NCTS: Hidden Markov Models and Its Application

Ming-Kai Lin, Yan-Yu Chen, Po-Ying Chen,
Chi-Hao Fang, Wen-Tai Hsu
Mentors: Gi-Ren Liu (NCKU), Yuan-Chung Sheu (NCTU)

August 9, 2019

Contents

1 Motivations

- Sleep stage is related to health. We want to estimate the real sleep stages with the observation data so as to enhance the convenience and efficiency of diagnosis.
- This project will use the hidden Markov models and codebook method to classify the observation states, compute the transition and the emission matrix, and predict the hidden states.

2 Codebook

Goal: Given a time sequence $X = \{X_t\}_{t=1}^{T}$, how to find a function f such that $f : X \to \{1, 2, ..., K\}$ for some predetermined $K \in \mathbb{N}$?

Definition 1 (Method 1: K-means)**.** $\forall k \in \{1, 2, ..., K\}$, we define $C_k^{(i)} := \{x_j \in X \mid \left\|x_j - \mu_k^{(i)}\right\| \leq \left\|x_j - \mu_{k'}^{(i)}\right\|, \quad \forall k' \neq k\}$.
Repeat this process; then, we set $f(x_j) = k$, if $x_j \in C_k^{(\text{final})}$. The function f classifies all elements in X into K distinctive classes.

- Advantage: Low complexity and no requirement of the labels information
- Disadvantage: Different initial mean μ_k^1 causes different clustering, and low stability of the algorithm.

After having labeled(T_1 datas) by the experienced, cluster the others by K-mean.

Definition 2 (Method 2: Adjusted K-means)**.** Given a time sequence $X = \{x_t\}_{t=1}^{T_1+T_2}$
$\forall y \in Y = \{1, 2, ..., 5\}$
$X_y := \{x_t \ : \ 1 \leq t \leq T_1, y_t = y\}$
Apply K-mean to each X_y, generates
$\gamma_y : \{\mu_1^y, \mu_2^y, \mu_3^y\}$
Redenote $\{\mu_1, ..., \mu_{15}\}$
Coding $\{x_t\}_{t=T_1+1}^{T_1+T_2}$

Next, we take the advantage of the time information.
ex:

$$\begin{matrix} ..., A\ , & (B,B) & , & (B,B) & , & (B,B) & , A\ , ... \\ & \in X_{B,in} & & \in X_{B,mid} & & \in X_{B,out} & \end{matrix}$$

where $X_B = \{x_t : 1 \leq t \leq T, y_t = B\} = X_{B,in} \bigcup X_{B,mid} \bigcup X_{B,out}$.
Apply K-means to $X_{B,in}$, $X_{B,mid}$, $X_{B,out}$ to get γ_B.

3 Markov Chain Models

Notations:

- $S_X = \{s_1, s_2, ..., s_n\}$: the state space
- $\{X_t\}_{t=1}^T$: a sequence of S_X-valued random variables defined on a suitable probability space
- $\{x_t\}_{t=1}^T$: a sequence of the outcomes of $\{X_t\}_{t=1}^T$
- $\pi = (\pi(i))_{i=1}^n = (\pi(s_i))_{i=1}^n$: the initial distribution of $\{X_t\}_{t=1}^T$
- $A = (a_{ij}) \in \mathbb{R}^{n\times n}$: the transition matrix of $\{X_t\}_{t=1}^T$, where $a_{ij} = a_{s_i s_j} = \mathbb{P}(X_t = s_j | X_{t-1} = s_i)$ is the probability of going from state s_i at time $t-1$ to state s_j at time t
- $\{X_1^t = x_1^t\}$: the abbreviation of $\{X_1 = x_1, X_2 = x_2, ..., X_t = x_t\}$ for $t = 1, 2, ..., T$.

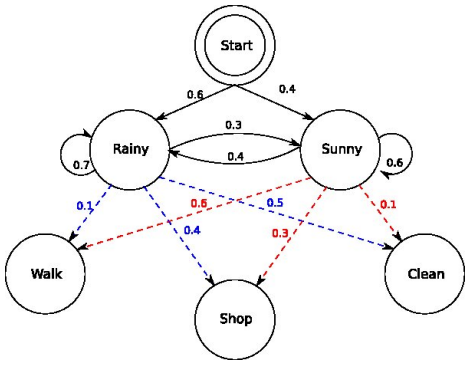
Definition 3 (Markov Property)**.** A process $\{X_t\}_{t=1}^T$ is called a *Markov chain* if it satisfies $\mathbb{P}(X_t = x_t | X_1^{t-1} = x_1^{t-1}) = \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1})$ for $t = 2, 3, ..., T$.

Thus, we can derive the joint probability distribution of $(X_1, X_2, ..., X_T)$ by
$\mathbb{P}(X_1^T = x_1^T)$
$= \mathbb{P}(X_1 = x_1) \prod_{t=2}^T \mathbb{P}(X_t = x_t | X_1^{t-1} = x_1^{t-1})$ *Tower Law*
$= \mathbb{P}(X_1 = x_1) \prod_{t=2}^T \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1})$ *Markov Property*
$= \pi(x_1) \prod_{t=2}^T a_{x_{t-1} x_t}$.
Furthermore, the result is also called *the time-homogeneous property.*

4 Hidden Markov Models

Notations:

