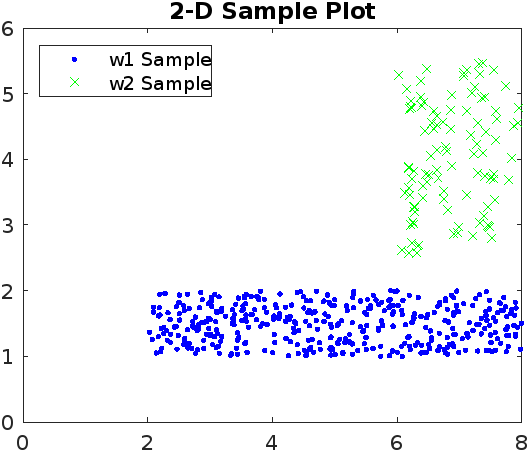
**Υπολογιστική Εργασία (Matlab)**

**Αναγνώριση Προτύπων**

Ονοματεπώνυμο: Γαρουφαλή Μαρία Νεφέλη (3129)

Στο 1ο μέρος πρέπει να κατασκευάσουμε τα δεδομένα που θα χρησιμοποιήσουμε στην εργασία. Επιλέγουμε την συνάρτηση *rand* που κατασκευάζει ομοιόμορφα κατανεμημένα δεδομένα. Τέλος, σχεδιάζουμε τις δύο κλάσεις.



Για το 2ο μέρος της εργασίας, αρχικά υπολογίζουμε βασικά μεγέθη που προκύπτουν από τις δύο κλάσεις και θα μας χρειαστούν στην συνέχεια. Αυτά είναι:

* Oι μέσες τιμές των δύο κλάσεων: **m1**, **m2**
* Τα μητρώα συνδιασποράς των δύο κλάσεων: **s1**, **s2**
* Συναρτήσεις πιθανοφάνειας: **p(11)**, **p(12)** , **p(21)** , **p(22)**

Ταξινομητής Ευκλείδειας Απόστασης

Στο συγκεκριμένο ερώτημα μας ζητείται να ταξινομήσουμε τα δεδομένα των δύο κλάσεων βάσει της απόστασής τους από το μέσο διάνυσμα της κάθε κλάσης. Για τον σκοπό αυτόν φτιάχνουμε τέσσερεις πίνακες στους οποίους θα αποθηκεύσουμε τα καλώς και τα κακώς ταξινομημένα δεδομένα της κάθε κλάσης, ώστε να μπορούμε να τα μετρήσουμε και να συμπεράνουμε το ποσοστό λάθους του αλγορίθμου. Όσον αφορά την διαδικασία ταξινόμησης, επαναληπτικά υπολογίζουμε την ευκλείδεια απόσταση κάθε στοιχείου της κάθε κλάσης από τα δύο μέσα διανύσματα και βάσει της μικρότερης κατατάσσουμε το στοιχείο στην κλάση ω1 ή την κλάση ω2.

ΣΤΑΤΙΣΤΙΚΑ

Το ποσοστό λάθους του συγκεκριμένου αλγορίθμου ταξινόμησης για τα συγκεκριμένα δεδομένα μετρήθηκε 6.4%.

Ταξινομητής απόστασης Mahalanobis

Στο συγκεκριμένο ερώτημα μας ζητείται να ταξινομήσουμε τα δεδομένα χρησιμοποιώντας την απόσταση Mahalanobis και κοινό πίνακα συνδιασποράς για τις δύο κλάσεις, που προκύπτει ως ο **σταθμισμένος μέσος**  των επιμέρους πινάκων συνδιασποράς. Η διαδικασία είναι παρόμοια με πριν. Επαναληπτικά υπολογίζουμε την απόσταση Mahalanobis για κάθε σημείο από τις δύο κλάσεις.

ΣΤΑΤΙΣΤΙΚΑ

Το ποσοστό λάθους του συγκεκριμένου αλγορίθμου για τα συγκεκριμένα δεδομένα μετρήθηκε 1.2%, σαφώς καλύτερο από την ευκλείδεια απόσταση.

Bayesian ταξινομητής

Στο συγκεκριμένο ερώτημα μας ζητείται η ταξινόμηση των στοιχείων με τον Bayesian ταξινομητή, χρησιμοποιώντας δεδομένα που υπολογίσαμε στην αρχή του δεύτερου μέρους και τους πίνακες συναδιασποράς ξεχωριστά για κάθε κλάση.

ΣΤΑΤΙΣΤΙΚΑ

Το ποσοστό λάθους του συγκεκριμένου αλγορίθμου για τα συγκεκριμένα δεδομένα μετρήθηκε 0%. Το αποτέλεσμα αυτό μας δείχνει ότι ο Bayesian ταξινομητής διαχωρίζει σωστά τα δεδομένα.

ΣΧΟΛΙΑ

Παρατηρήσαμε πως η απόσταση Mahalanobis είναι πιο ακριβής στον διαχωρισμό των δεδομένων από ότι είναι η Ευκλείδεια απόσταση. Αυτό συμβαίνει διότι, σε αντίθεση με την Ευκλείδεια, η απόσταση Mahalanobis λαμβάνει υπόψιν της και τον συσχετισμό των δεδομένων. Η μόνη περίπτωση που προτιμάμε την Ευκλείδεια απόσταση είναι όταν δεν υπάρχει συσχετισμός μεταξύ των δεδομένων, και αυτό χάριν ευκολίας του υπολογισμού. Ακόμα και σε αυτήν την περίπτωση όμως, η απόσταση Mahalanobis θα δώσει το ίδιο αποτέλεσμα με την Ευκλείδεια απόσταση.

