

# **ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

## **ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ** **ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

***ΡΟΗ Σ***

9ο Εξάμηνο

**“Αναγνώριση Προτύπων”**

Προπαρασκευή 3ης Εργαστηριακής Άσκησης:  
***Αναγνώριση Συναισθήματος στη Μουσική***

**ΜΑΡΟΥΓΚΑΣ ΙΣΙΔΩΡΟΣ 03113048**

**ΜΠΑΚΟΣ ΠΑΝΑΓΙΩΤΗΣ 03113014**

*Ακαδημαϊκό Έτος 2017-2018*

## Εισαγωγή

Σκοπός της άσκησης αυτή είναι η υλοποίηση ενός συστήματος αυτόματης ταξινόμησης μουσικών αποσπασμάτων βάσει του συναισθήματος που το απόσπασμα αυτό επιδιώκει να προκαλέσει στο δέκτη.

Στο πλαίσιο αυτό, το συναίσθημα αναπαριστάται με τη χρήση 2 χαρακτηριστικών σε raw μορφή με τα μεγέθη Χαρά/Λύπη (Valence) και Ενεργοποίηση-/Απενεργοποίηση (Activation) κι αυτό καθώς οφείλονται σε διεγέρσεις διαφορετικών περιοχών νευρώνων του εγκεφάλου.

## Εκτέλεση

### **Βήμα 1**

Ζητείται να εισάγουμε τα δεδομένα στο Matlab, και να τα αποθηκεύσουμε εκ νέου αλλάζοντας το format τους.

Συγκεκριμένα, ζητείται να κάνουμε subsampling, να μειώσουμε τα bits/sample στο μισό, καθώς και να μετατρέψουμε το αρχείο από stereo σε mono. Για την τελευταία μετατροπή: χρησιμοποιούμε τη μέση τιμή των δύο καναλιών της στερεοφωνικής εκτέλεσης, ενώ για τη μείωση της δειγματοληψίας και του ρυθμού κβάντισης τις εντολές decimate και τα ορίσματα της audiowrite του Matlab, αντιστοίχως.

```
%% Step 1 - Pre-process Data
basepath = './PRcourse_Lab3_data/PRcourse_Lab3_data/MusicFileSamples/';
music_contents = folder_load(basepath);

mkdir('./PRcourse_Lab3_data/PRcourse_Lab3_data/MonoSamples')
for i=1:size(music_contents,1)
    cname = music_contents(i).name;
    full_name = strcat(basepath,cname);
    [stereo,Fs] = audioread(full_name);

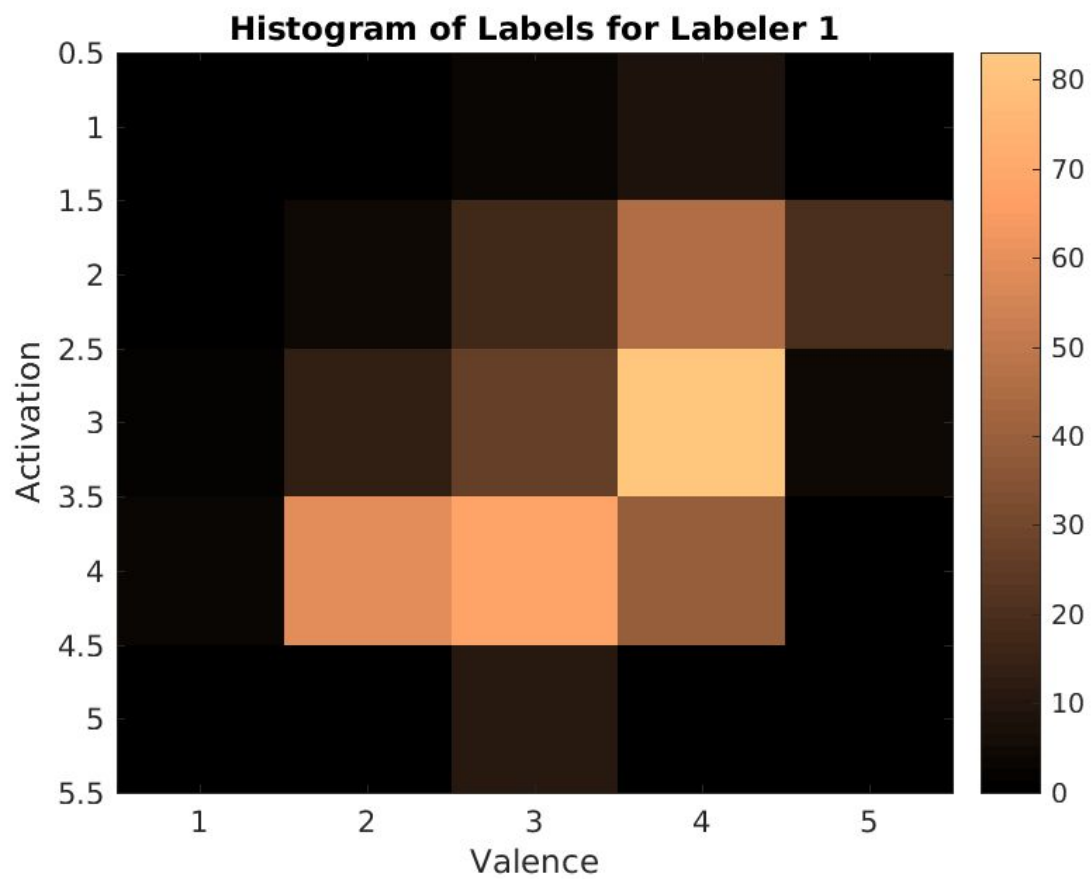
    % From fs=44.10 kHz, 16 bits/sample, stereo
    % To fs=22.05 kHz, 8 bits/sample, mono
    mono = (stereo(:,1) + stereo(:,2))/2; % stereo->mono
    mono = decimate(mono,2); % 44.10->22.05kHz
    % 16->8 bits per sample
    audiowrite(['./PRcourse_Lab3_data/MonoSamples\',cname],mono,Fs/2,'BitsPerSample',8);
end
```

