



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

Αλέξανδρος Αριστόβουλος

1063199

Εργαστηριακή άσκηση 5

Ερώτημα 1:

Διαβάζουμε τα παραδείγματα με τη συνάρτηση ReadDiabetes ()

```
Tot=768;  
[x,c] = ReadDiabetes (Tot);
```

Τοποθετούμε τα παραδείγματα στο πίνακα των υγιών αν έχουν την τιμή 1 στο πίνακα c ενώ τα τοποθετούμε στον πίνακα των αρρώστων αν έχουν τη τιμή 2.

```
% αριθμός υγιών  
y_counter=1;  
% αριθμός αρρώστων  
a_counter=1;  
  
for k=1:Tot  
    if c(k)==1  
        %πίνακας με τους υγιείς  
        y_arr(:,y_counter)=x(:,k);  
        y_counter=y_counter+1;  
  
    else  
        %πίνακας με τους αρρώστους  
        a_arr(:,a_counter)=x(:,k);  
        a_counter=a_counter+1;  
    end  
end
```

Ερώτημα 2:

Για να βρούμε τους συντελεστές κάναμε τη συνάρτηση του Ho-Kashyap να επιστρέφει τα w.

```
[Rc,Rep,w] = HoKa(y_arr,a_arr,0.5,200);  
disp("Συντελεστές των γραμμικών συναρτήσεων απόφασης")  
disp(w)
```

Συντελεστές των γραμμικών συναρτήσεων απόφασης

```
-0.0026  
-0.0008  
0.0003  
0.0000  
0.0000  
-0.0022  
-0.0182  
-0.0002  
0.1953
```

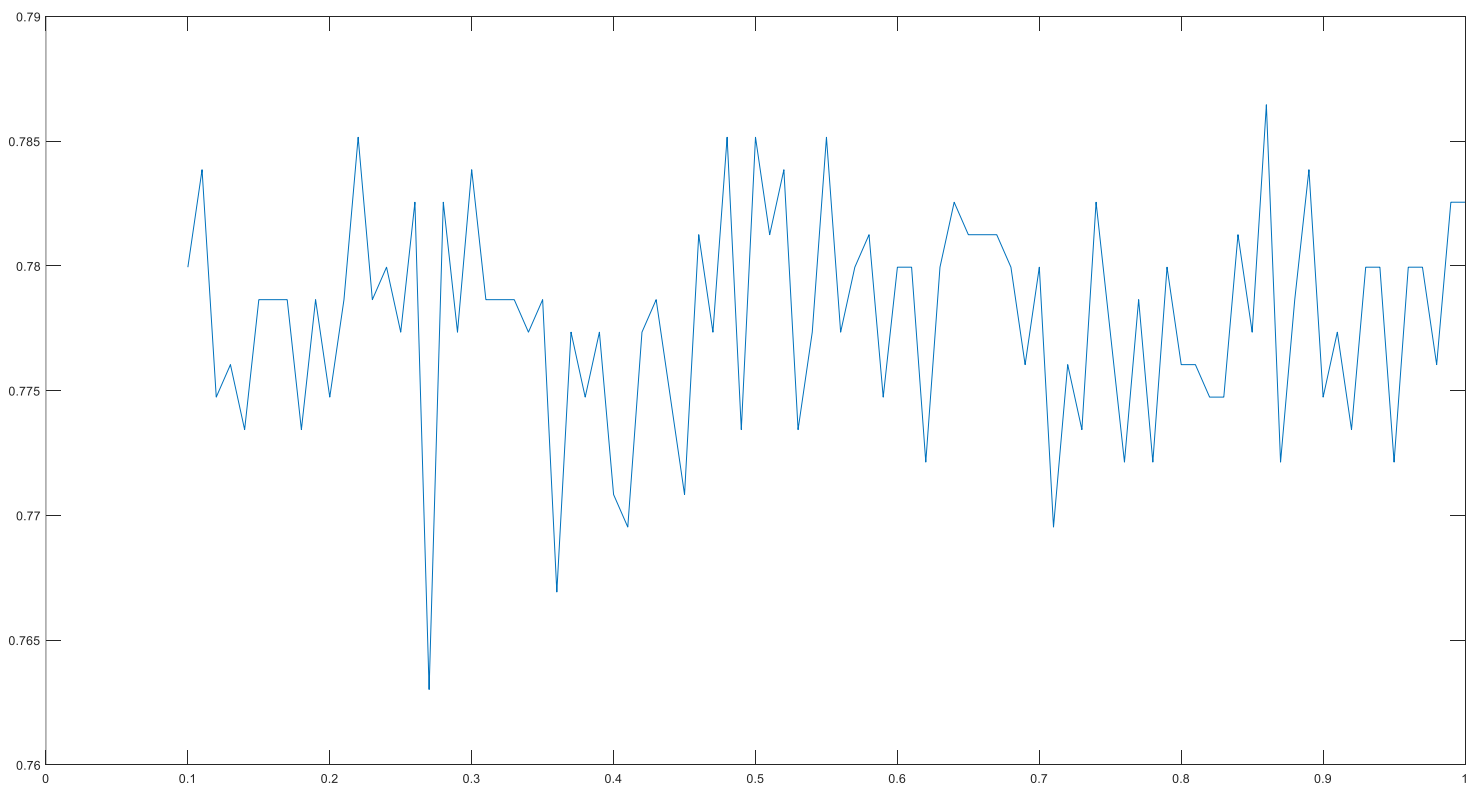
Ερώτημα 3:

Υπολογίζουμε το σφάλμα με τη χρήση των R_c και R_p

```
error=1-((Rc(1)+Rc(2))/(Rep(1)+Rep(2)));  
disp("Σφάλμα του συστήματος ταξινόμησης")  
disp(error)
```

Ερώτημα 4:

Για να κατασκευάσουμε την γραφική παράσταση που δίνει την αξιοπιστία του γραμμικού συστήματος ταξινόμησης για διαφορετικές τιμές του συντελεστή εκπαίδευσης καλούμε επαναληπτικά τη συνάρτηση Ho-Kashygar για διαφορετικούς συντελεστές και τυπώνουμε τα αποτελέσματα αξιοπιστίας.



```

counter=1;
max_rel=0;
for a=1:-0.01:0.1
    [Rc,Rep,~] = HoKa(y_arr,a_arr,a,100);
    reliability=(Rc(1)+Rc(2))/(Rep(1)+Rep(2));
    if reliability>max_rel
        %μαξ αξιοπιστία
        max_rel=reliability;
    end
    rel_arr(counter)=reliability;
    sintelestis(counter)=a;
    counter=counter+1;
end
figure('Name','Αξιοπιστία του γραμμικού συστήματος ταξινόμησης
συναρτήσεως του συντελεστή εκπαίδευσης','NumberTitle','off')
plot(sintelestis,rel_arr)

```

Για το διάγραμμα του αριθμού των επαναλήψεων του αλγόριθμου συναρτήσεως του συντελεστή εκπαίδευσης δημιουργήσαμε νέα συνάρτηση Ho-Kashgar η οποία πέρα από τα Rc, Rep, w επιστρέφει και τα σφάλματα της συνάρτησης καθώς και των αριθμό των επαναλήψεων.

Η συνάρτηση HoKa1 είναι η εξής:

```

function [Rc,Rep,w,sf_Hoka,i] = HoKa1(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) )
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 ;
k=mod(MaxRep ,5000);
%if k==0
while( i < MaxRep )

    % printf( 'Step %d\n', i ) ;
    %printf( 'Classification 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=i+1 ;
    Rc(1) = NoGreatValMat( b1(1:NumOfP1), 0.0 ) ;

    Rc(2) = NoGreatValMat( b1(NumOfP1+1:TotPat), 0.0 ) ;
    sf_Hoka(i)=1-((Rc(1)+Rc(2))/(Rep(1)+Rep(2)));
end

```

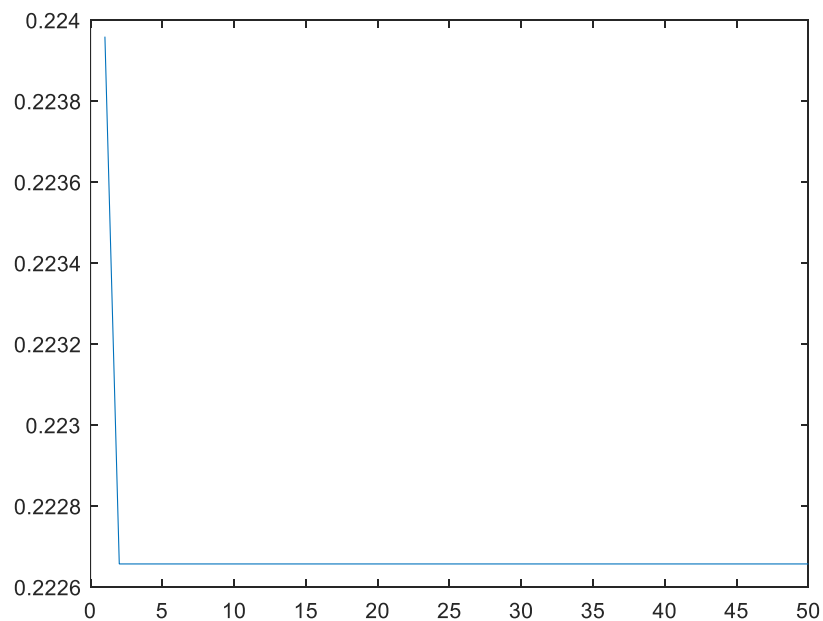
Την καλούμε ως εξής:

```

[Rc,Rep,w,sf_Hoka,i] = HoKal(y_arr,a_arr,max_rel,50);
figure('Name','Αριθμός των επαναλήψεων του αλγόριθμου συναρτήσει του
συντελεστή εκπαίδευσης','NumberTitle','off')
plot(1:1:50, sf_Hoka)

```

και παίρνουμε την εξής γραφική:



Παρατηρούμε ότι έπειτα από 2 επαναλήψεις το σφάλμα σταματά να μειώνεται.

Ερώτημα 5:

- Ο Ho-Kashgar βγάζει πολύ πιο αξιόπιστα αποτελέσματα από το Perceptron.
- Ο Ho-Kashgar βασίζεται λιγότερο στην τυχαιότητα από ότι ο Perceptron καθώς τα αποτελέσματα βγαίνουν πολύ πιο κοντά κάθε φορά.
- Ο Ho-Kashgar είναι πολύ πιο γρήγορος καθώς χρειάζονται πολύ λιγότερες επαναλήψεις για να φτάσουμε σε ελαχιστοποίηση του σφάλματος.