Μια ακόμα παρατήρηση που κάναμε ήταν η συγκριτικά καλύτερη επίδοση του Bayesian ταξινομητή και από τις δύο παραπάνω μεθόδους. Για τον υπολογισμό των συναρτήσεων πιθανοφάνειας χρησιμοποιήσαμε το **maximum likelihood estimation**. Η ML εκτίμηση έχει μερικές πολύ επιθυμητές ιδιότητες που ενδεχομένως εξηγούν την παρατήρησή μας.

Αρχικά, η εκτίμηση μέγιστης πιθανοφάνειας είναι ασυμπτωτικά αμερόληπτη, δηλαδή η εκτίμηση συγκλίνει, ως προς την μέση τιμή της, στην πραγματική τιμή.

Δεύτερον, η εκτίμηση μέγιστης πιθανοφάνειας είναι ασυμπτωτικά συνεπής, δηλαδή η εκτίμηση συγκλίνει ως προς την πιθανότητα . Με άλλα λόγια για μεγάλο N, η διασπορά εκτίμησης τείνει προς το 0.

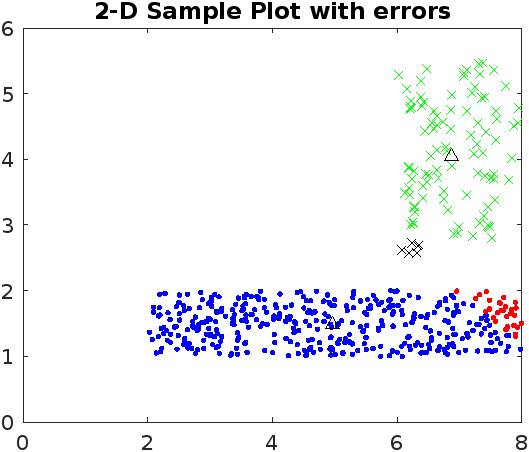
Έπειτα, η εκτίμηση μέγιστης πιθανοφάνειας είναι ασυμπτωτικά αποτελεσματική, δηλαδή επιτυγχάνει το κάτω φράγμα Cramer-Rao.

Συνοψίζοντας, η **maximum likelihood estimation**, για μεγάλες τιμές του Ν (όπως και στην περίπτωσή μας), είναι αμερόληπτη, κανονικά κατανεμημένη και έχει την ελάχιστη δυνατή διασπορά.

Στο διάγραμμα που ακολουθεί απεικονίζονται όλα τα ερωτήματα του Β μέρους. Συγκεκριμένα, φαίνονται οι δύο κλάσεις καθώς και οι αστοχίες των ταξινομητών Ευκλείδειας απόστασης και Mahalanobis.

Υπόμνημα

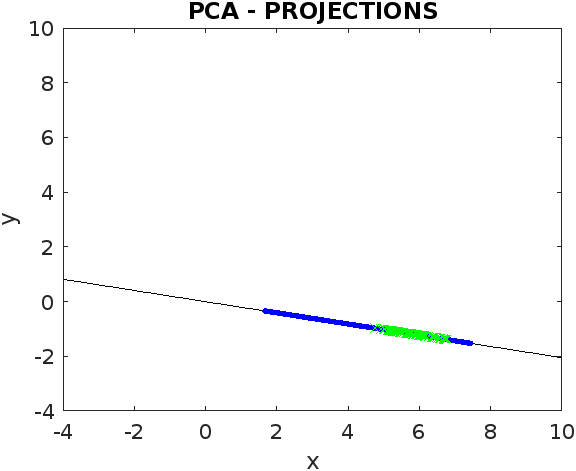
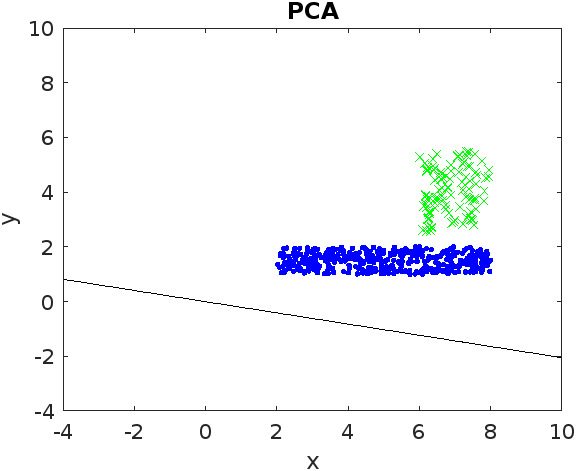
|  |  |
| --- | --- |
| ● | Κλάση ω1 |
| x | Κλάση ω2 |
| Δ | Μέσα διανύσματα κλάσεων |
| Κόκκινο χρώμα | Λάθη ευκλείδειας ταξινόμησης |
| Μαύρο χρώμα | Λάθη Mahalanobis ταξινόμησης |



Στο 3ο μέρος της εργασίας θα προβάλλουμε τα δεδομένα μας στον μονοδιάστατο χώρο χρησιμοποιώντας τους αλγορίθμους PCA, LDA. Στην συνέχεια θα τα διαχωρίσουμε με το κριτήριο της ευκλείδειας απόστασης.

