

Halifax, Nova Scotia B3H 3C3

CANADA

[rdawson@husky1.stmarys.ca](mailto:rdawson@husky1.stmarys.ca)