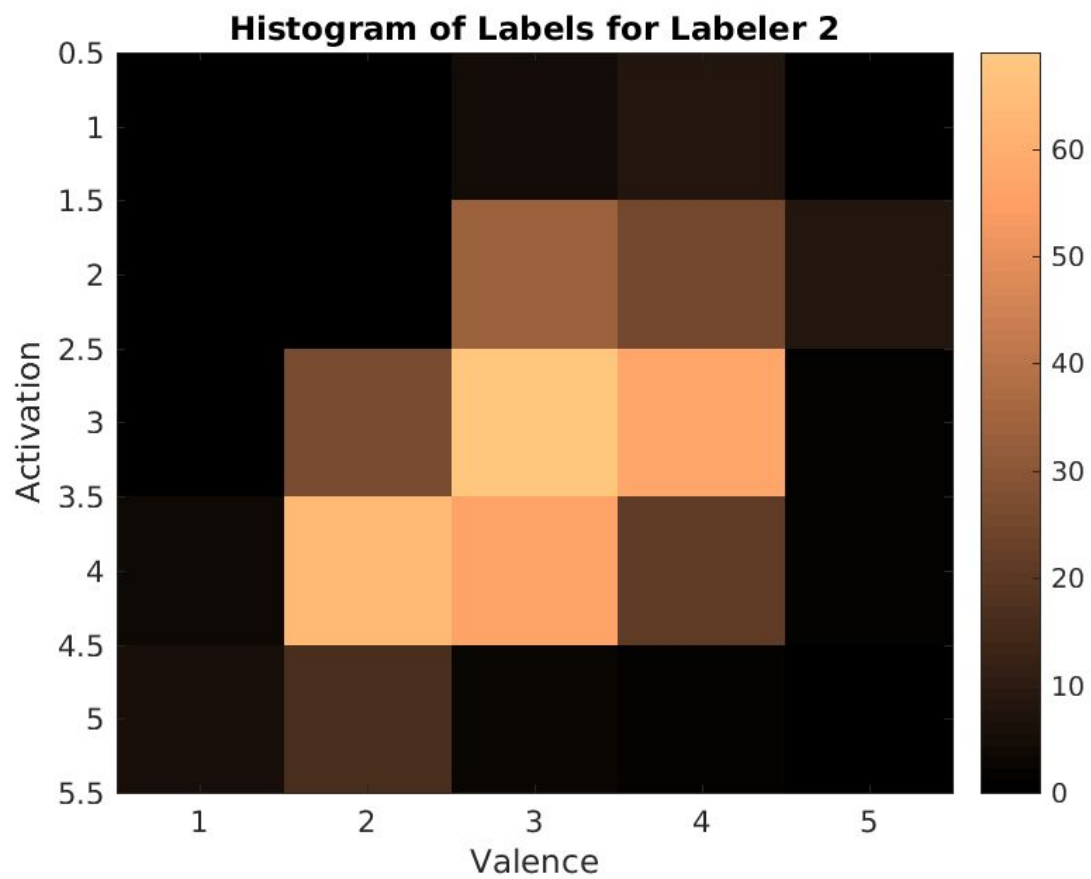
## Βήμα 2

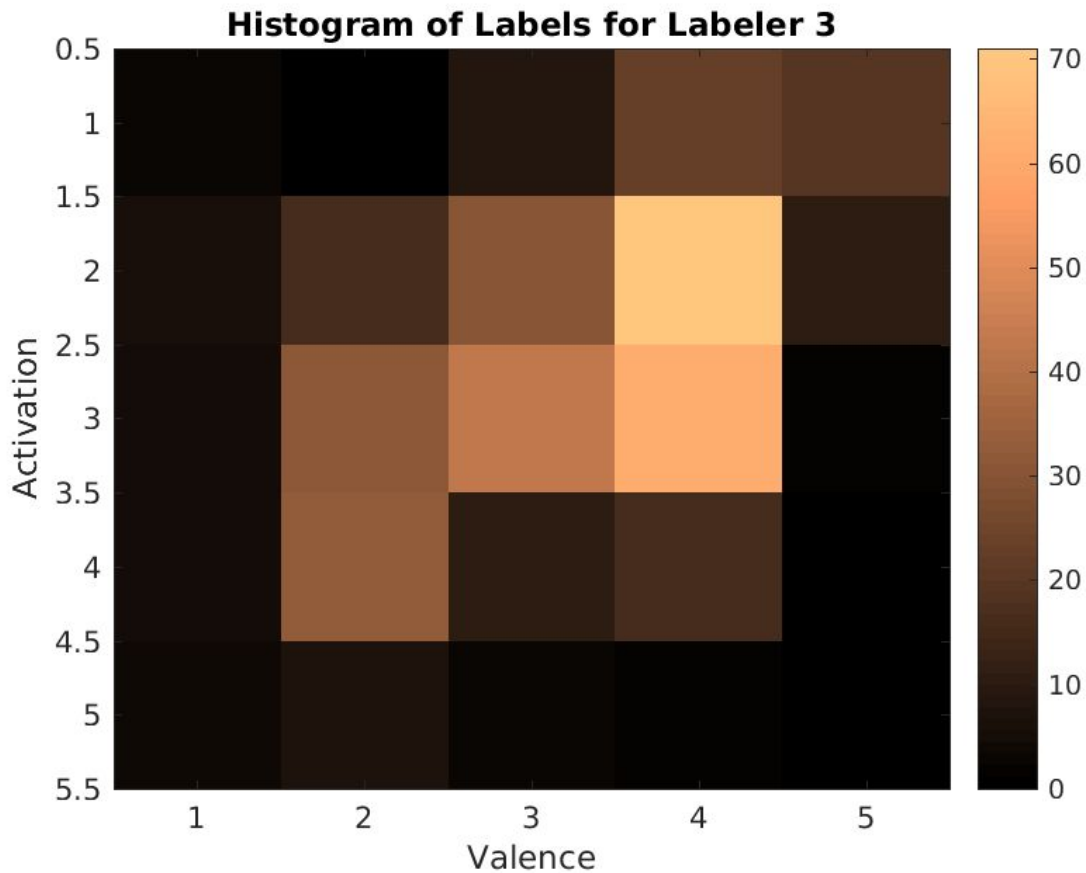
Ζητείται να υπολογίσουμε τη μέση τιμή και την τυπική απόκλιση του Valence και του Activation για κάθε έναν από τους τρεις επισημειωτές Labeler1, Labeler2, Labeler3 που μας δίνονται.