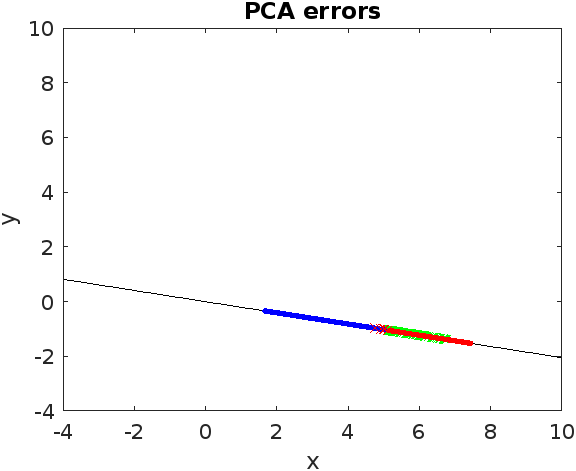
PCA

Στο πρώτο ερώτημα μας ζητείται να βρούμε την μονοδιάστατη προβολή των δεδομένων βάσει του αλγορίθμου PCA. Για τον σκοπό αυτόν, υπολογίζουμε τον Covariance Matrix αφού, κατά πάσα πιθανότητα ο μέσος όρος των δειγματικών σημείων θα είναι διάφορος του μηδενός. Ύστερα βρίσκουμε την μέγιστη ιδιοτιμή του και βάσει αυτής, την ευθεία προβολής των δεδομένων.



Παρατηρούμε πώς στον μονοδιάστατο χώρο προβολής υπάρχει επικάλυψη των δεδομένων. Αυτό σημαίνει ότι ο αλγόριθμος δεν πρόκειται να διαχωρίσει σωστά τα δεδομένα.

Στο επόμενο ερώτημα μας ζητείται ο διαχωρισμός των δεδομένων με τον ταξινομητή Ευκλείδειας απόστασης και ο υπολογισμός του σφάλματος ταξινόμησης. Αρχικά θα υπολογίσουμε τα μέσα διανύσματα των δύο κλάσεων στα projected δεδομένα. Έπειτα, επαναληπτικά θα υπολογίσουμε την Ευκλείδεια απόσταση για κάθε σημείο και θα το κατατάξουμε στην κλάση που είναι πιο κοντά.



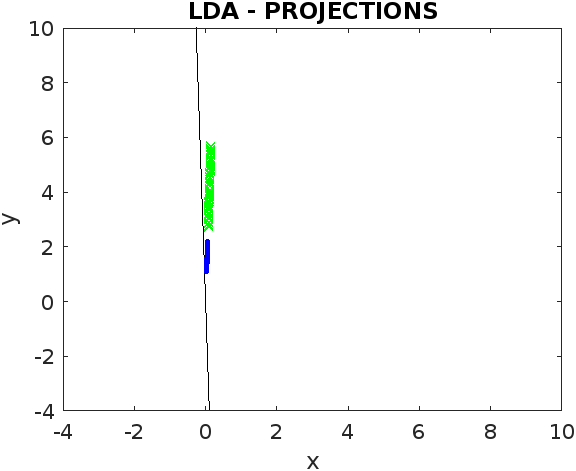
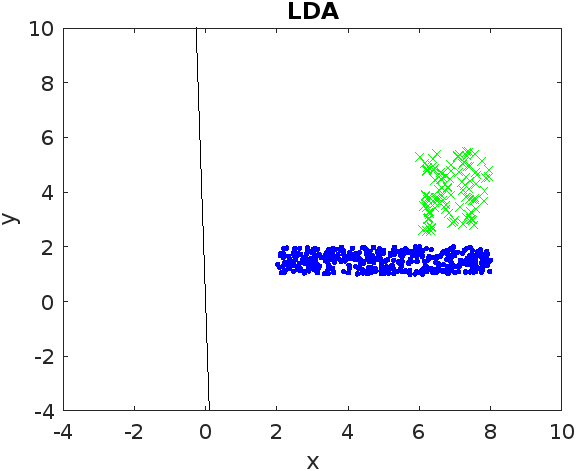
Στο παραπάνω διάγραμμα φαίνονται με κόκκινο χρώμα οι αστοχίες της Ευκλείδειας ταξινόμησης στα projected δεδομένα από τον PCA.

ΣΤΑΤΙΣΤΙΚΑ

Το σφάλμα ταξινόμησης της συγκεκριμένης διαδικασίας στα συγκεκριμένα δεδομένα μετρήθηκε 31.6%.

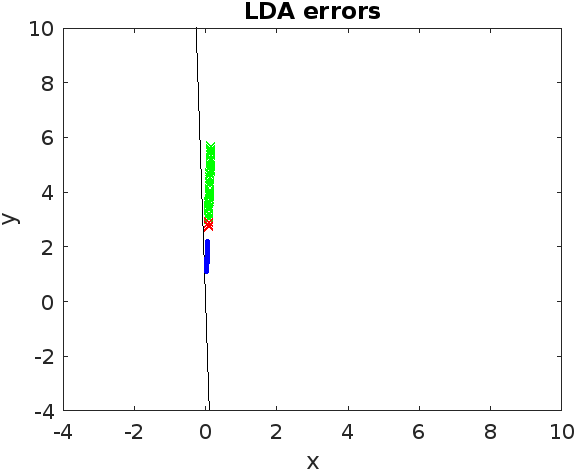
LDA

Στο τρίτο ερώτημα μας ζητείται να βρούμε την μονοδιάστατη προβολή των δεδομένων βάσει του αλγορίθμου LDA. Έτσι υπολογίζουμε την μέση τιμή των δεδομένων για κάθε κλάση ξεχωριστά. Στη συνέχεια βρίσκουμε τους πίνακες Σ1 και Σ2 και με κατάλληλους υπολογισμούς το διάνυσμα προβολής. Από αυτό αντλούμε την ευθεία προβολής.



