ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΠΑΝΕΠΙΣΤΗΜΙΟΥ ΠΑΤΡΩΝ

ΕΡΓΑΣΤΗΡΙΑΚΕΣ ΑΣΚΗΣΕΙΣ

ΑΝΑΓΝΩΡΙΣΗΣ ΠΡΟΤΥΠΩΝ Ι

ΒΑΓΓΕΛΗΣ ΔΕΡΜΑΤΑΣ
ΕΠΙΚΟΥΡΟΣ ΚΑΘΗΓΗΤΗΣ

ΠΑΤΡΑ ΜΑΡΤΗΣ 2003

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ MATLAB

1. Πίνακες

Α. Δημιουργία Πινάκων

Ο πιο εύκολος τρόπος για την δημιουργία ενός πίνακα αριθμών στην MATLAB, είναι εισάγοντας τα στοιχεία του χωριζόμενα με κενά ή με κόμματα, μέσα σε αγκύλες και χρησιμοποιώντας το ελληνικό ερωτηματικό (;) για την δήλωση του τέλους της κάθε γραμμής του. Για παράδειγμα, εισάγοντας την εντολή :

```
A = [1 2 3 ; 4 5 6 ; 7 8 0]
```

το αποτέλεσμα στην έξοδο θα είναι : $A = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 0 \end{matrix}$

Το κάθε στοιχείο ενός πίνακα μπορεί να προσπελαστεί κάνοντας χρήση της συγκεκριμένης στήλης και γραμμής που κατέχει αυτό. Για παράδειγμα το στοιχείο του πιο πάνω πίνακα *A* που βρίσκεται στην δεύτερη γραμμή και τρίτη στήλη αναφέρεται ως A(2,3) και είναι ίσο με 6 :

```
A(2,3) = 6
```

Β. Πράξεις Πινάκων

Ανάστροφος
Ο ειδικός χαρακτήρας (`′`) υποδηλώνει τον ανάστροφο ενός πίνακα. Η δήλωση :
`B = A′` (όπου A ο πιο πάνω πίνακας) έχει ως αποτέλεσμα:

$$B = \begin{matrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 0 \end{matrix}$$

και ο `x = [ -1 0 2 ] ′`

έχει ως αποτέλεσμα $x' = \begin{matrix} -1 \\ 0 \\ 2 \end{matrix}$

Αντίστροφος
Ο υπολογισμός του αντίστροφου ενός πίνακα στην MATLAB γίνεται με τη βοήθεια της εντολής `inv()`.
Για παράδειγμα ο αντίστροφος του πίνακα A δίνεται από την εντολή `inv(A)`.

Πρόσθεση και αφαίρεση Πινάκων
Η πρόσθεση και η αφαίρεση πινάκων γίνεται με τα σύμβολα (+) και (-) . Συγκεκριμένα η δήλωση :

```
C = A + B
```

έχει ως αποτέλεσμα τον πίνακα $C = \begin{matrix} 2 & 6 & 10 \\ 6 & 10 & 14 \\ 10 & 14 & 0 \end{matrix}$

Πρόσθεση και αφαίρεση υφίσταται και μεταξύ πινάκων και αριθμών για παράδειγμα ενός πίνακα 1x1. Σε αυτή την περίπτωση ο αριθμός αυτός προστίθεται ή αφαιρείται σε / από κάθε στοιχείο του πίνακα Έτσι το :

```
y = x - 1
```

δίνει : $y = \begin{matrix} -2 \\ -1 \\ 1 \end{matrix}$

Πολλαπλασιασμός και διαίρεση Πινάκων
Η πράξη του πολλαπλασιασμού στους πίνακες γίνεται με το (*). Δηλαδή ο πολλαπλασιασμός του πίνακα A με τον B δηλώνεται στην MATLAB με την :

```
C = A * B
```

Ομοίως και για την διαίρεση των πινάκων με το σύμβολο (/) .

```
C = A / B
```

Υπάρχουν επίσης και οι πράξεις (.*) και (./) οι οποίες δηλώνουν πολλαπλασιασμό και διαίρεση αντίστοιχα μεταξύ των στοιχείων των πινάκων. Για παράδειγμα αν :

```
x = [ 1 2 3 ] ;  y = [ 4 5 6 ] ;
```

τότε η πράξη

```
z = x  .*  y
```

έχει αποτέλεσμα z = 4 10 18

Ομοίως το αποτέλεσμα της πράξης `z = x ./ y`

είναι z = 0.25 0.40 0.50

Απομόνωση γραμμών και στηλών
Η απομόνωση των γραμμών ή στηλών ενός πίνακα γίνεται απλά με τον τελεστή εύρους (**:**). Για παράδειγμα έστω ότι έχουμε τον πίνακα *A* και θέλουμε να εξάγουμε την πρώτη στήλη. Αυτό γίνεται με την εντολή `A(:,1)` ή με την `A(1:3,1)`. Ομοίως για την απομόνωση της δεύτερης γραμμής του ίδιου πίνακα μπορούμε να χρησιμοποιήσουμε την εντολή `A(2,:)` ή την `A(2,1:3)`.

2. Αρχεία

Στην MATLAB για την προσπέλαση αρχείων χρησιμοποιούνται οι εντολές `fopen` και `fclose`.

```
f1 = fopen('path\όνομα αρχείου','r');
fclose(f1);
```

Όπου `f1` είναι ο δείκτης του συγκεκριμένου αρχείου και η επιλογή *r* δηλώνει το άνοιγμα του αρχείου για διάβασμα.

Για την ανάκτηση των δεδομένων ενός ASCII αρχείου χρησιμοποιούμε την εντολή `fscanf`. Η σύνταξη αυτής είναι η ακόλουθη :

```
[A,COUNT] = fscanf ( FID,FORMAT,SIZE)
```

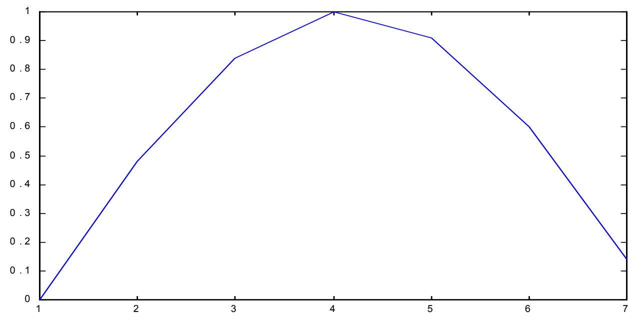
όπου
FID είναι ο pointer που είναι συσχετισμένος με το αρχείο που θέλουμε να διαβάσουμε.
FORMAT δηλώνει τον τύπο των δεδομένων του αρχείου και ακολουθεί την τυποποίηση της scanf της γλώσσας προγραμματισμού C.
SIZE δηλώνει τον αριθμό των στοιχείων που θα διαβάσουμε με την fscanf
A είναι ο πίνακας που θα αποθηκευτούν τα SIZE στοιχεία του αρχείου

3. Γραφικές παραστάσεις

Η εντολή `plot` δημιουργεί γραφικές παραστάσεις με άξονες x και y. Έστω *Y* είναι ένα διάνυσμα με τιμές : `[0 0.48 0.84 1.0 0.91 0.6 0.14]`. Η απεικόνιση των τιμών αυτών γίνεται ως εξής :

```
Y = [0  0.48  0.84  1.0  0.91  0.6  0.14] ;
plot (Y)
```

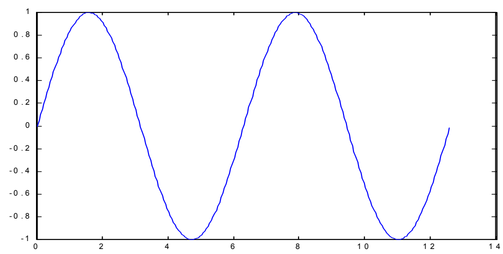
Το αποτέλεσμα φαίνεται στο σχήμα :



Στην περίπτωση που δύο διανύσματα X και Y έχουν το ίδιο μήκος, τότε η εντολή `plot (X, Y)` δημιουργεί μια γραφική παράσταση με τα στοιχεία του X στον x άξονα και τα στοιχεία του Y στον y. Για παράδειγμα :

```
t1 = 0:0.05:4*pi ;
y1 = sin(t1) ;
plot (t1,y1)
```

Έχει ως αποτέλεσμα την γραφική παράσταση :



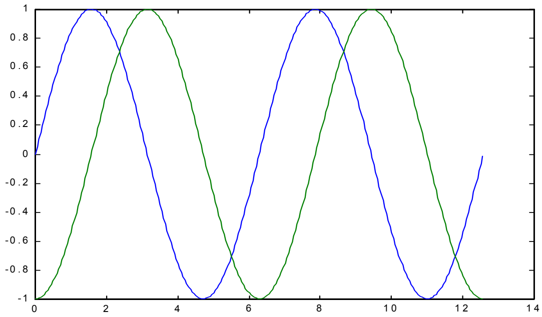
Για την ταυτόχρονη απεικόνιση δύο ή περισσοτέρων γραφικών παραστάσεων υπάρχουν δύο εναλλακτικοί τρόποι :

1) Με την εντολή `hold on` μεταξύ των αντίστοιχων εντολών `plot`.
2) Με μία εντολή `plot` ενσωματώνοντας στα ορίσματα της τα διανύσματα που θέλουμε να απεικονίσουμε.

