

Universidad Autónoma de Baja California

Instituto de Ingeniería

Maestría en Ciencias e Ingeniería



Tarea 2. Implementación de QDC y LDC

Materia: Modelos Inteligentes de Aprendizaje

presentado por

Emmanuel López Hernández

Director: Dr. Félix Fernando González Navarro

Mexicali, Baja California. Marzo 2020

Tabla de contenido

Figuras	2
Tablas.....	3
Introducción	4
Fundamento Teórico	4
Discriminante Lineal	4
Discriminante Cuadrático	5
Desarrollo	5
PARTE 1.....	5
PARTE 2.....	10
Resultados	14
PARTE 1.....	14
PARTE 2.....	16
Conclusiones.....	16
Bibliografía.....	17

Figuras

Figura 1. Teorema de Bayes (Rodrigo, 2016).	4
Figura 2.Desarrollo de LDC a partir del Teorema de Bayes (Rodrigo, 2016).	5
Figura 3. Desarrollo de QDC (Rodrigo, 2016).	5
Figura 4. Elección del porcentaje para training y test.	5
Figura 5. Obtención de labels de clase 1 y 2, así como sus índices.	6
Figura 6. Obtención de matrices de datos de training.	6
Figura 7. Clasificación a Priori.	6
Figura 8. Obtención de las matrices de covarianza de training.	6
Figura 9. Aplicación del “saneamiento” de las matrices de covarianza.	6
Figura 10. Calculo de etiquetas de test, así como del data set de test.	6
Figura 11. Calculo de QDC sin regularizar.	7
Figura 12. Cálculo de LDC sin regularizar.	7
Figura 13. Estimación de la mayor clase.	7
Figura 14. Obtención de la tasa de reconocimiento de QDC y LDC sin regularizar.	8
Figura 15. Regularización de QDC.	8
Figura 16. Clasificación de test para QDC.	8
Figura 17. Cálculo de la tasa de reconocimiento de QDC.	9
Figura 18. Se despliega la gráfica de la tasa de reconocimiento de QDC.	9
Figura 19. Regularización de LDC.	9
Figura 20. Clasificación de test para LDC.	9
Figura 21. Cálculo de la tasa de reconocimiento de LDC.	10
Figura 22. Se despliega la gráfica de la tasa de reconocimiento de LDC.	10
Figura 23. Tasa de reconocimiento de QDC y LDC sin regularizar.	10
Figura 24. Carpeta de “círculos_cuadrados”	10
Figura 25. Carpetas de test y training.	10
Figura 26. Imágenes dentro de la carpeta training.	11
Figura 27. Imágenes dentro de la carpeta test.	11
Figura 28. Carga de imagen y conversión a escala de grises y “double” para Training.	11
Figura 29. Clasificación de los archivos de training, en un vector de clases de 0 y 1.	11
Figura 30. Carga de imagen y conversión a escala de grises y “double” para Test.	12
Figura 31. Obtención de PCA a partir de la Matriz resultante de Training.	12
Figura 32. Producto de los Componentes Principales por la matriz resultante de Training y Test.	12
Figura 33. Clasificación de etiquetas e índices para training.	12
Figura 34. Clasificación a priori de training.	12
Figura 35. Separación de los datos de training en clases 0 y 1.	12
Figura 36. Obtención de la matriz de covarianza de todo el set de datos de training y su matriz saneada.	13
Figura 37. Cálculo de LDC por data set de la clase 0 y 1.	13
Figura 38. Clasificación de los resultados de LDC.	13
Figura 39.Obtención de la tasa de reconocimiento.	13
Figura 40. Se envía a pantalla si la imagen se trata de un Rectángulo o de un Óvalo.	14
Figura 41. Valores predichos vs resultados reales a graficar.	14
Figura 42. Tasa de Reconocimiento para QDC y LDC.	15
Figura 43. Tasa de reconocimiento de QDC y LDC sin regularizar,	15
Figura 44. Reconocimiento acertado de las imágenes.	16

Figura 45. Gráfica del reconocimiento de las imágenes..... 16

Tablas

Tabla 1. Tasa de Reconocimiento de QDC..... 14

Tabla 2. Tasa de Reconocimiento de LDC..... 15

Introducción

La presente investigación se desarrollará en dos partes:

PARTE UNO

Implementar los algoritmos de Discriminante Lineal y Discriminante Cuadrático. Para lo cual hay que:

1. Considerar el “saneamiento” de las matrices de covarianza utilizando el método de **Eigendecomposition**.
2. Implementar la regularización de las matrices de covarianza para el caso cuadrático y lineal.

