



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

**Αλέξανδρος Αριστόβουλος**

**1063199**

## **Εργαστηριακή άσκηση 2**

### **Ερώτημα 1:**

Σε αυτή τη συνάρτηση προσθέτουμε τις τιμές των μετρήσεων ανά κατηγορία και έπειτα διαιρούμε με τον αριθμό των μετρήσεων της κάθε κατηγορίας. Έτσι βρίσκουμε τη μέση τιμή της κάθε μέτρησης για κάθε κατηγορία.

```
function [centers,lenOfClasses] = CalculateVirtualCenter(x,c)
%#
%# [centers] = CalculateVirtualCenter(x,c)
%#
%# Input
%# x: Pattern Vectors
%# c: Classes
%# Output
%# centers: the calculated ideal center for each property for each category
%# lenOfClasses: the number of occurrences for each class

NumOfClass = max(c) ;
[rownum,colnum]=size(x);

%create the array where we will store the average values
centers = zeros(rownum,NumOfClass);

%add every number of the rows of each case in one cell
for i = 1:rownum
    for j = 1:colnum
        centers(i,c(j)) = centers(i,c(j)) + x(i,j);
    end
end

%get the number of occurrences for each class
lenOfClasses = zeros(1,NumOfClass);
for index = 1:NumOfClass
    lenOfClasses(index) = sum(c==index);
end
```

```

%divide with each case with the number of occurrences to get the average
for index = 1:NumOfClass
    %dont divide if we have 0 occurrences
    if lenOfClasses(index) ~= 0
        centers(:,index) = centers(:,index)./lenOfClasses(index);
    end
end
end

```

Τιμές εικονικού προτύπου:

1.5187	1.5186	1.5180	0	1.5189	1.5175	1.5171
13.2423	13.1117	13.4371	0	12.8277	14.6467	14.4421
3.5524	3.0021	3.5435	0	0.7738	1.3056	0.5383
1.1639	1.4082	1.2012	0	2.0338	1.3667	2.1228
72.6191	72.5980	72.4047	0	72.3662	73.2067	72.9659
0.4474	0.5211	0.4065	0	1.4700	0	0.3252
8.7973	9.0737	8.7829	0	10.1238	9.3567	8.4914
0.0127	0.0503	0.0088	0	0.1877	0	1.0400
0.0570	0.0797	0.0571	0	0.0608	0	0.0134

## Ερώτημα 2:

Χρησιμοποιούμε τη συνάρτηση ClassMinDistEuclOne για να βρούμε τις σωστές ταξινομήσεις με τη μέθοδο C και U. Το σφάλμα για τη κάθε μέθοδο ισούται με 1 μείον το άθροισμα των σωστών ταξινομήσεων της μεθόδου διά τον αριθμό των παραδειγμάτων.

```

disp("Ερώτημα 2")
[Rc,Ru,Rep] = ClassMinDistEuclOne(x,c);
sumRc = 0;
sumRu = 0;
for i = 1:length(Rc)
    sumRc = sumRc + Rc(i);
    sumRu = sumRu + Ru(i);
end
minError = 1-sumRc/length(x);
maxError = 1-sumRu/length(x);
disp("Ελάχιστο σφάλμα του συστήματος ταξινόμησης")
disp(minError)
disp("Μέγιστο σφάλμα του συστήματος ταξινόμησης")
disp(maxError)

```

Ερώτημα 2

Ελάχιστο σφάλμα του συστήματος ταξινόμησης (μέθοδος C)  
0.5187

Μέγιστο σφάλμα του συστήματος ταξινόμησης (μέθοδος U)  
0.5514

### Ερώτημα 3:

Σε αυτή τη συνάρτηση παίρνουμε σαν κέντρο της κατηγορίας το παράδειγμα που απέχει λιγότερη συνολική απόσταση από τα υπόλοιπα παραδείγματα της ίδιας κατηγορίας.

```
function [centers,indexes] = FindBestCenterInExamples(x,c)
%#
%# [centers] = CalculateVirtualCenter(x,c)
%#
%# Input
%# x: Pattern Vectors
%# c: Classes
%# Output
%# centers: the center (taken from the examples) for each category
%# indexes: the position in x where we find the best example

NumOfClass = max(c) ;
[rownum,colnum]=size(x);

%get the number of occurrences for each class
lenOfClasses = zeros(1,NumOfClass);
for index = 1:NumOfClass
    lenOfClasses(index) = sum(c==index);
end

%this is so we know where each category starts and ends
startFlags = zeros(NumOfClass+1,1);
%first element always starts at 1
startFlags(1) = 1;
for i = 2:NumOfClass+1
    startFlags(i) = startFlags(i-1) + lenOfClasses(i-1);
end

%this is where we store the minimum distance for each category
minDistance = zeros(NumOfClass,1);

%find the indexes where we have the minimum distance from each example of
%each category
indexes = zeros(NumOfClass,1);
for i=1:colnum
    distance = 0;
    for j=startFlags(c(i)):startFlags(c(i)+1)-1
        distance = distance + (x(:,i) - x(:,j))' * (x(:,i) - x(:,j));
    end
    %if you find an example with less distance from the other examples in
    %this category update the best example and the minimum distance
    if minDistance(c(i),1) > distance || minDistance(c(i),1) == 0
        indexes(c(i),1) = i;
        minDistance(c(i),1) = distance;
    end
end
```

