ทฤษฎีพื้นฐานของ Hidden Markov Model

Basic theory for Hidden Markov Models

นาย เฉลิมวุฒิ ไวชนะ
รศ.ดร. สมชาย จิตตะพันธ์กุล
ห้องปฏิบัติการวิจัยโทรคมนาคม

1. บทนำ

ผลที่ได้จากโปรเซสทั่วไปจะมีลักษณะเฉพาะเหมือนสัญญาณต่าง ๆ ซึ่งสัญญาณเหล่านี้จะเป็น discrete, continuous หรือจะเป็นสัญญาณที่ปราศจากการรบกวนต่าง ๆ (pure signal) หรือจะเป็นสัญญาณที่ถูกรบกวนโดยแหล่งกำเนิดอื่น ๆ หรือผลจากการบิดเบือนของการส่ง (transmission distortion) หรือเกิดการสะท้อนกลับ ฯลฯ สัญญาณต่าง ๆ เหล่านี้ก็จะมีลักษณะเฉพาะเป็นของตัวเองเสมอ

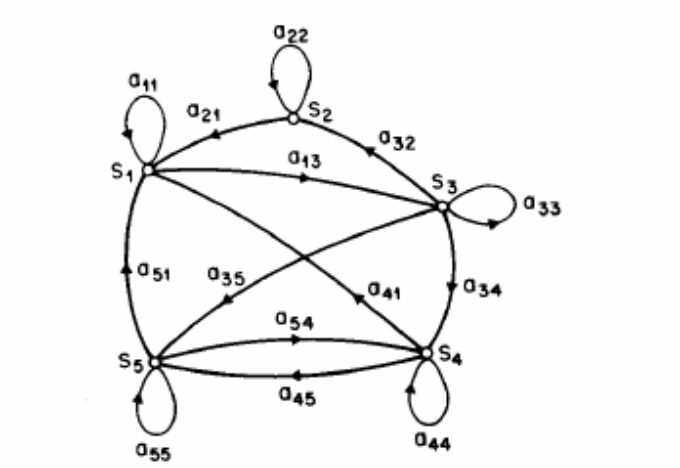
หนึ่งปัญหาที่น่าสนใจก็คือลักษณะเฉพาะของสัญญาณในเทอมของโครงสร้างสัญญาณ (signal model) ซึ่งมีอยู่หลายเหตุผลที่อธิบายว่าทำไมจึงมีคนสนใจในการประยุกต์โครงสร้างสัญญาณ หนึ่งในเหตุผลทั้งหมดก็คือโครงสร้างสัญญาณนั้นสามารถบอกรากฐานสำคัญเพื่อใช้สมมติรูปร่างของระบบ ตัวอย่างเช่น ถ้าหากเราต้องการปรับปรุงสัญญาณเสียงพูดที่ถูกรบกวนจาก noise และการบิดเบือนของการส่ง เราสามารถใช้โครงสร้างสัญญาณในการออกแบบระบบเพื่อกำจัด noise และลบล้างการบิดเบือนของการส่ง เหตุผลข้อที่สองได้อธิบายว่าทำไมโครงสร้างสัญญาณจึงมีความสำคัญ นั่นคือโครงสร้างสัญญาณทำให้เราได้เรียนรู้ถึงแหล่งกำเนิดสัญญาณมากมาย คุณสมบัตินี้มีความสำคัญอย่างมากเพราะต้นทุนของการสร้างสัญญาณจากแหล่งกำเนิดจริงนั้นมีค่าสูง ในกรณีของโครงสร้างสัญญาณที่ดี เราสามารถจำลองแหล่งกำเนิดและเรียนรู้จากมันได้มากเท่าที่จะเป็นไปได้จากการจำลองแหล่งกำเนิด สุดท้ายนี้ เหตุผลสำคัญทั้งหมดที่ว่าทำไมโครงสร้างสัญญาณจึงสำคัญ นั่นคือมันให้ผลการทดลองที่ดีมากและสามารถทำให้เราเข้าใจระบบที่ทดลอง เช่น ระบบการคาดเดา (prediction system) ระบบการรู้จำ (recognition system) ระบบการหาเอกลักษณ์ (identification system) เป็นต้น