Για παράδειγμα η διαδικασία για την ταυτόχρονη απεικόνιση της προηγούμενης γραφικής του ημίτονου με την γραφική του καθυστερημένου ημίτονου κατά $\pi/2$ είναι η εξής :

```
t = 0:0.05:4*pi ;
y1 = sin(t) ;
y2 = sin(t - pi/2) ;
plot (t,y1, t, y2) ;
```

Και το αποτέλεσμα είναι το παρακάτω :



Γενικότερα για την απεικόνιση μιας συνάρτησης *f(x)*, χρησιμοποιείται η εντολή `fplot` :

```
fplot ('f(x)',[xmin xmax]) ;
```

ΕΚΤΕΛΕΣΗ ΤΗΣ ΑΣΚΗΣΗΣ

(α) Να δημιουργηθούν οι 3x3 πίνακες ***A***, ***B*** με:

$$A = \begin{matrix} 1 & 1 & 1 \\ 2 & 2 & 2 \\ 1 & 1 & 1 \end{matrix} \quad , \quad B = \begin{matrix} 2 & 23 & 1 \\ 4 & 6 & 3 \\ 6 & -26 & 5 \end{matrix}$$

και να γίνουν οι εξής πράξεις :

```
1.E = (A * B ′ ) * A ′ + B – A * B
2.F = (A .* B) + 3 * B
```

3. Να βρεθεί ο γενικευμένος αντίστοφος πίνακας (ψευδοαντίστροφος) του πίνακα A:

$$A = \begin{matrix} 1 & 2 & 3 & -9 & 5 & 6 \\ 2 & 2 & 3 & 5 & 2 & 7 \\ 1 & 4 & 1 & 3 & 1 & 1 \end{matrix}$$

που δίνεται από την σχέση $B = (A'A)^{-1}A'$

(β) Να λυθεί το παρακάτω σύστημα :

```
 5x - 2y + 3z - w = 6
 1x + 2y - 3z     = 9
-3x +  y      -2w =-1
 4x + 3y -  z +5w =-7
```

(γ) Να γίνουν οι γραφικές παραστάσεις των πιο κάτω συναρτήσεων, στο διάστημα [0,1]:

$$f(x) = \tan(1 + e^{x^2})$$
$$g(x) = x^3 + \sin(x^2) + 5$$

(δ) Στο αρχείο «`\\ee-ntserver\PatRec\data.dat`» βρίσκονται αποθηκευμένα πρότυπα τεσσάρων κατηγοριών. Κάθε κατηγορία αποτελείται από δι-διδιάστατα πρότυπα. Κάθε γραμμή του αρχείου περιέχει και ένα πρότυπο ενώ κάθε στήλη περιέχει την x και την y διάσταση του αντίστοιχου πρότυπου. Στο αρχείο είναι τοποθετημένα ακολουθιακά πέντε πρότυπα για κάθε μία κατηγορία.

1) Να εισαχθούν τα παραδείγματα κάθε κατηγορίας σε ένα πίνακα διαστάσεων 5x8.
2) Να γίνει απεικόνιση των παραδειγμάτων κάθε κατηγορίας ξεχωριστά.
3) Να γίνει απεικόνιση όλων των παραδειγμάτων μαζί.
4) Να ελεγχθεί ποιες κατηγορίες (ανά δύο) είναι γραμμικά διαχωρίσιμες. Ποιές είναι οι ευθείες που διαχωρίζουν γραμμικά τις κατηγορίες;

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 1

ΕΙΣΑΓΩΓΗ ΣΤΗΝ ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ MATLAB

Αριθμός Ομάδας:

Ονοματεπώνυμο:

ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2

ΤΑΞΙΝΟΜΗΣΗ ΠΡΟΤΥΠΩΝ ΜΕ ΣΥΝΑΡΤΗΣΕΙΣ ΑΠΟΣΤΑΣΗΣ

1. Η ταξινόμηση της μικρότερης απόστασης προτύπων

Το πλέον διαδεδομένο κριτήριο ταξινόμησης με σύγκριση πρότυπων είναι η εύρεση της κατηγορίας εκείνης το πρωτότυπο της οποίας έχει την μικρότερη απόσταση από το άγνωστο παραμετρικό διάνυσμα. Το κριτήριο ταξινόμησης ονομάζεται ταξινόμηση της μικρότερης απόστασης (minimum distance pattern classification):

$$\omega_i = \underset{i}{\arg\min}\, d(x, y_i)$$

y_i είναι το πρότυπο αναφοράς της ω_i κατηγορίας και
x είναι το παραμετρικό διάνυσμα του άγνωστου πρότυπου.

2. Επιλογή ενός παραδείγματος σαν πρωτότυπο κατηγορίας

Η επιλογή του πλέον κατάλληλου πρότυπου πραγματοποιείται συνήθως όταν υπάρχουν σοβαροί περιορισμοί στην ταχύτητα απόκρισης του συστήματος ταξινόμησης οπότε δεν μπορούμε να χρησιμοποιήσουμε όλα τα παραδείγματα σαν πρωτότυπα, ή τα πρότυπα των κατηγοριών είναι σαφώς διαχωρισμένα.

Ο συνήθης τρόπος προσδιορισμού ενός βέλτιστου πρωτότυπου από τα παραδείγματα πραγματοποιείται με την εύρεση του πρότυπου εκείνου που έχει την μικρότερη αθροιστικά απόσταση από τα παραδείγματα της κατηγορίας του:

$$y_i = \underset{x_{ik}}{\arg\min} \sum_{j=1}^{N_i} d(x_{ij}, x_{ik})$$

x_{ik}, είναι το j παράδειγμα της της κατηγορίας ω_i,
N_i, είναι ο αριθμός των διαθέσιμων παραδειγμάτων της κατηγορίας ω_i

3. Υπολογισμός εικονικού πρωτότυπου

Η εύρεση του διανύσματος το οποίο έχει την μικρότερη αθροιστικά απόσταση από τα παραδείγματα της κατηγορίας του αποτελεί μία καλύτερη μέθοδο υπολογισμού του πρωτότυπου:

$$y_i = \arg\min_x \sum_{j=1}^{N_i} d(x_{ij}, x)$$

Αναλυτική λύση της εξίσωσης δεν είναι δυνατή για όλες της συναρτήσεις απόστασης. Αυτός είναι και ο σημαντικότερος λόγος για τον οποίο καταφεύγουμε συνήθως στην λύση επιλογής του βέλτιστου παραδείγματος.

Στην περίπτωση κατά την οποία χρησιμοποιούμε σαν συνάρτηση απόστασης το τετράγωνο της Ευκλείδειας απόστασης τότε αποδεικνύεται εύκολα ότι το μέσο διάνυσμα είναι εκείνο το οποίο ελαχιστοποιεί την αθροιστική απόσταση του από τα πρότυπα εκπαίδευσης.

Στο πρόγραμμα που ακολουθεί δίνεται η συνάρτηση `ClassMinDistEuclOne` η οποία υπολογίζει τα πρότυπα αναφοράς για κάθε μία κατηγορία προτύπων, και το σφάλμα ταξινόμησης με κριτήριο την ελάχιστη απόσταση.

ΠΡΟΓΡΑΜΜΑ ΣΕ MATLAB

```
function [Rc,Ru,Rep] = ClassMinDistEuclOne(x,c)
%#
%#   [Rc,Ru,Rep] = ClassMinDistEuclOne(x,c)
%#   Pattern Recognition:
%#       Distance measure:      Euclidean function
%#       Prototypes:            One Center
%#       Classification rule:   Minimum Distance
%#
%#   Input
%#       x: Pattern Vectors
%#       c: Classes
%#   Output
%#       Rc: Correct classification rate using the C-method
%#       Ru: Correct classification rate using the U-method
%#       Rep: Pattern vectors on each class
%#
NumOfClass = max(c) ;
NumOfPatterns = columns(x) ;
Mean = zeros(rows(x),NumOfClass) ;
Rep = zeros(NumOfClass,1) ;
%#
%#   C-Error
%#
Rc = zeros(NumOfClass,1) ;
for j = 1:NumOfPatterns
    k = c(j) ;
    Rep(k,1) = Rep(k,1) + 1 ;
    Mean(:,k) = Mean(:,k) + x(:,j) ;
end
for j=1:NumOfClass
    Mean(:,j) = Mean(:,j) / Rep(j,1) ;
```

```
end
for i = 1:NumOfPatterns
    for j = 1:NumOfClass
        Dist(j) = (x(:,i) - Mean(:,j))' * ( x(:,i) - Mean(:,j) ) ;
    end
    Rec = ArgMin(Dist) ;
    if (Rec == c(i))
       Rc(Rec) = Rc(Rec) + 1 ;
    end
end
%#
%#  U-Error
%#
Ru = zeros(NumOfClass,1) ;
for j = 1:NumOfPatterns
    k = c(j) ;
    Mt = Mean(:,k) ;
    Mean(:,k)= Rep(k,1)/(Rep(k,1)-1) * (Mean(:,k)-x(:,j)/Rep(k,1)) ;
    for i = 1:NumOfClass
        Dist(i)=(x(:,j)-Mean(:,i))' * (x(:,j)-Mean(:,i)) ;
    end
    [tmp,Rec] = min(Dist) ;
    if (Rec == k)
       Ru(Rec) = Ru(Rec) + 1 ;
    end
    Mean(:,k) = Mt ;
end
```

Κατανοήστε και εξηγήστε στον επιβλέποντα την δομή του προγράμματος, την χρήση και το περιεχόμενο των μεταβλητών του προγράμματος.