Παρατηρούμε πώς στον μονοδιάστατο χώρο προβολής δεν υπάρχει επικάλυψη των δεδομένων. Αυτό σημαίνει πως υπάρχει η δυνατότητα να διαχωριστούν τα δεδομένα, ακόμα και πλήρως.

Στο επόμενο ερώτημα μας ζητείται ο διαχωρισμός των δεδομένων με τον ταξινομητή Ευκλείδειας απόστασης και ο υπολογισμός του σφάλματος ταξινόμησης. Αρχικά θα υπολογίσουμε τα μέσα διανύσματα των δύο κλάσεων στα projected δεδομένα. Έπειτα, επαναληπτικά θα υπολογίσουμε την Ευκλείδεια απόσταση για κάθε σημείο και θα το κατατάξουμε στην κλάση που είναι πιο κοντά.



Στο παραπάνω διάγραμμα φαίνονται με κόκκινο χρώμα οι αστοχίες της Ευκλείδειας ταξινόμησης στα projected δεδομένα από τον LDA.

ΣΤΑΤΙΣΤΙΚΑ

Το σφάλμα ταξινόμησης της συγκεκριμένης διαδικασίας στα συγκεκριμένα δεδομένα μετρήθηκε 1.2%.

ΣΧΟΛΙΑ

Εύκολα παρατηρούμε ότι ο LDA στα συγκεκριμένα δεδομένα ήταν πολύ αποδοτικότερος σε σχέση με τον PCA. Αναλύοντας τις ιδιότητες του κάθε αλγορίθμου θα καταλάβουμε γιατί.

Ο PCA είναι μια μέθοδος γραμμικού μετασχηματισμού των δεδομένων χωρίς επίβλεψη. Ο συγκεκριμένος αλγόριθμος αντιμετωπίζει ολόκληρο το σύνολο των δεδομένων ως μία κλάση. Στόχος του είναι να μειώσει τον αριθμό των διαστάσεων της κλάσης, βρίσκοντας την μέγιστη διακύμανση των δεδομένων. Ουσιαστικά δηλαδή, ο PCA ψάχνει τον άξονα προβολής όπου τα δεδομένα θα είναι περισσότερο διασκορπισμένα.

Ο LDA από την άλλη είναι μια μέθοδος με επίβλεψη, στόχος της οποίας είναι η εύρεση ενός υποχώρου με την μέγιστη δυνατότητα διαχωρισμού των δεδομένων. Σε αντίθεση με τον PCA ο LDA έχει ένα βήμα «προεπεξεργασίας» στο οποίο υπολογίζει τα μέσα διανύσματα της κάθε κλάσης καθώς και τους πίνακες συνδιασποράς.

Καταλαβαίνουμε λοιπόν ότι ο LDA είναι πιο χρήσιμος και αποτελεσματικότερος αν θέλουμε να διαχωρίσουμε τα δεδομένα .

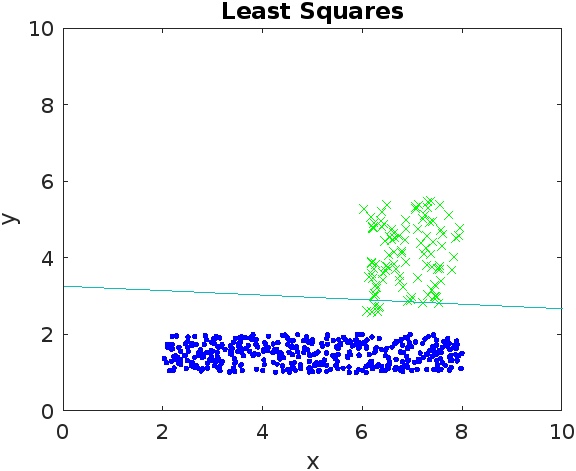
Τέλος, αξίζει να αναφέρουμε ότι ο PCA και ο LDA δεν είναι αλγόριθμοι ταξινόμησης, αλλά αλγόριθμοι μείωσης των διαστάσεων των δεδομένων. Έτσι λοιπόν, έχει σημασία και ο αλγόριθμος ταξινόμησης που επιλέγεται μετά την δράση του PCA ή του LDA.

Αυτό το παρατηρήσαμε άλλωστε και στο παράδειγμά μας όπου ο LDA κατάφερε να διαχωρίσει πλήρως τις δύο κλάσεις, αλλά ο αλγόριθμος ταξινόμησης που επιλέχθηκε για να τις ταξινομήσει είχε σφάλμα. (Ευκλείδεια Απόσταση)

Στο 4ο μέρος της εργασίας ζητείται να υλοποιήσουμε γραμμικούς ταξινομητές στον δισδιάστατο χώρο, με βάση τους αλγορίθμους Ελαχίστων Τετραγώνων και Perceptron.