เหล่านี้เป็นทางเลือกที่เป็นไปได้หลายทาง ที่นำมาใช้สำหรับเลือกชนิดของโครงสร้างสัญญาณเพื่อใช้หาลักษณะเฉพาะในคุณสมบัติของสัญญาณ เราสามารถแบ่งชนิดของโครงสร้างสัญญาณได้เป็น 2 ชนิดคือ ประเภทของ Deterministic model และประเภทของ Statistical model ประโยชน์ทั่วไปที่ได้จาก Deterministic model คือคุณสมบัติเฉพาะบางอย่างของสัญญาณ เช่น สัญญาณนั้นเป็น Sine wave หรือเป็นผลรวมของ exponential เป็นต้น ในกรณีนี้ เราต้องการทราบรายละเอียดต่าง ๆ ของโครงสร้างสัญญาณ เช่น ค่าแอมพลิจูด (amplitude) ความถี่ เป็นต้น อีกหนึ่งประเภทของโครงสร้างสัญญาณเป็นกลุ่มของ Statistical model ซึ่งหาลักษณะเฉพาะของสัญญาณจากคุณสมบัติของ Statistic ตัวอย่างของ Statistical model เช่น Gaussian process, Poisson process, Markov process และ hidden Markov process เป็นต้น

ในรายงานนี้จะกล่าวถึงพื้นฐานสำคัญของทฤษฎี Hidden Markov Model ปัญหาพื้นฐานของทฤษฎีนี้ วิธีการคำนวณหาค่าพารามิเตอร์ต่าง ๆ การแก้ปัญหาและการปรับปรุงโครงสร้างให้ดีขึ้น รวมถึงความสามารถในการนำไปประยุกต์ใช้

2. Markov Process

พิจารณาระบบที่อธิบายถึงช่วงเวลาหนึ่งของกลุ่มสถานะที่แน่นอนจำนวน N สถานะ คือ S_1 ถึง S_N ที่แสดงให้เห็นดังรูปที่ 1 โดยกำหนดให้ N = 5 และค่า a_{ij} เป็นค่าความน่าจะเป็นในการเปลี่ยนสถานะหนึ่งไปยังอีกสถานะหนึ่ง (โดยที่ i เป็นสถานะต้นและ j เป็นสถานะปลาย)



รูปที่ 1 A Markov chain with 5 states (labeled S_1 to S_5) with selected state transitions

เรากำหนดให้

- t คือลำดับเวลาหนึ่งของการเปลี่ยนสถานะ (discrete time)
- q_t เป็นสถานะปัจจุบัน ณ เวลา t

$$P[q_t = S_j \mid q_{t-1}, q_{t-2} = S_k, ...] = P[q_t = S_j \mid q_{t-1} = S_i] \tag{1}$$

เราพิจารณาสมการที่ (1) นี้ จะเห็นว่าสมการด้านขวามือจะไม่ขึ้นอยู่กับเวลา ดังนั้นเราจะได้ค่าความน่าจะเป็นของ a_{ij} คือ

$$a_{ij} = P[q_t = S_j \mid q_{t-1} = S_i], \quad 1 \le i, j \le N \tag{2}$$

$$a_{ij} \ge 0 \tag{3a}$$

$$\sum_{j=1}^{N} a_{ij} = 1 \tag{3b}$$

พิจารณาตัวอย่าง Markov model ของสภาพอากาศที่มี 3 สถานะคือ ฝนตก เมฆมาก และท้องฟ้าโปร่ง โดยเราจะสมมติสภาพอากาศจะเป็นอย่างไรในหนึ่งวัน กำหนดให้แต่ละสถานะเป็นดังนี้

State 1: rain

State 2: cloudy

State 3: sunny

เราจะกำหนดค่าความน่าจะเป็นของการเปลี่ยนสถานะซึ่งแทนด้วย matrix A

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

สมมติให้สภาพอากาศในวันที่ 1 เป็นท้องฟ้าโปร่ง (state 3) แล้วเราจะถามว่าค่าความน่าจะเป็นที่สภาพอากาศอีก 7 วันจะมีสภาพอากาศเป็น “sun-sun-rain-rain-sun-cloudy-sun” มีค่าเท่าไหร่ เราจะกำหนดให้ลำดับข้อมูลการเปลี่ยนสถานะเหล่านี้แทนด้วย O โดยที่ $O = \{S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3\}$ ซึ่งตรงกับวันที่ t = 1, 2, … , 8 เราจะได้ค่าความน่าจะเป็นที่สภาพอากาศจะเป็นไปตาม O คือ

