# CLP Lab 3 Report

Albert Aparicio Isarn
albert.aparicio.isarn@alu-etsetb.upc.edu

Héctor Esteban
hect.esteban@gmail.com

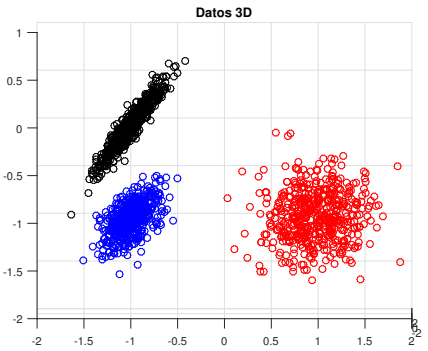
## 1. Selección de características con bases de datos gaussianas

La tabla 1 muestra los errores LC y QC obtenidos en entreno y en test para cada una de las tres dimensiones. La semilla utilizada es la número **2**.

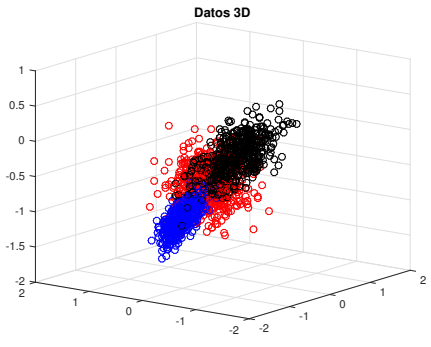
| **Fase** | **Training** | | | **Test** | | |
|---|---|---|---|---|---|---|
| **Dimensión** / **Clasificador** | **1D** | **2D** | **3D** | **1D** | **2D** | **3D** |
| **Lineal (LC)** | 0,00933333 | 0,01 | 0,004 | 0,00533333 | 0,00533333 | 0 |
| **Cuadrático (QC)** | 0,00466667 | 0,00133333 | 0 | 0,00266667 | 0 | 0 |

Cuadro 1: Errores LC y QC obtenidos en entreno y en test para cada una de las tres dimensiones. $SNR = 10dB$

Con el método MDA y semilla 2, dos proyecciones podrían ser las mostradas en las figuras 1a y 1b.




(a) Proyección 2D con los *clusters* separados.




(b) Proyección 2D con los *clusters* superpuestos.

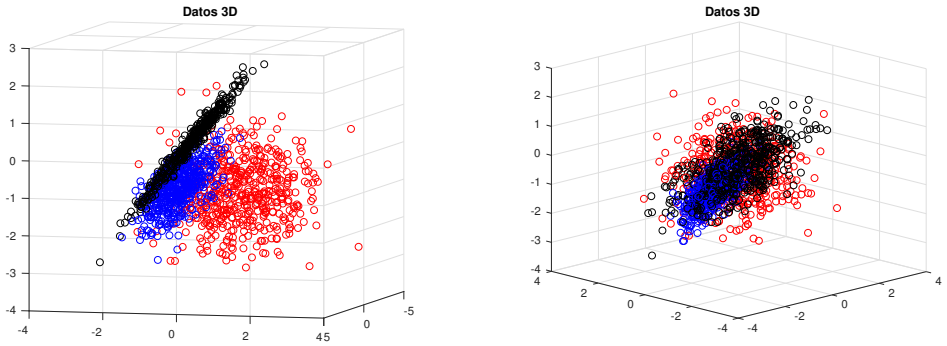
Figura 1: Proyecciones 2D para MDA, semilla 2, $SNR = 10dB$

A continuación se presentan los resultados de los dos apartados anteriores, pero seleccionando $SNR = 0dB$.

| **Fase** | **Training** | | | **Test** | | |
|---|---|---|---|---|---|---|
| **Dimensión** / **Clasificador** | **1D** | **2D** | **3D** | **1D** | **2D** | **3D** |
| **Lineal (LC)** | 0,194667 | 0,184 | 0,156 | 0,181333 | 0,154667 | 0,133333 |
| **Cuadrático (QC)** | 0,189333 | 0,166667 | 0,0733333 | 0,157333 | 0,138667 | 0,096 |

Cuadro 2: Errores LC y QC obtenidos en entreno y en test para cada una de las tres dimensiones. $SNR = 0dB$

Para los casos de 3D, las probabilidades de error son las mismas, ya que el procesado es el mismo (no hay reducción de características). En los casos en que sí se reducen, el MDA da menores probabilidades de error tanto en 2D como en 1D.