ΑΛΓΟΡΙΘΜΟΣ ΕΛΑΧΙΣΤΩΝ ΤΕΤΡΑΓΩΝΩΝ

Αρχικά κατασκευάζουμε τους κατάλληλους πίνακες, δηλαδή τον πίνακα **X**, και τον πίνακα **Y**. Στην συνέχεια, κάνοντας κατάλληλες πράξεις καταλήγουμε στο διάνυσμα **w**, το οποίο αναπαριστά την ευθεία διαχωρισμού.



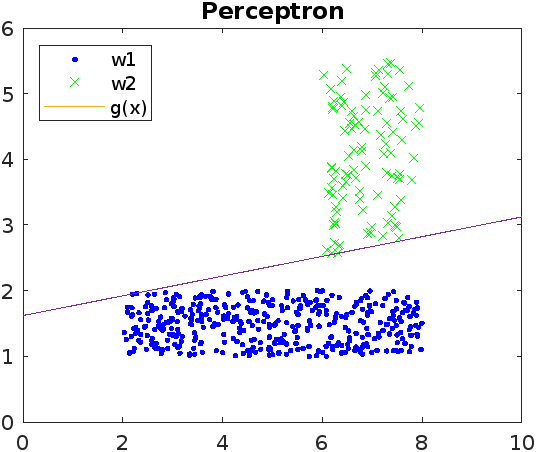
Στο παραπάνω σχήμα φαίνεται η ευθεία που προκύπτει, καθώς και πώς αυτή διαχωρίζει τα δεδομένα εκπαίδευσης.

ΣΤΑΤΙΣΤΙΚΑ

Για τον γραμμικό αυτόν ταξινομητή με αυτά τα δεδομένα, το σφάλμα ταξινόμησης μετρήθηκε 1.6%, ενώ το σφάλμα τετραγώνων μετρήθηκε 350.0643.

PERCEPTRON

Το πρώτο μας βήμα είναι να προσδιορίσουμε το αρχικό διάνυσμα το οποίο δεν ταξινομεί σωστά τα δεδομένα εκπαίδευσης **w**, καθώς και τον συντελεστή **ρ**. Συγκεκριμένα ως αρχικό διάνυσμα επιλέχθηκε το **w=[1; 0; 7]** και **ρ=1/50**. Στην συνέχεια, επαναληπτικά ελέγχουμε αν τα στοιχεία μας ταξινομούνται σωστά με το συγκεκριμένο w. Όσο αυτά παραμένουν αταξινόμητα, «διορθώνουμε» το w μέχρι αυτό να ταξινομεί σωστά όλα τα δεδομένα μας.



Στο παρακάτω σχήμα φαίνεται η ευθεία διαχωρισμού που προέκυψε μετά από **124 επαναλήψεις** για το συγκεκριμένο αρχικό διάνυσμα και την συγκεκριμένη παράμετρο ρυθμού εκμάθησης. Εφόσον τα δεδομένα μας είναι γραμμικά διαχωρίσιμα, ο perceptron είναι εγγυημένο ότι θα συγκλίνει. Έτσι το σφάλμα ταξινόμησης του αλγορίθμου είναι 0%, δηλαδή δεν υπάρχει σφάλμα.

ΚΩΔΙΚΑΣ ΥΛΟΠΟΙΗΣΗΣ (Matlab)

close all

clc;clear;

rng default

%create w1 samples

point\_a = 2; point\_b = 8;

point\_c = 1; point\_d= 2;

x1 = point\_a + (point\_b-point\_a).\*rand(400,1);

x2 = point\_c + (point\_d-point\_c).\*rand(400,1);

w1 = [x1 ,x2];

%create w2 samples

point\_a = 6; point\_b = 8;

point\_c = 2.5; point\_d= 5.5;

x1 = point\_a + (point\_b-point\_a).\*rand(100,1);

x2 = point\_c + (point\_d-point\_c).\*rand(100,1);

w2 = [x1 ,x2];

%plot samples

figure(1)

plot(w1(:,1),w1(:,2),'b.', w2(:,1),w2(:,2),'gx', 0,0);

title('2-D Sample Plot');

legend('w1 Sample','w2 Sample','Location','northwest');

%Meros B

%B1

%find mean values of samples

m1 = mean(w1);

m2 = mean(w2);

%find covariances

s1 = cov(w1);

s2 = cov(w2);

p11 = [];

p12 = [];

for i = 1:400

p11 = [p11,(1/(2\*pi\*sqrt(abs(det(s1))))) \* exp(-(1/2)\*(w1(i,:) - m1)\*inv(s1)\*(w1(i,:) - m1)')];

p12 = [p12,(1/(2\*pi\*sqrt(abs(det(s2))))) \* exp(-(1/2)\*(w1(i,:) - m2)\*inv(s2)\*(w1(i,:) - m2)')];

end

p21 = [];

p22 = [];

for i = 1:100

p22 = [p22,(1/(2\*pi\*sqrt(abs(det(s2))))) \* exp(-(1/2)\*(w2(i,:) - m2)\*inv(s2)\*(w2(i,:) - m2)')];

p21 = [p21,(1/(2\*pi\*sqrt(abs(det(s1))))) \* exp(-(1/2)\*(w2(i,:) - m1)\*inv(s1)\*(w2(i,:) - m1)')];

end

%B2

w1\_right = [];

w2\_right = [];

w1\_wrong = [];

w2\_wrong = [];