ΕΚΤΕΛΕΣΗ ΤΗΣ ΑΣΚΗΣΗΣ

1. Διάβασμα παραδειγμάτων τύπων γυαλιού

Στο αρχείο «`\\ee-ntserver\PatRec\glass.data`» υπάρχουν σε φόρμα "ASCII" δεδομένα από μετρήσεις της χημικής σύστασης 214 κομματιών γυαλιού. Οι τρείς πρώτες γραμμές του αρχείου είναι οι ακόλουθες:

```
1 1.52101 13.64 4.49 1.10 71.78 0.06 8.75 0.00 0.00 1
2 1.51761 13.89 3.60 1.36 72.73 0.48 7.83 0.00 0.00 1
3 1.51618 13.53 3.55 1.54 72.99 0.39 7.78 0.00 0.00 1
```

Οι αριθμοί κάθε γραμμής παριστούν τα ακόλουθα στοιχεία:

1. Αυξων αριθμός μέτρησης.
2. Βαθμός διάθλασης.
3. Περιεκτικότητα σε Na.
4. Περιεκτικότητα σε Mg.
5. Περιεκτικότητα σε Al.
6. Περιεκτικότητα σε Si.
7. Περιεκτικότητα σε K.
8. Περιεκτικότητα σε Ca.
9. Περιεκτικότητα σε Ba.
10. Περιεκτικότητα σε Fe.
11. Ο τύπος του Γυαλιού.

Ο κωδικός αριθμός του τελευταίου πεδίου περιγράφει τον τύπο του γυαλιού, ως εξής:

1. Γυαλί καλής ποιότητας από παράθυρο σπιτιού.
2. Γυαλί κοινής ποιότητας από παράθυρο σπιτιού.
3. Γυαλί καλής ποιότητας από παράθυρο αυτοκινήτου.
4. (δεν χρησιμοποιείται τέτοιος κωδικός).
5. Γυαλί δοχείου.
6. Γυαλί από επιτραπέζιο σκεύος.
7. Γυαλί από φανάρι αυτοκινήτου.

Τοποθετείστε τα παραδείγματα του αρχείου σε πίνακα x και την κατηγορία κάθε πρότυπου σε πίνακα c, χρησιμοποιώντας την συνάρτηση `ReadGlass`:

```
function [x,c] = ReadGlass(Tot)
%#
%#  [x,c] = ReadGlass(Tot)
%#   Read the Glass database stored in the file "glass.data"
%#  Input:
%#      Tot: Number of Patterns
%#  Output:
%#      x: Patterns matrix
%#      c: Classes integer
if ( Tot > 214 )
   Tot = 214 ;
```

```
end
f1 = fopen( '\\ee-ntserver\PatRec\glass.data', 'r' ) ;
x = zeros(9,Tot) ;
c = zeros(1,Tot) ;
for j = 1:Tot
   b = fscanf( f1, '%d', 1 ) ;
   for i = 1:9
      x(i,j) = fscanf( f1, '%f', 1 ) ;
   end
   c(1,j) = fscanf( f1, '%d', 1 ) ;
end
fclose(f1) ;
```

1. Υπολογίστε το βέλτιστο εικονικό πρωτότυπο

Για κάθε μία από τις κατηγορίες υπολογίστε το εικονικό πρότυπο που έχει την μικρότερη αθροιστική απόσταση από τα παραδείγματα της κατηγορίας του. Σαν συνάρτηση απόστασης χρησιμοποιείστε το τετράγωνο της Ευκλείδειας απόστασης.

2. Υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης

Χρησιμοποιώντας όλα τα παραδείγματα υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης. Το μέγιστο σφάλμα του συστήματος ταξινόμησης να το υπολογίστε με την U-μέθοδο (leaving one-out).

3. Επιλέξατε σαν πρωτότυπο ένα παράδειγμα

Για κάθε μία από τις κατηγορίες υπολογίστε το παράδειγμα που έχει την μικρότερη αθροιστική απόσταση από τα υπόλοιπα παραδείγματα της κατηγορίας του. Σαν συνάρτηση απόστασης χρησιμοποιείστε το τετράγωνο της Ευκλείδειας απόστασης.

4. Υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης

Χρησιμοποιώντας όλα τα παραδείγματα υπολογίστε το ελάχιστο και το μέγιστο σφάλμα του συστήματος ταξινόμησης.

5. Τι παρατηρείτε από τα αποτελέσματα των μετρήσεων

Περιγράψτε και αιτιολογείστε τα αποτελέσματα που πήρατε από τις μετρήσεις του ελάχιστου και του μέγιστου σφάλματος.

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2

ΤΑΞΙΝΟΜΗΣΗ ΠΡΟΤΥΠΩΝ ΜΕ ΣΥΝΑΡΤΗΣΕΙΣ ΑΠΟΣΤΑΣΗΣ

Αριθμός Ομάδας:

Ονοματεπώνυμο:

ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 3

Ο ΑΛΓΟΡΙΘΜΟΣ K-MEANS ΚΑΙ ΤΟ ΚΡΙΤΗΡΙΟ ΤΑΞΙΝΟΜΗΣΗΣ ΤΩΝ Κ-ΠΛΗΣΙΕΣΤΕΡΩΝ ΠΡΩΤΟΤΥΠΩΝ

1. Η ταξινόμηση των Κ-πλησιέστερων πρωτότυπων

Στην περίπτωση κατά την οποία η περιγραφή των κατηγοριών γίνεται με περισσότερα του ενός πρωτότυπα, τότε εκτός από το κριτήριο ταξινόμησης με την μικρότερη απόσταση έχουμε την δυνατότητα να χρησιμοποιήσουμε και το κριτήριο ταξινόμησης των Κ-πλησιέστερων πρωτότυπων (K-nearest neighbor classification). Το κριτήριο αυτό χρησιμοποιείται σπανιότερα, αλλά πολλές φορές δίνει μικρότερο σφάλμα ταξινομήσεων από το κριτήριο ταξινόμησης της μικρότερης απόστασης. Η ταξινόμηση πραγματοποιείται με την εύρεση των Κ πρωτότυπων που παρουσιάζουν την μικρότερη απόσταση από το άγνωστο πρότυπο και ακολουθεί η ταξινόμηση στην κατηγορία εκείνη τα πρωτότυπα της οποίας υπερτερούν σε αριθμό στο σύνολο των Κ-πλησιέστερων πρωτότυπων.

2.Επιλογή k-πρότυπων από παραδείγματα και ο αλγόριθμος k-means.

Το πρόβλημα της εκπαίδευσης συστήματος ταξινόμησης από παραδείγματα λύνεται συνήθως με την μέθοδος επιλογής k-πρότυπων από τα παραδείγματα. Η μέθοδος αυτή είναι η πλέον διαδεδομένη στις εφαρμογές και αποτελεί συνήθως έναν συμβιβασμό ανάμεσα στην υπολογιστική πολυπλοκότητα ενός συστήματος ταξινόμησης που χρησιμοποιεί όλα τα διαθέσιμα παραδείγματα για πρωτότυπα (δίνοντας υψηλή αξιοπιστία ταξινόμησης) και του συστήματος ταξινόμησης που χρησιμοποιεί μονάχα ένα πρότυπο αναφοράς και παρουσιάζει χαμηλή υπολογιστική πολυπλοκότητα και συνήθως μικρότερη αξιοπιστία ταξινόμησης.

Οι μέθοδοι υπολογισμού των πρωτότυπων ονομάζονται αλγόριθμοι του τύπου k-μέσων (k-means). Οι αλγόριθμοι αυτοί βασίζονται στην επανάληψη τριών βημάτων και παρουσιάζουν ευρεία διάδοση σε πρακτικά συστήματα διότι έχει αποδειχθεί ότι οι πλειονοτητα των αλγόριθμων του τύπου Κ-μέσων συγκλίνουν. Μειονέκτημα του αλγόριθμου είναι ότι η σύγκλιση επιτυγχάνεται σε ένα τοπικό ελάχιστο του αθροίσματος των αποστάσεων των προτύπων από τα πρωτότυπα.

- ***Αρχική επιλογή***. Ξεκινάμε με μία τυχαία επιλογή των πρωτότυπων $z_1,\ldots, z_k$ από τα πρότυπα των παραδειγμάτων.
- ***Ταξινόμηση προτύπων***. Βάσει του κριτηρίου ταξινόμησης τα πρότυπα της κατηγορίας ταξινομούνται σε μία από τις Κ εικονικές κατηγορίες που ορίζονται από τα πρωτότυπα.
- ***Υπολογισμός πρωτότυπων***. Σε κάθε εικονική κατηγορία ορίζουμε το πρωτότυπο σαν το πρότυπο του παραδείγματος που ανήκει στην εικονική κατηγορία και ελαχιστοποιεί το άθροισμα των αποστάσεων του από τα υπόλοιπα πρότυπα της κατηγορίας.
- ***Ελεγχος του κριτηρίου σύγκλισης***. Αν κατά την διαδοχική εκτίμηση των πρωτότυπων αυτά παρέμειναν ίδια, τότε ο αλγόριθμος τερματίζει, διαφορετικά επαναλαμβάνεται από το δεύτερο βήμα.