(a) Proyección 2D con los *clusters* separados.

(b) Proyección 2D con los *clusters* superpuestos.

Figura 2: Proyecciones 2D para MDA, semilla 2, $SNR = 0dB$

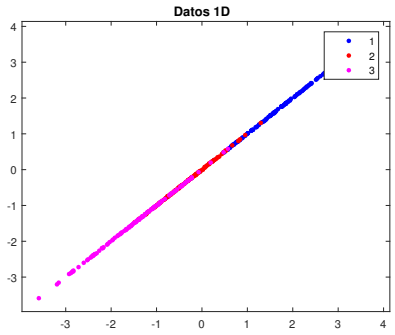
En las figuras 3 y 4 se muestran los *scatters* para los métodos PCA y MDA para 1 y 2 dimensiones.

Para 1D, el MDA compacta mejor los datos, reduciendo la superposición de los datos. El PCA muestra los datos muy superpuestos.

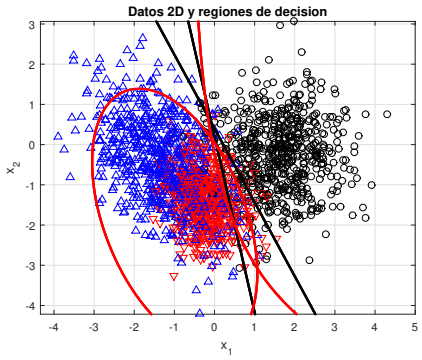
En el caso 2D, todas las clases están superpuestas, aunque en MDA lo están en menor grado. Además, en MDA las clases están más compactadas. En el scatter de PCA, hay mayor dispersión de datos.

A continuación se presentan los resultados del apartado 4 y 5 de la práctica, usando distribuciones gausianas con las medias alineadas.

Usando la semilla 2, los vectores de las medias son los que se muestran en la ecuación (1):

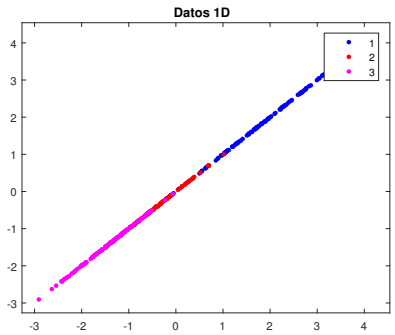



(a) Proyección en 1D.

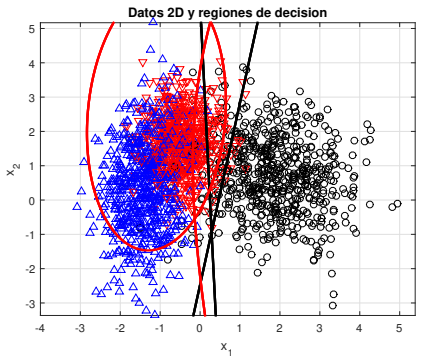



(b) Proyección en 2D.

Figura 3: Proyecciones en 1D y 2D para PCA, semilla 2, $SNR = 0dB$




(a) Proyección en 1D.




(b) Proyección en 2D.

Figura 4: Proyecciones en 1D y 2D para MDA, semilla 2, $SNR = 0dB$

$$\begin{pmatrix} 0{,}99631702257037136 & -0{,}013479105534217606 & -0{,}084680010926471622 \\ 0 & 0 & 0 \\ -0{,}99631702257037136 & 0{,}013479105534217606 & 0{,}084680010926471622 \end{pmatrix} \quad (1)$$

El rango de la matriz $S_b$ teóricamente deberia ser 1 porque las medias están alineadas. Sin embargo, MATLAB da como resultado 2, debido a la estimación de las medias. Usando MDA, con 1 característica bastaria.

A continuación se presentan los resultados de los apartados iniciales, usando $SNR = \{-5, 5\}\, dB$.

