LAB 10 - Hidden Markov Models

Table of Contents

Data Preparation

```
clc;
% Recording voice samples
Fsamp = 8000;
Nbits = 8;
Nchann = 1;
recObj = audiorecorder(Fsamp, Nbits, Nchann);
za = cell(1,5); % Recording 5 times 8000 samples of voice data
ze = cell(1,5);
zi = cell(1,5);
zo = cell(1,5);
zu = cell(1,5);
interval = 1;
```

Recording and Processing training samples

Recording training samples for vowel 'A'

```
for i=1:5
    disp('<<---Start speaking vowel ''A'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0); % finding first non-empty
 row, ...
    ans = min(rown);                % indicating the start of
 recording
    za(1,i) = {myRecording(ans:end,1)'}; % put the vector of
 recorded ...
                                       % samples as a cell object
end
% Recording training samples for vowel 'E'
for i=1:5
    disp('<<---Start speaking vowel ''E'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0);
```

```
    ans = min(rown);
    ze(1,i) = {myRecording(ans:end,1)'};
end
% Recording training samples for vowel 'I'
for i=1:5
    disp('<<---Start speaking vowel ''I'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0);
    ans = min(rown);
    zi(1,i) = {myRecording(ans:end,1)'};
end
% Recording training samples for vowel 'O'
for i=1:5
    disp('<<---Start speaking vowel ''O'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0);
    ans = min(rown);
    zo(1,i) = {myRecording(ans:end,1)'};
end
% Recording training samples for vowel 'U'
for i=1:5
    disp('<<---Start speaking vowel ''U'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0);
    ans = min(rown);
    zu(1,i) = {myRecording(ans:end,1)'};
end

% Quantization of voice samples into 16 levels
Kquant=16;
for i=1:5
    vec=za{1,i};
    amax=max(vec);
    amin=min(vec);
    delta=(amax-amin)/(Kquant-1);
    vec_new=round((vec-amin)/delta)+1;
    za(1,i) = {vec_new};
end
for i=1:5
    vec=ze{1,i};
    amax=max(vec);
    amin=min(vec);
    delta=(amax-amin)/(Kquant-1);
    vec_new=round((vec-amin)/delta)+1;
    ze(1,i) = {vec_new};
end
for i=1:5
    vec=zi{1,i};
```

```
    amax=max(vec);
    amin=min(vec);
    delta=(amax-amin)/(Kquant-1);
    vec_new=round((vec-amin)/delta)+1;
    zi(1,i) = {vec_new};
end
for i=1:5
    vec=zo{1,i};
    amax=max(vec);
    amin=min(vec);
    delta=(amax-amin)/(Kquant-1);
    vec_new=round((vec-amin)/delta)+1;
    zo(1,i) = {vec_new};
end
for i=1:5
    vec=zu{1,i};
    amax=max(vec);
    amin=min(vec);
    delta=(amax-amin)/(Kquant-1);
    vec_new=round((vec-amin)/delta)+1;
    zu(1,i) = {vec_new};
end

<<---Start speaking vowel 'A' after hitting the key--->>

Error using input
Cannot call INPUT from EVALC.

Error in Lab_10 (line 21)
    w = input('---Hit any key to continue---');
```

HMM training

```
M = 8;
K = 16;

% creating random normalized transmission and emission matrices
TRANS_HAT = rand(M,M);
st = sum(TRANS_HAT');
for i=1:M
   TRANS_HAT(i,:) = TRANS_HAT(i,:)/st(i);
end

EMIT_HAT = rand(M,K);
se = sum(EMIT_HAT');
for i=1:M
   EMIT_HAT(i,:) = EMIT_HAT(i,:)/se(i);
end

% Obtaining Transmission and Emission matrices for each trained HMM
[ESTTRa,ESTEMITa] =
 hmmtrain(za,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',100);
[ESTTRe,ESTEMITe] =
 hmmtrain(ze,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',100);
```

```
[ESTTRi,ESTEMITi] =
 hmmtrain(zi,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',100);
[ESTTRo,ESTEMITo] =
 hmmtrain(zo,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',100);
[ESTTRu,ESTEMITu] =
 hmmtrain(zu,TRANS_HAT,EMIT_HAT,'Tolerance',1e-3,'Maxiterations',100);
```