- $S_X = \{s_1, ...s_n\}$: the hidden state space
- $S_Y = \{v_1, ...v_m\}$: the observation state space
- $\{X_t\}_{t=1}^T$: a hidden state process, that is, a sequence of random variables taking values in S_X
- $\{x_t\}_{t=1}^T$: a realization of the hidden state process $\{X_t\}_{t=1}^T$
- $\{Y_t\}_{t=1}^T$: an observation state process, that is, a sequence of random variables taking values in S_Y
- $\{y_t\}_{t=1}^T$: a realization of the observation state process $\{Y_t\}_{t=1}^T$
- $\pi = (\pi(i))_{i=1}^n = (\pi(s_i))_{i=1}^n$: the initial distribution of $\{X_t\}_{t=1}^T$



Notations:

- $A = (a_{ij}) \in \mathbb{R}^{n\times n}$: the transition matrix of $\{X_t\}_{t=1}^T$, where $a_{ij} = a_{s_i s_j} = \mathbb{P}(X_t = s_j | X_{t-1} = s_i)$ is the probability of going from state s_i at time $t-1$ to state s_j at time t
- $B = (b_j(k)) \in \mathbb{R}^{n\times m}$: the emission matrix of $\{Y_t\}_{t=1}^T$, where $b_j(k) = b_{s_j}(v_k) = \mathbb{P}(Y_t = v_k | X_t = s_j)$ is the probability of observing the state v_k given by the hidden state s_j

Definition 4 (Hidden Markov Model)**.** A process $\{(X_t, Y_t)\}_{t=1}^T$ is called a *hidden Markov model* if it satisfies

$$\begin{aligned}&\mathbb{P}(X_t = x_t, Y_t = y_t | X_1^{t-1} = x_1^{t-1}, Y_1^{t-1} = y_1^{t-1})\\ &= \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}) \mathbb{P}(Y_t = y_t | X_t = x_t)\end{aligned}$$

for $t = 2, 3, ..., T$.

With the previous notations, it can be written as $\mathbb{P}(X_t = x_t, Y_t = y_t | X_1^{t-1} = x_1^{t-1}, Y_1^{t-1} = y_1^{t-1}) = a_{x_{t-1}x_t} b_{x_t}(y_t)$.

Some observations of the hidden Markov model:

- $\{(X_t, Y_t)\}_{t=1}^T$ is a two-dimensional Markov chain.
- $\{X_t\}_{t=1}^T$ is a Markov chain with initial distribution π and transition matrix A.
- $Y_1, ..., Y_T$ are conditionally independent given $X_1, ..., X_T$, that is, $\mathbb{P}(Y_1^T = y_1^T | X_1^T = x_1^T) = \prod_{t=1}^T b_{x_t}(y_t)$.
- The joint probability distribution of $(X_1, Y_1, ..., X_T, Y_T)$ is $\mathbb{P}_\lambda(X_1^T = x_1^T, Y_1^T = y_1^T) = \pi(x_1) b_{x_1}(y_1) \prod_{t=2}^T a_{x_{t-1}x_t} b_{x_t}(y_t)$, where $\lambda = (\pi, A, B)$ is the parameter of the hidden Markov model.

Here is an example of the hidden Markov model.
Three problems of the hidden Markov models:

1. Scoring: Given the observation $\{y_t\}_{t=1}^T$ and the parameter λ, how do we compute the score $P_\lambda(Y_1^T = y_1^T)$?

2. Decoding: Given the observation $\{y_t\}_{t=1}^T$ and the parameter λ, how do we choose the corresponding optimal hidden state sequence that is the best explanation of the observations?

3. Estimation: How do we adjust the parameter λ to maximize $P_\lambda(Y_1^T = y_1^T)$?

Since we would like to predict the subjects' hidden states, our experiment will follow by estimating the sleeping data, scoring by the forward and backward algorithm with rescaling, and decoding the hidden state sequence via the Baum-Welch algorithm.

Proposition 1 (Estimation)**.** *Given realizations $\{x_t\}_{t=1}^T$ and $\{y_t\}_{t=1}^T$, we estimate the transition matrix $\hat{A} = (\hat{a}_{ij}) \in \mathbb{R}^{n\times n}$ and the emission matrix $\hat{B} = (\hat{b}_j(k)) \in \mathbb{R}^{n\times m}$ by*

$$\hat{a}_{ij} = \frac{\#(\{t \mid X_{t+1} = s_j, X_t = s_i\})}{\#(\{t \mid X_t = s_i\})}$$

and

$$\hat{b}_j(k) = \frac{\#(\{t \mid Y_t = v_k, X_t = s_j\})}{\#(\{t \mid X_t = s_j\})},$$

where $\#(\,\cdot\,)$ means the number of the events.

5 Baum-Welch Algorithm

Given a realization $\{y_t\}_{t=1}^T$ of the observation process $\{Y_t\}_{t=1}^T$, we would like to predict "the most possible outcome" of the hidden state X_t at time t, denoted by $\hat{x}_t$. That is, $\hat{x}_t = \arg\max_{s_j\in S_x} \mathbb{P}(X_t = s_j|Y_1^T = y_1^T)$. With some knowledge of the conditional probability, we can derive that

