LABORATORY EXPERIENCE #10
(Hidden Markov Machines)

Introduction

Hidden Markov Machines are models in which we supposed to be observing a discrete variable that is derived by a finite state machine in which the **Markov property** holds. Then, this variable will depend on the state of the hidden machine and on the measurement error, that is modelled as a zero-mean Gaussian variable.
An HMM is defined by the **transition matrix**, that defines the probability to jump from one state to another inside the machine, the **emission matrix**, that defines the probability of having a certain output given that the machine is at a specific state, and the **prior probabilities** vector.
The experiment was focused on train a set of HMMs in order to recognize a vowel from a recorded sample.

Data preparation

Matlab provides a set of functions to record and load voice samples. For each vowel, a sequence of one second lasting samples were recorded to train a specific HMM. Once the voice data has been recorded, the quantization levels were reduced from 256 to 16 in order to simplify the problem.

Algorithms implementation and results

For each vowel, a different HMM was trained, obtaining 5 couples of transition matrixes and emission matrixes. For this purpose *'hmmtrain'* Matlab function has been used. By default it uses the **Baum-Welch** algorithm.
Then each of the 5 HMMs was tested with a new set of vowels samples, in order to obtain a 5x5 matrix (5 is the number of vowels) and evaluate the probability that a test sample is obtain by a specific HMM.
If the HMM are well defined, we should have the maximum values of these probabilities on the main diagonal in each row of the matrix. The result of the experiment was successful for 3 vowels over the 5 tested, with a strike probability of 60%.

	HMMa	HMMe	HMMi	HMMo	HMMu
'a'	-15032,46	-26198,51	-24207,80	-18865,67	-21566,57
'e'	-20076,10	-21313,54	-19982,45	-25502,30	-18591,26
'i'	-26374,70	-19233,43	-17727,28	-30357,32	-18623,78
'o'	-23293,07	-22966,49	-21771,48	-26815,14	-20269,46
'u'	-20031,44	-21990,66	-20201,71	-25087,02	-19258,61

Figure 1: Resulting matrix of probabilities using testing vowels samples different from training samples

Repeating the experiment testing the HMMs with samples taken from the training sets, depending on which sample we take, we can obtain lower or better results.

	HMMa	HMMe	HMMi	HMMo	HMMu
‘a’	-13594,12	-27122,15	-24974,84	-16020,11	-22350,47
‘e’	-28955,58	-20201,78	-18845,65	-32135,05	-20017,79
‘i’	-27228,52	-14174,36	-12322,07	-32480,49	-16791,04
‘o’	-15846,80	-30107,48	-25801,64	-14444,23	-25593,13
‘u’	-20941,11	-20213,29	-19372,95	-28973,30	-17334,05