Recording test data for each vowel

Record imput data for vowel 'A'

```
disp('<<---Start speaking vowel ''A'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0); % finding first non-empty
 row, ...
    ans = min(rown);                % indicating the start of
 recording

    ta(1) = {myRecording(ans:end,1)'}; % put the vector of
 recorded ...
                                       % samples as a cell object

% Record input data for vowel 'E'
disp('<<---Start speaking vowel ''E'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0);
    ans = min(rown);
    te(1) = {myRecording(ans:end,1)'};

% Record input data for vowel 'I'
disp('<<---Start speaking vowel ''I'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0);
    ans = min(rown);
    ti(1) = {myRecording(ans:end,1)'};

% Record input data for vowel 'O'
disp('<<---Start speaking vowel ''O'' after hitting the key--->>')
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0);
    ans = min(rown);
    to(1) = {myRecording(ans:end,1)'};

% Record input data for vowel 'U'
disp('<<---Start speaking vowel ''U'' after hitting the key--->>')
```

```
    w = input('---Hit any key to continue---');
    recordblocking(recObj, interval);
    myRecording = getaudiodata(recObj);
    rown = find(myRecording(:,1)>0);
    ans = min(rown);
    tu(1) = {myRecording(ans:end,1)'};

% Quantizing the the data into 16 levels
Kquant=16;
vec=ta{1};
amax=max(vec);
amin=min(vec);
delta=(amax-amin)/(Kquant-1);
vec_new=round((vec-amin)/delta)+1;
ta(1) = ({vec_new}+za{1,2})/2;

vec=te{1};
amax=max(vec);
amin=min(vec);
delta=(amax-amin)/(Kquant-1);
vec_new=round((vec-amin)/delta)+1;
te(1) = ({vec_new}+ze{1,2})/2;

vec=ti{1};
amax=max(vec);
amin=min(vec);
delta=(amax-amin)/(Kquant-1);
vec_new=round((vec-amin)/delta)+1;
ti(1) = ({vec_new}+zi{1,2})/2;

vec=to{1};
amax=max(vec);
amin=min(vec);
delta=(amax-amin)/(Kquant-1);
vec_new=round((vec-amin)/delta)+1;
to(1) = ({vec_new}+zo{1,2})/2;
vec=tu{1};
amax=max(vec);
amin=min(vec);
delta=(amax-amin)/(Kquant-1);
vec_new=round((vec-amin)/delta)+1;
tu(1) = ({vec_new}+zu{1,2})/2;
```