ΠΡΟΓΡΑΜΜΑ ΣΕ MATLAB

```
function [Alphabet] = KMeans(Vector,KM,Threshold)
%#
%#  [Alphabet] = KMeans(Vectors,KM,Threshord)
%#
%#  Input
%#      Vector: Pattern Vectors
%#      KM: Size of the alphabet
%#      Threshold: Covergence threshold
%#  Output
%#      Alphabet: Vector alphabet
%#
L = size(Vector) ;
N = L(1,1) ;
NumOfVect = columns(Vector) ;
if ( NumOfVect <= KM )
   printf( 'The alphabet is greater than the number of Vectors\n' ) ;
   return ;
end
Cluster = zeros(NumOfVect,1) ;
Alphabet = zeros(N,KM) ;
Error = 1.0E28 ;
prError = 1.0E30 ;
for i=1:KM
   while( 1 )
      Alphabet(:,i) = Vector(:,floor(NumOfVect*rand(1,1))+1) ;
      Same = 0 ;
      for j=1:i-1
         if ( Alphabet(:,j) == Alphabet(:,i) )
            Same = 1 ;
            break ;
         end
      end
   if ( Same == 0 )
      break ;
   end
   end
end
while( (prError-Error)/Error > Threshold )
   prError = Error ;
   Error = 0.0 ;
   NewAlphabet = zeros(N,KM) ;
   Rep = zeros(KM,1) ;
   for i = 1:NumOfVect
      for j = 1:KM
         Dist(j)=(Vector(:,i)-Alphabet(:,j))'*(Vector(:,i)-Alphabet(:,j)) ;
      end
      Rec = ArgMin(Dist) ;
      Error = Error + min(Dist) ;
      Cluster(i) = Rec ;
      Rep(Rec) = Rep(Rec)+1 ;
      NewAlphabet(:,Rec) = NewAlphabet(:,Rec) + Vector(:,i) ;
   end
   for i = 1:KM
      Alphabet(:,i) = NewAlphabet(:,i) / Rep(i) ;
   end
end
```

Κατανοήστε και εξηγήστε στον επιβλέποντα την δομή του προγράμματος, την χρήση και το περιεχόμενο των μεταβλητών του προγράμματος.

ΕΚΤΕΛΕΣΗ ΤΗΣ ΑΣΚΗΣΗΣ

1. Διάβασμα παραδειγμάτων ανίχνευσης τοπικής συμπεριφοράς της Ιονόσφαιρας

Στο αρχείο «`\\ee-ntserver\PatRec\ionosphere.data`» υπάρχουν σε φόρμα "ASCII" δεδομένα που προέχονται από ανακλάσεις σημάτων στην ιονόσφαιρα. Συγκεκριμένα σήματα ειδικής μορφής στέλνονται στην ιονόσφαιρα από 16 κεραίες υψηλών συχνοτήτων. Από τα ανακλούμενα σήματα κατατάσσουμε την σύσταση της ιονόσφαιρας στην συγκεκριμένη θέση και χρόνο (η σύσταση μεταβάλλεται συνεχώς) σε "καλής" και "κακής" σύστασης. Υπάρχουν συνολικά 351 μετρήσεις που ταξινομούνται σε δύο κατηγορίες.

Οι τρείς πρώτες γραμμές του αρχείου είναι οι ακόλουθες:

```
1 0 0.99539 -0.05889 0.85243 0.02306 0.83398 -0.37708 1 0.03760 0.85243 -0.17755 0.59755 -
0.44945 0.60536 -0.38223 0.84356 -0.38542 0.58212 -0.32192 0.56971 -0.29674 0.36946 -0.47357
0.56811 -0.51171 0.41078 -0.46168 0.21266 -0.34090 0.42267 -0.54487 0.18641 -0.45300 g

1 0 1 -0.18829 0.93035 -0.36156 -0.10868 -0.93597 1 -0.04549 0.50874 -0.67743 0.34432 -0.69707
-0.51685 -0.97515 0.05499 -0.62237 0.33109 -1 -0.13151 -0.45300 -0.18056 -0.35734 -0.20332 -
0.26569 -0.20468 -0.18401 -0.19040 -0.11593 -0.16626 -0.06288 -0.13738 -0.02447 b

1 0 1 -0.03365 1 0.00485 1 -0.12062 0.88965 0.01198 0.73082 0.05346 0.85443 0.00827 0.54591
0.00299 0.83775 -0.13644 0.75535 -0.08540 0.70887 -0.27502 0.43385 -0.12062 0.57528 -0.40220
0.58984 -0.22145 0.43100 -0.17365 0.60436 -0.24180 0.56045 -0.38238 g
```

Οι αριθμοί κάθε γραμμής παριστούν τα ακόλουθα στοιχεία:

Πεδίο	***Περιεχόμενο του πεδίου***
1-34	η διανυσματική παράσταση του πρότυπου
35	"b" για το καλό πρότυπο και "g" για το κακό πρότυπο

Τοποθετείστε τα παραδείγματα του αρχείου σε πίνακα x και την κατηγορία κάθε πρότυπου σε πίνακα c, χρησιμοποιώντας την συνάρτηση `ReadGlass`:

```
function [x,c] = ReadIonosphere(Tot)
%#
%#  [x,c] = ReadIonosphere(Tot)
%#   Read the Ionosphere database stored in the file "ionosphere.data"
%#  Input:
%#      Tot: Number of Patterns
%#  Output:
%#      x: Patterns matrix
%#      c: Classes integer
if ( Tot > 351 )
   Tot = 351;
end
f1 = fopen( '\\ee-ntserver\PatRec\ionosphere.data', 'r' ) ;
x = zeros(34,Tot) ;
c = zeros(1,Tot) ;
for j = 1:Tot
   for i = 1:34
      x(i,j) = fscanf( f1, '%f', 1 ) ;
```

```
    end
    br = fscanf( f1, '%s', 1 ) ;
    if ( strcmp( br, 'b' ) == 1 )
       c(1,j) = 1 ;
    else
       c(1,j) = 2 ;
    end
end
fclose(f1) ;
```

2. Υπολογίστε τρία εικονικά πρωτότυπα από τα παραδείγματα

Για κάθε μία από τις δύο κατηγορίες προτύπων υπολογίστε τρία εικονικά πρωτότυπα από τα παραδείγματα με την βοήθεια του αλγόριθμου K-means.

3. Υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης

Χρησιμοποιώντας τα εικονικά πρότυπα που βρήκατε στο προηγούμενο βήμα και όλα τα παραδείγματα υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης που χρησιμοποιεί το κριτήριο του πλησιέστερου γείτονα.

4. Επιλέξατε τρία παραδείγματα σαν πρωτότυπα

Τροποποιήστε την συνάρτηση `KMeans(Vector,KM,Threshold)` έτσι ώστε να επιλέγει `KM` πρωτότυπα από τα παραδείγματα κάθε κατηγορίας. Σαν συνάρτηση απόστασης χρησιμοποιείστε το τετράγωνο της Ευκλείδιας απόστασης. Με την βοήθεια της συνάρτησης που θα κατασκευάσετε επιλέξατε τρία παραδείγματα από κάθε κατηγορία σαν πρωτότυπα.

5. Υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης

Χρησιμοποιώντας όλα τα παραδείγματα υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης για τα πρωτότυπα που υπολογίσατε στο προηγούμενο βήμα.

6. Τι παρατηρείτε από τα αποτελέσματα των μετρήσεων

Συγκρίνατε και αιτιολογείστε τα αποτελέσματα που πήρατε από τις μετρήσεις του ελάχιστου σφάλματος για τις διαφορετικές μεθόδους εκτίμησης των πρωτότυπων κάθε κατηγορίας.

***ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ***

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 3

Ο ΑΛΓΟΡΙΘΜΟΣ K-MEANS ΚΑΙ ΤΟ ΚΡΙΤΗΡΙΟ ΤΑΞΙΝΟΜΗΣΗΣ ΤΩΝ Κ-ΠΛΗΣΙΕΣΤΕΡΩΝ ΠΡΩΤΟΤΥΠΩΝ

Αριθμός Ομάδας:

Ονοματεπώνυμο:

<u>ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ</u>

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 4

Ο ΑΛΓΟΡΙΘΜΟΣ PERCPTRON

1. Ο αλγόριθμος Perceptron

Εστω σύστημα ταξινόμησης Ν κατηγοριών για το οποίο υπάρχουν Μ παραδείγματα σε κάθε μία από τις κατηγορίες του. Αν χρησιμοποιήσουμε γραμμικές συναρτήσεις απόφασης για τις οποίες ισχύει ότι $g_{ij}(\boldsymbol{x})=-g_{ji}(\boldsymbol{x})$ τότε ο ακόλουθος επαναληπτικός αλγόριθμος συγκλίνει ***αν*** τα παραδείγματα είναι γραμμικά διαχωρίσιμα.

- ***Αρχικές τιμές***. Τοποθετούμε τυχαίες τιμές στους συντελεστές των N(N-1)/2 συναρτήσεων απόφασης.
- ***Ταξινόμηση***. Επιλέγουμε τυχαία μία κατηγορία ω_i και από αυτήν το πρότυπο της $\boldsymbol{x}_m$. Υπολογίζουμε όλες τις συναρτήσεις απόφασης $g_{ij}(\boldsymbol{x}_m)$. Για όσες από αυτές η συνάρτηση απόφασης δεν είναι θετική επαναπροσδιορίζουμε του αντίστοιχους συντελεστές ως εξής: $w_{ij} = w_{ij} + a\,(\boldsymbol{x}_m, 1)$, όπου a είναι θετικός πραγματικός αριθμός.
- ***Ελεγχος σύγκλισης***. Αν με τον έλεγχο όλων των παραδειγμάτων εκπαίδευσης δεν πραγματοποιηθεί επαναπροσδιορισμός των συντελεστών των συναρτήσεων απόφασης ο αλγόριθμος τερματίζει. Διαφορετικά επαναλαμβάνεται το προηγούμενο βήμα.