Figure 2: Example of probability matrix using samples from the training sets

```
clear all
close all
clc

Fsamp=8000;
Nbits=8;
Nchann=1;
interval=1;
recObj = audiorecorder(Fsamp, Nbits, Nchann);

disp('Start speaking "a" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
a1 = getaudiodata(recObj);
save('a1.mat','a1');
disp('Start speaking "a" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
a2 = getaudiodata(recObj);
save('a2.mat','a2');
disp('Start speaking "a" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
a3 = getaudiodata(recObj);
save('a3.mat','a3');
disp('Start speaking "a" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
a4 = getaudiodata(recObj);
save('a4.mat','a4');
disp('Start speaking "a" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
a5 = getaudiodata(recObj);
save('a5.mat','a5');

disp('Start speaking "e" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
e1 = getaudiodata(recObj);
save('e1.mat','e1');
disp('Start speaking "e" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
e2 = getaudiodata(recObj);
save('e2.mat','e2');
disp('Start speaking "e" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
e3 = getaudiodata(recObj);
save('e3.mat','e3');
disp('Start speaking "e" after hitting the key')
```

```
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
e4 = getaudiodata(recObj);
save('e4.mat','e4');
disp('Start speaking "e" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
e5 = getaudiodata(recObj);
save('e5.mat','e5');


disp('Start speaking "i" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
i1 = getaudiodata(recObj);
save('i1.mat','i1');
disp('Start speaking "i" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
i2 = getaudiodata(recObj);
save('i2.mat','i2');
disp('Start speaking "i" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
i3 = getaudiodata(recObj);
save('i3.mat','i3');
disp('Start speaking "i" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
i4 = getaudiodata(recObj);
save('i4.mat','i4');
disp('Start speaking "i" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
i5 = getaudiodata(recObj);
save('i5.mat','i5');

disp('Start speaking "o" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
o1 = getaudiodata(recObj);
save('o1.mat','o1');
disp('Start speaking "o" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
o2 = getaudiodata(recObj);
save('o2.mat','o2');
disp('Start speaking "o" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
o3 = getaudiodata(recObj);
save('o3.mat','o3');
disp('Start speaking "o" after hitting the key')
w = input('Hit any key to continue ');
```

```
recordblocking(recObj, interval);
o4 = getaudiodata(recObj);
save('o4.mat','o4');
disp('Start speaking "o" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
o5 = getaudiodata(recObj);
save('o5.mat','o5');

disp('Start speaking "u" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
u1 = getaudiodata(recObj);
save('u1.mat','u1');
disp('Start speaking "u" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
u2 = getaudiodata(recObj);
save('u2.mat','u2');
disp('Start speaking "u" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
u3 = getaudiodata(recObj);
save('u3.mat','u3');
disp('Start speaking "u" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
u4 = getaudiodata(recObj);
save('u4.mat','u4');
disp('Start speaking "u" after hitting the key')
w = input('Hit any key to continue ');
recordblocking(recObj, interval);
u5 = getaudiodata(recObj);
save('u5.mat','u5');





y=myRecording
%[y,Fs] = audioread('AleAndroid01.wav');
%y=mean(y,2); % converting from stereo to mono
N_file=length(y);

%normalizing the signal
% y=(y-min(y))/(max(y)-min(y));
% m_y=mean(abs(y));
% y=y-m_y;
```

Published with MATLAB® R2016a

```
clear all
close all
clc

load('a1.mat')
load('a2.mat')
load('a3.mat')
load('a4.mat')
load('a5.mat')
load('e1.mat')
load('e2.mat')
load('e3.mat')
load('e4.mat')
load('e5.mat')
load('i1.mat')
load('i2.mat')
load('i3.mat')
load('i4.mat')
load('i5.mat')
load('o1.mat')
load('o2.mat')
load('o3.mat')
load('o4.mat')
load('o5.mat')
load('u1.mat')
load('u2.mat')
load('u3.mat')
load('u4.mat')
load('u5.mat')

Kquant=16;

za=[a1,a2,a3,a4,a5];
ze=[e1,e2,e3,e4,e5];
zi=[i1,i2,i3,i4,i5];
zo=[o1,o2,o3,o4,o5];
zu=[u1,u2,u3,u4,u5];

for ii=1:5
    amax=max(za(:,ii));
    amin=min(za(:,ii));
    delta=(amax-amin)/(Kquant-1);%quantization interval
    za(:,ii)=round((za(:,ii)-amin)/delta)+1;%quantized signal
end

for ii=1:5
    amax=max(ze(:,ii));
    amin=min(ze(:,ii));
    delta=(amax-amin)/(Kquant-1);%quantization interval
    ze(:,ii)=round((ze(:,ii)-amin)/delta)+1;%quantized signal
end
```

```
for ii=1:5
    amax=max(zi(:,ii));
    amin=min(zi(:,ii));
    delta=(amax-amin)/(Kquant-1);%quantization interval
    zi(:,ii)=round((zi(:,ii)-amin)/delta)+1;%quantized signal
end

for ii=1:5
    amax=max(zo(:,ii));
    amin=min(zo(:,ii));
    delta=(amax-amin)/(Kquant-1);%quantization interval
    zo(:,ii)=round((zo(:,ii)-amin)/delta)+1;%quantized signal
end

for ii=1:5
    amax=max(zu(:,ii));
    amin=min(zu(:,ii));
    delta=(amax-amin)/(Kquant-1);%quantization interval
    zu(:,ii)=round((zu(:,ii)-amin)/delta)+1;%quantized signal
end

save('za.mat','za');
save('ze.mat','ze');
save('zo.mat','zo');
save('zi.mat','zi');
save('zu.mat','zu');
```