%Taksinomisi stoixeiwn W1 me euclidean distance

for i = 1:400

d1 = norm(w1(i,:) - m1);

d2 = norm(w1(i,:) - m2);

if d1<=d2

w1\_right = [w1\_right;w1(i,:)];

else

w1\_wrong = [w1\_wrong;w1(i,:)];

end

end

%Taksinomisi stoixeiwn W2 me euclidean distance

for i = 1:100

d1 = norm(w2(i,:) - m1);

d2 = norm(w2(i,:) - m2);

if d1<d2

w2\_wrong = [w2\_wrong;w2(i,:)];

else

w2\_right = [w2\_right;w2(i,:)];

end

end

%Ypologismos sfalmatos taksinomisis

error = (size(w1\_wrong,1) + size(w2\_wrong,1)) / 500;

disp(['Euclidean distance error : ',num2str(error\*100),'%']);

%Plot dedomena klasewn & lathi euclidean taksinomisis

figure(2)

plot(w1(:,1),w1(:,2),'b.',w2(:,1),w2(:,2),'gx', 0,0);

if size(w1\_wrong,1) ~= 0

hold on;

plot(w1\_wrong(:,1),w1\_wrong(:,2),'r.', 0,0);

end

if size(w2\_wrong,1) ~= 0

hold on;

plot(w2\_wrong(:,1),w2\_wrong(:,2),'rx', 0,0);

end

% Plot mesa dianysmata klasewn

plot(m1(:,1), m1(:,2), 'k^', 0,0);

hold on;

plot(m2(:,1), m2(:,2), 'k^', 0,0);

%B3

s = (0.8\*s1)+(0.2\*s2);

w1\_right = [];

w2\_right = [];

w1\_wrong = [];

w2\_wrong = [];

%Taksinomisi stoixeiwn W1 me mahalanobis distance

for i = 1:400

d1 = sqrt((w1(i,:) - m1)\*inv(s)\*(w1(i,:) - m1)');

d2 = sqrt((w1(i,:) - m2)\*inv(s)\*(w1(i,:) - m2)');

if d1<=d2

w1\_right = [w1\_right;w1(i,:)];

else

w1\_wrong = [w1\_wrong;w1(i,:)];

end

end

%Taksinomisi stoixeiwn W2 me mahalanobis distance

for i = 1:100

d1 = sqrt((w2(i,:) - m1)\*inv(s)\*(w2(i,:) - m1)');

d2 = sqrt((w2(i,:) - m2)\*inv(s)\*(w2(i,:) - m2)');

if d1<d2

w2\_wrong = [w2\_wrong;w2(i,:)];

else

w2\_right = [w2\_right;w2(i,:)];

end

end

%Ypologismos sfalmatos mahalanobis distance

error = (size(w1\_wrong,1) + size(w2\_wrong,1)) / 500;

disp(['Mahalanobis distance error : ',num2str(error\*100),'%']);

%Plot lathi mahalanobis distance

if size(w1\_wrong,1) ~= 0

%hold on;

plot(w1\_wrong(:,1),w1\_wrong(:,2),'k.', 0,0);

end

if size(w2\_wrong,1) ~= 0

%hold on;

plot(w2\_wrong(:,1),w2\_wrong(:,2),'kx', 0,0);

end

title('2-D Sample Plot with errors');

legend('w1 Sample','w2 Sample','Euc. errors w1','Euc. errors w2', 'Mah. erros w1','Mah. erros w2','Location','northwest');

%B4

w1\_right = [];

w2\_right = [];

w1\_wrong = [];

w2\_wrong = [];

for i = 1:400

if p11>=p12

w1\_right = [w1\_right;w1(i,:)];

else

w1\_wrong = [w1\_wrong;w1(i,:)];

end

end

for i = 1:100

if p21>p22

w2\_wrong = [w2\_wrong;w2(i,:)];

else

w2\_right = [w2\_right;w2(i,:)];

end

end

error = (size(w1\_wrong,1) + size(w2\_wrong,1)) / 500;

disp(['Bayesian classification error : ',num2str(error\*100),'%']);

% Meros C

%C1

figure(3);

projectedPCAw1 = w1;

projectedPCAw2 = w2;

%Pinakas Dedomenwn

M = [w1;w2];

%Covariance Pinaka

Matrix = cov(M);

%Idiotimes

eigen = eig(Matrix);

%Max idiotimi

eigen\_max = max(eigen);

colV = [1;1];

colV = eigen\_max\*colV;

%Evresi eytheias diaxwrismou

z = Matrix\colV;

syms x y;

% Plot eytheias diaxwrismou & deigmatwn

f1(x,y) = z(1,:)\*x + z(2,:)\*y;

h = ezplot(f1,[-4,10]);

set(h, 'Color', 'k');

hold on

plot(w1(:,1),w1(:,2),'b.');

hold on;

plot(w2(:,1),w2(:,2),'gx');

title('PCA');

% Plot eytheias diaxwrismou

figure

f1(x,y) = z(1,:)\*x + z(2,:)\*y;

h = ezplot(f1,[-4,10]);

set(h, 'Color', 'k');

hold on

slope = -z(1,:)/z(2,:);

%Evresi projected points gia W1

for i = 1:400

b0 = -(-1/slope)\*w1(i,1) + w1(i,2);

xIntersection = (b0)/ (slope - (-1/slope));

yIntersection = (-1/slope) \* xIntersection + b0;

projectedPCAw1(i,1)=xIntersection;

projectedPCAw1(i,2)=yIntersection;

plot(xIntersection, yIntersection, 'b.')