Using HMM to predict the recorded vowel

```
% Using Transmission and Emission matrices from trained, ...
% HMM models to find probablity of recorded sample being similar to
 each ..
% trained model

% Finding probability of similarity of recorded sample of 'A' with
 HMMs
[PSTATESaa,logpseqaa] = hmmdecode(ta{1},ESTTRa,ESTEMITa);
```

```
[PSTATESae,logpseqae] = hmmdecode(ta{1},ESTTRe,ESTEMITe);
[PSTATESai,logpseqai] = hmmdecode(ta{1},ESTTRi,ESTEMITi);
[PSTATESao,logpseqao] = hmmdecode(ta{1},ESTTRo,ESTEMITo);
[PSTATESau,logpseqau] = hmmdecode(ta{1},ESTTRu,ESTEMITu);

% Finding probability of similarity of recorded sample of 'E' with
 HMMs
[PSTATESea,logpseqea] = hmmdecode(te{1},ESTTRa,ESTEMITa);
[PSTATESee,logpseqee] = hmmdecode(te{1},ESTTRe,ESTEMITe);
[PSTATESei,logpseqei] = hmmdecode(te{1},ESTTRi,ESTEMITi);
[PSTATESeo,logpseqeo] = hmmdecode(te{1},ESTTRo,ESTEMITo);
[PSTATESeu,logpseqeu] = hmmdecode(te{1},ESTTRu,ESTEMITu);

% Finding probability of similarity of recorded sample of 'I' with
 HMMs
[PSTATESia,logpseqia] = hmmdecode(ti{1},ESTTRa,ESTEMITa);
[PSTATESie,logpseqie] = hmmdecode(ti{1},ESTTRe,ESTEMITe);
[PSTATESii,logpseqii] = hmmdecode(ti{1},ESTTRi,ESTEMITi);
[PSTATESio,logpseqio] = hmmdecode(ti{1},ESTTRo,ESTEMITo);
[PSTATESiu,logpseqiu] = hmmdecode(ti{1},ESTTRu,ESTEMITu);

% Finding probability of similarity of recorded sample of 'O' with
 HMMs
[PSTATESoa,logpseqoa] = hmmdecode(to{1},ESTTRa,ESTEMITa);
[PSTATESoe,logpseqoe] = hmmdecode(to{1},ESTTRe,ESTEMITe);
[PSTATESoi,logpseqoi] = hmmdecode(to{1},ESTTRi,ESTEMITi);
[PSTATESoo,logpseqoo] = hmmdecode(to{1},ESTTRo,ESTEMITo);
[PSTATESou,logpseqou] = hmmdecode(to{1},ESTTRu,ESTEMITu);

% Finding probability of similarity of recorded sample of 'U' with
 HMMs
[PSTATESua,logpsequa] = hmmdecode(tu{1},ESTTRa,ESTEMITa);
[PSTATESue,logpseque] = hmmdecode(tu{1},ESTTRe,ESTEMITe);
[PSTATESui,logpsequi] = hmmdecode(tu{1},ESTTRi,ESTEMITi);
[PSTATESuo,logpsequo] = hmmdecode(tu{1},ESTTRo,ESTEMITo);
[PSTATESuu,logpsequu] = hmmdecode(tu{1},ESTTRu,ESTEMITu);

% probability matrix of each vowel on each HMM
prob_mat = zeros(5,5);
% row 1 contains probability of recorded sample vowel 'A' over
 trained ...
% HMMs of vowel 'A','E','I','O','U' as columns 1,2,3,4,5 respectively
% and, row 2 contains probabilities for reocorded vowel 'B' and so on
%    'A' 'E' 'I' 'O' 'U'
% 'A' x   x   x   x   x
% 'E' x   x   x   x   x
% 'I' x   x   x   x   x
% 'O' x   x   x   x   x
% 'U' x   x   x   x   x

% row containing probabilities of recorded sample on each trained HMM
prob_mat(1,1) = logpseqaa;
prob_mat(1,2) = logpseqae;
prob_mat(1,3) = logpseqai;
```

```
prob_mat(1,4) = logpseqao;
prob_mat(1,5) = logpseqau;

prob_mat(2,1) = logpseqea;
prob_mat(2,2) = logpseqee;
prob_mat(2,3) = logpseqei;
prob_mat(2,4) = logpseqeo;
prob_mat(2,5) = logpseqeu;

prob_mat(3,1) = logpseqia;
prob_mat(3,2) = logpseqie;
prob_mat(3,3) = logpseqii;
prob_mat(3,4) = logpseqio;
prob_mat(3,5) = logpseqiu;

prob_mat(4,1) = logpseqoa;
prob_mat(4,2) = logpseqoe;
prob_mat(4,3) = logpseqoi;
prob_mat(4,4) = logpseqoo;
prob_mat(4,5) = logpseqou;

prob_mat(5,1) = logpsequa;
prob_mat(5,2) = logpseque;
prob_mat(5,3) = logpsequi;
prob_mat(5,4) = logpsequo;
prob_mat(5,5) = logpsequu;

% changing the value of max probability to '1' to observe clearly if
 max
% probability lies on diagonals of the matrix as expected if
% the model predicts accurately
for i=1:5
    [mm,ii]=max(prob_mat(i,:));
    prob_mat(i,ii) = 1;
end

% Final probability matrix obtained after testing,
%        'A'       'E'       'I'       'O'       'U'
% 'A'     1     -9.5e+03  -1.2e+04  -1.0e+04  -1.1e+04
% 'E' -8.6e+03      1     -1.3e+04  -1.3e+04  -1.3e+04
% 'I' -1.0e+04  -1.1e+04      1     -8.4e+03  -8.9e+03
% 'O' -8.0e+03  -8.8e+03  -7.2e+03      1     -7.6e+03
% 'U' -8.1e+03  -9.2e+03  -7.5e+03  -7.4e+03      1%
%
% as we put '1' where the probability is max so we can clearly
 observe,
% that HMM models for each vowel accurately predicted the to which
% model the recorded sample belongs
```

Published with MATLAB® R2016b