

CLP Lab 2 Report

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

Héctor Esteban
hect.esteban@gmail.com

1. Clasificación aplicada a la base de datos Phoneme

En la figura 1 se pueden ver los espectros de los fonemas.

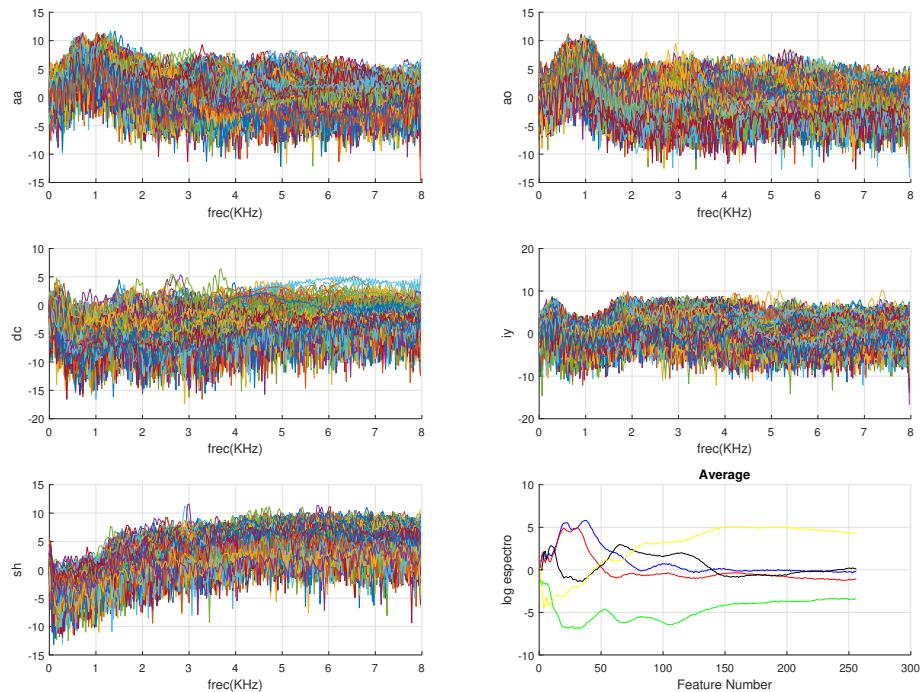


Figura 1: Espectros de los fonemas de la base de datos. En la figura inferior derecha se puede ver el espectro de los promedios de cada fonema.

1.1. Clasificación con 64 características

En la tabla 1 se muestran las probabilidades de error para los clasificadores Lineal y Quadrático, tanto en entrenamiento (Train) como en Test.

Fase Clasificador \	Train	Test
Lineal (LC)	0,073588342440801452	0,08851063829787234
Cuadrático (QC)	0,049908925318761385	0,10553191489361702

Cuadro 1: Probabilidades del error de clasificación de Train y de Test, obtenidas mediante LC y QC usando todas las características.

Las matrices de confusión obtenidas con los clasificadores Lineal y Quadrático en test se encuentran en las ecuaciones (1) y (2), respectivamente.

$$\begin{pmatrix} 129 & 52 & 0 & 0 & 0 \\ 30 & 237 & 0 & 0 & 0 \\ 0 & 0 & 190 & 5 & 2 \\ 0 & 0 & 2 & 301 & 0 \\ 0 & 0 & 0 & 0 & 227 \end{pmatrix} \quad (1)$$

$$\begin{pmatrix} 122 & 59 & 0 & 0 & 0 \\ 49 & 218 & 0 & 0 & 0 \\ 0 & 0 & 193 & 3 & 1 \\ 0 & 0 & 15 & 284 & 4 \\ 0 & 0 & 0 & 0 & 227 \end{pmatrix} \quad (2)$$

La matriz de confusión del clasificador cuadrático en test tiene mayor índice de errores, debida a la tendencia a sobreentrenar que tiene este clasificador.

La matriz del clasificador lineal tiene menos errores en test, cosa que también se refleja en sus probabilidades de error, ya que por sus pocos grados de libertad, debe generalizar más en la clasificación.

1.2. Clasificación con 2 características

Para esta sección se han probado dos pares de características: el par (8, 40) y el par (22, 64).

Para las componentes 22 y 64, se han elegido componentes que separan bien todas las componentes a la vez. Sin embargo, el resultado es que los pares de clases 1-2 y 4-5 aparecen mezclados.

Eligiendo las componentes 8 y 40, las clases 4 y 5 se han separado mejor, aunque la pareja 1-2 sigue apareciendo mezclada.

Fase Clasificador	Train	Test
Lineal (LC)	0,26156648451730419	0,26978723404255317
Cuadrático (QC)	0,253551912568306	0,26638297872340427

Cuadro 2: Probabilidades del error de clasificación para el par (8, 40).