$$\begin{aligned} P(O \mid Model) &= P[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \mid Model] \\ &= P[S_3] \cdot P[S_3 \mid S_3] \cdot P[S_3 \mid S_3] \cdot P[S_1 \mid S_3] \\ &\quad \cdot P[S_1 \mid S_1] \cdot P[S_3 \mid S_1] \cdot P[S_2 \mid S_3] \cdot P[S_3 \mid S_2] \\ &= \pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{11} \cdot a_{13} \cdot a_{32} \cdot a_{23} \\ &= 1 \cdot (0.8)(0.8)(0.1)(0.4)(0.3)(0.1)(0.2) \\ &= 1.536 \times 10^{-4} \end{aligned}$$

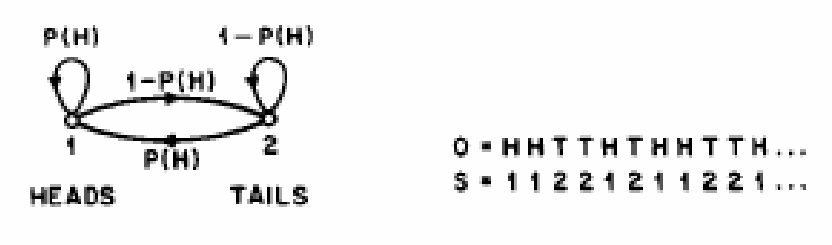
โดยที่เรากำหนดค่า π เป็นค่าความน่าจะเป็นตั้งต้นของสถานะโดยที่

$$\pi_i = P[q_1 = S_i], \qquad 1 \le i \le N \tag{4}$$

3. Hidden Markov Model

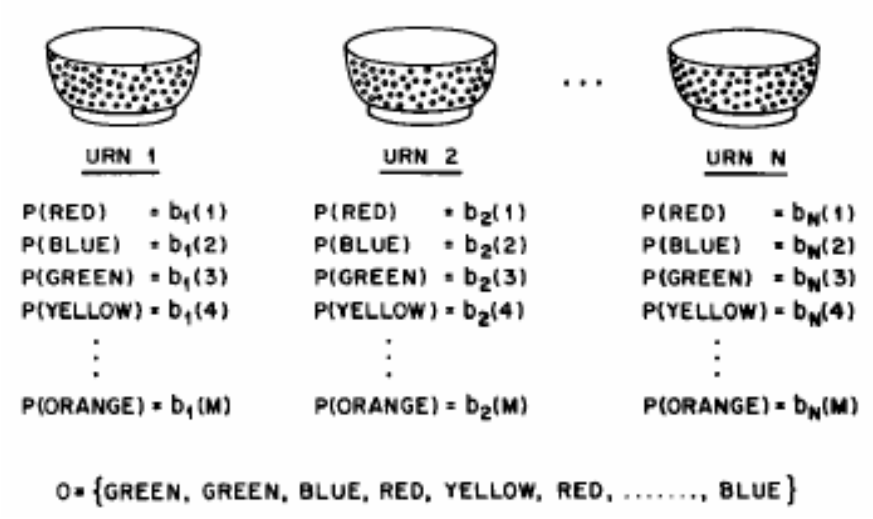
เราได้พิจารณา Markov model ว่ามันสามารถสังเกตเหตุการณ์ที่เกิดขึ้นในการเปลี่ยนแปลงสถานะได้ ซึ่งเหตุผลนี้ที่ทำให้ Markov model มีข้อจำกัดในการนำไปใช้มากเกินไป ในส่วนนี้เราได้เพิ่มเติมแนวความคิดของ Markov model รวมเข้ากับกรณีที่ข้อมูลของสถานะนั้นเป็น probabilistic function ซึ่งเราสามารถเรียกโมเดลนี้ว่า hidden Markov model คุณสมบัติของโมเดลนี้คือ เราไม่สามารถสังเกตเหตุการณ์ที่เกินขึ้นในโปรเซสได้ (hidden) เพื่อความเข้าใจได้ง่ายขึ้น เราจะพิจารณาโมเดลของการทดลองทายลูกเต๋า (High Low) ในการทายลูกเต๋าแต่ละครั้งลูกเต๋าจะถูกเขย่าในถ้วยซึ่งเราไม่สามารถสังเกตเห็นได้ว่าข้างในนั้นเกิดอะไรขึ้นบ้าง สิ่งเดียวที่เราสังเกตได้ก็คือผลลัพธ์ที่ได้เมื่อผู้เขย่าเปิดฝาถ้วยออกแล้วเท่านั้น