Published with MATLAB® R2016a

```
clear all
close all
clc

load('za.mat');
load('ze.mat');
load('zi.mat');
load('zo.mat');
load('zu.mat');

K=16; % number of quantization levels
M=8; % number of states
TRANS_HAT=rand(M,M)
for ii=1:M
    TRANS_HAT(ii,:)=(TRANS_HAT(ii,:))/sum((TRANS_HAT(ii,:)));
end
EMIT_HAT=rand(M,K);
for ii=1:M
    EMIT_HAT(ii,:)=(EMIT_HAT(ii,:))/sum((EMIT_HAT(ii,:)));
end

[ESTTRa,ESTEMITa]=hmmtrain(za,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',
save('ESTTRa.mat','ESTTRa');
save('ESTEMITa.mat','ESTEMITa');
[ESTTRe,ESTEMITe]=hmmtrain(ze,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',
save('ESTTRe.mat','ESTTRe');
save('ESTEMITe.mat','ESTEMITe');
[ESTTRi,ESTEMITi]=hmmtrain(zi,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',
save('ESTTRi.mat','ESTTRi');
save('ESTEMITi.mat','ESTEMITi');
[ESTTRo,ESTEMITo]=hmmtrain(zo,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',
save('ESTTRo.mat','ESTTRo');
save('ESTEMITo.mat','ESTEMITo');
[ESTTRu,ESTEMITu]=hmmtrain(zu,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',
save('ESTTRu.mat','ESTTRu');
save('ESTEMITu.mat','ESTEMITu');
```

