EEEM030 Assignment 2: Speech recognition

Kongkea Ouch

Abstract

This purpose of this assignment is to train an existing Hidden Markov Model with new observations by calculating several values using the Baum-Welch formulas, which belong to Expectation-Maximization algorithm. All calculations and plotting are performed in Matlab program with provided inputs of the model's matrix of state-transition probabilities, matrix of parameters and sequence of observations. At the end, the outcome of the computations will lead to another model reinforced from new observations and its performance in comparison to the original model will be discussed.

Introduction to Hidden Markov Model

A Markov chain is described by Rabiner [1] as a model that generates stochastic sequences of states with each state possibly deriving value from anything such as symbols, phonemes, words, or even the weather. With a Markov chain, we can directly observe the state sequence. However, for a model which we cannot see the underlying state sequence except the outputs, it is called the Hidden Markov Model (HMM) which is based on a Markov chain [1]. The standard HMM's the state sequence is discrete while the output probabilities of observations can be generated from a discrete distribution or a continuous distribution based on an HMM lecture by Dr Jackson [2].

Plot of state topology and output probability density functions b_i

As the continuous HMM is considered in this assignment, the parameters of this continuous model (λ) are put together in the lecture [2] in below forms:

Matrix of state-transition probabilities,

$$A = \{\pi_i, a_{ij}, \eta_i\} = \{P(x_t = j|\, x_{t-1} = i\} \qquad \text{for } 1 \le i, j \le N \qquad (1)$$

Continuous output probability densities,

$$B = \{b_i(o_t)\} = \{p(o_t|x_t = i)\} \qquad \text{for } 1 \le i \le N$$

where $b_i(o_t)$ is the output probability density (pdf) of an observation o_t can be generated from a Gaussian distribution of state i

$$b_i(o_t) = \mathcal{N}(o_t, \mu_i, \Sigma_i)$$

$$= \frac{1}{\sqrt{2\pi\,\Sigma_i}} \exp\left(\frac{-(o-\mu_i)^2}{2\Sigma_i}\right) \qquad (2)$$

evaluated at o_t with mean μ_i and variance Σ_i

Given A:

0	0.44	0.56	0
0 0	0.92 0.04	0.06 0.93	0.02 0.03
0	0	0	0

Table 1: State matrix

We can apply (1) to break down and visualize A in state topology accordingly:

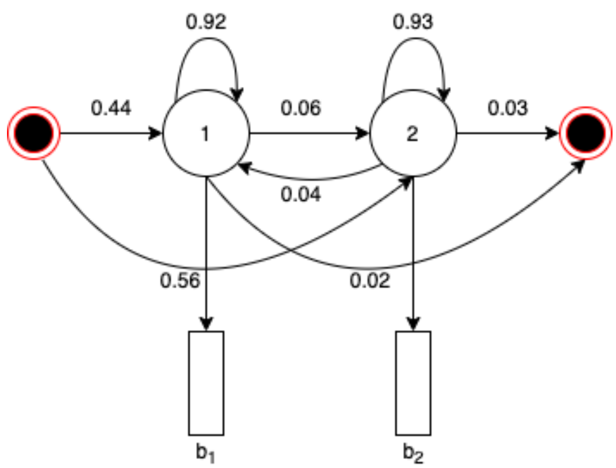

Figure 1: State topology

We do not know what discrete state sequence takes place to produce observations which is why each b_i is seen as empty. However, we can still determine and plot the output pdf of each state provided:

State i	1	2
Mean μ_i	1.00	4.00
Variance Σ_i	1.44	0.49

Table 2: Output parameters

The means, variances and values to be evaluated at (-3 to 10 with an increment of 0.1) are put into equation (2) to generate the output pdf values and then plotting proceeds.