ปัญหาที่น่าสนใจในการสร้าง HMM ของการทอยเหรียญ (Coin Toss Model) ขึ้นมาเพื่อบ่งบอกลำดับของผลลัพธ์ (หัวและก้อย) นั้น ปัญหาแรกคือการตัดสินใจว่าสถานะในโมเดลเกี่ยวข้องกับอะไร และการตัดสินใจต่อไปว่าจะต้องมีจำนวนสถานะเท่าไหร่ในโมเดล ในกรณีของการทอยเหรียญ 1 เหรียญ เราได้กำหนดจำนวนสถานะของโมเดลได้ 2 สถานะ คือ หัวและก้อย โมเดลนี้ได้แสดงในรูปที่ 2 ซึ่งโมเดลนี้เป็นแค่ Markov model เท่านั้นเพราะเราสามารถสังเกตเหตุการณ์ในการเปลี่ยนสถานะได้จากหัวไปก้อยหรือก้อยไปหัวได้ แต่ที่น่าสนใจก็คือเมื่อเราใช้ hidden Markov model ในเหตุการณ์นี้ เราจะได้โมเดลที่มี 1 สถานะ (สถานะของเหรียญที่ถูกไบแอส) และมีตัวแปรที่ไม่รู้ค่าก็คือ การไบแอสของเหรียญ



รูปที่ 2 1-coin model

เพื่อเป็นการแสดงแนวความคิดเพิ่มเติมของ HMM เราจะยกเอาโมเดลของลูกบอลในถ้วยขึ้นมาอธิบาย สมมติให้มีจำนวนถ้วยอยู่ N ใบ และในถ้วยแต่ละใบจะมีลูกบอล M สี ซึ่งจำนวนลูกบอลแต่ละสีในแต่ละถ้วยนั้นไม่เท่ากัน ในการทดลองเราจะหยิบลูกบอลขึ้นมาโดยไม่รู้ว่าหยิบมาจากถ้วยใด เพราะฉะนั้นสิ่งที่เราจะรู้เพียงอย่างเดียวคือลูกบอลที่หยิบขึ้นมามีสีอะไร ซึ่งโมเดลนี้ได้แสดงในรูปที่ 3



รูปที่ 3 An N-state urn and ball model which illustrates the general case of a discrete symbol HMM

จะเห็นได้ว่าแต่ละสถานะนั้นไม่ใช่สีของลูกบอลแต่เป็นถ้วยแต่ละใบ เมื่อผลลัพธ์ที่ได้เป็นสีของลูกบอลเราจึงไม่มีทางรู้เลยว่าการเปลี่ยนแปลงสถานะนั้นเป็นไปในรูปแบบใด เพราะเราไม่รู้ว่าหยิบลูกบอลมาจากถ้วยใบใด และไม่รู้เลยว่าข้างในโมเดลเป็นแบบใดด้วย จากโมเดลของลูกบอลในถ้วยได้ให้แนวความคิดว่า HMM คืออะไรและมันสามารถนำไปประยุกต์ใช้ได้อย่างไร

4. ส่วนประกอบของ HMM

ส่วนประกอบต่าง ๆ ที่ HMM จะต้องมีคือ

- N คือจำนวนของสถานะในโมเดล เช่น จำนวนถ้วยที่ใส่ลูกบอล
- M คือจำนวนชนิดของข้อมูล เช่น จำนวนสีของลูกบอล
- ค่าการกระจายความน่าจะเป็นของการเปลี่ยนสถานะ ($A = \{a_{ij}\}$) โดยที่

$$a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i], \quad 1 \le i, j \le N \tag{5}$$

- ค่าการกระจายความน่าจะเป็นของข้อมูลในสถานะ j ($B = \{b_j(k)\}$) โดยที่

$$b_j(k) = P[v_k \text{ at } t \mid q_t = S_j], \quad 1 \le j \le N, \quad 1 \le k \le M \tag{6}$$