Las probabilidades de error de los clasificadores LC y QC se muestran en las tablas 3 y 4

| **Fase** | **Training** | | | **Test** | | |
|---|---|---|---|---|---|---|
| **Dimensión / Clasificador** | **1D** | **2D** | **3D** | **1D** | **2D** | **3D** |
| **Lineal (LC)** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Cuadrático (QC)** | 0 | 0 | 0 | 0 | 0 | 0 |

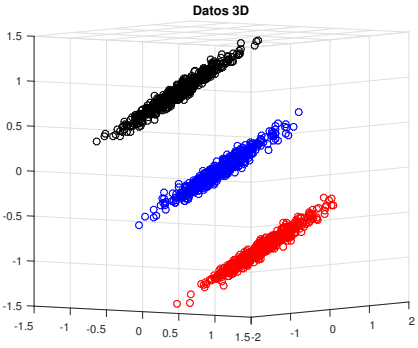
Cuadro 3: Errores LC y QC obtenidos en entreno y en test para cada una de las tres dimensiones. $SNR = 5dB$. Se ha usado la semilla 3

| **Fase** | **Training** | | | **Test** | | |
|---|---|---|---|---|---|---|
| **Dimensión / Clasificador** | **1D** | **2D** | **3D** | **1D** | **2D** | **3D** |
| **Lineal (LC)** | 0,674 | 0,660667 | 0 | 0,674667 | 0,666667 | 0 |
| **Cuadrático (QC)** | 0,67 | 0,658667 | 0 | 0,650667 | 0,658667 | 0 |

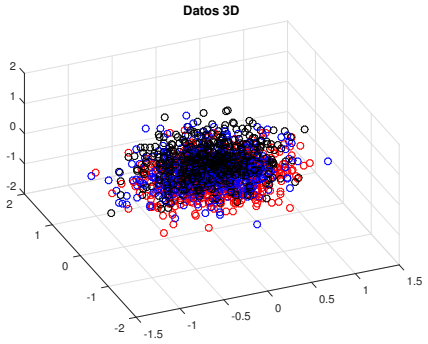
Cuadro 4: Errores LC y QC obtenidos en entreno y en test para cada una de las tres dimensiones. $SNR = -5dB$. Se ha usado la semilla 3

En las figuras 5 y 6 se muestran las proyecciones 2D para el método MDA.

El método MDA resulta ventajoso cuando hay poca separación entre clases, de manera que a la hora de reducir características, se trata de maximizar esta separación.

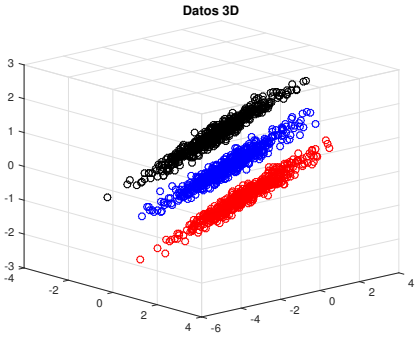



(a) Proyección 2D con los *clusters* separados.

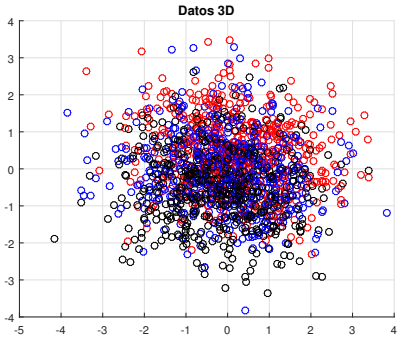



(b) Proyección 2D con los *clusters* superpuestos.

Figura 5: Proyecciones 2D para MDA, semilla 3, $SNR = 5dB$




(a) Proyección 2D con los *clusters* separados.




(b) Proyección 2D con los *clusters* superpuestos.

Figura 6: Proyecciones 2D para MDA, semilla 3, $SNR = -5dB$

## 2. MDA en clasificación

El gráfico de los errores de clasificación para $d' = 1 : 256$ se muestra en la figura 7.