1.2.1. Par (8, 40)

Las probabilidades de error obtenidas en este caso se encuentran en la tabla 2. En la figura 2 se muestra el dibujo del scatter con las fronteras de clasificación.

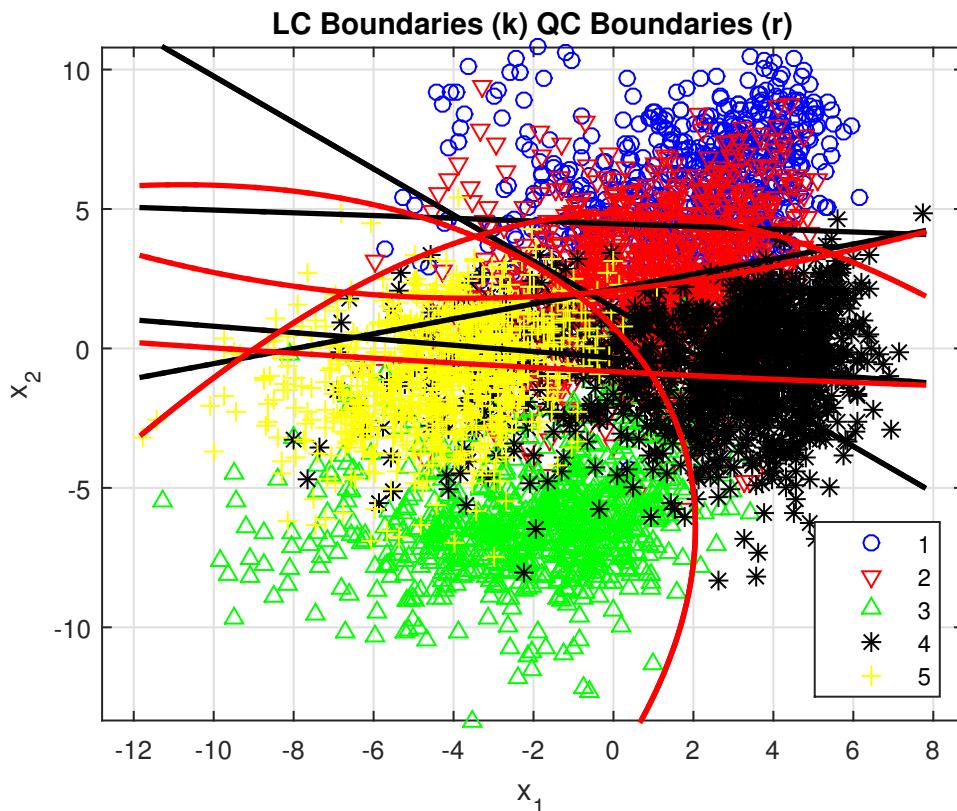


Figura 2: *Scatter plot* con las fronteras de clasificación obtenidas para la clase ‘aa’ respecto al resto de clases.

1.2.2. Par (22, 64)

Las probabilidades de error obtenidas en este caso se encuentran en la tabla 3.

Fase Clasificador \	Train	Test
Lineal (LC)	0,28998178506375227	0,28085106382978725
Cuadrático (QC)	0,2783242258652095	0,27148936170212767

Cuadro 3: Probabilidades del error de clasificación para el par (22, 64).

En la figura 3 se muestra el dibujo del scatter con las fronteras de clasificación.