Reportar los hallazgos en el presente documento de resultados en PDF que detalle lo realizado.

PARTE DOS

Implementar un sistema de reconocimiento de imágenes. La tarea será reconocer entre rectángulos y óvalos. Los datos de entrenamiento y prueba son “círculos_cuadrados.zip”.

1. Utilizar LDC.
2. Utilizar PCS como método de extracción de características.

Como nota, cabe recordar, que los cálculos y/o medidas que se hagan para training –i.e. matrices de covarianza, medias, etc.- deberán ser usados para preprocesar los datos de test. No se deben recalcular en los datos de test.

Fundamento Teórico

Discriminante Lineal

El Análisis Discriminante Lineal (LDA) es un método de clasificación supervisado donde intervienen variables cualitativas, donde dos o más grupos son conocidos a priori, y además, en uno de ellos, se clasifican nuevas observaciones en función de sus características; mediante el uso del Teorema de Bayes. En este análisis, se estima la probabilidad de que una observación, a partir del valor de los predictores, sea parte de cada una de las clases de la variable cualitativa, $P(Y=k|X=x)$. Asignando la observación a la clase k , para la que la probabilidad predicha es mayor (Rodrigo, 2016).

Presenta las siguientes ventajas en contraposición con la Regresión logística:

- Cuando las clases están muy separadas, en la Regresión logística los parámetros estimados se vuelven inestables, mientras que esto no sucede en LDA.
- Lo mismo sucede cuando si el número de observaciones es bajo y la distribución de los predictores es aproximadamente normal en cada clase, en LDA hay mayor estabilidad.

De acuerdo al Teorema de Bayes:

$$P(B|A) = \frac{P(AB)}{P(A)}$$

Figura 1. Teorema de Bayes (Rodrigo, 2016).

A partir de dos eventos A y B, se establece la probabilidad de que B ocurra habiendo ocurrido A: $P(B|A)$ es igual a la probabilidad de que A y B ocurran al mismo tiempo: $P(AB)/P(A)$.

Tomando como base este Teorema, se puede conocer la probabilidad a posteriori para cada clase; y por lo tanto obtener la siguiente ecuación:

$$P(Y = k|X = x) = \frac{\pi_k P(X = x|Y = k)}{\sum_{j=1}^K \pi_j P(X = x|Y = j)} = \frac{\pi_k f_k(x)}{\sum_{j=1}^K \pi_j f_j(x)}$$

Figura 2. Desarrollo de LDC a partir del Teorema de Bayes (Rodrigo, 2016).

Discriminante Cuadrático

El Análisis Discriminante Cuadrático (QDA) es parecido al LDA, con la diferencia de que QDA considera que cada clase k, tiene su propia matriz de covarianza (Σ_k), y por lo tanto, la función discriminante toma forma cuadrática (Rodrigo, 2016):

$$\log(P(Y = k|X = x)) = -\frac{1}{2} \log|\Sigma_k| - \frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log(\pi_k)$$

Figura 3. Desarrollo de QDC (Rodrigo, 2016).

De manera, que para calcular la probabilidad a posteriori, es necesario estimar, para cada clase, (Σ_k), μ_k y π_k a partir de la muestra; y cada nueva observación se clasifica en la clase para la que el valor de la probabilidad a posteriori sea mayor.

Desarrollo

PARTE 1

Para el desarrollo de la presente investigación, en lo que corresponde a la primera parte, se utilizaron los datos de Wisconsin para la implementación de la Regresión Cuadrática y la Regresión Lineal.

```
%Elige el porcentaje que desees.
trn_input=input('Seleccione el porcentaje: (1)70-30, (2)80-20, (3)90-10: ');
if trn_input == 1
    percent=0.7;
elseif trn_input == 2
    percent=0.8;
else
    percent=0.9;
end
```

Figura 4. Elección del porcentaje para training y test.

Al ejecutar el programa, se pregunta al usuario qué porcentaje de training y test se desea, para las presentes pruebas se utilizó el porcentaje de 80% de training y 20% de test.

Mediante cálculos, se obtienen los vectores de clase 1 y 2 para los datos de training, así como sus índices:

```
%labels trn values and indexes
label_tr_c1 = label_training(label_training==1);
lbl_tr_c1_indx = find(label_training==1);

label_tr_c2 = label_training(label_training==2);
lbl_tr_c2_indx = find(label_training==2);
```

Figura 5. Obtención de labels de clase 1 y 2, así como sus índices.

También se obtiene la matriz de datos de training y se separa en otras dos matrices, una para clase 1 y la otra para clase 2:

```
%matrices de datos de training
training_set = datos((1:training_percent), :);

tr_data_c1 = training_set(lbl_tr_c1_indx, :);
tr_data_c2 = training_set(lbl_tr_c2_indx, :);
```

Figura 6. Obtención de matrices de datos de training.