Ο αλγόριθμος perceptron υπολογίζει τις συναρτήσεις απόφασης για σύστημα ταξινόμησης Ν κατηγοριών. Ο υπολογισμός των συναρτήσεων ανάμεσα σε δύο κατηγορίες εξαρτάται μόνο από τα αντίστοιχα παραδείγματα των κατηγοριών. Συνεπώς μπορούμε κάθε φορά να υπολογίσουμε ανεξάρτητα τους συντελεστές των συναρτήσεων απόφασης γιά κάθε ζευγάρι κατηγοριών. Υπολογίζουμε δηλαδή τους συντελεστές της συνάρτησης $g_{ij}(\boldsymbol{x})$ από τα αντίστοιχα παραδείγματα των κατηγοριών ω_i και ω_j.

Για να βελτιώσουμε την ταχύτητα σύγκλισης του αλγόριθμου, αρχικά τοποθετούμε μία σχετικά μεγάλη αρχική τιμή στον συντελεστή εκπαίδευσης, π.χ. α=1. Στην συνέχεια επιλέγοντας τυχαία παραδείγματα εκτελούμε το επαναληπτικό τμήμα του αλγόριθμου perceptron. Οταν ολοκληρωθεί ένας ολόκληρος κύκλος ελέγχου των παραδειγμάτων και δεν πραγματοποιηθεί καμμία μεταβολή των συντελεστών της συνάρτησης απόφασης ο αλγόριθμος τερματίζει. Αν το πρότυπο κάποιου παραδείγματος δεν ταξινομηθεί σωστά, τότε το διάνυσμα των συντελεστών βαρύτητας που έχουν αρνητική τιμή επαναπροσδιορίζονται και επαναλαμβάνεται ο κύκλος ελέγχου των παραδειγμάτων. Αν μετά από κάποιες επαναλήψεις όλα τα πρότυπα ταξινομηθούν σωστά τότε το ελάχιστο σφάλμα του συστήματος ταξινόμησης είναι 0% και γνωρίζουμε επίσης ότι τα πρότυπα είναι γραμμικά διαχωρίσιμα. Αν το επαναληπτικό τμήμα του αλγόριθμου δεν τερματίζει τότε ελαττώνουμε τον συντελεστή εκπαίδευσης α και επαναλαμβάνουμε τον αλγόριθμο. Η διαδικασία μείωσης του συντελεστή εκπαίδευσης επαναλαμβάνεται ξανά αν ο αλγόριθμος δεν συγκλίνει. Αν η αριθμητική τιμή του συντελεστή εκπαίδευσης μειωθεί σημαντικά χωρίς να πετύχουμε σύγκλιση, τότε μπορούμε να ισχυριστούμε με αρκετά μεγάλη βεβαιότητα ότι τα πρότυπα δεν είναι γραμμικά διαχωρίσιμα.

ΠΡΟΓΡΑΜΜΑ ΣΕ MATLAB

```
function [Rc,Rep] = Perceptron(x1,x2,Lr,MaxRep)
%#
%#  [Rc,Rep] = Perceptron(x,c,Lr)
%#  Input
%#      x1: Pattern Vectors for the first class
%#      x2: Pattern Vectors for the second class
%#      Lr: Learning rate
%#      MaxRep: Maximum repetitions
%#  Output
%#      Rc: Correct classification rate using the C-method
%#      Rep: Pattern vectors on each class
NumOfP1 = columns(x1) ;
NumOfP2 = columns(x2) ;
Weights = 2 * rand(rows(x1)+1,1) - 1 ;
Rep = [NumOfP1,NumOfP2] ;
TotPat = sum(Rep) ;
Rc = zeros(2,1) ;
if ( rows(x1) ~= rows(x2) )
      printf( 'Error in vectors x1, x2\n' ) ;
end
%#  C-Error
for j = 1:MaxRep
      if ( rand() > 0.5 )
            k = floor(rand() * NumOfP1 + 1) ;
            Pat = [x1(:,k);1] ;
            Score = Weights' * Pat ;
            if ( Score < 0 )
                  Weights = Weights + Lr * Pat ;
            end
      else
      k = floor(rand() * NumOfP2 + 1) ;
      Pat = [x2(:,k);1] ;
      Score = Weights' * Pat ;
      if ( Score >= 0 )
         Weights = Weights - Lr * Pat ;
      end
      end
      Rc = zeros(2,1) ;
      for i=1:NumOfP1
            if ( Weights' * [x1(:,i);1] >= 0 )
                  Rc(1) = Rc(1) + 1 ;
            end
      end
   for i=1:NumOfP2
      if ( Weights' * [x2(:,i);1] < 0 )
         Rc(2) = Rc(2) + 1 ;
      end
   end
printf( '%8d', sum(Rc) ) ;
fflush(stdout) ;
      if ( sum(Rc) == TotPat )
            break ;
      end
end
printf( '\n' ) ;
```

Κατανοήστε και εξηγήστε στον επιβλέποντα την δομή του προγράμματος, την χρήση και το περιεχόμενο των μεταβλητών του προγράμματος.

ΕΚΤΕΛΕΣΗ ΤΗΣ ΑΣΚΗΣΗΣ

1. Διάβασμα προτύπων από ιατρικές εργαστηριακές εξετάσεις

Στο αρχείο "\\ee-ntserver\PatRec\ionosphere.data" υπάρχουν σε φόρμα "ASCII" δεδομένα που προέχονται από εξετάσεις αίματος ανθρώπων που έχουν εξετασθεί για προβληματική λειτουργία του ήπατος (συκώτι). Οι μετρήσεις αυτές έχουν πραγματοποιηθεί στο BUPA Medical Research Ltd. Σε κάθε γραμμή του αρχείου υπάρχουν πέντε αριθμοί που προέρχονται από εξετάσεις αίματος οι οποίες δίνουν σημαντικές πληροφορίες στους γιατρούς για την διάγνωση δυσλειτουργιών του ήπατος οι οποίες προέρχονται κυρίως από την συστηματική, υπερβολική και μακροχρόνια κατανάλωση οινοπνευματοδών ποτών.
Στο αρχείο υπάρχουν συνολικά 345 πρότυπα.

Οι τρείς πρώτες γραμμές του αρχείου είναι οι ακόλουθες:

```
85 92 45 27 31 0.0 1
85 64 59 32 23 0.0 2
86 54 33 16 54 0.0 2
```

Οι αριθμοί σε κάθε γραμμή παριστούν τα ακόλουθα στοιχεία:

Πεδίο	***Περιεχόμενο του πεδίου***
1	Μέση τιμή της συγκέντρωσης λευκών και ερυθρών αιμοσφαιρίων στο αίμα
2	Συγκέντρωση αλκαλικής φωσφατάσης, alkaline phosphotase
3	Συγκέντρωση της τρανσαμινάσης sgpt, alamine aminotransferase
4	Συγκέντρωση της τρανσαμινάσης sgot, aspartate aminotransferase
5	Συγκέντρωση της γ.GT, gamma-glutamyl transpeptidase
6	Ισοδύναμη ημερήσια κατανάλωση μεταλλικών κουτιών μπύρας (250 gr.)
7	Πεδίο τελικής διάγνωσης: 1 ο εξεταζόμενος είναι υγιής 2 ο εξεταζόμενος πάσχει από κίρρωση του ήπατος

Τοποθετείστε τα παραδείγματα του αρχείου που αντιστοιχούν σε υγιείς στον πίνακα X και τα παραδείγματα που αντιστοιχούν σε ασθενείς στον πίνακα C.

```
function [x,c] = ReadLiver(Tot)
%#
%#  [x,c] = ReadLiver(Tot)
%#   Read the Liver database stored in the file "liverdisorder.data"
%#  Input:
%#      Tot: Number of Patterns
%#  Output:
%#      x: Patterns matrix
%#      c: Classes integer
if ( Tot > 345 )
   Tot = 345 ;
end
f1 = fopen( '../DATA/liverdisorder.data', 'r' ) ;
x = zeros(6,Tot) ;
c = zeros(1,Tot) ;
for j = 1:Tot
        for i = 1:6
        x(i,j) = fscanf( f1, '%f', 1 ) ;
        end
        c(1,j) = fscanf( f1, '%d', 1 ) ;
end
fclose(f1) ;
```

2. Υπολογίστε τους συντελεστές των γραμμικών συναρτήσεων απόφασης με τον αλγόριθμο Perceptron

Για να κατασκευάσετε ένα αυτόματο διαγνωστικό σύστημα για δυσλειτουργίες του ήπατος χρησιμοποιώντας γραμμικές συναρτήσεις απόφασης πρέπει αρχικά να υπολογίσετε τους συντελεστές των γραμμικών συναρτήσεων. Επιλέξτε έναν συντελεστή εκπαίδευσης και υπολογίστε το διάνυσμα των συντελεστών της γραμμικής συνάρτησης απόφασης. *Δώστε προσοχή στο κριτήριο σύγκλισης διότι αν τα πρότυπα δεν είναι γραμμικά διαχωρίσιμα ο αλγόριθμος δεν τερματίζει.*

3. Υπολογίστε το σφάλμα του συστήματος ταξινόμησης

Χρησιμοποιώντας όλα τα παραδείγματα εκπαίδευσης υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης.

4. Μελετήστε την επίδραση που έχει ο συντελεστής εκπαίδευσης στον αλγόριθμο Perceptron

Για να μελετήσετε την επίδραση που έχει η επιλογή του συντελεστή εκπαίδευσης στον αλγόριθμο perceptron κατασκευάστε την γραφική παράσταση που δίνει την αξιοπιστία του γραμμικού συστήματος ταξινόμησης για διαφορετικές τιμές του συντελεστή εκπαίδευσης. Κατασκευάστε επίσης και τον αριθμό των επαναλήψεων του αλγόριθμου σαν συνάρτηση του συντελεστή εκπαίδευσης.