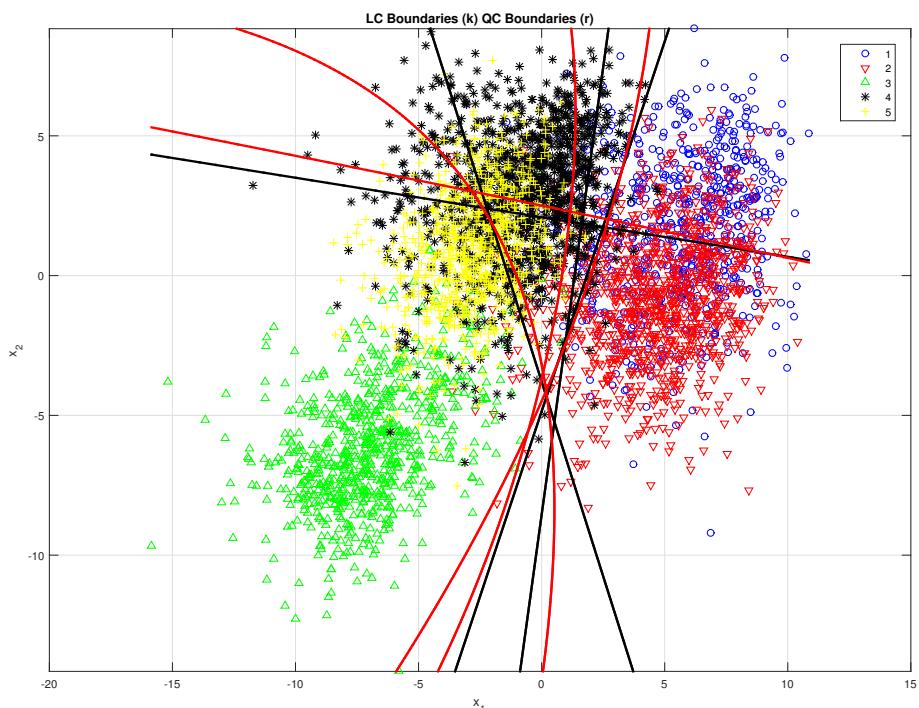


Figura 3: *Scatter plot* con las fronteras de clasificación obtenidas para la clase ‘aa’ respecto al resto de clases.

2. Reducción de dimensión mediante PCA

En las figuras 4 y 5 se muestran las gráficas de los errores de clasificación para 64 y 256 características, respectivamente.

La probabilidad de error del clasificador lineal se queda 'estancada' con menos características que el cuadrático, ya que tiene menos grados de libertad y requiere menos características para entrenar el clasificador correctamente.

El código que hace las proyecciones PCA se ha hecho con un bucle paralelo (**parfor**) de MATLAB, de manera que el tiempo de ejecución se reduce aproximadamente a razón del número de procesadores de que disponga el ordenador.

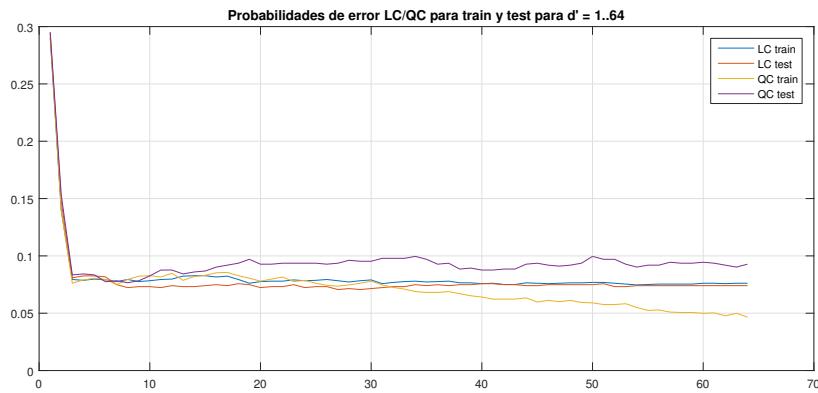


Figura 4: Gráfica de probabilidades de error con 64 características.

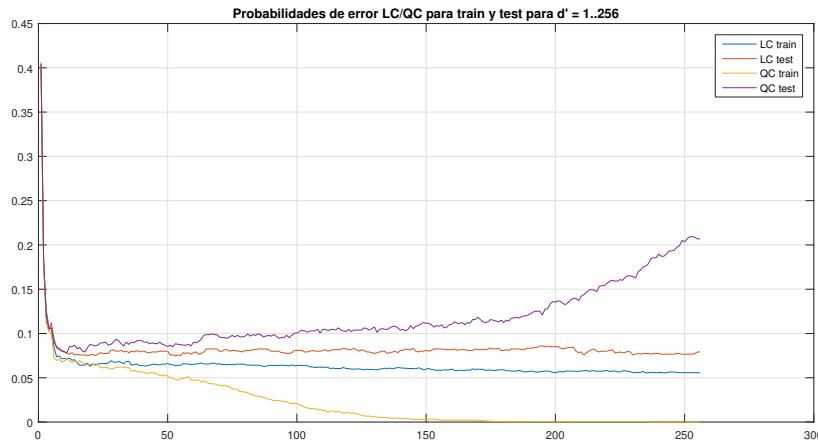


Figura 5: Gráfica de probabilidades de error con 256 características.

A continuación se incluye el código de la función `prac2_fonemas_PCA.m`

```

% prac2_fonemas_PCA.m
clear;
close all; % close all previous figures

%% Options / Initialitation
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(dc);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 = pca(X_train);
%% Data projection
X_train_proj = X_train * W;
X_test_proj = X_test * W;

LC_train_Pe = zeros(N_coor,1);
LC_test_Pe = zeros(N_coor,1);
QC_train_Pe = zeros(N_coor,1);
QC_test_Pe = zeros(N_coor,1);

% TODO Select d columns, compute error probabilities and plot graphics
tic
parfor d=1: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', N_coor));
grid on
legend('LC train', 'LC test' , 'QC train', 'QC test', 'best');

hold off
./prac2_fonemas_PCA.m

```