Se realiza la clasificación a priori para c1 y c2:

```
%Clasificación a priori
class_priori_1 = length(label_tr_c1)/length(label_training);
class_priori_2 = length(label_tr_c2)/length(label_training);
```

Figura 7. Clasificación a Priori.

Se obtiene la matriz de covarianza del set de training, así como las matrices de covarianza de las sub matrices para clase 1 y clase 2 de training:

```
%matriz de covarianza de todo el dataset de training
sigma = cov(training_set);

%matrices de covarianza c1 y c2
sigma_c1 = cov(tr_data_c1);
sigma_c2 = cov(tr_data_c2);
```

Figura 8. Obtención de las matrices de covarianza de training.

Mediante una función creada en matlab llamada “sanear()”, se sanean los valores negativos y se reconstruye la matriz.

```
%% Se sanean los valores negativos, y se reconstruye la matriz. %%
ssaneado = sanear(sigma);
ssaneado_c1 = sanear(sigma_c1);
ssaneado_c2 = sanear(sigma_c2);
```

Figura 9. Aplicación del “saneamiento” de las matrices de covarianza.

Se calculan también las etiquetas de test, así como el data set para test:

```
%% TEST %%
label_test = labels((training_percent+1:end), :);
test_percent = sizerow - training_percent;
test_set = datos((training_percent+1:end), :);
```

Figura 10. Calculo de etiquetas de test, así como del data set de test.

Se realiza el cálculo de QDC para la matriz de test, utilizando los valores de las matrices saneadas de covarianza de training de clase 1 y 2, así como los valores de las medias de c1 y c2 para training:

```
for i = 1:testlen
    X = test_set(i, :);
    QDC_c1(i) = -0.5 * X * E_inv_C1 * X' + M_C1 * E_inv_C1 * X' - 0.5 * M_C1 * E_inv_C1 * M_C1' - 0.5 * log(det(E_C1)) + log(class_priori_1);
    QDC_c2(i) = -0.5 * X * E_inv_C2 * X' + M_C2 * E_inv_C2 * X' - 0.5 * M_C2 * E_inv_C2 * M_C2' - 0.5 * log(det(E_C2)) + log(class_priori_2);
end
```

Figura 11. Cálculo de QDC sin regularizar.

Donde:

- M_C1 y M_C2. Son las medias de las matrices de clase 1 y clase 2, respectivamente.
- E_inv_C1 y E_inv_C2. Son las inversas de las matrices saneadas de clase 1 y 2.
- E_C1 y E_C2. Son las matrices saneadas de clase 1 y 2.
- class_priori_1 y class_priori_2, es la probabilidad de las clases a priori.

Además, se calcula LDC:

```
for i = 1:testlen
    X = test_set(i, :);
    LDC_c1(i) = M_C1 * inv(ssaneado) * X' - 0.5 * M_C1 * inv(ssaneado) * M_C1' + log(class_priori_1);
    LDC_c2(i) = M_C2 * inv(ssaneado) * X' - 0.5 * M_C2 * inv(ssaneado) * M_C2' + log(class_priori_2);
end
```

Figura 12. Cálculo de LDC sin regularizar.

Donde:

- M_C1 y M_C2. Son las medias de las matrices de clase 1 y clase 2, respectivamente.
- inv(ssaneado). Es la inversa de la matriz completa saneada de training.
- ssaneado. Es la matriz completa saneada de training.
- class_priori_1 y class_priori_2, es la probabilidad de las clases a priori.

En este punto, los valores de clase 1, se convertirán en clase 0; y los valores de clase 2, se convertirán en clase 1.

Con esto, se procede a calcular cuál clase es mayor, si QDC_c1 > QDC_c2, será clase 0, de lo contrario, será clase 1; lo mismo para LDC, se compara si LDC_c1 > LDC_c2, será clase 0, de lo contrario, será clase 1.

```
%Una vez que se calcula QDC y DLC se estima cual clase es mayor
lbl_test_QDC = [];
lbl_test_LDC = [];
for m=1:length(QDC_c1)
    if QDC_c1(m) > QDC_c2(m)
        lbl_test_QDC(m)=0;
    else
        lbl_test_QDC(m)=1;
    end

    if LDC_c1(m) > LDC_c2(m)
        lbl_test_LDC(m)=0;
    else
        lbl_test_LDC(m)=1;
    end
end
```

Figura 13. Estimación de la mayor clase.