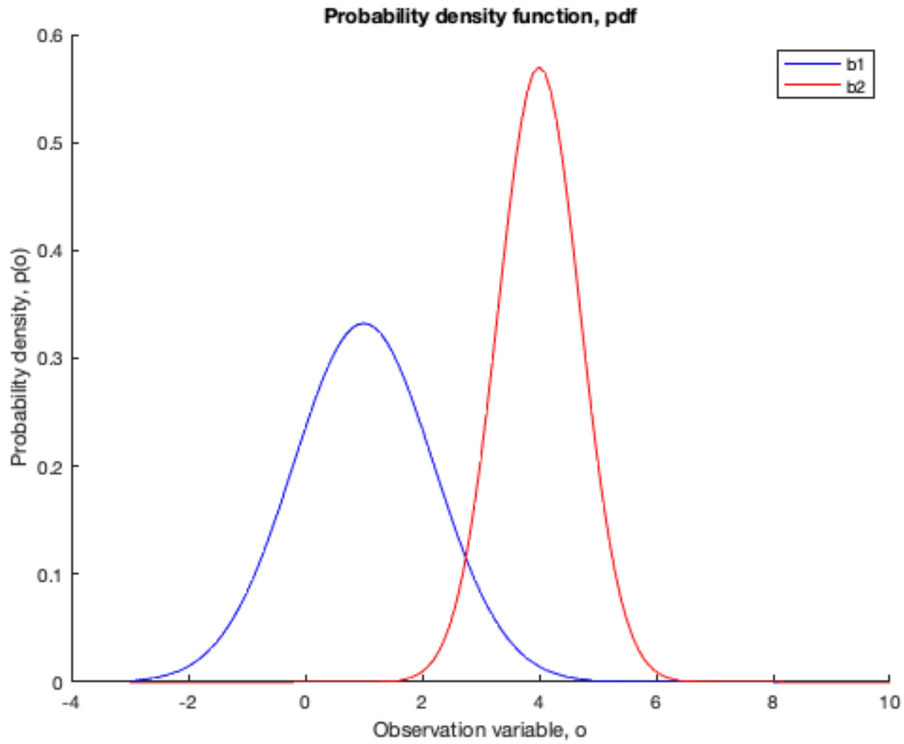

Figure 2: Output pdfs b_i

Both output pdfs reflect the means and variances from table 2 with state 2's probability densities having higher mean than state 1 while deviating less than state 1. Notably, state 1 and 2 share some intersecting distribution which means some observations which are less likely to be generated by a state are better classified under another state.

Output probability density for each time frame and state $\boldsymbol{b_i(o_t)}$

Given $O = \{3.8, 4.2, 3.4, -0.4, 1.9, 3.0, 1.6, 1.9, 5.0\}$ for $t = 1$ with $T = 9$, we can evaluate $b_i(o_t)$ for state 1 and 2 at each frame $t = 1 \dots 9$ by using equation (2) and plot:

$$B = \begin{bmatrix} 0.022 & 0.009 & 0.045 & 0.168 & 0.250 & 0.082 & 0.293 & 0.250 & 0.001 \\ 0.547 & 0.547 & 0.394 & 1.5e^{-9} & 0.006 & 0.205 & 0.001 & 0.006 & 0.205 \end{bmatrix}$$