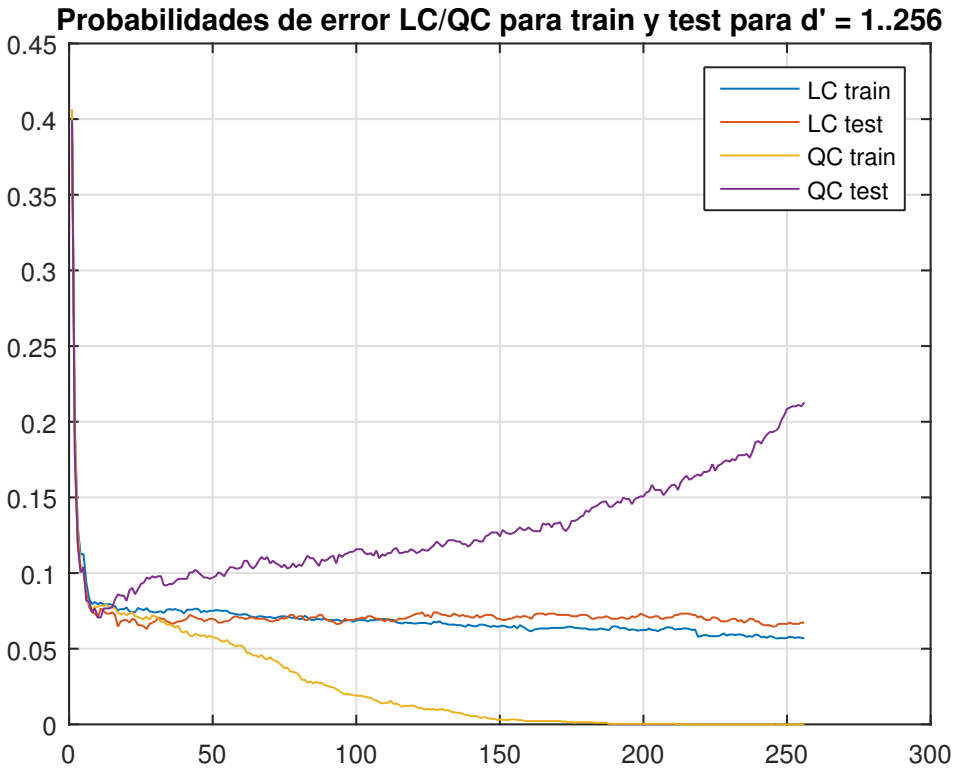
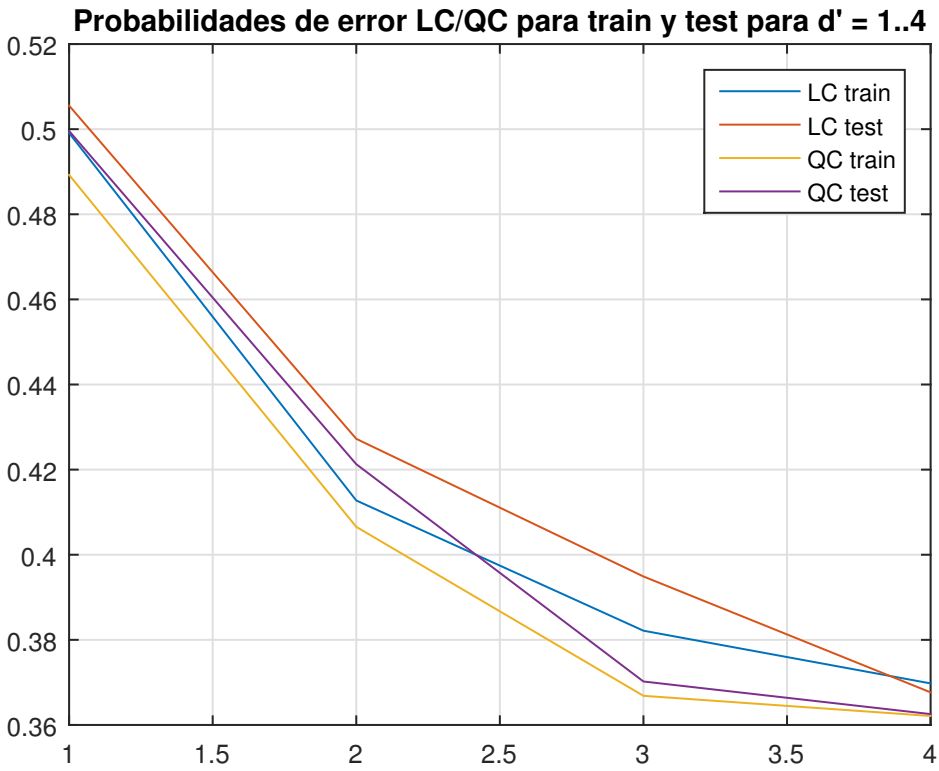



Figura 7: Gráficos de los errores de clasificación para $d' = 1 : 256$.