Published with MATLAB® R2016a

```matlab
clear all
close all
clc

load('ESTTRa.mat');
load('ESTEMITa.mat');
load('ESTTRe.mat');
load('ESTEMITe.mat');
load('ESTTRi.mat');
load('ESTEMITi.mat');
load('ESTTRo.mat');
load('ESTEMITo.mat');
load('ESTTRu.mat');
load('ESTEMITu.mat');

load('za.mat');
load('ze.mat');
load('zi.mat');
load('zo.mat');
load('zu.mat');
%
% Kquant=16
%
% Fsamp=8000;
% Nbits=8;
% Nchann=1;
% interval=1;
% recObj = audiorecorder(Fsamp, Nbits, Nchann);
%
% disp('Start speaking "a" after hitting the key')
% w = input('Hit any key to continue ');
% recordblocking(recObj, interval);
% ta = getaudiodata(recObj);
% amax=max(ta);
% amin=min(ta);
% delta=(amax-amin)/(Kquant-1);%quantization interval
% ta=round((ta-amin)/delta)+1;%quantized signal
% save('ta.mat','ta');
%
% disp('Start speaking "e" after hitting the key')
% w = input('Hit any key to continue ');
% recordblocking(recObj, interval);
% te = getaudiodata(recObj);
% amax=max(te);
% amin=min(te);
% delta=(amax-amin)/(Kquant-1);%quantization interval
% te=round((te-amin)/delta)+1;%quantized signal
% save('te.mat','te');
%
% disp('Start speaking "i" after hitting the key')
% w = input('Hit any key to continue ');
% recordblocking(recObj, interval);
% ti = getaudiodata(recObj);
% amax=max(ti);
% amin=min(ti);
% delta=(amax-amin)/(Kquant-1);%quantization interval
% ti=round((ti-amin)/delta)+1;%quantized signal
% save('ti.mat','ti');
%
% disp('Start speaking "o" after hitting the key')
% w = input('Hit any key to continue ');
% recordblocking(recObj, interval);
% to = getaudiodata(recObj);
% amax=max(to);
% amin=min(to);
% delta=(amax-amin)/(Kquant-1);%quantization interval
% to=round((to-amin)/delta)+1;%quantized signal
% save('to.mat','to');
%
% disp('Start speaking "u" after hitting the key')
% w = input('Hit any key to continue ');
% recordblocking(recObj, interval);
% tu = getaudiodata(recObj);
% amax=max(tu);
% amin=min(tu);
% delta=(amax-amin)/(Kquant-1);%quantization interval
% tu=round((tu-amin)/delta)+1;%quantized signal
% save('tu.mat','tu');


ta=za(:,5);
te=ze(:,5);
ti=zi(:,5);
to=zo(:,5);
tu=zu(:,5);


[PSTATESaa,logpseqaa] = hmmdecode(ta',ESTTRa,ESTEMITa);
paa=logpseqaa;
[PSTATESaa,logpseqae] = hmmdecode(ta',ESTTRe,ESTEMITe);
pae=logpseqae;
[PSTATESaa,logpseqai] = hmmdecode(ta',ESTTRi,ESTEMITi);
pai=logpseqai;
[PSTATESaa,logpseqao] = hmmdecode(ta',ESTTRo,ESTEMITo);
pao=logpseqao;
[PTATESaa,logpseqau] = hmmdecode(ta',ESTTRu,ESTEMITu);
pau=logpseqau;
row1=[paa,pae,pai,pao,pau];


[PSTATESaa,logpseqea] = hmmdecode(te',ESTTRa,ESTEMITa);
pea=logpseqea;
[PSTATESaa,logpseqee] = hmmdecode(te',ESTTRe,ESTEMITe);
pee=logpseqee;
[PSTATESaa,logpseqei] = hmmdecode(te',ESTTRi,ESTEMITi);
pei=logpseqei;
```

```
[PSTATESaa,logpseqeo] = hmmdecode(te',ESTTRo,ESTEMITo);
peo=logpseqeo;
[PTATESaa,logpseqeu] = hmmdecode(te',ESTTRu,ESTEMITu);
peu=logpseqeu;
row2=[pea,pee,pei,peo,peu];


[PSTATESaa,logpseqia] = hmmdecode(ti',ESTTRa,ESTEMITa);
pia=logpseqia;
[PSTATESaa,logpseqie] = hmmdecode(ti',ESTTRe,ESTEMITe);
pie=logpseqie;
[PSTATESaa,logpseqii] = hmmdecode(ti',ESTTRi,ESTEMITi);
pii=logpseqii;
[PSTATESaa,logpseqio] = hmmdecode(ti',ESTTRo,ESTEMITo);
pio=logpseqio;
[PTATESaa,logpseqiu] = hmmdecode(ti',ESTTRu,ESTEMITu);
piu=logpseqiu;
row3=[pia,pie,pii,pio,piu];


[PSTATESaa,logpseqoa] = hmmdecode(to',ESTTRa,ESTEMITa);
poa=logpseqoa;
[PSTATESaa,logpseqoe] = hmmdecode(to',ESTTRe,ESTEMITe);
poe=logpseqoe;
[PSTATESaa,logpseqoi] = hmmdecode(to',ESTTRi,ESTEMITi);
poi=logpseqoi;
[PSTATESaa,logpseqoo] = hmmdecode(to',ESTTRo,ESTEMITo);
poo=logpseqoo;
[PTATESaa,logpseqou] = hmmdecode(to',ESTTRu,ESTEMITu);
pou=logpseqou;
row4=[poa,poe,poi,poo,pou];


[PSTATESaa,logpsequa] = hmmdecode(tu',ESTTRa,ESTEMITa);
pua=logpsequa;
[PSTATESaa,logpseque] = hmmdecode(tu',ESTTRe,ESTEMITe);
pue=logpseque;
[PSTATESaa,logpsequi] = hmmdecode(tu',ESTTRi,ESTEMITi);
pui=logpsequi;
[PSTATESaa,logpsequo] = hmmdecode(tu',ESTTRo,ESTEMITo);
puo=logpsequo;
[PTATESaa,logpsequu] = hmmdecode(tu',ESTTRu,ESTEMITu);
puu=logpsequu;
row5=[pua,pue,pui,puo,puu];

probs=[row1;row2;row3;row4;row5];

n_strikes=0;
for ii=1:5
    [M,I] = max(probs(ii,:))
    if ii == I
        n_strikes=n_strikes+1;
    end
end
```

```
p_strikes=n_strikes/5;
```

Published with MATLAB® R2016a