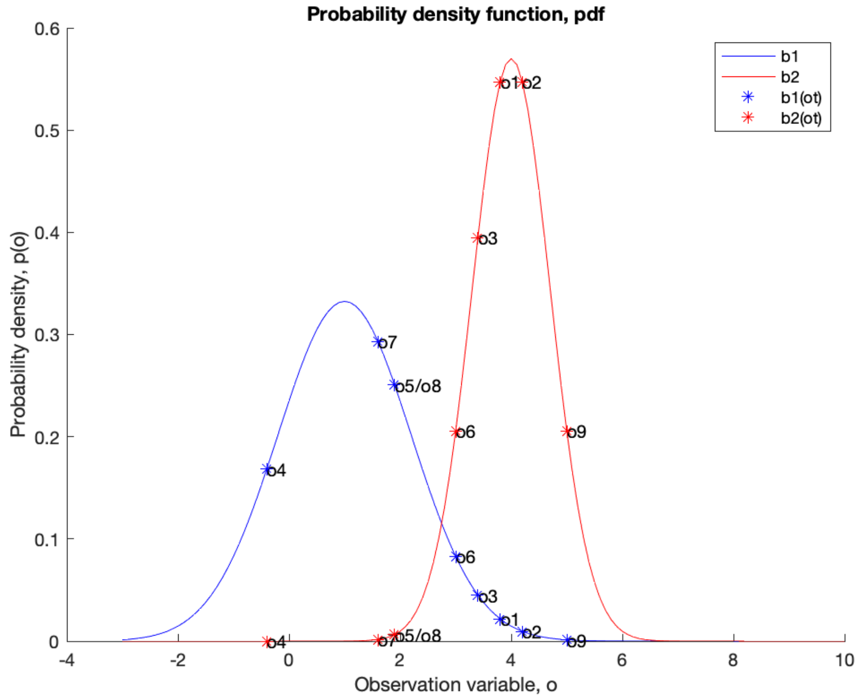

Figure 3: Output pdfs $b_i(o_t)$

State 2 likely generates o_1 (3.8), o_2(4.2), o_3 (3.4), o_6 (3.0), o_9 (5.0) as these values average close to state 2's mean of 4.00 but deviate further from state 1. Similarly, o_4 (−0.4), o_5 (1.9), o_7(1.6), o_8 (1.9) are mostly likely to be produced by state 1 as they all resonate with state 1's mean of 1 and deviate away from state 2.

Forward likelihoods $\boldsymbol{\alpha_t(i)}$ and overall likelihood of the observations $\boldsymbol{P(\mathcal{O}|\lambda)}$

Gold & Morgan [3] suggest using either forward or backward algorithm (dynamic programming) to efficiently make use of HMM state's conditional dependence and sum over all possible unknown paths that could lead to the sequence of observations.

Forward likelihood [2] is represented by cell $\boldsymbol{\alpha_t(i)}$ which is calculated by summing over all probabilities of extended paths from left to right side and fold them into a trellis:

1. Initialize at $t = 1$,

$\alpha_1(i) = \pi_i b_i(o_1)$ for $1 \leq i \leq N$

2. Recur for $t = \{2,3, \ldots, T\}$,

$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(o_t)$ for $1 \leq j \leq N$

3. Terminate with overall likelihood,

$P(\mathcal{O}|\lambda)$ = $\sum_{i=1}^{N} \alpha_T(i)\eta_i$

$\alpha_1(1) = 0.009$

$\alpha_1(2) = 0.306$

$\alpha_2(1) = 2.003e^{-04}$

$\alpha_2(2) = 0.156$

$\alpha_3(1) = \ 2.894e^{-04}$

$\alpha_3(2) = 0.057$

$\alpha_4(1) = 4.309e^{-04}$

$\alpha_4(2) = 8.006e^{-11}$

$\alpha_5(1) = \ 9.949e^{-05}$

$\alpha_5(2) = \ 1.637e^{-07}$

$\alpha_6(1) = \ 7.588e^{-06}$

$\alpha_6(2) = \ 1.257e^{-06}$

$\alpha_7(1) = \ 2.063e^{-06}$

$\alpha_7(2) = \ 2.594e^{-09}$

$\alpha_8(1) = \ 4.763e^{-07}$

$\alpha_8(2) = \ 7.989e^{-10}$

$\alpha_9(1) = \ 5.632e^{-10}$

$\alpha_9(2) = \ 6.023e^{-09}$

$P(\mathcal{O}|\lambda) = 1.919e^{-10}$ (3)

Backward likelihoods $\boldsymbol{\beta_t(i)}$

The backward likelihoods [2] are computed in a similar manner to forward likelihoods and should result in the same overall likelihood as well except they proceed from right to left side $T, t-1, \ldots, 1$:

1. Initialize at $t = T$,

$\beta_T(i) = \ \eta_i$ for $1 \ \leq i \ \leq N$

2. Recur for $t = \{2,3, \ldots, T\}$,

$\beta_t(i) = \ \sum_{j=1}^{N} \alpha_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j)$ for $1 \ \leq i \leq N$

