Machine Learning for Signal Processing

Homework-1

Osman Kaan Kurtça 040160090

Aziz Gündoğdu 040160112

<u>Load_Data Function:</u>

```
function [smagNote, smagMusic, sphaseMusic] = load_data()
% smagNote: 1025 x 11 matrix containing the mean spectrum magnitudes
of the notes. A correct sequence of the notes is REQUIRED. (From left
to right: e f g a b c d e2 f2 g2 a2)
% smagMusic: 1025 x K matrix containing the spectrum magnitueds of the
music after STFT.
% sphaseMusic: 1025 x K matrix containing the spectrum phases of the
music after STFT.

%% Load Spectrum Magnitudes of Notes
% Fill your code here to return 'smagNote'
notesfolder = 'notes15';
listname = dir(fullfile(notesfolder,'*.wav'));
notes=[];
for k = 1:length(listname)
    [s, fs] = audioread([notesfolder filesep listname(k).name]);
    s = s(:, 1);
    s = resample(s, 16000, fs);
    spectrum = stft(s', 2048, 256, 0, hann(2048));
    %Find the central frame
    middle = ceil(size(spectrum, 2) /2);
    note = abs(spectrum(:, middle));
    %Clean up everything more than 40 db below the peak
    note(find(note<max(note(:))/100)) = 0 ;
    note = note/norm(note);
    %normalize the note to unit length
    notes = [notes, note];
end
smagNote=notes;
%% Load Spectrum Magnitues and Phases of The Provided Music
% Fill your code here to return 'smagMusic' and 'sphaseMusic'
[y,Fs] = audioread('polyushka.wav');
spectrum=stft(y', 2048, 256, 0, hann(2048));
smagMusic=abs(spectrum);
sphaseMusic=spectrum./smagMusic;
```

<u>Cod Summary:</u> With the load_data function, we have taken the spectrogram of the music and note data (with the stft function) and made it suitable for matrix operations. We found both the amplitude and the phase of the music data separately because we will need the phase information when recompose the music.

Stft Function

We used the handwritten stft function to take the spectrograms of the signals and to take the original form of the spectrogrammed signals without changing it. İf the input of function contains real value it takes the spectrogram. Otherwise, it applies inverse transformation

Synthesize_Music Function

```
function [synMusic] = synthesize_music(smagMusicProj,sphaseMusic)
%% Argument Descriptions
% Required Input Arguments:
% sphaseMusic: 1025 x K matrix containing the spectrum phases of the music after STFT.
% smagMusicProj: 1025 x K matrix, reconstructed version of smagMusic using transMatT

% Required Output Arguments:
% synMusic: N x 1 music signal reconstructed using STFT.

%% Music synthesis
% Fill your code here to return 'synMusic'
r=smagMusicProj.*sphaseMusic;
synMusic=stft(r,2048,256,0,hann(2048));
synMusic=transpose(synMusic);
```

Code Summary: In this function; First,
The amplitude matrix, which we reconstruct with the help of note matrices and the pseudo inverse algorithm, is multiplied by the phase matrix. After that, with this complex valued matrix we find the reconstructed music signal using stft function. Finally, we take the transpose of signal to transform the row matrix.

Run_problem_master_file Function Problem-1

```
%% Load Notes and Music
clc; clear all; close all;

[smagNote, smagMusic, sphaseMusic] = load_data();
W=pinv(smagNote)*smagMusic;
W=max(W,0);
save("problem2_1.mat","W");
%% Synthesize Music
% Use the 'synthesize_music' function here.
smagMusicProj=smagNote*W;
synMusic=synthesize_music(smagMusicProj,sphaseMusic);

% Use 'wavwrite' function to write the synthesized music as 'problem2_1_synthesis.wav' to the 'results' folder.
filename='C:\Users\osman\Documents\MATLAB\MLSP_HW!\results\problem2_1_synthesis.wav';
fs=44100/2;
audiowrite(filename,synMusic,fs);
```

Code Summary: Firstly, Using load_data function, we got spectrogram of the signals. Then, applying pseude inverse algorithm, Weight matrix (W) have found. All negative values in matrix are set the zero with max function. and W is stored. Using W matrix, we obtained the reconstructed amplitude matrix of spectrogram. Finally, we have found and stored reconstructed music signal with synthesize_music and audiowrite function.

Run_problem_master_file Function Problem-2

```
%Find and store the transformation matrix
[p1,Fs1]=audioread('C:\Users\osman\Documents\MATLAB\MLSP_HW!\audio\s
ilentnight_piano.aif');
p1=p1(:,1);
spectrum_p1=stft(p1', 1024, 256, 0, hann(1024));
smagMusic_p1=abs(spectrum_p1);

[g1,Fs2]=audioread('C:\Users\osman\Documents\MATLAB\MLSP_HW!\audio\s
ilentnight_guitar.aif');
g1=g1(:,1);
spectrum_g1=stft(g1', 1024, 256, 0, hann(1024));
smagMusic_g1=abs(spectrum_g1);

W1=pinv(smagMusic_p1)*smagMusic_g1;
W1=max(W1,0);

[p2,Fs3] =
audioread('C:\Users\osman\Documents\MATLAB\MLSP_HW!\audio\littlestar
_piano.aif');
p2=p2(:,1);
spectrum_p2=stft(p2', 1024, 256, 0, hann(1024));
smagMusic_p2=abs(spectrum_p2);
sphaseMusic_p2=spectrum_p2./smagMusic_p2;
add=zeros(513,2075);
smagMusic_p2= [smagMusic_p2 add];
sphaseMusic_p2=[sphaseMusic_p2 add];

% Apply the transformation matrix to audio C and store the created
music using 'synthesize_music' function.
smagMusic_g2=smagMusic_p2*W1;

r=smagMusic_g2.*sphaseMusic_p2;
synMusic_g2=stft(r,1024,256,0,hann(1024));
synMusic_g2=transpose(synMusic_g2);

% Use 'wavwrite' function to write the synthesized music as
'problem2_2_synthesis.wav' to the 'results' folder.
filename='C:\Users\osman\Documents\MATLAB\MLSP_HW!\results\problem2_
2_synthesis.wav';
fs=44100;
audiowrite(filename,synMusic_g2,fs);
```

Code Summary: We have found the transformation matrix that converts to amplitude of the spectrogram for silent night piano version to amplitude of the spectrogram for silent night guitar version using pseudo inverse algorithm. After that, applying this transformation matrix to amplitude of spectrogram for little star piano version, We got the little star guitar version and stored it in results folder.

Note: We added 2075 sero column matrix to amplitude of spectrogram of little star piano and We adjusted its dimensions so that it can be multiplied by the transformation matrix(W2).

Workspace and Results Folder