Ακολουθούν τα αποτελέσματα καθώς και η αναπαράσταση των co-occurence πινάκων σε 2D ιστογράμματα.

<u>Valence</u>		<u>Activation</u>	
Mean	Std	Mean	Std
3.3447	0.8921	3.2015	0.9152
3.0316	0.8721	3.2464	0.9283
3.2524	1.0583	2.659	1.0173







**Ο πίνακας co-occurrence** είναι ένας πίνακας ο οποίος υποδεικνύει πόσες φορές τα δύο μεγέθη που αντιστοιχούν στις δύο διαστάσεις του λαμβάνουν ένα δεδομένο ζεύγος τιμών.

Βάσει αυτών, παρατηρούμε ότι ενώ οι τρεις επισημειωτές έχουν παρόμοια εικόνα (δηλαδή υψηλό αριθμό εμφανίσεων κυρίως των μεσαίων τιμών), δε δίνουν όλοι ακριβώς το ίδιο αποτέλεσμα.

```
%% Step 2 - Per Labeler Statistics
basepath = './PRcourse_Lab3_data/PRcourse_Lab3_data/EmotionLabellingData/';
lab_file = folder_load(basepath);
for j=1:1:size(lab_file,1)
    cname = lab_file(j).name;
    full_name = strcat(basepath,cname);
    curr_labeler = load(full_name);
    curr_labeler = curr_labeler.labelList;
    Labelers(:,j) = curr_labeler;
end
```

```

% a) mean and b) standard deviation calculation
mean_act(j) = mean(cell2mat({curr_labeler.activation}));
std_act(j) = std(cell2mat({curr_labeler.activation}));
mean_val(j) = mean(cell2mat({curr_labeler.valence}));
std_val(j) = std(cell2mat({curr_labeler.valence}));

% c) co-occurrence matrix & histogram
temp_co_occur = zeros(5,5);
for k=1:1:size(curr_labeler,1)
    curr_act = curr_labeler(k).activation;
    curr_val = curr_labeler(k).valence;
    temp_co_occur(curr_act,curr_val) = temp_co_occur(curr_act,curr_val) + 1;
end
co_occur{jj} = temp_co_occur;

figure;imagesc(temp_co_occur);
colormap('copper');colorbar;
xlabel('Valence');ylabel('Activation');
title(['Histogram of Labels for Labeler ',num2str(jj)]);
title1 = ['histogram_',num2str(jj),'.png'];
saveas(gcf,title1);
end

```

### Βήμα 3

Με τον όρο συμφωνία μεταξύ των επισημειωτών εννοούμε το αν οι επισημειωτές δίνουν το ίδιο αποτέλεσμα για ένα ορισμένο στοιχείο.

Η παρατηρούμενη συμφωνία (Observed Agreement) είναι απλώς η διαφορά της τιμής των επισημειωτών, ενώ η συνολική κατά μέσο όρο συμφωνία δίνεται από τον τύπο