end

%set the centers while ignoring the non existant category 4

centers = zeros(rownum, NumOfClass);

for i = 1:NumOfClass

if i ~= 4

centers(:,i) = x(:,indexes(i,1));

end

end

### Ερώτημα 3

Τιμές προτύπου από παραδείγματα:

1.5187	1.5181	1.5166	0	1.5206	1.5189	1.5172
13.1900	12.9600	13.4100	0	12.8500	14.9900	14.7500
3.3700	2.9600	3.3900	0	1.6100	0.7800	0
1.1800	1.4300	1.2800	0	2.1700	1.7400	2.0000
72.7200	72.9200	72.6400	0	72.1800	72.5000	73.0200
0.5700	0.6000	0.5200	0	0.7600	0	0
8.8300	8.7900	8.6500	0	9.7000	9.9500	8.5300
0	0.1400	0	0	0.2400	0	1.5900
0.1600	0	0	0	0.5100	0	0.0800

Θέσεις των ιδανικών παραδειγμάτων για κάθε κατηγορία:

47  
100  
157  
0  
175  
182  
194

Για τη κατηγορία 4 δεν έχουμε παραδείγματα και για αυτό η θέση του κέντρου-παραδείγματος είναι 0.

### Ερώτημα 4:

Δημιουργούμε τη συνάρτηση ClassMinDistEuclTwo για να βρούμε τις σωστές ταξινομήσεις με τη μέθοδο C και U. Αντίθετα με τη ClassMinDistEuclOne δεν χρησιμοποιούμε εικονικά κέντρα αλλά το καλύτερο παράδειγμα σαν κέντρο. Για αυτό το λόγο καλούμε μέσα της τη συνάρτηση FindBestCenterInExamples που προαναφέρθηκε. Το σφάλμα για τις κάθε μεθόδους C και U ισούται με 1 μείον το άθροισμα των σωστών ταξινομήσεων της κάθε μεθόδου διά τον αριθμό των παραδειγμάτων.

function [Rc,Ru,Rep] = ClassMinDistEuclTwo(x,c)

%# This is to return correct classifications using examples as centers

%# [Rc,Ru,Rep] = ClassMinDistEuclOne(x,c)

%# Pattern Recognition:

%# Distance measure: Euclidian

%# 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] = size(x) ;
[centers,~] = FindBestCenterInExamples(x,c);
Rep = zeros(NumOfClass,1) ;

```

```

%#
%# C-Error
%#

```

```

Rc = zeros(NumOfClass,1) ;
for i = 1:NumOfPatterns
    for j = 1:NumOfClass
        Dist(j) = (x(:,i) - centers(:,j))' * ( x(:,i) - centers(:,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) ;
    xRemoved = x;
    xRemoved(:,j) = [];
    cRemoved = c;
    cRemoved(:,j) = [];
    [centers,~] = FindBestCenterInExamples(xRemoved,cRemoved);
    for i = 1:NumOfClass
        Dist(i) = (x(:,j) - centers(:,i))' * ( x(:,j) - centers(:,i) ) ;
    end
    Rec = ArgMin(Dist) ;
    if (Rec == k)
        Ru(Rec) = Ru(Rec) + 1 ;
    end
end

```

end

#### Ερώτημα 4

Ελάχιστο σφάλμα του συστήματος ταξινόμησης (μέθοδος C)

0.5748

Μέγιστο σφάλμα του συστήματος ταξινόμησης (μέθοδος U)

0.5841

#### Ερώτημα 5:

Παρατηρούμε ότι:

- Το σφάλμα της μεθόδου U είναι πάντα μεγαλύτερο από το σφάλμα της μεθόδου C αφού λαμβάνουμε υπόψιν μας ένα λιγότερο παράδειγμα στην εκπαίδευση.
- Το σφάλμα χρησιμοποιώντας σαν κέντρα τα παραδείγματα είναι μεγαλύτερο από ότι αν υπολογίσουμε εικονικό κέντρο. Αυτό συμβαίνει γιατί είναι σχεδόν αδύνατο κάποιο παράδειγμα να είναι στη θέση του εικονικού κέντρου. Ακόμα, όσο πιο πολύ διασκορπισμένα είναι τα παραδείγματα τότε τόσο πιο μακριά θα είναι το κοντινότερο παράδειγμα από το εικονικό κέντρο. Έτσι είναι πιο πιθανό να ταξινομήσουμε λάθος τα παραδείγματα λόγω αυτής της απόκλισης.