5. Τι παρατηρείτε από τα αποτελέσματα των μετρήσεων

Περιγράψτε και αιτιολογείστε τα αποτελέσματα που πήρατε από τις μετρήσεις του ελάχιστου σφάλματος.

ΤΜΗΜΑ ΗΛΕΚΤΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 4

Ο ΑΛΓΟΡΙΘΜΟΣ PERCPTRON

Αριθμός Ομάδας:

Ονοματεπώνυμο:

<u>ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ</u>

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 5

Ο ΑΛΓΟΡΙΘΜΟΣ HO-KASHYAP

1. Μη γραμμικά διαχωρίσιμα πρότυπα

Ο αλγόριθμος Ho-Kasyhap λύνει αποτελεσματικά το πρόβλημα τις εκπαίδευσης των συναρτήσεων απόφασης στην περίπτωση των μη γραμμικά διαχωρίσιμων προτύπων. Ο αλγόριθμος ελαχιστοποίησης του σφάλματος εκτίμησης είναι ένας επαναληπτικός αλγόριθμος ο οποίος συγκλίνει ακόμα και στις περιπτώσεις στις οποίες τα πρότυπα των παραδειγμάτων δεν είναι γραμμικά διαχωρισμένα.

Ας δούμε όμως πως προκύπτει ο αλγόριθμος. Οπως έχει προαναφερθεί μελετάμε το πρόβλημα ορισμού της συνάρτησης απόφασης δύο κατηγοριών μία και ο υπολογισμός των συντελεστών εξαρτάται μονάχα από τα παραδείγματα δύο κατηγοριών κάθε φορά. Εστω $g_{12}(\boldsymbol{x}) = -g_{21}(\boldsymbol{x})$ και N_1, N_2 είναι ο αριθμός των παραδειγμάτων για τις δύο κατηγορίες αντίστοιχα.

Ορίζω σαν $\boldsymbol{x}$ την επέκταση του διανύσματος κάθε πρότυπου που προκύπτει με την προσθήκη μιάς νέας διάστασης με τιμή 1, $\boldsymbol{x} = (\boldsymbol{x}', 1)$ όπου $\boldsymbol{x}'$ είναι το διάνυσμα του πρότυπου. Επίσης για κάθε ένα από τα διανύσματα εκπαίδευσης της κατηγορίας ω2 αντιστρέφω το πρόσημο των συντελεστών $\boldsymbol{x} = -\boldsymbol{x}$. Η μετατροπή αυτή μετασχηματίζει το πρόβλημα εκπαίδευσης του συστήματος στην εύρεση των συντελεστών $\boldsymbol{w}$ οι οποίοι να ικανοποιούν τις ακόλουθες ανισότητες:

$$\boldsymbol{x}_i \boldsymbol{w} > 0,\ \ i = 1,...,M,\ M = N_1 + N_2$$

Ολες τις συναρτήσεις μπορούμε να τις ομαδοποιήσουμε σε:

$$\boldsymbol{X}\,\boldsymbol{w} > \boldsymbol{0},\ \ \boldsymbol{X} = (\,\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_M\,)$$

Το πρόβλημα επίλυσης της ανίσωσης μπορεί να αντικατασταθεί από ένα πρόβλημα επίλυσης της εξίσωσης: $\boldsymbol{X}\,\boldsymbol{w} = \boldsymbol{b}$, $\boldsymbol{b}$ διάνυσμα θετικών αριθμών.

Αναλυτική λύση της εξίσωσης δεν μπορεί να επιτευχθεί, γιαυτό τον λόγο προσπαθούμε να επιτύχουμε μία επαναληπτική λύση η οποία να βελτιστοποιεί διαδοχικά μία συνάρτηση κριτηρίου. Η συνάρτηση αυτή θα μας δείχνει ταυτόχρονα πόσο έχουμε πλησιάσει την επιθυμούμενη λύση του προβλήματος σε κάθε επανάληψη του αλγόριθμου. Η συνάρτηση σφάλματος δίνεται από την εξίσωση:

$$Er(X,w,b) = \frac{1}{2}\sum_i (w^T x_i - b_i)^2 = \frac{1}{2}\| Xw - b \|^2$$

Η βέλτιστη τιμή του $\boldsymbol{w}$ προκύπτει από την ελαχιστοποίηση της συνάρτησης του σφάλματος.

Επειδή μπορούμε να ελαχιστοποιήσουμε την συνάρτηση μεταβάλλοντας δύο ανεξάρτητες μεταβλητές τις $\boldsymbol{w}$, και $\boldsymbol{b}$ υπολογίζουμε την παράγωγο της συνάρτησης σφάλματος ως προς αυτές τις μεταβλητές.

$$\frac{\partial E(X,w,b)}{\partial w} = X^T (Xw - b)$$

Με γνωστή την τιμή ***b***, η ελάχιστη τιμή της συνάρτησης βρίσκεται στο σημείο στο οποίο ισχύει:

$$\frac{\partial E(X,w,b)}{\partial b} = -(Xw-b)$$
$$\frac{\partial E(X,w,b)}{\partial w} = 0 \Rightarrow X^T(Xw-b) = 0 \Rightarrow w = (X^T X)^{-1} X^T b$$

Η έκφραση ($\boldsymbol{X}^{\mathrm{T}}$ $\boldsymbol{X}$)$^{-1}$ $\boldsymbol{X}^{\mathrm{T}}$ ονομάζεται γενικευμένος αντίστροφος πίνακας του ***X***. Ο συντελεστής ***b*** επαναπροσδιορίζεται βάσει της μεθόδου βηματικής σύγκλισης:

$$b = b - c\frac{\partial E(X,w,b)}{\partial b}$$

Για να έχουμε αποδεκτή λύση δεν πραγματοποιούμε την απλή αντικατάσταση της παραγώγου που έχουμε υπολογίσει διότι γνωρίζουμε ότι οι συνιστώσες του διανύσματος ***b*** πρέπει να είναι πάντα θετικοί αριθμοί. Με την αναδρομική σχέση που δίνουμε στην επόμενη εξίσωση επαναπροσδιορίζουμε τους συντελεστές του διανύσματος ***b*** μόνο στην περίπτωση κατά την οποία αυτοί θα αυξηθούν αριθμητικά σε κάποιες από τις συνιστώσες τους.

$$\boldsymbol{b} = \boldsymbol{b} + \mathrm{c}\,(\,\boldsymbol{X}\,\boldsymbol{w} - \boldsymbol{b} + |\boldsymbol{X}\,\boldsymbol{w} - \boldsymbol{b}|)$$

Με την τεχνική αυτή προσεγγίζουμε την επίλυση της ανίσωσης που επιθυμούμε, ικανοποιώντας ταυτόχρονα και τους περιορισμούς που θέσαμε.

Με βάση τα παραπάνω παραθέτουμε τον αλγόριθμο υπολογισμού των συναρτήσεων απόφασης από παραδείγματα που βασίζεται στην ελαχιστοποίηση του σφάλματος εκτίμησης. Συνήθης επιλογή του συντελεστή c είναι κάποιος μικρός θετικός πραγματικός αριθμός $0 < c < 1$. Για αυτές τις τιμές ο αλγόριθμος, που είναι γνωστός και με το όνομα Ho-Kashyap, συγκλίνει ακόμα και αν τα διανύσματα δεν είναι γραμμικά διαχωρίσιμα.

- ***Αρχικές τιμές***. Τοποθετούμε τυχαίους μικρούς θετικούς αριθμούς στους συντελεστές του διανύσματος ***b***.
- ***Υπολογισμός των συντελεστών της συνάρτησης απόφασης***. Με την σχέση που ακολουθεί πραγματοποιούμε μία εκτίμηση του ***w***: $\boldsymbol{w} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\,\boldsymbol{b}$.
- ***Ελεγχος σύγκλισης***. Αν $\boldsymbol{X}\,\boldsymbol{w} > \boldsymbol{0}$ τότε τα παραδείγματα είναι γραμμικά διαχωρίσιμα οπότε και ο αλγόριθμος τερματίζει. Αν η παραπάνω συνθήκη δεν ικανοποιείται και οι μεταβολές του ***w*** κατά δυο διαδοχικές προσεγγίσεις έχουν μειωθεί σημαντικά τότε έχουμε κατάσταση σύγκλισης του αλγόριθμου. Σε αυτή την περίπτωση γνωρίζουμε ότι τα παραδείγματα δεν είναι γραμμικά διαχωρίσιμα.
- ***Επαναπροσδιορισμός του b.*** Η νέα τιμή του διανύσματος ***b*** υπολογίζεται από την ακόλουθη αναδρομική σχέση: $\boldsymbol{b} = \boldsymbol{b} + \mathrm{c}\,(\,\boldsymbol{X}\,\boldsymbol{w} - \boldsymbol{b} + |\boldsymbol{X}\,\boldsymbol{w} - \boldsymbol{b}|)$.
- ***Επανάληψη του αλγόριθμου***. Το επαναληπτικό τμήμα του αλγόριθμου εκτελείται από το δεύτερο βήμα.