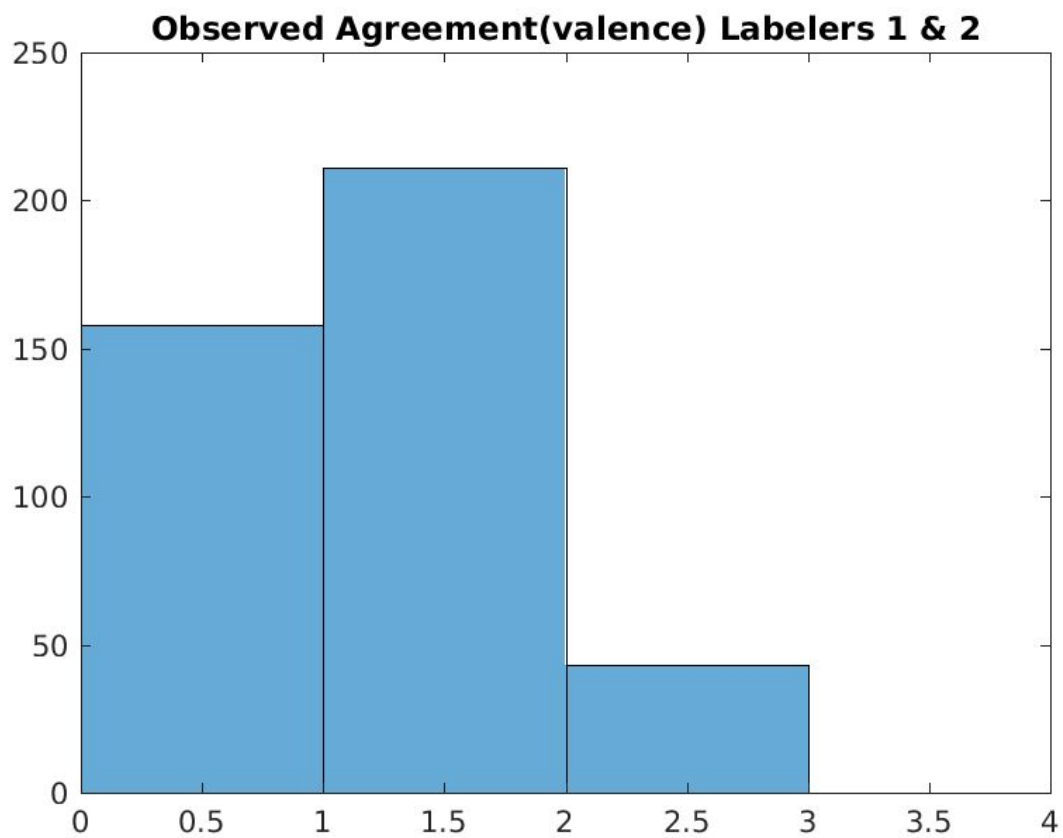
$$1 - E\left\{\frac{|I_{ij} - I_{ik}|}{4}\right\}$$

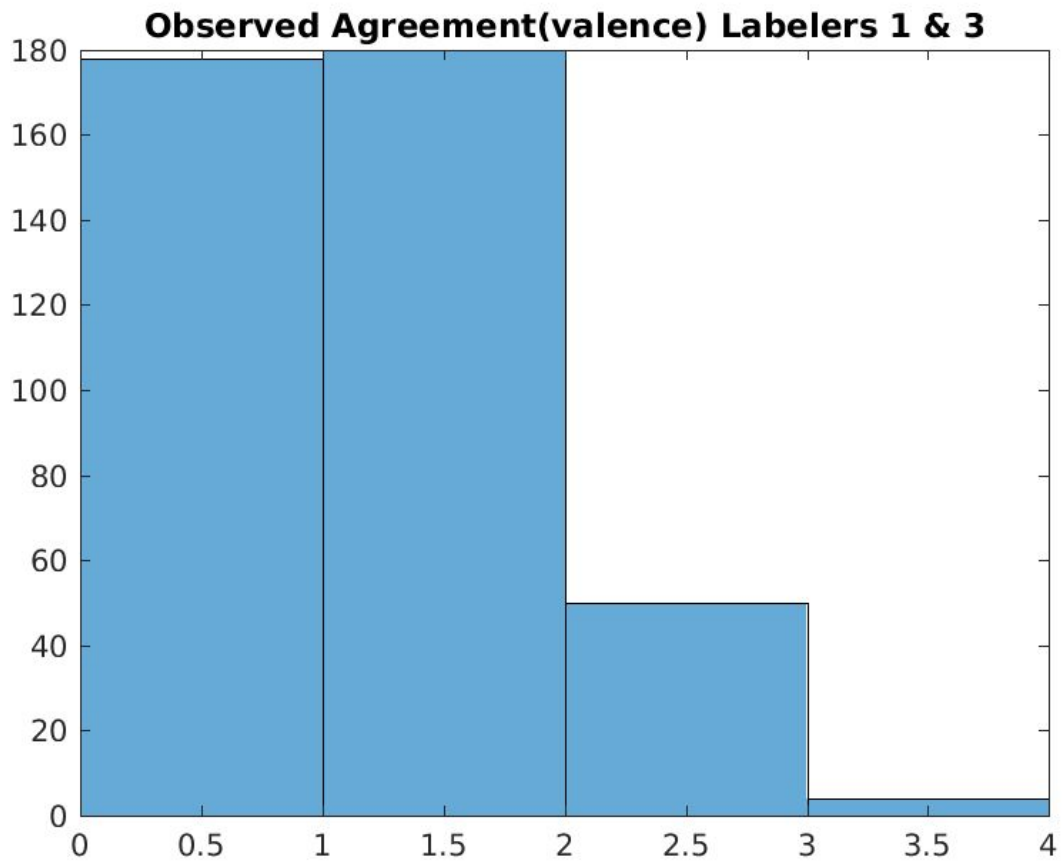
Ακολουθούν τα αποτελέσματα για τη συνολική κατά μέσο όρο συμφωνία, καθώς και τα ιστογράμματα των διαφορών για τα Valence και Activation για κάθε ζεύγος επισημειωτών.