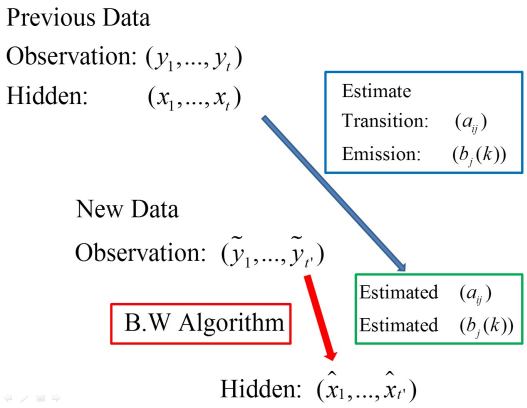
$$\begin{aligned}
\mathbb{P}(X_t = s_j|Y_1^T = y_1^T) &= \frac{\mathbb{P}(Y_1^t = y_1^t, X_t = s_j, Y_{t+1}^T = y_{t+1}^T)}{\mathbb{P}(Y_1^T = y_1^T)} \\
&= \frac{\mathbb{P}(Y_1^t = y_1^t, X_t = s_j)}{\mathbb{P}(Y_1^T = y_1^T)} \frac{\mathbb{P}(Y_1^t = y_1^t, X_t = s_j, Y_{t+1}^T = y_{t+1}^T)}{\mathbb{P}(Y_1^t = y_1^t, X_t = s_j)} \\
&= C \cdot \mathbb{P}(Y_1^t = y_1^t, X_t = s_j)\mathbb{P}(Y_{t+1}^T = y_{t+1}^T|Y_1^t = y_1^t, X_t = s_j),
\end{aligned}$$

where $C = (\mathbb{P}(Y_1^T = y_1^T))^{-1}$ is independent of the choices of s_j.

Therefore, we are curious about the following two quantities.

Definition 5.

- $\alpha_t(j) = \mathbb{P}(Y_1^t = y_1^t, X_t = s_j)$: the probability of the *prior* observation sequence from 1 to t and the hidden state s_j at time t
- $\beta_t(j) = \mathbb{P}(Y_{t+1}^T = y_{t+1}^T|X_t = s_j)$: the probability of the *later* observation sequence from $t+1$ to T given the hidden state s_j at time t



Then, we now introduce *the Baum-Welch Algorithm* to decode the optimal hidden state process $\{X_t\}_{t=1}^T$.

Theorem 1 (Baum-Welch Algorithm)**.** $\hat{x}_t = \arg\max_{j\in\{1,2,...,n\}} \alpha_t(j)\beta_t(j)$ *for* $t = 1, 2, ..., T$.

The procedure of Baum-Welch algorithm is demonstrated by the following figure.

6 Forward Algorithm

To calculate $\alpha_t(j) = \mathbb{P}(Y_1^t = y_1^t, X_t = s_j) =$ the probability of the prior observation sequence from 1 to t and the hidden state s_j at time t
(a) Initiation: $t = 1$
$\alpha_1(j) = \mathbb{P}(Y_1 = y_1, X_1 = s_j) = \pi(j)b_j(y_1)$ for $j = 1, 2, ..., n$

(b) Induction: $t = 2, 3, ..., T$

$$\begin{aligned}
\alpha_t(j) =& \mathbb{P}(Y_1^t = y_1^t, X_t = s_j) \\
=& \sum_{i=1}^{n} \mathbb{P}(Y_1^{t-1} = y_1^{t-1}, X_{t-1} = s_i, Y_t^T = y_t^T, X_t = s_j) \\
& \textit{by the partition on } X_{t-1} \\
=& \sum_{i=1}^{n} \mathbb{P}(Y_1^{t-1} = y_1^{t-1}, X_{t-1} = s_i) \\
& \times \mathbb{P}(Y_t^T = y_t^T, X_t = s_j | Y_1^{t-1} = y_1^{t-1}, X_{t-1} = s_i) \\
& \textit{by the conditional probability} \\
=& \sum_{i=1}^{n} \mathbb{P}(Y_1^{t-1} = y_1^{t-1}, X_{t-1} = s_i) \\
& \times \mathbb{P}(Y_t^T = y_t^T, X_t = s_j | Y_1^{t-1} = y_1^{t-1}, X_{t-1} = s_i) \\
& \textit{by the hidden Markov model} \\
=& \sum_{i=1}^{n} \alpha_{t-1}(i) a_{ij} b_j(y_t)
\end{aligned}$$

(c) Termination:

$$\mathbb{P}(Y_1^T = y_1^T) = \sum_{i=1}^{N} \alpha_T(i)$$

We conclude the forward algorithm by the following pseudocode.

Theorem 2 (Forward Algorithm)**.**

1. *initial* $\alpha_1(j) = \pi(j) b_j(y_1)$ *for* $j = 1, 2, ..., n$
2. *for* $t = 2 : T$
   $\alpha_t(j) = \sum_{i=1}^{n} \alpha_{t-1}(i) a_{ij} b_j(y_t)$ *for* $j = 1, 2, ..., n$

7 Backward Algorithm

To calculate $\beta_t(j) = \mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_t = s_j)$= the probability of the later observation sequence from $t+1$ to T given the hidden state s_j at time t
(a) Initiation: $t = T$,
$\beta_T(j) = 1$ for $j = 1, 2, ..., n$

(b) Induction: $t = T-1, T-2, ..., 1$

$$\begin{aligned}
\beta_t(j) =& \mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_t = s_j) \\
=& \sum_{i=1}^n \mathbb{P}(X_{t+1} = s_i, Y_{t+1}^T = y_{t+1}^T | X_t = s_j) \\
=& \sum_{i=1}^n \mathbb{P}(X_{t+1} = s_i, Y_{t+1} = y_{t+1} | X_t = s_j) \\
& \times \mathbb{P}(Y_{t+2}^T = y_{t+2}^T | X_t = s_j, X_{t+1} = s_i, Y_{t+1} = y_{t+1}) \\
=& \sum_{i=1}^n \mathbb{P}(X_{t+1} = s_i, Y_{t+1} = y_{t+1} | X_t = s_j) \\
& \times \mathbb{P}(Y_{t+2}^T = y_{t+2}^T | X_t = s_j, X_{t+1} = s_i, Y_{t+1} = y_{t+1}) \\
& \textit{by the hidden Markov model} \\
=& \sum_{i=1}^n \mathbb{P}(X_{t+1} = s_i, Y_{t+1} = y_{t+1} | X_t = s_j) \mathbb{P}(Y_{t+2}^T = y_{t+2}^T | X_{t+1} = s_i) \\
=& \sum_{i=1}^n a_{ji} b_i(y_{t+1}) \beta_{t+1}(i)
\end{aligned}$$