Esto es, para poder comparar contra la clasificación de test y ver qué tan acertada es la tasa de reconocimiento tanto de QDC como de LDC.

Se calcula la tasa de reconocimiento de QDC y LDC sin regularizar.

```
%Tasa de reconocimiento QDC y LDC, sin regularización
positiveLDC = [];
positiveQDC = [];
labelTst = label_test';
labelTst(labelTst == 1) = 0;
labelTst(labelTst == 2) = 1;
TstTasaQDC = [];
TstTasaLDC = [];

for a = 1:length(lbl_test_QDC)
    if lbl_test_QDC(1, a) == labelTst(1, a)
        positiveQDC(a)=1;
    end
    if lbl_test_LDC(1, a) == labelTst(1, a)
        positiveLDC(a)=1;
    end
end
TstTasaQDC = (sum(positiveQDC)/length(lbl_test_QDC)) * 100;
TstTasaLDC = (sum(positiveLDC)/length(lbl_test_LDC)) * 100;
```

Figura 14. Obtención de la tasa de reconocimiento de QDC y LDC sin regularizar.

Ahora, se procede a Regularizar en QDC para clase 1 y clase 2, tomando como valor el rango de la variable “alpha”, que va desde 0 hasta 1, en intervalos de 0.1.

```
for a = 1: length(alpha)
    QDC_Reg_C1 = alpha(a) * ssaneado_c1 + (1 - alpha(a)) * ssaneado;
    QDC_Reg_C2 = alpha(a) * ssaneado_c2 + (1 - alpha(a)) * ssaneado;

    for i = 1:testlen
        X = test_set(i, :);
        QDC_c1_reg(i) = -0.5 * X * inv(QDC_Reg_C1) * X' + M_C1 * inv(QDC_Reg_C1) * X' - 0.5 * M_C1 * inv(QDC_Reg_C1) * M_C1';
        QDC_c2_reg(i) = -0.5 * X * inv(QDC_Reg_C2) * X' + M_C2 * inv(QDC_Reg_C2) * X' - 0.5 * M_C2 * inv(QDC_Reg_C2) * M_C2';
    end

    QDC_c1_end(a, :) = QDC_c1_reg;
    QDC_c2_end(a, :) = QDC_c2_reg;
end
```

Figura 15. Regularización de QDC.

Se clasifica QDC con los valores de test, pero en vez de 1 y 2, serán 0 y 1 respectivamente.

```
%Clasificación de test regularizado para QDC
for v = 1:length(alpha)
    for m=1:length(QDC_c1_reg)

        if QDC_c1_end(v, m)>QDC_c2_end(v, m)
            reg_QDC(m)=0;
        else
            reg_QDC(m)=1;
        end
    end
    lbl_reg_QDC(v, :) = reg_QDC;
end
```

Figura 16. Clasificación de test para QDC.

Se calcula la tasa de reconocimiento de la matriz de QDC:

```

for b = 1:length(alpha)
    for a = 1:length(lbl_reg_QDC)
        if lbl_reg_QDC(b, a) == lblQDCBinary(1, a)
            aciertosQDC(a)=1;
        end
    end
    tasa_QDC(b) = (sum(aciertosQDC)/length(lbl_reg_QDC)) * 100;
    aciertosQDC = [];
end

```

Figura 17. Cálculo de la tasa de reconocimiento de QDC.

Por último, se grafica la tasa de reconocimiento de la matriz de QDC:

```

%% GRÁFICAS %%
subplot(2, 1, 1);
plot(trans_alpha(:, 1), tasa_QDC, 'r -o');hold on;
xlabel('Valor de Alpha');
ylabel('% de reconocimiento');
title('Tasa de Reconocimiento QDC');

```

Figura 18. Se despliega la gráfica de la tasa de reconocimiento de QDC.

Se procede también a Regularizar en LDC para clase 1 y clase 2, tomando como valor el rango de la variable “gamma”, que va desde 0 hasta 1, en intervalos de 0.1.

```

for b = 1: length(gama)
    LDC_Reg = gama(b) * ssaneado + (1 - gama(b)) * (std(training_set).^2);

    for i = 1:testlen
        X = test_set(i, :);
        LDC_c1_reg(i) = M_C1 * inv(LDC_Reg) * X' -0.5 * M_C1 * inv(LDC_Reg) * M_C1' + log(class_priori_1);
        LDC_c2_reg(i) = M_C2 * inv(LDC_Reg) * X' -0.5 * M_C2 * inv(LDC_Reg) * M_C2' + log(class_priori_2);
    end
    LDC_c1_end(b, :) = LDC_c1_reg;
    LDC_c2_end(b, :) = LDC_c2_reg;
end

```

Figura 19. Regularización de LDC.