<u>Expected Agreement (Activation)</u>		
L1-L2	L1-L3	L2-L3
0.8459	0.7961	0.8046

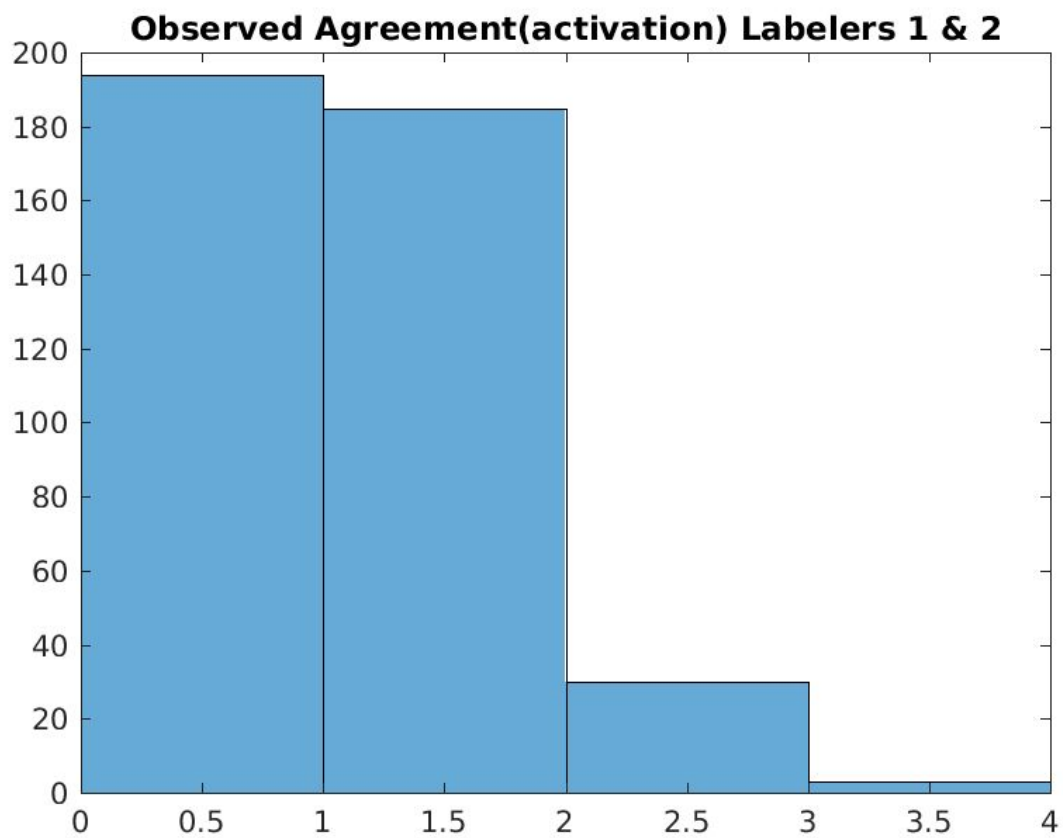
<u>Expected Agreement (Valence)</u>		
L1-L2	L1-L3	L2-L3
0.8198	0.8228	0.8331