(c) Termination:

$$\begin{aligned}
\mathbb{P}(Y_1^T = y_1^T) &= \sum_{i=1}^n \mathbb{P}(X_1 = s_i, Y_1^T = y_1^T) \\
&= \sum_{i=1}^n \mathbb{P}(X_1 = s_i, Y_1 = y_1) \cdot \mathbb{P}(Y_2^T = y_2^T | X_1 = s_i, Y_1 = y_1) \\
&\quad \text{by the hidden Markov model} \\
&= \sum_{i=1}^n \mathbb{P}(X_1 = s_i, Y_1 = y_1) \cdot \mathbb{P}(Y_2^T = y_2^T | X_1 = s_i) \\
&= \sum_{i=1}^n \pi(i) b_i(y_1) \beta_1(i)
\end{aligned}$$

We conclude the backward algorithm by the following pseudocode.

Theorem 3 (Backward Algorithm)**.**

1. *initial* $\beta_T(j) = 1$ *for* $j = 1, 2, ..., n$
2. *for* $t = T-1 : 1$
   $\beta_t(j) = \sum_{i=1}^n a_{ji} b_j(y_{t+1}) \beta_{t+1}(i)$ *for* $j = 1, 2, ..., n$

8 Rescaling

Goal: to mitigate floating point error as t is large(resp. small) $\alpha_t(j)$ (resp.$\beta_t(j)$) is small enough to make $\alpha_t(j) \cdot \beta_t(j)$ machine zero.
We multiply $\alpha_t(j)$ by a constant depending only on time.

Set

$$C_t = \left(\sum_{i=1}^{n} \tilde{\alpha}_t(i) \right)^{-1} \quad \forall\ 2 \leq t \leq T$$

where $\tilde{\alpha}_t(j)$ is defined as:

$$\tilde{\alpha}_t(j) = \begin{cases} \alpha_t(j) & \text{if } t = 1 \\ \sum_{i=1}^{n} \hat{\alpha}_{t-1}(i) a_{ij} b_j(y_t) & \text{if } 2 \leq t \leq T \end{cases}$$

where $\hat{\alpha}_t(i)$, the value we recorded, is the normalized value of $\tilde{\alpha}_t(i)$, i.e.

$$\hat{\alpha}_t(i) = C_t \cdot \tilde{\alpha}_t(i) \text{ and } \sum_{i=1}^{n} \hat{\alpha}_t(i) = 1$$

And we have the following rescaling of $\alpha_t(i)$, and $\beta_t(i)$:

$$\hat{\alpha}_t(i) = \prod_{k=1}^{t} C_k \cdot \alpha_t(i) \text{ for } i = 1, 2, ..., n,\ t = 1, 2, ..., T$$

$$\hat{\beta}_t(i) = \prod_{k=t}^{T} C_k \cdot \beta_t(i) \text{ for } i = 1, 2, ..., n,\ t = 1, 2, ..., T$$

The algorithm in code goes as follows:

$$\alpha_1 = \tilde{\alpha}_1 \xrightarrow{C_1} \hat{\alpha}_1 \xrightarrow{F.W.} \tilde{\alpha}_2 \xrightarrow{C_2} \hat{\alpha}_2 \xrightarrow{F.W.} ... \xrightarrow{C_{T-1}} \hat{\alpha}_{T-1} \xrightarrow{F.W.} \tilde{\alpha}_T \xrightarrow{C_T} \hat{\alpha}_T$$

$$\beta_T = \tilde{\beta}_T \xrightarrow{C_T} \hat{\beta}_T \xrightarrow{B.W.} \tilde{\beta}_{T-1} \xrightarrow{C_{T-1}} \hat{\beta}_{T-1} \xrightarrow{B.W.} ... \xrightarrow{C_2} \hat{\beta}_2 \xrightarrow{B.W.} \tilde{\beta}_1 \xrightarrow{C_1} \hat{\beta}_1$$

Theorem 4. *The Baum-Welch algorithm can be improved by*

$$\hat{x}_t = \arg\max_j \alpha_t(j)\beta_t(j) = \arg\max_j \hat{\alpha}_t(j)\hat{\beta}_t(j)$$

Proof. Since

$$\hat{\alpha}_t(j)\hat{\beta}_t(j) = \prod_{k=1}^{t} C_k \alpha_t(i) \prod_{k=t}^{T} C_k \beta_t(i) = C \cdot C_t \left[\alpha_t(j)\beta_t(j)\right]$$

where $C = \prod_{k=1}^{T} C_k$, and C, C_t are independent of the choice of j. □

9 Second-order Hidden Markov Models

Now we consider an extension of previous model, we assume that $\{X_t\}_{t=1}^T$ is a second-order Markov chain. Also, we impose three assumptions of the extension:

1. $\mathbb{P}(X_t = x_t | X_{t-2}^{t-1} = x_{t-2}^{t-1}, Y_1^{t-1} = y_1^{t-1}) = \mathbb{P}(X_t = x_t | X_{t-2}^{t-1} = x_{t-2}^{t-1})$
2. $\mathbb{P}(Y_t = y_t | X_1^t = x_1^t, Y_{t-1} = y_{t-1}) = \mathbb{P}(Y_t = y_t | X_t = x_t, Y_{t-1} = y_{t-1})$
3. $\mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_{t-1}^t = x_{t-1}^t, Y_1^t = y_1^t)$
   $= \mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_{t-1}^t = x_{t-1}^t, Y_t = y_t)$

Notations:

- $A' = (a_{ij}) \in \mathbb{R}^{n \times n}$: the second stage transition matrix where $a_{ij} = a_{s_i s_j} = \mathbb{P}(X_2 = s_j | X_1 = s_i)$ is the probability of going from state s_i at time 1 to state s_j at time 2
- $A = (a_{ijk}) \in \mathbb{R}^{n \times n \times n}$: the transition matrix of $\{X_t\}_{t=2}^T$ where $a_{ijk} = a_{s_i s_j s_k} = \mathbb{P}(X_t = s_k | X_{t-2} = s_i, X_{t-1} = s_j)$ is the probability of going from state s_i at time $t-2$ and state s_j at time $t-1$ to state s_k at time t
- $B' = (b_j(k)) \in \mathbb{R}^{n \times m}$: the first-observation emission matrix, where $b_j(k) = b_{s_j}(v_k) = \mathbb{P}(Y_1 = v_k | X_1 = s_j)$ is the probability of observing the state v_k given by the hidden state s_j
- $B = (b_j(k|l)) \in \mathbb{R}^{n \times m \times m}$: the emission matrix of $\{Y_t\}_{t=2}^T$, where $b_j(k|l) = b_{s_j}(v_k | v_l) = \mathbb{P}(Y_t = v_k | X_t = s_j, Y_{t-1} = v_l)$ is the probability of observing the state v_k at time t given by the hidden state s_j at time t and the observation state v_l at time $t-1$

As the previous discussion, we would like to extend the Baum-Welch algorithm to the second-order hidden Markov model. Let $\gamma_t(i,j) = \mathbb{P}(X_{t-1} = s_i, X_t = s_j | Y_1^T = y_1^T)$.

Theorem 5 (Extended Baum-Welch Algorithm)**.** $(\hat{x}_{t-1}, \hat{x}_t) = \arg\max_{s_i, s_j \in S_X} \gamma_t(i,j)$ *for* $t = 2, ...T-1$, $\hat{x}_1 = \arg\max_{s_j \in S_X} \sum_{k=1}^n \gamma_2(j,k)$, *and* $\hat{x}_T = \arg\max_{s_j \in S_X} \sum_{i=1}^n \gamma_T(i,j)$.

Since $\hat{x}_t$ will be predicted twice, we let $\hat{x}_t^{(1)} = \arg\max_{s_j \in S_X} \sum_{s_i \in S_X} \gamma_t(i,j)$ and $\hat{x}_t^{(2)} = \arg\max_{s_j \in S_X} \sum_{s_k \in S_X} \gamma_{t+1}(j,k)$; then choose $\hat{x}_t = \arg\max_{\hat{x}_t^{(1)}, \hat{x}_t^{(2)}} \{\sum_{i=1}^n \gamma_t(i, \hat{x}_t^{(1)}), \sum_{k=1}^n \gamma_{t+1}(\hat{x}_t^{(2)}, k)\}$ for $t = 2, 3, ..., T-1$.

With some knowledge of the conditional probability, we can derive that

$$\begin{aligned}
\gamma_t(i,j) =& \frac{\mathbb{P}(X_{t-1} = s_i, X_t = s_j, Y_1^t = y_1^t, Y_{t+1}^T = y_{t+1}^T)}{\mathbb{P}(Y_1^T = y_1^T)} \\
=& \frac{\mathbb{P}(X_{t-1} = s_i, X_t = s_j, Y_1^t = y_1^t)}{\mathbb{P}(Y_1^T = y_1^T)} \\
& \times \mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_{t-1} = s_i, X_t = s_j, Y_1^t = y_1^t) \\
=& C \times \mathbb{P}(X_{t-1} = s_i, X_t = s_j, Y_1^t = y_1^t) \\
& \times \mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_{t-1} = s_i, X_t = s_j, Y_t = y_t),
\end{aligned}$$

where $C = (\mathbb{P}(Y_1^T = y_1^T))^{-1}$ is independent of the choices of s_i and s_j.

Hence, we are curious about the following two quantities.

Definition 6.

- $\alpha_t(i,j) = \mathbb{P}(X_{t-1} = s_i, X_t = s_j, Y_1^t = y_1^t)$: the probability of the *prior* observation y_1^t, the hidden state s_j at time t, and s_i at time $t-1$.
- $\beta_t(i,j) = \mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_{t-1} = s_i, X_t = s_j, Y_t = y_t)$: the probability of the *later* observation y_{t+1}^T given the hidden state s_j at time t