end

%Evresi projected points gia W2

for i = 1:100

b0 = -(-1/slope)\*w2(i,1) + w2(i,2);

xIntersection = (b0)/ (slope - (-1/slope));

yIntersection = (-1/slope) \* xIntersection + b0;

projectedPCAw2(i,1)=xIntersection;

projectedPCAw2(i,2)=yIntersection;

plot(xIntersection, yIntersection, 'gx')

end

title('PCA - PROJECTIONS')

%C2

w1\_wrong = [];

w1\_right = [];

w2\_wrong = [];

w2\_right = [];

mesos\_projectedPCA\_w1 = mean(projectedPCAw1);

mesos\_projectedPCA\_w2 = mean(projectedPCAw2);

figure

%Plot eytheias diaxwrismou

f1(x,y) = z(1,:)\*x + z(2,:)\*y;

h = ezplot(f1,[-4,10]);

set(h, 'Color', 'k');

hold on

%Evresi projected points gia W1

for i = 1:400

b0 = -(-1/slope)\*w1(i,1) + w1(i,2);

xIntersection = (b0)/ (slope - (-1/slope));

yIntersection = (-1/slope) \* xIntersection + b0;

projectedPCAw1(i,1)=xIntersection;

projectedPCAw1(i,2)=yIntersection;

plot(xIntersection, yIntersection, 'b.')

%Taksinomisi projected points

d1 = norm(projectedPCAw1(i,:) - mesos\_projectedPCA\_w1);

d2 = norm(projectedPCAw1(i,:) - mesos\_projectedPCA\_w2);

if d1<=d2

w1\_right = [w1\_right;projectedPCAw1(i,:)];

else

w1\_wrong = [w1\_wrong;projectedPCAw1(i,:)];

end

end

%Evresi projected points gia W2

for i = 1:100

b0 = -(-1/slope)\*w2(i,1) + w2(i,2);

xIntersection = (b0)/ (slope - (-1/slope));

yIntersection = (-1/slope) \* xIntersection + b0;

projectedPCAw2(i,1)=xIntersection;

projectedPCAw2(i,2)=yIntersection;

plot(xIntersection, yIntersection, 'gx')

%Taksinomisi projected points

d1 = norm(projectedPCAw2(i,:) - mesos\_projectedPCA\_w1);

d2 = norm(projectedPCAw2(i,:) - mesos\_projectedPCA\_w2);

if d1<d2

w2\_wrong = [w2\_wrong;projectedPCAw2(i,:)];

else

w2\_right = [w2\_right;projectedPCAw2(i,:)];

end

end

%Plot PCA me lathi taksinomisis

title('PCA errors');

if size(w1\_wrong,1) ~= 0

hold on;

plot(w1\_wrong(:,1),w1\_wrong(:,2),'r.', 0,0);

end

if size(w2\_wrong,1) ~= 0

hold on;

plot(w2\_wrong(:,1),w2\_wrong(:,2),'rx', 0,0);

end

disp(['Pososto lathous PCA : ',num2str( ((size(w1\_wrong,1)+size(w2\_wrong,1))/500)\*100 ),'%']);

%C3

figure

% P(W1) & P(W2)

Pithanotita\_W1 = 400/500;

Pithanotita\_W2 = 100/500;

% Pinakes dedomenwn

projectedLDAw1 = w1;

projectedLDAw2 = w2;

Pinakas\_1\_lda = w1-m1;

Pinakas\_2\_lda = w2-m2;

Pinakas\_1\_lda = Pithanotita\_W1\*(transpose(Pinakas\_1\_lda)\*Pinakas\_1\_lda);

Pinakas\_2\_lda = Pithanotita\_W2\*(transpose(Pinakas\_2\_lda)\*Pinakas\_2\_lda);

Sw = Pithanotita\_W1\*Pinakas\_1\_lda+Pithanotita\_W2\*Pinakas\_2\_lda;

wproj = inv(Sw)\*(transpose(m1)-transpose(m2));

slope\_lda = -wproj(1,:)/wproj(2,:);

slope2 = -wproj(1,:)/wproj(2,:);

syms x y;

%Plot LDA eytheias diaxwrismou & klasewn

f3(x,y) = (-1/slope2)\*x + y;

h = ezplot(f3,[-4,10]);

set(h, 'Color', 'k');

hold on

plot(w1(:,1),w1(:,2),'b.');

hold on;

plot(w2(:,1),w2(:,2),'gx');

title('LDA');

%Plot eytheias diaxwrismou

figure

f3(x,y) = (-1/slope2)\*x + y;

h = ezplot(f3,[-4,10]);

set(h, 'Color', 'k');

hold on

%Evresi projected points gia W1

for i = 1:400

b0 = -(slope2)\*w1(i,1) + w1(i,2);

xIntersection = -(b0)/ ((slope2)-(-1/slope2));

yIntersection = (slope2) \* xIntersection + b0;

projectedLDAw1(i,1)=xIntersection;

projectedLDAw1(i,2)=yIntersection;

plot(xIntersection, yIntersection, 'b.')

end

%Evresi projected points gia W2

for i = 1:100

b0 = -(slope2)\*w2(i,1) + w2(i,2);

xIntersection = -(b0)/ ((slope2)-(-1/slope2));

yIntersection = (slope2) \* xIntersection + b0;

projectedLDAw2(i,1)=xIntersection;

projectedLDAw2(i,2)=yIntersection;

plot(xIntersection, yIntersection, 'gx')

end

title('LDA - PROJECTIONS')