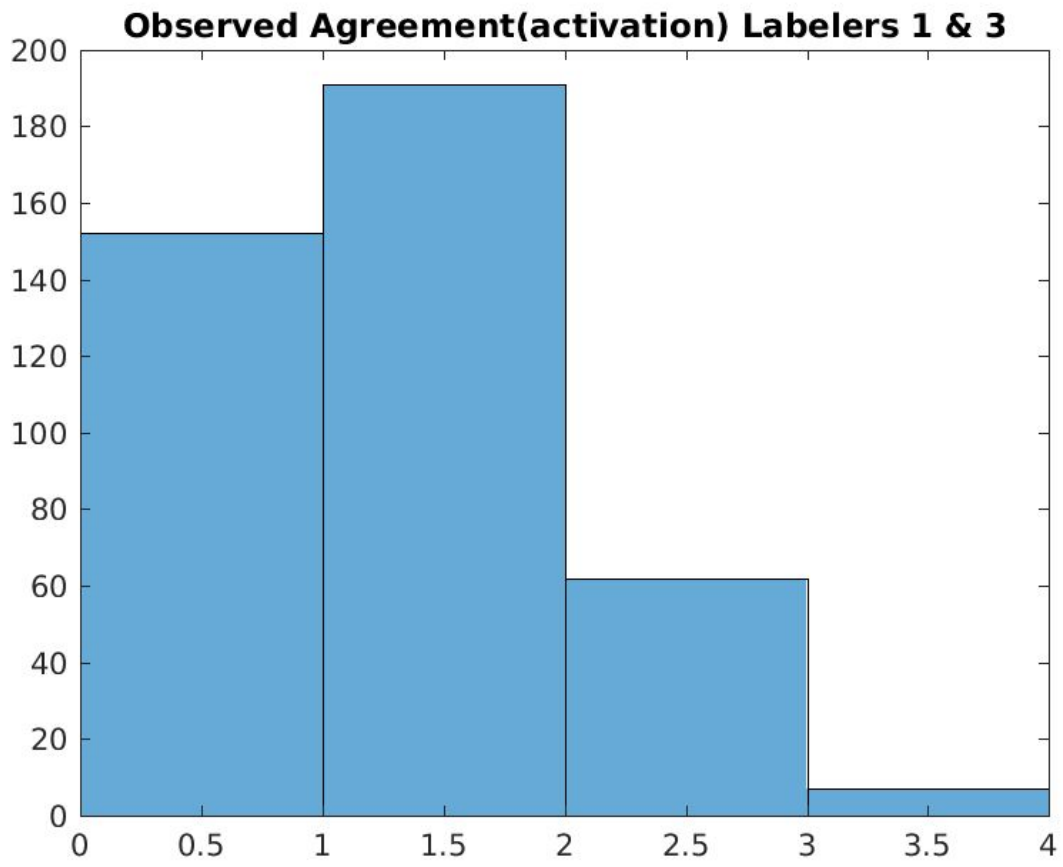


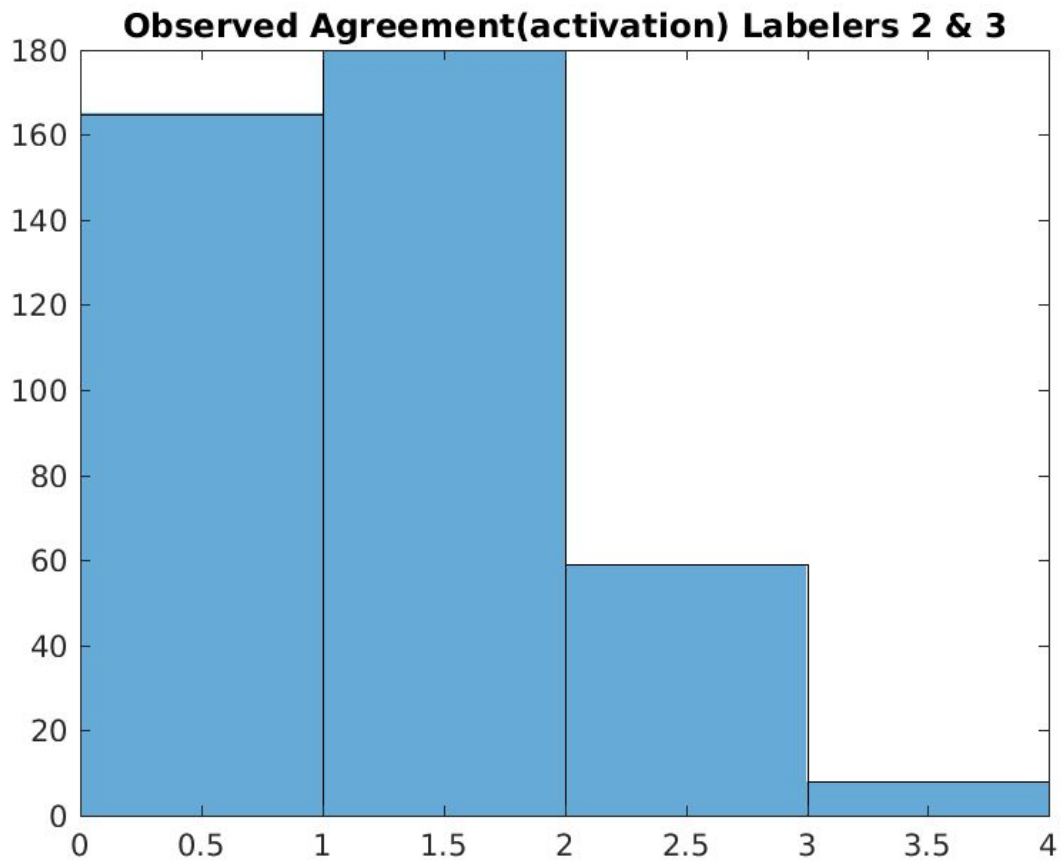












*%% Step 3 - Observed Agreement*

*combos = [1 2; 1 3; 2 3];*

*for k=1:1:size(combos,1)*

*x = combos(k,1);*

*y = combos(k,2);*

*ObsAgr\_activ = abs(cell2mat({Labelers(:,x).activation}) - cell2mat({Labelers(:,y).activation}));*

*ObsAgr\_val = abs(cell2mat({Labelers(:,x).valence}) - cell2mat({Labelers(:,y).valence}));*

*TotObsAgr\_activ(k) = 1-mean(ObsAgr\_activ)/4;*

*TotObsAgr\_val(k) = 1-mean(ObsAgr\_val)/4;*

*disp(['Observed Agreement ( activation ) between Labeler',num2str(x),' and Labeler',num2str(y),' is',num2str(TotObsAgr\_activ(k))]);*

*disp(['Observed Agreement ( valence ) between Labeler',num2str(x),' and Labeler',num2str(y),' is',num2str(TotObsAgr\_val(k))]);*

```

histo_activ = zeros(5,1);
histo_val = zeros(5,1);
for j=1:size(ObsAgr_activ,2)
    histo_activ( ObsAgr_activ(j) + 1) = histo_activ( ObsAgr_activ(j) + 1) +1;
    histo_val( ObsAgr_val(j) + 1) = histo_val( ObsAgr_val(j) + 1) +1;
end
figure;histogram(ObsAgr_activ,0:4);
title(['Observed Agreement(activation) Labelers ',num2str(x),' & ',num2str(y)]);
title1 = ['ObsAgr_activ_',num2str(x),num2str(y),'.png'];
saveas(gcf,title1);
figure;histogram(ObsAgr_val,0:4);
title(['Observed Agreement(valence) Labelers ',num2str(x),' & ',num2str(y)]);
title1 = ['ObsAgr_val_',num2str(x),num2str(y),'.png'];
saveas(gcf,title1);
end

```

#### Βήμα 4

Ζητείται να υπολογιστεί ο συντελεστής **Krippendorff's alpha** για ordinal δεδομένα σε κάθε διάσταση.

Ορίζεται ως:

$$\alpha = 1 - \frac{\text{Observed Agreement}}{\text{Observed Disagreement}}$$

όπου

**η παρατηρούμενη διαφωνία** είναι η μέση τιμή όλων των σημείων διαφωνίας των δεδομένων και

**η αναμενόμενη διαφωνία** είναι η μέση τιμή της διαφοράς μεταξύ δύο τιμών  $i$  και  $k$  σε όλα τα πιθανά ζευγάρια  $n(n-1)$  τιμών.