To calculate $\alpha_t(i,j) = \mathbb{P}(X_{t-1} = s_i, X_t = s_j, Y_1^t = y_1^t) =$ the probability of the partial observation y_1^t and state s_i at time $t-1$ and state s_j at time t.
(a) Initiation: $t = 2$,

$$\begin{aligned}
\alpha_2(i,j) =& \mathbb{P}(X_1 = s_i, X_2 = s_j, Y_1^2 = y_1^2) \\
=& \mathbb{P}(Y_2 = y_2 | X_1 = s_i, X_2 = s_j, Y_1 = y_1) \\
& \times \mathbb{P}(X_2 = s_j | X_1 = s_i, Y_1 = y_1) \mathbb{P}(Y_1 = y_1 | X_1 = s_i) \mathbb{P}(X_1 = s_i) \\
=& \mathbb{P}(Y_2 = y_2 | X_2 = s_j, Y_1 = y_1) \mathbb{P}(X_2 = s_j | X_1 = s_i) \\
& \times \mathbb{P}(Y_1 = y_1 | X_1 = s_i) \mathbb{P}(X_1 = s_i) \\
=& b_j(y_2|y_1) a_{ij} b_i(y_1) \pi(i)
\end{aligned}$$

(b) Induction: $t = 3, 4, ..., T$

$$\begin{aligned}
\alpha_t(j,k) =& \mathbb{P}(X_{t-1} = s_j, X_t = s_k, Y_1^t = y_1^t) \\
=& \sum_{s_i \in S_X} \mathbb{P}(X_{t-2} = s_i, X_{t-1} = s_j, X_t = s_k, Y_1^t = y_1^t) \\
=& \sum_{s_i \in S_X} \mathbb{P}(X_{t-2} = s_i, X_{t-1} = s_j, Y_1^{t-1} = y_1^{t-1}) \\
& \times \mathbb{P}(X_t = s_k, Y_t = y_t | X_{t-2} = s_i, X_{t-1} = s_j, Y_1^{t-1} = y_1^{t-1}) \\
=& \sum_{s_i \in S_X} \alpha_{t-1}(i,j) \mathbb{P}(X_t = s_k | X_{t-2} = s_i, X_{t-1} = s_j, Y_1^{t-1} = y_1^{t-1}) \\
& \times \mathbb{P}(Y_t = y_t | X_{t-2} = s_i, X_{t-1} = s_j, X_t = s_k, Y_1^{t-1} = y_1^{t-1}) \\
=& \sum_{s_i \in S_X} \alpha_{t-1}(i,j) a_{ijk} b_k(y_t | y_{t-1})
\end{aligned}$$

(c) Termination:

$$\mathbb{P}(Y_1^T = y_1^T) = \sum_{s_i, s_j \in S_X} \alpha_T(i,j)$$

We conclude the forward algorithm by the following pseudocode.

Theorem 6 (Extended Forward Algorithm)**.**

1. *initial:* $\alpha_2(i,j) = \pi(i) a_{ij} b_i(y_1) b_j(y_2|y_1)$

2. *for* $t = 3{:}T$
   $\alpha_t(j,k) = \sum_{s_i \in S_X} \alpha_{t-1}(i,j) a_{ijk} b_k(y_t|y_{t-1})$

To calculate $\beta_t(i,j) = \mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_{t-1} = s_i, X_t = s_j, Y_1^t = y_1^t) =$ the probability of the partial observation sequence from time $t+1$ to T, given state s_i at time $t-1$, state s_j at time t and observation at time t.
(a) Initiation: $t = T$

$$\beta_T(i,j) = 1$$

(b) Induction: $t = 2, 3, ..., T-1$

$$\begin{aligned}
\beta_t(i,j) =& \mathbb{P}(Y_{t+1}^T = y_{t+1}^T | X_{t-1} = s_i, X_t = s_j, Y_t = y_t) \\
=& \sum_{s_k \in S_X} \mathbb{P}(Y_{t+1}^T = y_{t+1}^T, X_{t+1} = s_k | X_{t-1} = s_i, X_t = s_j, Y_t = y_t)
\end{aligned}$$

$$\begin{aligned}
=& \sum_{s_k \in S_X} \mathbb{P}(Y_{t+2}^T = y_{t+2}^T | X_{t-1} = s_i, X_t = s_j, X_{t+1} = s_k, Y_t^{t+1} = y_t^{t+1}) \\
& \times \mathbb{P}(Y_{t+1} = y_{t+1} | X_{t-1} = s_i, X_t = s_j, X_{t+1} = s_k, Y_t = y_t) \\
& \times \mathbb{P}(X_{t+1} = s_k | X_{t-1} = s_i, X_t = s_j, Y_t = y_t) \\
=& \sum_{s_k \in S_X} \mathbb{P}(Y_{t+2}^T = y_{t+2}^T | X_t = s_j, X_{t+1} = s_k, Y_t^{t+1} = y_t^{t+1}) \\
& \times \mathbb{P}(Y_{t+1} = y_{t+1} | X_{t+1} = s_k, Y_t = y_t) \mathbb{P}(X_{t+1} = s_k | X_{t-1} = s_i, X_t = s_j) \\
=& \sum_{s_k \in S_X} \beta_{t+1}(j,k) b_k(y_{t+1}|y_t) a_{ijk}
\end{aligned}$$

(c) Termination:

$$\begin{aligned}
\mathbb{P}(Y_1^T = y_1^T) &= \sum_{i,j=1}^{N} \mathbb{P}(X_1 = s_i, X_2 = s_j, Y_1^T = y_1^T) \\
&= \sum_{i,j=1}^{N} \mathbb{P}(Y_3^T = y_3^T | X_1 = s_i, X_2 = s_j, Y_1^2 = y_1^2) \\
&\quad \times \mathbb{P}(X_1 = s_i, X_2 = s_j, Y_1^2 = y_1^2) \\
&= \sum_{i,j=1}^{N} \beta_2(i,j) b_j(y_2|y_1) a_{ij} b_j(y_1) \pi(i)
\end{aligned}$$

We conclude the backward algorithm by the following pseudocode.