3. Terminate with overall likelihood,

$P(\mathcal{O}|\lambda)$ = $\sum_{i=1}^{N} \pi_i b_i(o_1)\, \beta_1(i)$

$\beta_9(1) = \ 0.020$

$\beta_9(2) = \ 0.030$

$\beta_8(1) = \ 3.934e^{-04}$

$\beta_8(2) = \ 0.005$

$\beta_7(1) = 9.300e^{-05}$

$\beta_7(2) = 3.770e^{-05}$

$\beta_6(1) = 2.510e^{-05}$

$\beta_6(2) = 1.147e^{-06}$

$\beta_5(1) = 1.929e^{-06}$

$\beta_5(2) = 3.024e^{-07}$

$\beta_4(1) = 4.454e^{-07}$

$\beta_4(2) = 2.114e^{-08}$

$\beta_3(1) = 6.898e^{-08}$

$\beta_3(2) = 2.999e^{-09}$

$\beta_2(1) = 2.926e^{-09}$

$\beta_2(2) = 1.225e^{-09}$

$\beta_1(1) = 6.578e^{-11}$

$\beta_1(2) = 6.245e^{-10}$

$P(\mathcal{O}|\lambda) = 1.919e^{-10}$ (4)

As expected, the overall likelihoods of the observations of (3) and (4) are the same.

Occupation likelihoods $\boldsymbol{\gamma_t(i)}$

The Baum-Welch expectation [2], a part of the Expectation-Maximization (EM) technique, is used to compute probability of a state occupancy being at time t which involves earlier forward and backward probabilities:

$$\gamma_t(i) = \frac{\alpha_t(i)\,\beta_t(i)}{P(\mathcal{O}|\lambda)}$$

$\gamma_1(1) = 0.003$

$\gamma_1(2) = 0.996$

$\gamma_2(1) = 0.003$

$\gamma_2(2) = 0.996$

$\gamma_3(1) = 0.104$

$\gamma_3(2) = 0.896$

$\gamma_4(1) = 1.000$

$\gamma_4(2) = 8.817e^{-09}$

$\gamma_5(1) = 0.999$

$\gamma_5(2) = 2.579e^{-04}$

$\gamma_6(1) = 0.992$

$\gamma_6(2) = 2.579e^{-04}$

$\gamma_7(1) = 0.999$

$\gamma_7(2) = 5.095e^{-04}$

$\gamma_8(1) = 0.976$

$\gamma_8(2) = 0.023$

$\gamma_9(1) = 0.058$

$\gamma_9(2) = 0.941$

Re-estimated means and variances

We can now start re-estimating the maximum likelihood of HMM's output means and variances by incorporating the calculated occupation likelihoods and given observations by using Baum-Welch's maximization [2]:

$$\hat{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) o_t}{\sum_{t=1}^{T} \gamma_t(i)}$$

$$\hat{\Sigma}_i = \frac{\sum_{t=1}^{T} \gamma_t(i)(o_t - \mu_i)(o_t - \mu_i)^T}{\sum_{t=1}^{T} \gamma_t(i)}$$