*Επειδή ο συντελεστής ζητείται να υπολογιστεί για ordinal data, ενώ τα δεδομένα μας είναι nominal, θα πρέπει αφού φτιάξουμε τον coincidence matrix, να μετατρέψουμε κατάλληλα τα δεδομένα μας, ώστε να χρησιμοποιούμε ordinal μετρική μεταξύ τους.*

Μετά τη διαδικασία αυτή (βρίσκεται στη συνάρτηση *kriAlpha*), λαμβάνουμε:

<b><u>Krippendorff's alpha</u></b>	
<b>Valence</b>	<b>Activation</b>
0.4747	0.4398

%% Step 4 - Krippendorff's Alpha ( code obtained from <https://goo.gl/PjXHkf> )

```
activ_labels = [cell2mat({Labelers(:,1).activation}) ; cell2mat({Labelers(:,2).activation}) ;
```

```
cell2mat({Labelers(:,3).activation})];
```

```
kalpha_activ = kriAlpha(activ_labels,'ordinal');
```

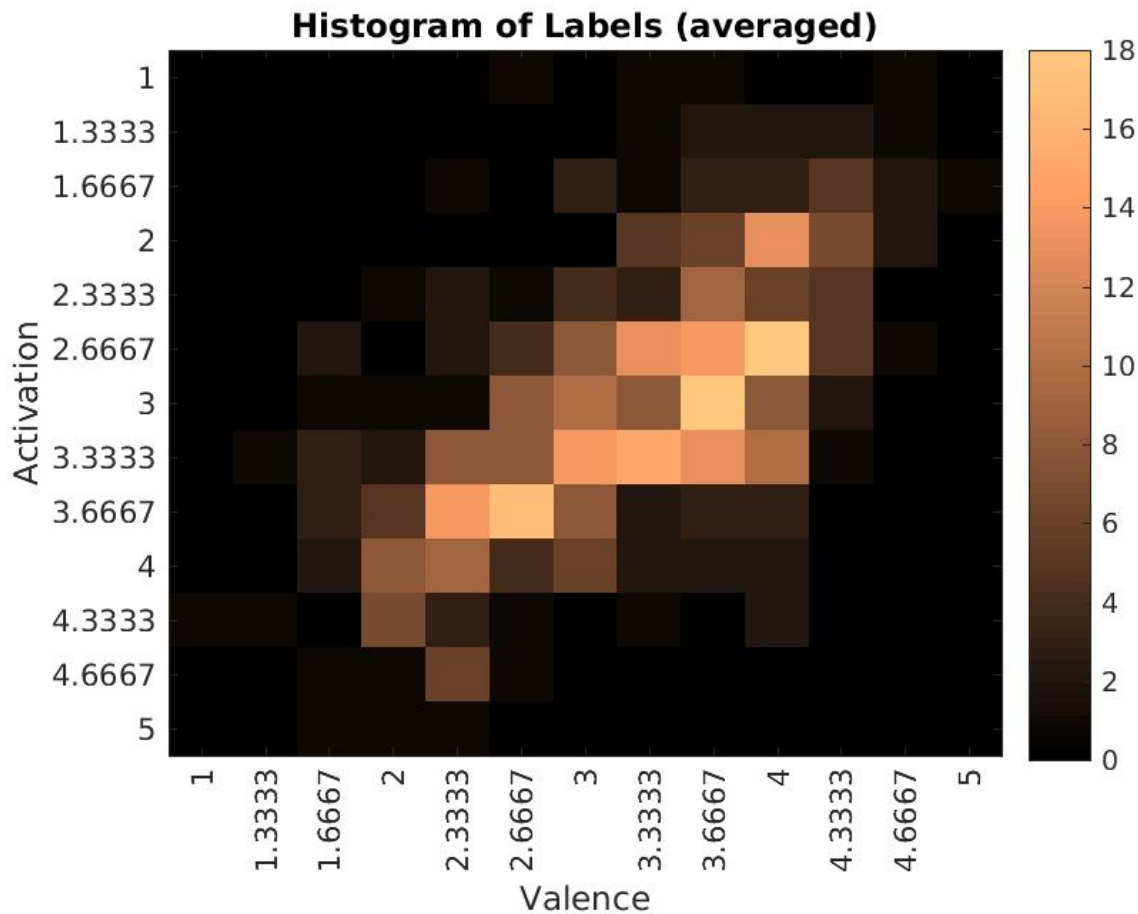
```
val_labels = [cell2mat({Labelers(:,1).valence}) ; cell2mat({Labelers(:,2).valence}) ; cell2mat({Labelers(:,3).valence})];
```

```
kalpha_val = kriAlpha(val_labels,'ordinal');
```



## Βήμα 5

Ζητείται να υπολογιστούν οι τελικές επισημειώσεις ως η μέση τιμή των τριών επισημειωτών για κάθε μέγεθος, και να παρασταθούν σε 2D ιστόγραμμα.



```
%% Step 5 - Mean labels
activation = mean(activ_labels);
valence = mean(val_labels);
vals = [1, 1.3333, 1.6667, 2, 2.3333, 2.6667, 3, 3.3333, 3.6667, 4, 4.3333, 4.6667, 5];
co_occurrence = zeros(13,13);
for j=1:1:size(activation,2)
    curr_act1 = round(activation(j),4);
    curr_act = find(vals==curr_act1);
    curr_val1 = round(valence(j),4);
    curr_val = find(vals==curr_val1);
    co_occurrence(curr_act,curr_val) = co_occurrence(curr_act,curr_val) + 1;
end
figure;imagesc(vals,vals,co_occurrence);
set(gca,'XTick',vals);set(gca,'YTick',vals);xtickangle(90);
colormap('copper');colorbar;
xlabel('Valence');ylabel('Activation');
title('Histogram of Labels (averaged)');
saveas(gcf,'histogram_avg.png');
```