Se clasifica LDC con los valores de test, pero en vez de 1 y 2, serán 0 y 1 respectivamente.

```

%Clasificación de test regularizado para LDC
for v = 1:length(gama)
    for m=1:length(LDC_c1_reg)

        if LDC_c1_end(v, m)>LDC_c2_end(v, m)
            reg_LDC(m)=0;
        else
            reg_LDC(m)=1;
        end
    end
    lbl_reg_LDC(v, :) = reg_LDC;
end

```

Figura 20. Clasificación de test para LDC.

Se calcula la tasa de reconocimiento de la matriz de LDC:

```

for b = 1:length(gama)
    for a = 1:length(lbl_reg_LDC)
        if lbl_reg_LDC(b, a) == lblLDCBinary(1, a)
            aciertosLDC(a)=1;
        end
    end
    tasa_LDC(b) = (sum(aciertosLDC)/length(lbl_reg_LDC)) * 100;
    aciertosLDC = [];
end

```

Figura 21. Cálculo de la tasa de reconocimiento de LDC.

Por último, se grafica la tasa de reconocimiento de la matriz de LDC:

```

subplot(2, 1, 2);
plot(trans_gamma(:, 1), tasa_LDC, 'b -o');hold on;
xlabel('Valor de Gamma');
ylabel('% de reconocimiento');
title('Tasa de Reconocimiento LDC');
hold off

```

Figura 22. Se despliega la gráfica de la tasa de reconocimiento de LDC.

En este punto, también se despliega en pantalla la tasa de reconocimiento de QDC y de LDC sin regularizar.

```

disp("La tasa de reconocimiento sin regularizar de QDC es de: " + TstTasaQDC);
disp("La tasa de reconocimiento sin regularizar de LDC es de: " + TstTasaLDC);

```

Figura 23. Tasa de reconocimiento de QDC y LDC sin regularizar.

PARTE 2

Para el desarrollo de esta actividad, se utilizan las imágenes de la carpeta de “círculos_cuadrados”, en donde las imágenes que tienen como prefijo “c”, son datos de training que para fines de la presente prueba, equivalen a rectángulos; mientras que las imágenes con prefijo “o”, son datos de training que equivalen a óvalos; por otra parte, las imágenes con prefijo “t”, pertenecen a los datos de test, en ellas hay tanto rectángulos como óvalos.

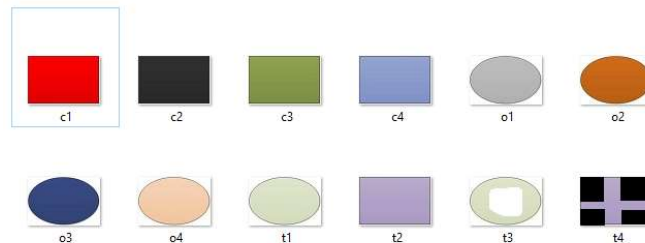


Figura 24. Carpeta de “círculos_cuadrados”

Se han separado de manera manual las imágenes correspondientes a test y training, y se han dispuesto en las siguientes carpetas:

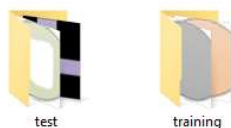


Figura 25. Carpetas de test y training.

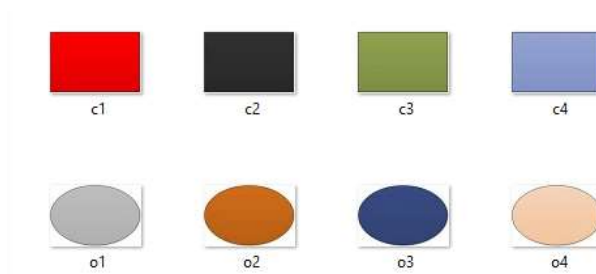


Figura 26. Imágenes dentro de la carpeta training.



Figura 27. Imágenes dentro de la carpeta test.