State i	1	2
Mean $\hat{\mu}_i$	1.674	4.089
Variance $\hat{\Sigma}_i$	1.846	0.378

Table 3: Re-estimated output parameters

Plot of the trained output pdfs

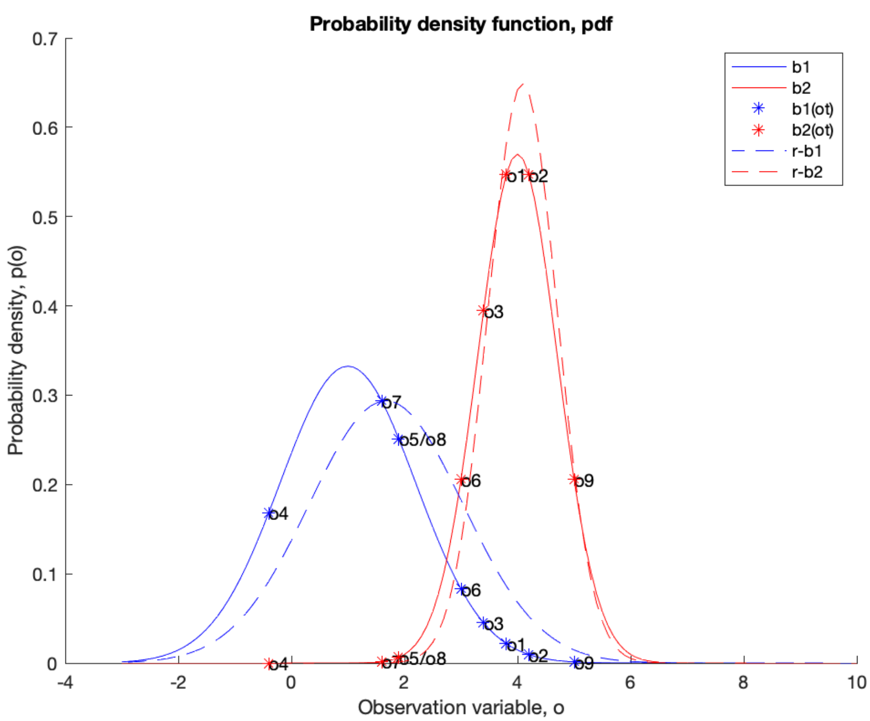

Figure 4: Trained output pdfs

Using the new means and variances, we can now plot the new pdfs in comparison to the original pdfs of figure 3 in a similar way. Looking at the graphs, the overall likelihood of new output pdfs seems to perform better than the initial output pdfs as both states' means increase. State 1's distribution spreads out significantly than before while state 2's distribution slightly gets closer around the slightly rose mean and still maintain similar shape with improvement in the right direction as compared to previous pdf. Hence, it can be inferred that this model has not reached its local optimum yet from only one training iteration and still need to learn from more samples that consistently represent the model and yield better re-estimation.

Conclusion

This assignment implements the EM training of the Hidden Markov Model which is ideal for basic speech recognition as it resembles the stochastic nature of human's utterance which can be broken down into probabilistic state sequence. From the re-estimated model's output parameters, it can be seen that the probability of the new model improves over the old one and reflects the assimilation of training data. Still, more quality iterations are needed so as to find the potential local optimum and enhance classification ability of both states especially state 1 that seems to falter in comparison to state 2 in the first iteration.

References

[1] L. Rabiner, "A tutorial on HMM and selected applications in speech recognition," *Proceeding-s of the IEEE*, vol. 77, no. 2, pp. 257-286, 1989.

[2] P. Jackson, "Hidden Markov Models," [Online]. Available: http://personal.ee.surrey.ac.uk/Per-sonal/P.Jackson/ISSPR/hmm_isspr11_hw.pdf. [Accessed 15 December 2020].

[3] B. Gold, N. Morgan and D. Ellis, Speech and Audio Signal Process, 2nd ed., New York: Wiley-Blackwell, 2011.