## Βήμα 6

Ζητείται να υπολογίσουμε τα χαρακτηριστικά με χρήση συναρτήσεων του **MIRToolbox**:

1. Auditory Roughness
2. Rythmic Periodicity Along Auditory Channels
3. Key Clarity
4. Modality
5. Spectral Novelty
6. Harmonic Change Detection Function (HCDF)

```
%% Step 6 - MIR Toolbox Feature Extraction
addpath(genpath('./MIRtoolbox1.7'))
basepath = './PRcourse_Lab3_data/PRcourse_Lab3_data/MonoSamples/';
music_contents = folder_load(basepath);

for i=1:size(music_contents,1)
    cname = music_contents(i).name;
    full_name = strcat(basepath,cname);

    curr_track = mirframe(full_name);

    % 1) Auditory Roughness
    roughness = mirgetdata(mirroughness(curr_track));
    rough_mean(1) = mean(roughness);
    rough_std = std(roughness);

    rough_med_pl = roughness;
    rough_med_pl(rough_med_pl < median(rough_med_pl)) = [];
    rough_mean(2) = mean(rough_med_pl);

    rough_med_mi = roughness;
    rough_med_mi(rough_med_mi > median(rough_med_mi)) = [];
    rough_mean(3) = mean(rough_med_mi);

    % 2) Rythmic Periodicity Along Auditory Channels
    fluctuation = mirgetdata(mirsum(mirfluctuation(curr_track)));
    fluct_max = max(fluctuation);
    fluct_mean = mean(fluctuation);

    % 3) Key Clarity
    keys = mirgetdata(mirkey(curr_track));
    key_mean = mean(keys);

    % 4) Modality
    mode = mirgetdata(mirmode(curr_track));
    modality = mean(mode);

    % 5) Spectral Novelty
    snov = mirgetdata(mirnovelty(curr_track));
```

```

sp_nov_mean = mean(snov);

% 6) Harmonic Change Detection Function (HCDF)
hcdf = mirgetdata(mirhcdf(curr_track));
hcdf_mean = mean(hcdf);

features(i).rough_mean1 = rough_mean(1);
features(i).rough_mean2 = rough_mean(2);
features(i).rough_mean3 = rough_mean(3);
features(i).rough_std = rough_std;
features(i).fluct_max = fluct_max;
features(i).fluct_mean = fluct_mean;
features(i).key_mean = key_mean;
features(i).modality = modality;
features(i).sp_nov_mean = sp_nov_mean;
features(i).hcdf_mean = hcdf_mean;
clc;
end

```

## Βήμα 7

Ζητείται να υπολογιστούν οι Mel Frequency Cepstral Coefficients (MFCCs), η πρώτη (delta) και η δεύτερη (delta-delta) παράγωγος αυτών.

Προκύπτει έτσι ένα feature vector μήκους 39.

Από το διάνυσμα αυτό εξάγουμε τις ακόλουθες τιμές:

- Μέση τιμή
- Τυπική απόκλιση
- Μέση τιμή του 10% των max τιμών
- Μέση τιμή του 10% των min τιμών

**Κάθε μία από αυτές είναι διαστατικότητας 39 για κάθε μουσικό απόσπασμα.**

```

%% Step 7 - MFCC Extraction
for i=1:1:size(music_contents,1)
    cname = music_contents(i).name;
    full_name = strcat(basepath,cname);

    curr_track = mirframe(full_name, 0.025, 's', 0.01, 's');

    mfcc = mirgetdata(mirmfcc(curr_track, 'Bands', 26, 'Rank', 1:13));
    mfcc_delta = mirgetdata(mirmfcc(curr_track, 'Bands', 26, 'Rank', 1:13, 'Delta', 1));
    mfcc_deltadelta = mirgetdata(mirmfcc(curr_track, 'Bands', 26, 'Rank', 1:13, 'Delta', 2));

    mfcc_mean = mean(mfcc);

```

```

mfcc_delta_mean = mean(mfcc_delta);
mfcc_deltadelta_mean = mean(mfcc_deltadelta);

mfcc_std = std(mfcc);
mfcc_delta_std = std(mfcc_delta);
mfcc_deltadelta_std = std(mfcc_deltadelta);

mfcc = sort(mfcc);
mfcc_delta = sort(mfcc_delta,2);
mfcc_deltadelta = sort(mfcc_deltadelta,2);

perc = round(0.1*size(mfcc,1));

mfcc_mean_low = mean(mfcc(1:perc,:));
mfcc_delta_mean_low = mean(mfcc_delta(1:perc,:));
mfcc_deltadelta_mean_low = mean(mfcc_deltadelta(1:perc,:));

mfcc_mean_high = mean(mfcc(end-perc:end,:));
mfcc_delta_mean_high = mean(mfcc_delta(end-perc:end,:));
mfcc_deltadelta_mean_high = mean(mfcc_deltadelta(end-perc:end,:));

mfcc_related.mfcc_mean = mfcc_mean;
mfcc_related.mfcc_delta_mean = mfcc_delta_mean;
mfcc_related.mfcc_deltadelta_mean = mfcc_deltadelta_mean;


mfcc_related.mfcc_mean_low = mfcc_mean_low;
mfcc_related.mfcc_delta_mean_low = mfcc_delta_mean_low;
mfcc_related.mfcc_deltadelta_mean_low = mfcc_deltadelta_mean_low;

mfcc_related.mfcc_mean_high = mfcc_mean_high;
mfcc_related.mfcc_delta_mean_high = mfcc_delta_mean_high;
mfcc_related.mfcc_deltadelta_mean_high = mfcc_deltadelta_mean_high;

features(i).mfcc_related = mfcc_related;
clc;
end

```