ΠΡΟΓΡΑΜΜΑ ΣΕ MATLAB

```
function [Rc,Rep] = HoKa(x1,x2,Lr,MaxRep)
%#
%#  [Rc,Rep] = HoKa(x1,x2,Lr,MaxRep)
%#
%#  Input
%#      x1: Pattern Vectors for the first class
%#      x2: Pattern Vectors for the second class
%#      Lr: Learning rate
%#      MaxRep: Maximum repetitions
%#  Output
%#      Rc: Correct classification rate using the C-method
%#      Rep: Pattern vectors on each class
%#
NumOfP1 = columns(x1) ;
x1 = [x1;ones(1,NumOfP1)] ;
NumOfP2 = columns(x2) ;
x2 = [x2;ones(1,NumOfP2)] ;
Rep = [NumOfP1,NumOfP2] ;
TotPat = sum(Rep) ;
Rc = zeros(2,1) ;
if ( rows(x1) ~= rows(x2) )
      printf( 'Error in vectors x1, x2\n' ) ;
end
z = [x1,-x2]';
Nv = columns(z) ;
Np = rows(z) ;
b = 0.1 * rand( Np, 1 ) ;
piz = inv( z' * z ) * z' ;
w = piz * b ;
b1 = z * w ;
e = b1 - b ;
i = 0 ;
while( i < MaxRep )
  printf( 'Step %d\n', i ) ;
  printf(     'Score:     %7.4f%%\n',(100*NoGreatValMat(b1,0.0))/(rows(b1)*
columns(b1)) ) ;
  if ( GreatValMat(b1,0.0) == 1 )
     printf( 'Linear Separation of classes in %d repetitions\n', i ) ;
     Rc = Rep ;
     return ;
  end
  ea = AbsMat(e) ;
  b = b + Lr * ( e + ea ) ;
  w = piz * b ;
  b1 = z * w ;
  e = b1 - b ;
  i++ ;
end
 printf( 'Original Ho-Kasyap not convergence in %d repetitions\n', i ) ;
 printf( 'Classification Score: %7.4f%%\n', ( 100 * NoGreatValMat( b1, 0.0
) ) / ( rows(b1) * columns(b1)) ) ;
Rc(1) = NoGreatValMat( b1(1:NumOfP1), 0.0 )  ;
Rc(2) = NoGreatValMat( b1(NumOfP1+1:TotPat), 0.0 )  ;
```

Κατανοήστε και εξηγήστε στον επιβλέποντα την δομή του προγράμματος, την χρήση και το περιεχόμενο των μεταβλητών του προγράμματος.

ΕΚΤΕΛΕΣΗ ΤΗΣ ΑΣΚΗΣΗΣ

1. Διάβασμα προτύπων από ιατρικών εργαστηριακές εξετάσεις για την διάγνωση ζακχαρώδη διαβήτη σε γυναίκες

Στο αρχείο `"\\ee-ntserver\PatRec\pima-indians-diabetes.data"` υπάρχουν σε φόρμα "ASCII" δεδομένα που προέρχονται από εξετάσεις αίματος γυναικών που έχουν εξετασθεί για πιθανή ασθένεια ζαχαρώδη διαβήτη. Οι μετρήσεις αυτές έχουν πραγματοποιηθεί σε 768 γυναίκες Ινδιάνους στις ΗΠΑ. Σε κάθε γραμμή του αρχείου υπάρχουν πληροφορίες που συσχετίζονται με κάθε εξεταζόμενη γυναίκα και περιέχει πληροφορίες που συσχετίζονται με τα συμπτώματα που ανιχνεύονται με φυσικές και εργαστηριακές εξετάσεις.

Οι τρείς πρώτες γραμμές του αρχείου είναι οι ακόλουθες:

```
6 148 72 35 0 33.6 0.627 50 1
1 85 66 29 0 26.6 0.351 31 0
8 183 64 0 0 23.3 0.672 32 1
```

Οι αριθμοί σε κάθε γραμμή παριστούν τα ακόλουθα στοιχεία:

Πεδίο	***Περιεχόμενο του πεδίου***
1	Αριθμός κυήσεων
2	Δοκιμασία φόρτισης του πλάσματος στο αίμα με γλυκόζη
3	Διαστολική πίεση του αίματος
4	Πάχος πτυχών του δέρματος στην περιοχή του τρικεφάλου
5	Συγκέντρωση ινσουλίνης στο αίμα
6	Μάζα σώματος
7	Συνάρτηση κληρονομικότητας
8	Ηλικία
9	Ταξινόμηση: 0 Υγιής, 1 διαβητικός

Τοποθετείστε τα παραδείγματα του αρχείου που αντιστοιχούν σε υγιείς στον πίνακα X και τα παραδείγματα που αντιστοιχούν σε ασθενείς στον πίνακα C.

2. Υπολογίστε τους συντελεστές των γραμμικών συναρτήσεων απόφασης με τον αλγόριθμο Ho-Kashyap.

Για να κατασκευάσετε ένα αυτόματο διαγνωστικό σύστημα για αυτόματη διάγνωση πάθησης διαβήτη χρησιμοποιώντας γραμμικές συναρτήσεις απόφασης πρέπει αρχικά να υπολογίσετε τους συντελεστές των γραμμικών συναρτήσεων. Επιλέξτε έναν συντελεστή εκπαίδευσης και υπολογίστε το διάνυσμα των συντελεστών της γραμμικής συνάρτησης απόφασης με την βοήθεια του αλγόριθμου Ho-Kashap.

3. Υπολογίστε το σφάλμα του συστήματος ταξινόμησης

Χρησιμοποιώντας όλα τα παραδείγματα εκπαίδευσης υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης.

4. Μελετήστε την επίδραση που έχει ο συντελεστής εκπαίδευσης στον αλγόριθμο Ho-kashyap

Κατασκευάστε την γραφική παράσταση που δίνει την αξιοπιστία του γραμμικού συστήματος ταξινόμησης για διαφορετικές τιμές του συντελεστή εκπαίδευσης. Κατασκευάστε επίσης και τον αριθμό των επαναλήψεων του αλγόριθμου σαν συνάρτηση του συντελεστή εκπαίδευσης.

5. Τι παρατηρείτε από τα αποτελέσματα των μετρήσεων

Περιγράψτε και αιτιολογείστε τα αποτελέσματα που πήρατε από τις μετρήσεις του ελάχιστου σφάλματος.

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 5

Ο ΑΛΓΟΡΙΘΜΟΣ HO-KASHYAP

Αριθμός Ομάδας:

Ονοματεπώνυμο:

<u>ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ</u>

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 6

ΤΟ ΠΟΛΥΕΠΙΠΕΔΟ ΔΙΚΤΥΟ PERCEPTRON ΚΑΙ Ο ΑΛΓΟΡΙΘΜΟΣ ΟΠΙΣΘΟΔΡΟΜΙΚΗΣ ΔΙΑΔΟΣΗΣ ΤΟΥ ΣΦΑΛΜΑΤΟΣ

1. Ο αλγόριθμος οπισθοδρομικής διάδοσης του σφάλματος στα νευρωνικά δίκτυα

Εστω ένα πολυεπίπεδο νευρωνικό δίκτυο που αποτελείται από νευρώνες perceptron.
Η συνάρτηση μεταφοράς κάθε νευρώνα αποτελείται από ένα γραμμικό και ένα μη γραμμικό τελεστή συνδεδεμένους σειριακά:

$$y_i = \sum_{j=1}^{Ni} w_{ij} x_j$$

$$o_i = f(y_i)$$

y_i , είναι η έξοδος του γραμμικού τελεστή στον νευρώνα I,
w_{ij} , είναι ο συντελεστής βαρύτητας της σύναψης που συνδέει την έξοδο του j νευρώνα με τον νευρώνα i.
x_j , είναι η στάθμη εξόδου του j νευρώνα.
o_i , είναι η στάθμη εξόδου του i νευρώνα.
f(x), είναι η μη-γραμμική συνάρτηση μεταφοράς του νευρώνα.

Ο υπολογισμός των συντελεστών βαρύτητας των συνάψεων οι οποίοι ελαχιστοποιούν το αθροιστικό σφάλμα από Ν αριθμό παραδειγμάτων πραγματοποιείται με τον αλγόριθμο οπισθοδρομικής διάδοσης του σφάλματος.
Αποδεικνύεται ότι αν η συνάρτηση του στιγμιαίου σφάλματος του ορίζεται σαν το τετράγωνο της Ευκλείδιας απόστασης της εξόδου του δικτύου από τις επιθυμούμενες τιμές για τυχαίο παράδειγμα (x_i, t_i) του μπορεί να μειωθεί όταν εφαρμόσουμε τον εξής επαναληπτικό αλγόριθμο.

$$E(W^{(t)}, x_i, t_i) = \sum_{j=1}^{M} (o_j - t_{ij})^2$$

$W^{(t)}$, είναι το διάνυσμα των συντελεστών βαρύτητας των συνάψεων του δικτύου.
o_j, είναι η j έξοδος του νευρώνα που βρίσκεται στο επίπεδο εξόδου.
t_{ij},είναι η j επιθυμούμενη έξοδος του i παραδείγματος.
x_i,είναι το διάνυσμα εισόδου στο δίκτυο γιά το i παράδειγμα.

- ***Αρχικές τιμές***. Τοποθετούμε τυχαίους μικρούς θετικούς αριθμούς στους συντελεστές βαρύτητας όλων των συνάψεων.
- ***Επιλογή τυχαίου παραδείγματος***. Με τυχαίο τρόπο επιλέγω ένα παράδειγμα εκπαίδευσης.
- ***Υπολογισμός στάθμης εξόδου σε κάθε νευρώνα του δικτύου.*** Υπολογίζω από χαμηλότερα επίπεδα πρός το επίπεδο εξόδου την εσωτερική κατάσταση κάθε νευρώνα και την στάθμη εξόδου του.
- ***Υπολογισμός των συντελεστών δ των νευρώνων.*** Υπολογίζω από το επίπεδο εξόδου πρός χαμηλότερα επίπεδα τους συντελεστές δ κάθε νευρώνα.
- ***Επαναπροσδιορισμός όλων των συντελεστών βαρύτητας των νευρώνων.*** Υπολογίζω τους νέους συντελεστές βαρύτητας όλων των συνάψεων του δικτύου.
- ***Ελεγχος σύγκλισης.*** Ελέγω τον ρυθμό εκπαίδευσης του δικτύου γιά να διαπιστώσω αν το νευρωνικό δίκτυο μαθαίνει. Αν οι μεταβολές των συντελεστών βαρύτητας παραμένει υψηλός τότε ο αλγόριθμος επαναλαμβάνεται από το δεύτερο βήμα, διαφορετικά τερματίζει.