Appendix: Matlab implementation code

```
close all
clear all
clc

% Entry probabilities
pi0 = [0.44 0.56];

% State-transition probabilities
a = [0.92 0.06;
     0.04 0.93];

% Exit probabilities
n = [0.02 0.03];

% Mean and variance defining the states'
% output probability densities bi(ot)
mean = [1.00 4.00];
variance = [1.44 0.49];
stdev = sqrt(variance);

% O: Observation sequence
O = [3.8 4.2 3.4 -0.4 1.9 3.0 1.6 1.9 5.0];

% Emitting states
N = length(mean);

% Time frames
T = length(O);

%% Plot output pdfs for each state
figure()
title("Probability density function, pdf")
xlabel("Observation variable, o");
ylabel("Probability density, p(o)")

x = -3:0.1:10;
y = zeros(N, length(x));
colors = {'b','r'};

for i = 1:N
    for t = 1:length(x)
        y(i,t) = (1 / sqrt(2*pi*variance(i))) * exp((-(x(t)-mean(i))^2) /
(2*variance(i)));
    end

    hold on
    plot(x,y(i,:),colors{i},'DisplayName',['b' num2str(i)]);
end

%% Output probability density for each time frame and state
b = zeros(N,T);
b_colors = {'b*','r*'};

for i = 1:N
```

```
    for t = 1:T
        b(i,t) = (1 / sqrt(2*pi*variance(i))) * exp((-(O(t)-mean(i))^2) /
(2*variance(i)));

        if t == 5
            o_label = 'o5/o8';
        elseif t == 8
            o_label = '';
        else
            o_label =   ['o' num2str(t)];
        end

        hold on
        text(O(t),b(i,t),o_label);
    end

    hold on
    plot(O,b(i,:),b_colors{i},'DisplayName',['b' num2str(i) '(ot)']);

end

%% Forward likelihoods and overall likelihood of the observations
alpha = zeros(T,N);

% Initialise at t = 1
for i = 1:N
    alpha(1,i) = pi0(i)*b(i,1);
end

% Recur for t = {2,3,...,T}
for t = 2:T
    for j = 1:N
        for i = 1:N
            alpha(t,j) = alpha(t,j) + alpha(t-1,i)*a(i,j);
        end

        alpha(t,j) = alpha(t,j)*b(j,t);
    end
end

% Finally get overall likelihood of the observations
P_fwd = 0;

for i = 1:N
    P_fwd = P_fwd + alpha(T,i)*n(i);
end

%% Backward likelihoods of the observations
beta = zeros(T,N);

% Initialise at t = T
for i = 1:N
    beta(T,i) = n(i);
end

% Recur for t = {T-1,T-2,...,1}
```

```
for t = T-1:-1:1
    for i = 1:N
        for j = 1:N
            beta(t,i) = beta(t,i) + a(i,j)*b(j,t+1)*beta(t+1,j);
        end
    end
end

% Finally get overall likelihood of the observations
P_bwd = 0;

for i = 1:N
    P_bwd = P_bwd + pi0(i)*b(i,1)*beta(1,i);
end

%% Occupation likelihoods
gamma = zeros(T,N);

for t = 1:T
    for i = 1:N
        gamma(t,i) = (alpha(t,i)*beta(t,i)) / P_fwd;
    end
end

%% Transition likelihood
xi = zeros(N,N,T);

for t = 2:T
    for i = 1:N
        for j = 1:N
            xi(i,j,t) = (alpha(t-1,i)*a(i,j)*b(j,t)*beta(t,j)) / P_fwd;
        end
    end
end

%% Re-estimated means and variances
r_mean = zeros(1,N);
r_variance = zeros(1,N);

r_mean_numerator = zeros(1,N);
r_variance_numerator = zeros(1,N);
sum_gamma = zeros(1,N);

for i = 1:N
    for t = 1:T
        r_mean_numerator(i) = r_mean_numerator(i) + gamma(t,i)*O(t);
        r_variance_numerator(i) = r_variance_numerator(i)  +
gamma(t,i)*(O(t)-mean(i))*(O(t)-mean(i))';

        sum_gamma(i) = sum_gamma(i) + gamma(t,i);
    end

    r_mean(i) = r_mean_numerator(i)/sum_gamma(i);
    r_variance(i) = r_variance_numerator(i)/sum_gamma(i);
end
```

```matlab
%% Re-estimated state-transition
r_a = zeros(N,N);

for i = 1:N
    for j = 1:N
        r_a(i,j) = sum(xi(i,j,:)) / sum(gamma(:,i));
    end
end


%% Plots of the pdfs from re-estimation
r_colors = {'b--','r--'};
r_y = zeros(N, length(x));
r_stdev = sqrt(r_variance);

for i = 1:N
    r_y(i,:) = normpdf(x,r_mean(i),r_stdev( i));
    hold on
    plot(x,r_y(i,:),r_colors{i},'DisplayName',['r-b' num2str(i)])
end

legend
```