- ค่าเริ่มต้นของสถานะ ($\pi = \{\pi_i\}$) โดยที่

$$\pi_i = P[q_1 = S_i], \quad 1 \le i \le N \tag{7}$$

เราสามารถบอกได้ว่าค่าต่าง ๆ เหล่านี้เป็นลักษณะเฉพาะของโมเดล ซึ่งค่า N และ M นั้นเรารู้ได้จากค่าตัวแปร A และ B ดังนั้นเราจึงละไว้ได้ ตัวแปรของโมเดลสามารถแสดงได้ดังนี้

$$\lambda = (A, B, \pi) \tag{8}$$

5. ปัญหาขั้นพื้นฐานใน HMM

ปัญหาที่ 1 : การคำนวณหาค่า $P(O \mid \lambda)$ โดยที่มีลำดับข้อมูลเป็น $O = O_1 O_2 ... O_T$ และมีโมเดลเป็น $\lambda = (A, B, \pi)$

ปัญหาที่ 2 : การเลือกเส้นทางของลำดับสถานะที่ให้ค่าความเป็นไปได้มากที่สุด $(Q = q_1 q_2 ... q_T)$

ปัญหาที่ 3 : การปรับค่าของตัวแปรต่าง ๆ ในโมเดล $\lambda = (A, B, \pi)$ เพื่อให้ค่า $P(O \mid \lambda)$ มากที่สุด

ในปัญหาแรกเป็นปัญหาเกี่ยวกับระบบการคำนวณ เราจะพิจารณาว่าทำไมปัญหานี้จึงเป็นปัญหาพื้นฐานใน HMM เนื่องจากโมเดลมีลักษณะเป็นหลายมิติ จึงมีสมการหลายชั้นซึ่งจะมีผลในการนำไปคำนวณโดยใช้คอมพิวเตอร์

ส่วนปัญหาที่ 2 เป็นปัญหาของส่วนที่เราไม่สามารถสังเกตได้ใน HMM ในการหาลำดับของสถานะทำไมเราจึงต้องหาลำดับของสถานะ เนื่องจากเราไม่มีทางรู้ได้เลยว่าลำดับของสถานะที่เกิดขึ้นเป็นอย่างไร เราจึงได้แต่สมมติขึ้นมาโดยที่ลำดับสถานะนั้นจะให้ค่าความเป็นไปได้มากที่สุด

ส่วนปัญหาที่ 3 เป็นการทำให้โมเดลมีความสมบูรณ์มากขึ้น โดยนำชุดข้อมูลแต่ละชุดผ่านการโปรเซสในโมเดล โมเดลจะทำการปรับค่าตัวแปรต่าง ๆ โดยอัตโนมัติ ทำให้มีผลลัพธ์ที่มีความแม่นยำเพิ่มขึ้น ซึ่งเราเรียกกระบวนการนี้ว่า training ยิ่งถ้ามีข้อมูลมากและเป็นข้อมูลที่มีความถูกต้องสูง ก็จะทำให้โมเดลมีประสิทธิภาพมากขึ้น

6. การแก้ปัญหาพื้นฐานใน HMM

การแก้ปัญหาที่ 1

เราจะทำการคำนวณหาค่าความน่าจะเป็นของลำดับข้อมูล $O = O_1 O_2 ... O_T$ จากโมเดล λ เราจะเขียนแทนค่านี้ด้วย $P(O \mid \lambda)$ โดยเราจะกำหนดให้ลำดับของสถานะคือ

$$Q = q_1 q_2 ... q_T \tag{9}$$

ให้ q_1 เป็นสถานะตั้งต้น และค่าความน่าจะเป็นของลำดับข้อมูล O โดยที่มีลำดับสถานะเป็นดังสมการที่ 9 คือ

$$P(O \mid Q, \lambda) = \prod_{t=1}^{T} P(O_t \mid q_t, \lambda) \tag{10a}$$

เรากำหนดให้ลำดับข้อมูลเป็นอิสระต่อกันเราจะได้

$$P(O \mid Q, \lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \cdot ... \cdot b_{q_T}(O_T) \tag{10b}$$