En el código se procede a extraer las imágenes de la carpeta de training, y convertirlas a escala de grises, para poder utilizarlas mejor, después se convierten en tipo “double”. También se obtienen los nombres de las imágenes dentro de la carpeta “training”, con el fin de poder transformar las que empiecen con “c” (rectángulos) en valores de 0, y las que empiecen con la letra “o” (óvalos) en 1, y almacenar en un vector de clases.

```
%% TRAINING %%
As = imageDatastore('circulos_cuadrados/training');
matrixA=[];
matrixv = [];
v=[];
for i = 1:size(As.Files,1)
    img = readimage(As,i);
    img = rgb2gray(img); %%Conversión de la imagen RGB a escala de grises.
    img= im2double(img);
    s = size(img);
    S = s(1)*s(2);
    str=strjoin(As.Files(i));
    fileTrn(i, :) = strsplit(str, '\'); %Para MacOS este signo debe ser '/'
    V = reshape(img,1,S);
    matrixA=[matrixA;V];
end
```

Figura 28. Carga de imagen y conversión a escala de grises y “double” para Training.

```
v=zeros(size(matrixA,1),1);
v(v==0)=1;
nomFilesTrn = [fileTrn(:, end)];
firstLetter = extractBefore(nomFilesTrn, 2);
betaClases = strrep(firstLetter, 'c', '0');
trnClases = strrep(betaClases, 'o', '1');
clases = str2double(trnClases);
```

Figura 29. Clasificación de los archivos de training, en un vector de clases de 0 y 1.

Se extraen ahora los datos de Test, y de igual forma, se convierten las imágenes en escala de grises, y en “double”, después se obtienen los nombres de las imágenes, los cuales servirán para mostrar en los resultados, de qué tipo de imagen se trata, de acuerdo a la predicción.

```

%% TEST %%
test = imageDatastore('circulos_cuadrados/test');
for i = 1:size(test.Files,1)
    img_test=readimage(test,i);
    img_test = rgb2gray(img_test); %%Conversión de la imagen RGB a escala de grises.
    img_test= im2double(img_test);
    s = size(img_test);
    S = s(1)*s(2);
    str=strjoin(test.Files(i));
    fileTst(i, :)=strsplit(str,'\'); %%Para MacOS este signo debe ser '/'
    V = reshape(img_test,1,S);
    matrixv=[matrixv;V];
end

```

Figura 30. Carga de imagen y conversión a escala de grises y “double” para Test.

Se aplica PCA a la matriz resultante de Training, para obtener los componentes principales.

```

nomFilesTst = [fileTst(:, end)];
trnX = pca(matrixA);

```

Figura 31. Obtención de PCA a partir de la Matriz resultante de Training.

También se multiplica el resultado de PCA por cada una de las matrices: de Training y de Test.

```

%Training set y Test set respectivamente
prodTrn = matrixA * trnX;
prodTst = matrixv * trnX;

```

Figura 32. Producto de los Componentes Principales por la matriz resultante de Training y Test.

Se obtienen las etiquetas de clases y los índices para training, para clase 0 y 1.

```

%labels trn values and indexes

label_tr_c1 = clases(clases==0);
lbl_tr_c1_indx = find(clases==0);

label_tr_c2 = clases(clases==1);
lbl_tr_c2_indx = find(clases==1);

```

Figura 33. Clasificación de etiquetas e índices para training.

Después se calcula la probabilidad a priori de las clases 0 y 1, así como la obtención de los datos de training separados en clases, y la matriz de covarianza y su saneamiento.

```

%Clasificación a priori
class_priori_1 = length(label_tr_c1)/length(clases);
class_priori_2 = length(label_tr_c2)/length(clases);

```

Figura 34. Clasificación a priori de training.

```

%Training data para c1 y c2
tr_data_c1 = prodTrn(lbl_tr_c1_indx, :);
tr_data_c2 = prodTrn(lbl_tr_c2_indx, :);

```

Figura 35. Separación de los datos de training en clases 0 y 1.

```

%matriz de covarianza de todo el dataset de training
sigma = cov(prodTrn);

%% Se sanean los valores negativos, y se reconstruye la matriz. %%
ssaneado = sanear(sigma);

```

Figura 36. Obtención de la matriz de covarianza de todo el set de datos de training y su matriz saneada.

Se calcula el LDC con los datos de Test, las medias de training clasificadas, y las matrices saneadas. Después se clasifican los resultados de LDC, para ambos resultados de LDC, y se determina si “LDC_c1 > LDC_c2”, se clasifica como 0, de otra manera, se pone 1.

```

%% Se calcula LDC por dataset_c1 y dataset_c2 %%
LDC_c1 = [];
LDC_c2 = [];
testlen = length(prodTst(:, 1));
M_C1 = mean(tr_data_c1);
M_C2 = mean(tr_data_c2);

for i = 1:testlen
    X = prodTst(i, :);
    LDC_c1(i) = M_C1 * inv(ssaneado) * X' - 0.5 * M_C1 * inv(ssaneado) * M_C1' + log(class_priori_1);
    LDC_c2(i) = M_C2 * inv(ssaneado) * X' - 0.5 * M_C2 * inv(ssaneado) * M_C2' + log(class_priori_2);
end

```

Figura 37. Cálculo de LDC por data set de la clase 0 y 1.

```

%Clasificación de test
lbl_test_LDC = [];
for m=1:length(LDC_c1)
    if LDC_c1(m) > LDC_c2(m)
        lbl_test_LDC(m)=0;
    else
        lbl_test_LDC(m)=1;
    end
end

```

Figura 38. Clasificación de los resultados de LDC.