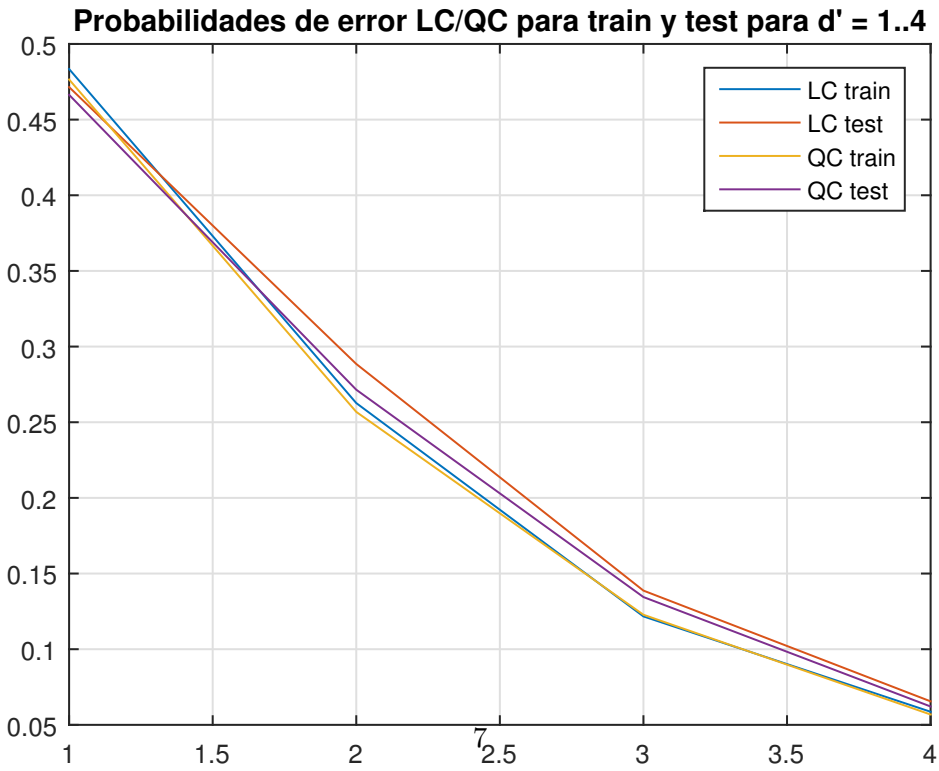
Para usar el método MDA, $d_{MAX}$ será 4, ya que MDA puede trabajar hasta c-1 (c numero de clases). Hemos visto que esto se da en el momento de calcular la matriz W, ya que tiene tamaño 256x4

Los gráficos de los errores de clasificación, para $d' = 1 : d_{MAX}$, para los métodos PCA y MDA se muestran en las figuras 8a y 8b.

Las probabilidades de error son menores en MDA, para $d' = 1 : 4$.




(a) Método PCA




(b) Método MDA

Figura 8: Gráficos de errores de clasificación para $d' = 1 : 4$ con los métodos PCA y MDA.