ค่าความน่าจะเป็นของชุดลำดับสถานะ Q สามารถเขียนได้ดังนี้

$$P(Q \mid \lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} ... a_{q_{T-1} q_T} \tag{11}$$

เพราะฉะนั้นค่า $P(O \mid \lambda)$ จะสามารถเขียนได้ดังนี้

$$P(O \mid \lambda) = \sum_{all Q} P(O \mid Q, \lambda) P(Q \mid \lambda) \tag{12}$$

$$= \sum_{q_1, q_2, ..., q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) ... a_{q_{T-1} q_T} b_{q_T}(O_T) \tag{13}$$

เนื่องจากว่าถ้าเราใช้คอมพิวเตอร์คำนวณหาค่า $P(O \mid \lambda)$ โดยวิธีนี้ คอมพิวเตอร์จะต้องใช้จำนวนคำสั่งถึง $2T \cdot N^T$ ครั้ง ถ้าสมมติกำหนดให้โมเดลมีสถานะ 5 สถานะ $(N = 5)$ และมีชุดลำดับข้อมูล 100 ข้อมูล $(T = 100)$ เพราะฉะนั้นคอมพิวเตอร์จะต้องให้คำสั่งในการคำนวณเท่ากับ $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ ครั้งเลย ทีเดียว เราจึงจำเป็นต้องใช้วิธีอื่น ๆ มาช่วยในการลดจำนวนคำสั่งในการคำนวณหาค่า $P(O \mid \lambda)$ ซึ่งในรายงานนี้ได้เสนอวิธีการลดจำนวนคำสั่งในการคำนวณโดยใช้วิธี Forward Procedure

Forward Procedure [2], [3] จะมีตัวแปรเพิ่มขึ้นมาเพื่อจัดรูประบบสมการใหม่ให้เข้าใจได้ง่ายขึ้นและลดรูปสมการด้วย ซึ่งตัวแปรนี้แทนด้วย

$$\alpha_t(i) = P(O_1 O_2 ... O_t, q_t = S_i \mid \lambda) \tag{14}$$

สำหรับวิธี Forward Procedure นั้นจะประกอบด้วย 3 ส่วนด้วยกัน มีดังนี้

1. Initialization :

$$\alpha_1(i) = \pi_i b_i(O_1), \qquad 1 \le i \le N \tag{15}$$

2. Induction :

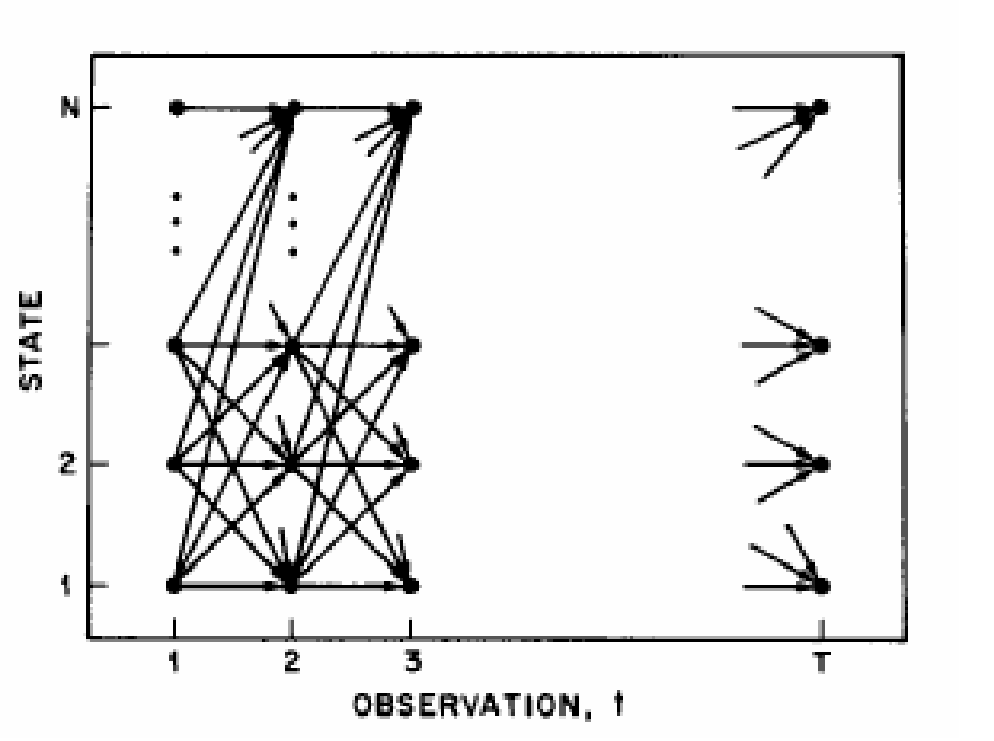
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(j) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1$$