Theorem 7 (Extended Backward Algorithm)**.**

1. *initial:* $\beta_T(i,j) = 1$

2. for t = T-1:1
$\beta_t(i,j) = \sum_{s_k \in S_X} a_{ijk} b_k(y_{t+1}|y_t)\beta_{t+1}(j,k)$

Similar to the original rescaling, we set

$$C_t = \left(\sum_{i,j=1}^{N} \tilde{\alpha}_t(i,j) \right)^{-1}$$

Proposition 2. $\hat{\alpha}_t(i,j) = \prod_{k=2}^{t} C_k \alpha_t(i.j)$ *for* $i = 1, 2, ..., n,\ t = 1, 2, ..., T$

Proposition 3. $\hat{\beta}_t(i,j) = \prod_{k=t}^{T} C_k \beta_t(i,j)$ *for* $i = 1, 2, ..., n,\ t = 1, 2, ..., T$

Note that $\hat{\alpha}_t(i,j)$ and $\hat{\beta}_t(i,j)$ are not small enough to be machine zero.

Theorem 8. *For these new (scaled) value, the Baum-Welch algorithm holds, that is,*

$$\hat{x}_t = \arg\max_{i,j} \alpha_t(i,j)\beta_t(i,j) = \arg\max_{i,j} \hat{\alpha}_t(i,j)\hat{\beta}_t(i,j).$$

Proof.

$$\begin{aligned} \hat{\alpha}_t(i,j)\hat{\beta}_t(i,j) &= (\prod_{k=2}^{t} C_k)\alpha_t(i.j)(\prod_{k=t}^{T} C_k)\beta_t(i,j) \\ &= C \cdot C_t \cdot \alpha_t(i.j) \cdot \beta_t(i,j), \end{aligned}$$

where $C = \prod_{k=2}^{T} C_k$. □

10 Experiment

Given 39 subjects' sleeping data, including feature and label,

$$\begin{aligned} \text{feature} &\in \mathbb{R}^{42308\times 10} \\ \text{feature 38} &\in \mathbb{R}^{41280\times 10} \\ \text{feature 39} &\in \mathbb{R}^{1028\times 10} \\ \text{label} &\in \{\text{Awake, REM, N1, N2, N3}\}^{42308} \\ \text{label 38} &\in \{\text{Awake, REM, N1, N2, N3}\}^{41280} \\ \text{label 39} &\in \{\text{Awake, REM, N1, N2, N3}\}^{1028} \end{aligned}$$

we apply method 1 and 2 and then compare the results.

- Use the first 38 subjects' data as a training set to run K-means.
- Set K=15.
- Estimate the first 38 subjects' observation state

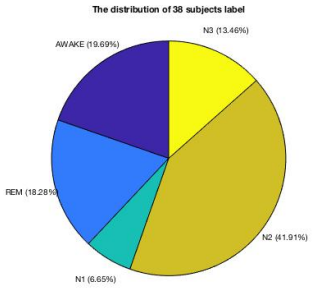


Figure 1: The distribution of 38 subjects' labels

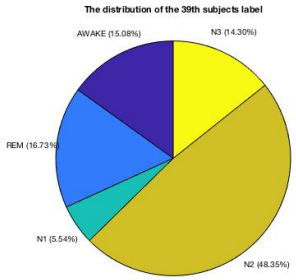


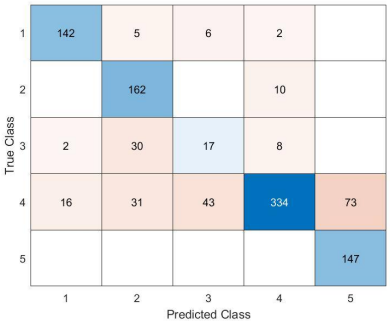
Figure 2: The distribution of the 39th subject's labels

- Use the 15 means to classify the 39^{th} subject's data and get the observation data.
- Use the first 38 subjects' data to compute the transition matrix and emission matrix.
- Apply to the 39^{th} subject's observation data and get the prediction of the 39^{th} subject's hidden state.
- Compute the accuracy .
- Get the first 38 subjects' labels and collect every category.
- Run K-means on each category (K_{new}=3).
- Use the 15 means to classify the 39^{th} subject's data and get the observation data.
- Use the first 38 subjects' data to compute the transition matrix and emission matrix.
- Apply to the 39^{th} subject's observation data and get the prediction of the 39^{th} subject's hidden state.
- Compute the accuracy.

11 Visualization

Accuracy of method 1 = 0.7802
Accuracy of method 1 = 0.8239

Accuracy of method 2 = 0.7879
Accuracy of method 2 = 0.8084



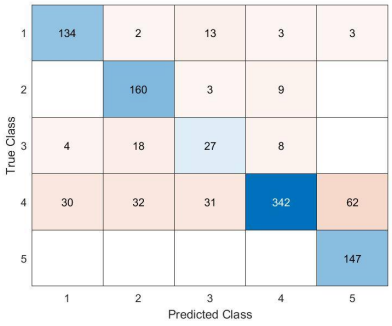
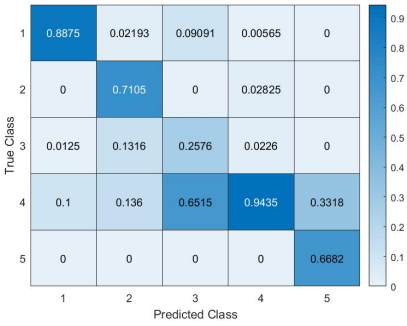