2. Υπολογισμός των συντελεστών δ των νευρώνων

Ο συντελεστής δ ορίζεται σαν την αρνητική παράγωγο της συνάρτησης του στιγμιαίου σφάλματος ως προς την εσωτερική κατάσταση του νευρώνα:

$$\delta_k = -\frac{\partial}{\partial y_k} E(W^{(t)}, x_i, t_i)$$

y_k, είναι η εσωτερική κατάσταση του k νευρώνα
Για τους νευρώνες εξόδου αποδυκνύεται ότι ο συντελεστής δ είναι:

$$\delta_k = -(t_{ik} - o_k)\frac{\partial f(o_k)}{\partial y_k}$$

Για τους κρυφούς νευρώνες αποδυκνύεται ότι ο συντελεστής δ είναι:

$$\delta_k = \frac{\partial f(o_k)}{\partial y_k}\sum_{m=1}^{M}\delta_m w_{mk}$$

Οπου δ_m, είναι η εσωτερική κατάσταση του m νευρώνα που βρίσκεται στο αμέσως υψηλότερο επίπεδο, και
w_{mk}, είναι ο συντελεστής βαρύτητας της σύναψης που συνδέει την έξοδο του νευρώνα k με τον νευρώνα m.

4. Επαναπροσδιορισμός των συντελεστών βαρύτητας των συνάψεων

Σε όλες τις περιπτώσεις οι νέοι συντελεστές βαρύτητας των συνάψεων δίνονται από την σχέση:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} + n\delta_i o_j$$

ΠΡΟΓΡΑΜΜΑ ΣΕ MATLAB

```
function [Recs,Sums] = MlpEbp1LBias(x,c,HidNodes,Lr,Tr,Step)
%#
%#  [Recs,Sums] = MlpEbp1LBias(x,c,HidNodes,Lr,Tr,Step)
%#      Back-propagation of error. One hidden layer with bias inputs
%#  Input
%#      x: Pattern Vectors
%#      c: Classes
%#      HidNodes: Number of Hidden nodes
%#      Lr: Learning rate
%#      Tr: Training epoches
%#      Step: Classification accuracy and Estimation Error
%#            after "Step" training cycles
%#  Output
%#      Recs: Correct classification rate using the C-method
%#      Sums: Error Function
NumOfClass = max(c) ;
NumOfPatterns = columns(x) ;
x = [ x ; 0.5 * ones( 1, NumOfPatterns ) ] ;
Dimens = rows(x) ;
Rec = zeros( 1, Tr/Step+1) ;
Sums = zeros( 1, Tr/Step+1) ;
HidWeights = rand(Dimens,HidNodes+1) - 0.5 ;
OutWeights = rand( HidNodes+1, NumOfClass ) - 0.5 ;
HidOut = 0.5 * ones( 1, HidNodes+1 ) ;
HidDelta = zeros( 1, HidNodes+1) ;
Out = zeros( 1, NumOfClass ) ;
OutDelta = zeros( 1, NumOfClass ) ;
mr = 0 ;
for m = 0:Tr
      Nxv = floor(rand(1) * NumOfPatterns ) + 1 ;
      Xinp = x(:,Nxv)' ;
      for j = 1:HidNodes
            HidOut(1,j) = 1 / ( exp(-(Xinp * HidWeights(:,j))) + 1 ) ;
      end
      for j = 1:NumOfClass
            Out(1,j) = 1 / ( exp(-(HidOut * OutWeights(:,j)) ) + 1 ) ;
      end
      Desired = zeros( 1, NumOfClass ) + 0.1 ;
      Desired(1,c(Nxv)) = 0.9 ;
      OutDelta = ( Desired - Out ) .* Out .* ( 1 - Out ) ;
      HidDelta = HidOut .* ( 1 - HidOut ) .* ( OutWeights * OutDelta' )' ;
      OutWeights = OutWeights + Lr * HidOut' * OutDelta ;
      HidWeights = HidWeights + Lr * Xinp' * HidDelta ;
      if( rem( m, Step ) == 0 )
            mr = mr + 1 ;
            Rc = zeros(NumOfClass,1) ;
            Sum = 0 ;
            for i = 1:NumOfPatterns
                  k = c(i) ;
                  Xinp = x(:,i)' ;
                  for j = 1:HidNodes
                        HidOut(1,j)=1/(exp(-(Xinp*HidWeights(:,j)))+1) ;
                  end
                  for j = 1:NumOfClass
                        Out(1,j)=1/(exp(-(HidOut*OutWeights(:,j)))+1) ;
                  end
                  for j = 1:NumOfClass
                        if ( k == j )
                              Sum = Sum + ( 0.9 - Out(1,j) )^2 ;
```

```
                        else
                              Sum = Sum + ( Out(1,j) - 0.1 )^2 ;
                        end
                  end
                  Rec = ArgMax(Out) ;
                  if (Rec == k )
                        Rc(Rec) = Rc(Rec) + 1 ;
                  end
            end
      Sums(1,mr) = Sum ;
      TotRec = sum(Rc') ;
      Recs(1,mr) = 100.0 * TotRec / NumOfPatterns ;
      if ( TotRec == NumOfPatterns )
            return ;
      end
      end
end
```

Κατανοήστε και εξηγήστε στον επιβλέποντα την δομή του προγράμματος, την χρήση και το περιεχόμενο των μεταβλητών του προγράμματος.

ΕΚΤΕΛΕΣΗ ΤΗΣ ΑΣΚΗΣΗΣ

1. Διάβασμα προτύπων από χημική ανάλυση διαφορετικών τύπων κρασιού

Στο αρχείο `"\\ee-ntserver\PatRec\wine.data"` υπάρχουν σε φόρμα "ASCII" δεδομένα που προέρχονται από χημική ανάλυση διαφορετικών τύπου κρασιών που προέχονται από την ίδια περιοχή της Ιταλίας αλλά από τρείς διαφορετικές καλιέργειες. Οι χημικές αναλύσεις περιγράφουν την περιεκτικότητα του κάθε δείγματος κρασιού σε δεκατρία διαφορετικά συστατικά του.

Οι τρείς πρώτες γραμμές του αρχείου είναι οι ακόλουθες:

```
1 14.23 1.71 2.43 15.6 127 2.8 3.06 .28 2.29 5.64 1.04 3.92 1065
1 13.2 1.78 2.14 11.2 100 2.65 2.76 .26 1.28 4.38 1.05 3.4 1050
1 13.16 2.36 2.67 18.6 101 2.8 3.24 .3 2.81 5.68 1.03 3.17 1185
```

Οι αριθμοί σε κάθε γραμμή παριστούν τα ακόλουθα στοιχεία:

Πεδίο	***Περιεχόμενο του πεδίου***
1	Ταξινόμηση καλιεργειών σταφυλιού σε τύπο 1,2,3
2-14	Μετρήσεις χημικής σύστασης

Τοποθετείστε τα παραδείγματα του αρχείου που αντιστοιχούν σε πίνακα δεδομένων X και την κατηγορία κάθε πρότυπου στον πίνακα C.

2. Εκπαιδεύστε πολυεπίπεδο νευρωνικό δίκτυο το οποίο να ταξινομεί την προέλευση του κρασιού.

Για να κατασκευάσετε ένα αυτόματο διαγνωστικό σύστημα το οποίο να ταξινομεί την κατηγορία σταφυλιών από την οποία παρασκευάστηκε το κρασί πρέπει να υπολογίσετε τους συντελεστές βαρύτητας των συνάψεων ενός νευρωνικού δικτύου. Κατασκευάστε δίκτυο δύο επιπέδων το οποίο να διαθέτει στο κρυφό επίπεδο πέντε νευρώνες. Επιλέξτε έναν συντελεστή εκπαίδευσης και εκπαιδεύστε το δίκτυο. Χρησιμοποιώντας όλα τα παραδείγματα εκπαίδευσης υπολογίστε το ελάχιστο σφάλμα του συστήματος ταξινόμησης.

3. Μελετήστε την επίδραση που έχει ο συντελεστής εκπαίδευσης στον αλγόριθμος οπισθοδρομικής διάδοσης του σφάλματος.

Για να μελετήσετε την επίδραση που έχει η επιλογή του συντελεστή εκπαίδευσης στον αλγόριθμο κατασκευάστε την γραφική παράσταση που δίνει την αξιοπιστία του νευρωνικού γραμμικού συστήματος ταξινόμησης για διαφορετικές τιμές του συντελεστή εκπαίδευσης.

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ

ΑΝΑΓΝΩΡΙΣΗ ΠΡΟΤΥΠΩΝ Ι

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 6

ΤΟ ΠΟΛΥΕΠΙΠΕΔΟ ΔΙΚΤΥΟ PERCEPTRON ΚΑΙ Ο ΑΛΓΟΡΙΘΜΟΣ ΟΠΙΣΘΟΔΡΟΜΙΚΗΣ ΔΙΑΔΟΣΗΣ ΤΟΥ ΣΦΑΛΜΑΤΟΣ

Αριθμός Ομάδας:

Ονοματεπώνυμο:

<u>ΜΕΤΡΗΣΕΙΣ ΚΑΙ ΠΑΡΑΤΗΡΗΣΕΙΣ</u>