%C4

w1\_wrong = [];

w1\_right = [];

w2\_wrong = [];

w2\_right = [];

mesos\_projectedLDA\_w1 = mean(projectedLDAw1);

mesos\_projectedLDA\_w2 = mean(projectedLDAw2);

figure

% Plot eytheia diaxwrismou

f3(x,y) = -(1/slope2)\*x + y;

h = ezplot(f3,[-4,10]);

set(h, 'Color', 'k');

hold on

% Evresi projected points W1

for i = 1:400

b0 = -(slope2)\*w1(i,1) + w1(i,2);

xIntersection = -(b0)/ ((slope2)-(-1/slope2));

yIntersection = (slope2) \* xIntersection + b0;

projectedLDAw1(i,1)=xIntersection;

projectedLDAw1(i,2)=yIntersection;

plot(xIntersection, yIntersection, 'b.')

% Taksinomisi projected points

d1 = norm(projectedLDAw1(i,:) - mesos\_projectedLDA\_w1);

d2 = norm(projectedLDAw1(i,:) - mesos\_projectedLDA\_w2);

if d1<=d2

w1\_right = [w1\_right;projectedLDAw1(i,:)];

else

w1\_wrong = [w1\_wrong;projectedLDAw1(i,:)];

end

end

%Evresi projected points W2

for i = 1:100

b0 = -(slope2)\*w2(i,1) + w2(i,2);

xIntersection = -(b0)/ ((slope2)- (-1/slope2));

yIntersection = (slope2) \* xIntersection + b0;

projectedLDAw2(i,1)=xIntersection;

projectedLDAw2(i,2)=yIntersection;

plot(xIntersection, yIntersection, 'gx')

%Taksinomisi projected points

d1 = norm(projectedLDAw2(i,:) - mesos\_projectedLDA\_w1);

d2 = norm(projectedLDAw2(i,:) - mesos\_projectedLDA\_w2);

if d1<d2

w2\_wrong = [w2\_wrong;projectedLDAw2(i,:)];

else

w2\_right = [w2\_right;projectedLDAw2(i,:)];

end

end

%Plot me lathi taksinomisis

title('LDA errors');

if size(w1\_wrong,1) ~= 0

hold on;

plot(w1\_wrong(:,1),w1\_wrong(:,2),'r.', 0,0);

end

if size(w2\_wrong,1) ~= 0

hold on;

plot(w2\_wrong(:,1),w2\_wrong(:,2),'rx', 0,0);

end

disp(['Pososto lathous LDA : ',num2str( ((size(w1\_wrong,1)+size(w2\_wrong,1))/500)\*100 ),'%']);

% Meros D

%D1 Minimum Least Squares

y = ones(500,1);

y(401:500) = -y(401:500);

X = ones(500,3);

X(:,1:2) = [w1;w2];

w = inv(X'\*X)\*X'\*y;

%Plot eytheias diaxwrismou & klasewn

figure;

plot(w1(:,1),w1(:,2),'b.',w2(:,1),w2(:,2),'gx', 0,0);

hold on;

syms x y;

f(x,y) = w(1)\*x + w(2)\*y + w(3);

ezplot(f,[0,10]);

title('Least Squares');

%Evresi sfalmatos tetragwnwn

J = [];

y = ones(500,1);

for i = 1:500

J = [J, (y(i) - X(i,:)\*w)^2];

end

lse = sum(J);

disp(['Minimum least Square, Sfalma Tetragwnwn: ',num2str(lse)]);

%Evresi sfalmatos taksinomisis

errors = 0;

for i = 1:400

if w'\*[w1(i,:),1]'<=0

errors = errors +1;

end

end

for i = 1:100

if w'\*[w2(i,:),1]' >= 0

errors = errors +1;

end

end

disp(['Minimum least Square, Error: ',num2str( (errors/500)\*100 ),'%']);

%D2 Perceptron

w = [1; 0; 7];

r = 1/50;

y = ones(500,1);

y(401:500) = -y(401:500);

X = [w1;w2];

X = [X,ones(500,1)];

Y = [];

e = 0;

a = 1;

syms x y;

%Epanaliptiko loop taksinomisi twn dedomenwn

%kai epanaprosdiorismou toy w

while length(Y)|| a ~= 0

a = 0;

Y = [];

for i = 1:500

if i<=400

if X(i,:)\*w<=0

Y = [Y;1\*X(i,:)];

end

else

if X(i,:)\*w>=0

Y = [Y;(-1)\*X(i,:)];

end

end

end

e = e +1;

sm = [];

for i = 1:size(Y,1)

sm = [sm ;Y(i,:)];

end

a = sum(sm);

w = w +r\*a';

end

%Plot eytheias diaxwrismou & klasewn

f(x,y) = w(1)\*x + w(2)\*y+w(3);

figure

plot(w1(:,1),w1(:,2),'b.',w2(:,1),w2(:,2),'gx', 0,0);

hold on;

xx = linspace(0,10,1000);

plot(xx,(xx\*w(1)+w(3))/(-w(2)), 0,0)

legend('w1','w2','g(x)','Location', 'northwest');

title('Perceptron');

disp(['Number of Iterations : ',num2str(e)]);