Código fuente de prac3_fonemas_MDA.m

```
% prac3_fonemas_MDA.m
clear;
close all;  % close all previous figures

%% Options / Initalitation
i_dib=0;          %0 NO /1 YES: plot spectrums
N_coor = 256;
V_coor=1:N_coor;              %64 to take all features set 1:64
% V_coor=[22 64];          % EXAMPLE: Selection of a subset of two
    features

N_feat=length(V_coor);
% class name: Labels:
% 1(aa);2(ao);3(dcl);4(iy);5(sh);
N_classes=5;
N_fft=256;             %256 (8KHz) 128 (4KHz), 64 (2KHz), 32(1khZ)
%% Database load
load BD_phoneme

%% MEAN IS REMOVED FROM DATABASE
X=X-ones(length(Labels),1)*mean(X);

%% Spectrum plot
if i_dib==1
    Frec_max=8*N_fft/256;      %Max frequency in KHz
    eje_frec=(0:N_fft-1)*Frec_max/N_fft;
    clases=['aa';'ao';'dc';'iy';'sh'];
    figure('name','LOG(Espectrum)')
    for i_clas=1:N_classes
        subplot(3,2,i_clas)
        hold on
        index=find(Labels==i_clas);
        for i1=1:length(index)
            plot(eje_frec,X(index(i1),1:N_fft));
        end
        hold off
        grid
        zoom on
        xlabel('frec(KHz)')
        ylabel(clases(i_clas,:));
    end
    subplot(3,2,N_classes+1)
    hold on
    i_color=['b' 'r' 'g' 'k' 'y'];
    for i_clas=1:N_classes
        index= Labels==i_clas;
        aux=mean(X(index,1:N_fft));
        plot(aux,i_color(i_clas));
    end
    hold off
    grid
    zoom on
```

```
    xlabel('Feature Number')
    ylabel('log espectro')
    title('Average');
    clear index aux i_color i_clas eje_frec Frec_max
end
% clear i_dib N_fft

%% Feature selection
if V_coor(1)~=0
    X=X(:,V_coor);  % Feature selection
end
% clear V_coor

%% Database partition
P_train=0.7;
Index_train=[];
Index_test=[];
for i_class=1:N_classes
    index=find(Labels==i_class);
    N_i_class=length(index);
    [I_train,I_test] = dividerand(N_i_class,P_train,1-P_train);
    Index_train=[Index_train;index(I_train)];
    Index_test=[Index_test;index(I_test)];
end
% Train Selection
X_train=X(Index_train,:);
Labels_train=Labels(Index_train);
% Test Selection and mixing
X_test=X(Index_test,:);
Labels_test=Labels(Index_test);
% clear Index_train Index_test index i_class N_i_class I_train I_test

%% Projections
W = mda_clp(X_train,Labels_train,N_classes);%pca(X_train);
W_size = size(W);

%% Data projection
X_train_proj = X_train * W;
X_test_proj = X_test * W;

LC_train_Pe = zeros(W_size(2),1);
LC_test_Pe = zeros(W_size(2),1);
QC_train_Pe = zeros(W_size(2),1);
QC_test_Pe = zeros(W_size(2),1);

% TODO Select d columns, compute error probabilities and plot graphics
tic
parfor d=1:W_size(2)%     N_coor
    %% Create a default (linear) discriminant analysis classifier:
    linclass = fitcdiscr(X_train_proj(:,1:d),Labels_train,'prior','
    empirical')

    Linear_out = predict(linclass,X_train_proj(:,1:d));
```

```
    Linear_Pe_train=sum(Labels_train ~= Linear_out)/length(Labels_train);
    fprintf(1,' error Linear train = %g   \n', Linear_Pe_train)

    LC_train_Pe(d) = Linear_Pe_train;

    Linear_out = predict(linclass,X_test_proj(:,1:d));
    Linear_Pe_test=sum(Labels_test ~= Linear_out)/length(Labels_test);
    fprintf(1,' error Linear test = %g   \n', Linear_Pe_test)

    LC_test_Pe(d) = Linear_Pe_test;

    %% Create a quadratic discriminant analysis classifier:
    quaclass = fitcdiscr(X_train_proj(:,1:d),Labels_train,'discrimType','
    quadratic','prior','empirical')

    Quadratic_out= predict(quaclass,X_train_proj(:,1:d));
    Quadratic_Pe_train=sum(Labels_train ~= Quadratic_out)/length(
    Labels_train);
    fprintf(1,' error Quadratic train = %g   \n', Quadratic_Pe_train)

    QC_train_Pe(d) = Quadratic_Pe_train;

    Quadratic_out= predict(quaclass,X_test_proj(:,1:d));
    Quadratic_Pe_test=sum(Labels_test ~= Quadratic_out)/length(
    Labels_test);
    fprintf(1,' error Quadratic test = %g   \n', Quadratic_Pe_test)

    QC_test_Pe(d) = Quadratic_Pe_test;

    %% Test confusion matrices
    CM_Linear_test=confusionmat(Labels_test,Linear_out)
    CM_Quadratic_test=confusionmat(Labels_test,Quadratic_out)

    % Print d'
    d
end
toc

plot(LC_train_Pe); hold on
plot(LC_test_Pe);
plot(QC_train_Pe);
plot(QC_test_Pe);

title(sprintf('Probabilidades de error LC/QC para train y test para d'' =
    1..%d', W_size(2)));
grid on
legend('LC train', 'LC test' , 'QC train', 'QC test', 'best');

hold off
```

../prac3_fonemas_MDA.m