Figure 3: confusion matrix of method 1Figure 4: confusion matrix of method 2



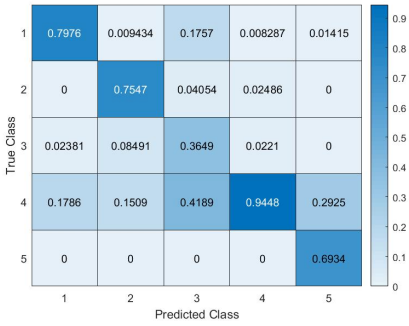
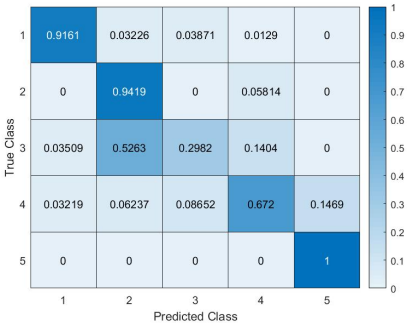


Figure 5: precision of method 1 Figure 6: precision of method 2



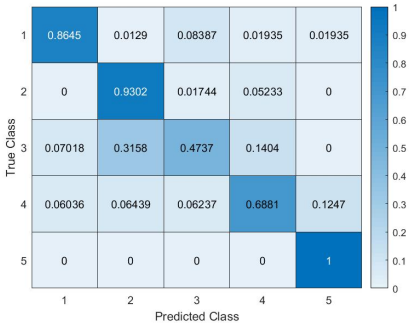


Figure 7: recall of method 1 Figure 8: recall of method 2

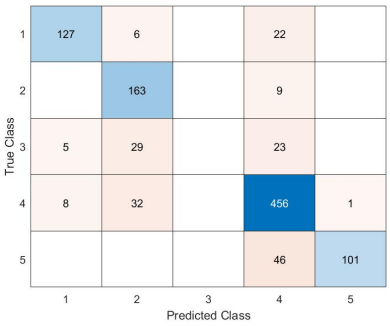


Figure 9: confusion matrix of method 1

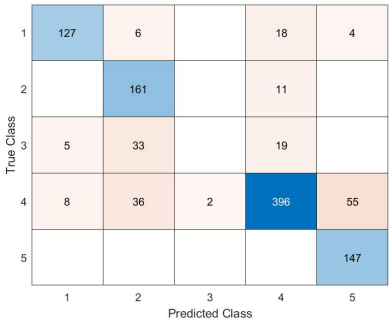


Figure 10: confusion matrix of method 2

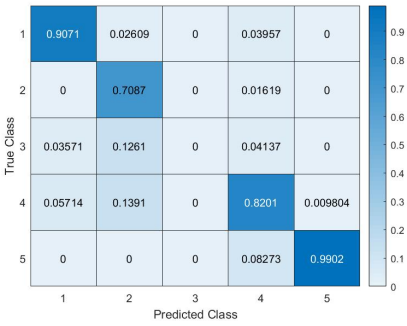


Figure 11: precision of method 1

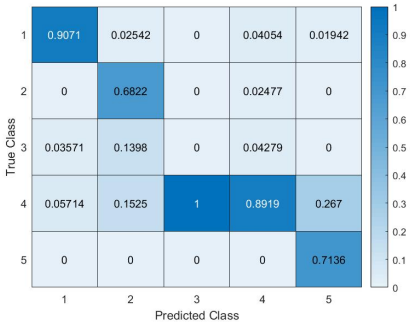


Figure 12: precision of method 2

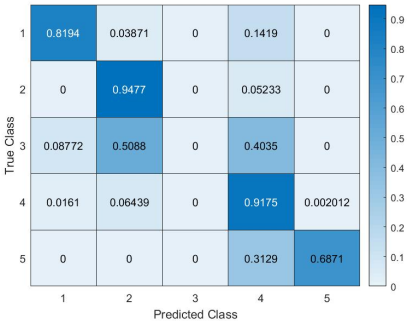


Figure 13: recall of method 1

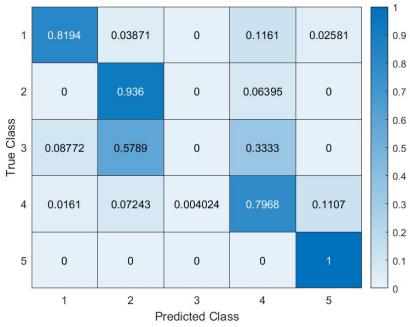


Figure 14: recall of method 2

12 Conclusion

For the first-order HMM

- From accuracy viewpoint, method 2 is better than method 1.
- From precision viewpoint, the performance on REM,N1,N2,N3 of method 2 is better than method 1.
- From recall viewpoint, the performance on N1,N2,N3 of method 2 is better than method 1.
- The performance on the third class of both methods is not as well as our expectation.

For the second-order HMM

- From accuracy viewpoint, method 1 is better than method 2.
- From precision viewpoint, the performance on AWAKE,REM,N3 of method 1 is equal or better than method 2.
- From recall viewpoint, the performance on AWAKE,REM,N2 of method 1 is better than method 2.
- The performance on the third class of both methods is not as well as our expectation.
- The accuracy of both method 1 and 2 are higher than first-order HMM.

13 Improvement

- Since our methods to classification don't take time effect into account, we can't figure out the time relation between each data point after clustering.
- One may use method 3 to take time effect into account and compare the result to the previous.
- Since the limit of the hidden Markov models, one may consider a more complicated model, take more information into account, and compare the result to the previous.