En este punto se obtiene la tasa de reconocimiento, y se envía a pantalla el resultado de la predicción, arrojando mensajes si el archivo tal, de la carpeta de “test” corresponde a un rectángulo o a un óvalo.

```

%Tasa de reconocimiento
test_real = [1 0 1 0];
aciertos = [];
for a = 1:length(lbl_test_LDC)
    if lbl_test_LDC(a) == test_real(a)
        aciertos(a)=1;
    end
end

```

Figura 39. Obtención de la tasa de reconocimiento.


```

tasa_rec = (length(aciertos)/length(lbl_test_LDC)) * 100;
disp("La tasa de reconocimiento es de " + tasa_rec + "%");

rec_circulo = [];
for n= 1: length(lbl_test_LDC)
    if lbl_test_LDC(n) == 0
        rec_circulo = "Rectángulo";
    else
        rec_circulo = "Óvalo";
    end
    disp(nomFilesTst(n) + " es un " + rec_circulo);
end

```

Figura 40. Se envía a pantalla si la imagen se trata de un Rectángulo o de un Óvalo.

Por último, se grafican los resultados predichos contra los resultados reales.

```

%% GRÁFICAS %%
subplot(2, 1, 1);
plot(lbl_test_LDC, 'r-o');hold on;
title('Valores reconocidos');
subplot(2, 1, 2);

plot(test_real, 'b-o');hold on;
title('Valores reales');
hold off

```

Figura 41. Valores predichos vs resultados reales a graficar.

Resultados

PARTE 1

Con los resultados de la tasa de reconocimiento de QDC, se puede observar el porcentaje de reconocimiento respecto a cada uno de los valores que toma Alpha, siendo 0 con el que más se acerca al 100%.

Tabla 1. Tasa de Reconocimiento de QDC

Valor de Alpha	Porcentaje de Reconocimiento
0	97.3684 %
0.1	95.6140 %
0.2	93.8596 %
0.3	93.8596 %
0.4	93.8596 %
0.5	92.9825 %
0.6	92.1053 %
0.7	92.1053 %
0.8	92.1053 %
0.9	94.7368 %
1	95.6140 %

Con los resultados de la tasa de reconocimiento de LDC, se puede observar el porcentaje de reconocimiento respecto a cada uno de los valores que toma Gamma, siendo 1 con el que más se

acerca al 100%. Con esto se comprueba que el porcentaje de reconocimiento de Alpha cuando es 0, es igual al porcentaje de reconocimiento de Gamma cuando es 1.

Tabla 2. Tasa de Reconocimiento de LDC

Valor de Gamma	Porcentaje de Reconocimiento
0	77.1930 %
0.1	92.1053 %
0.2	91.2281 %
0.3	91.2281 %
0.4	91.2281 %
0.5	91.2281 %
0.6	91.2281 %
0.7	91.2281 %
0.8	90.3509 %
0.9	90.3509 %
1	97.3684 %

Los resultados de las tablas anteriores pueden ser expresados gráficamente de la siguiente manera:

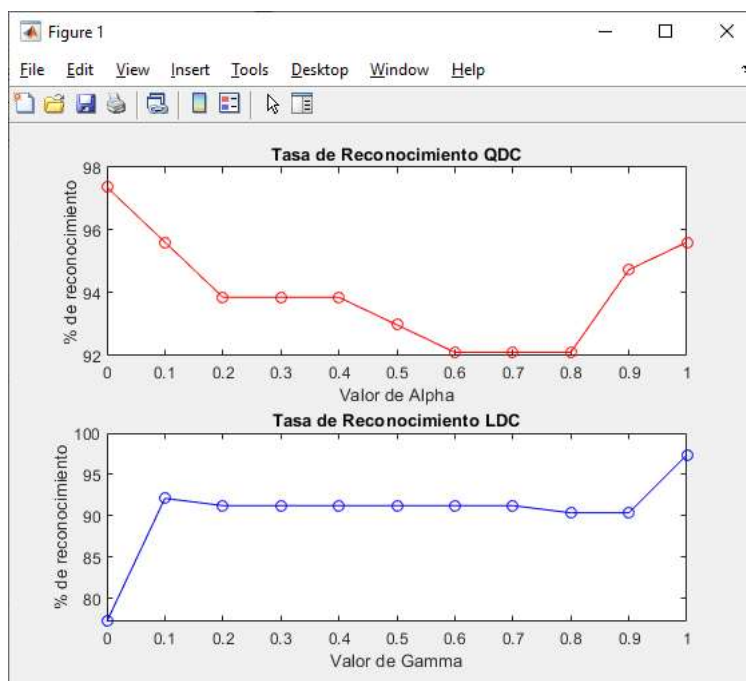


Figura 42. Tasa de Reconocimiento para QDC y LDC.

Mientras tanto, en consola se muestra el resultado del cálculo de las tasas de reconocimiento de QDC y LDC sin regularizar:

```
La tasa de reconocimiento sin regularizar de QDC es de: 95.614|
La tasa de reconocimiento sin regularizar de LDC es de: 97.3684
```

Figura 43. Tasa de reconocimiento de QDC y LDC sin regularizar,

PARTE 2

Para los resultados de la parte 2, se obtuvo que al ejecutar LDC (sin regularizar), se obtuvo una tasa de reconocimiento del 100%, ya que todas las figuras de la carpeta de “test” fueron identificadas correctamente:

```
La tasa de reconocimiento es de 100%
t1.jpg es un Óvalo
t2.jpg es un Rectángulo
t3.jpg es un Óvalo
t4.jpg es un Rectángulo
fx >>
```

Figura 44. Reconocimiento acertado de las imágenes.

Al graficar los valores predichos contra los valores reales, se observa que coinciden al 100%:

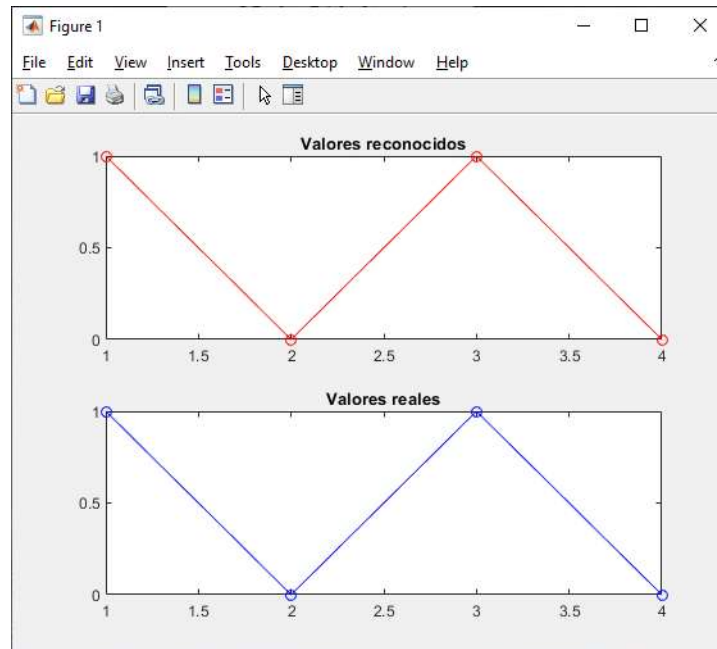


Figura 45. Gráfica del reconocimiento de las imágenes.

Conclusiones

PARTE 1

- Se ha comprobado que los mayores índices de aproximación del valor predicho con respecto al valor conocido de test, se dan cuando, para QDC regularizado, alpha vale 0; y para cuando LDC regularizado, gamma vale 1. De hecho, bajo estas condiciones, con alpha 0 y gamma 1, se obtiene el mismo porcentaje de desempeño.
- Los valores de la matriz de LDC regularizada, cuando gamma vale 0, son valores NaN, por los que al clasificarse, como LDC_c1_end y LDC_c2_end tienen el mismo valor, son clasificados como 1, esto es debido a la condición especificada: $LDC_c1_end(v, m) > LDC_c2_end(v, m)$; lo mismo sucede con los valores de QDC y LDC (regularizados y no regularizados) que ni uno ni otro es mayor, sino que ambos son iguales.
- Se ha comprobado que, sin regularizar, la mayor tasa de reconocimiento se da con LDC, que coincide con el porcentaje que se da en alpha 0 y gamma 1, es decir, 97.3684 %.

PARTE 2

- Con PCA se han podido extraer los componentes principales de training, con los cuales se calculan las nuevas matrices para Training y Test.
- Se comprueba que con LDC, se logra una excelente tasa de reconocimiento para este caso en concreto.

Bibliografía

Rodrigo, J. A. (2016). *www.cienciadedatos.net*. Obtenido de https://www.cienciadedatos.net/documentos/28_linear_discriminant_analysis_lda_y_quadratic_discriminant_analysis_qda