$$1 \le j \le N \tag{16}$$

3. Termination :

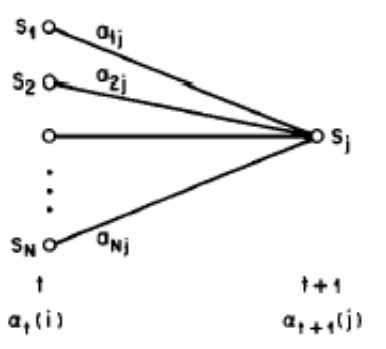
$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{17}$$

ในข้อ 1 เป็นส่วนของค่าเริ่มต้นของข้อมูลตัวแรก O_1 ของทุก ๆ สถานะในโมเดล และในข้อที่ 2 จะเป็นหัวใจสำคัญของ Forward Procedure ซึ่งเป็นวิธีการคำนวณหาค่า $\alpha_t(i)$ ของข้อมูลถัดจาก O_1 ทั้งหมดไปจนถึงตัวสุดท้าย เมื่อการคำนวณดำเนินไปจน $t = T-1$ ก็เป็นอันเสร็จสิ้นของการคำนวณโดยวิธี Forward Procedure โดยในข้อที่ 3 จะเป็นสมการที่เสร็จสมบูรณ์แล้วของค่า $P(O \mid \lambda)$ ซึ่งเท่ากับผลรวมของค่า $\alpha_T(i)$ ของข้อมูลตัวสุดท้ายของทุก ๆ สถานะในโมเดล เราสามารถแสดงลักษณะของการคำนวณโดยวิธี Forward Procedure ด้วยกราฟได้ดังรูปที่ 4



รูปที่ 4 Implementation of the computation of $\alpha_t(i)$ in terms of a lattice of observation t, and states i

และในส่วนย่อย ๆ ของกราฟในรูปที่ 4 มีองค์ประกอบดังรูปที่ 5



รูปที่ 5 Illustration of the sequence of operations required for the computation of the forward variable $\alpha_{t+1}(j)$

ถ้าเราใช้วิธี Forward Procedure ในการคำนวณหาค่า $P(O|\lambda)$ โดยใช้คอมพิวเตอร์ จะมีจำนวนคำสั่งในการคำนวณทั้งหมดเป็น N^2T เท่านั้น ซึ่งถ้าสมมติให้โมเดลมีสถานะทั้งหมด 5 สถานะ $(N=5)$ และมีชุดลำดับข้อมูล 100 ข้อมูล $(T=100)$ เพราะฉะนั้นคอมพิวเตอร์จะต้องให้คำสั่งในการคำนวณประมาณ 3000 คำสั่งเท่านั้นเอง ซึ่งเมื่อเทียบกับวิธีตรงแล้วจะแตกต่างกันมาก

การแก้ปัญหาที่ 2

สำหรับปัญหาในข้อที่ 2 นี้ เป็นการเลือกเส้นทางของการเปลี่ยนแปลงสถานะที่ให้ค่าความเป็นไปได้มากที่สุด ในรายงานนี้ได้เสนอวิธีของ Viterbi Algorithm [21], [22] ซึ่งเราจะกำหนดให้ชุดลำดับของการเปลี่ยนแปลงสถานะที่ดีที่สุดเป็น $Q=\{q_1q_2...q_T\}$ และให้ชุดลำดับของข้อมูลเป็น $O=\{O_1O_2...O_T\}$ และเราต้องการหาค่า

$$\delta_t(i)=\max_{q_1,q_2,...,q_{t-1}} P[q_1q_2...q_t=i,O_1O_2...O_t\mid\lambda] \tag{18}$$

ซึ่งค่า $\delta_t(i)$ เป็นค่าที่ดีที่สุดของหนึ่งเส้นทางในลำดับที่ t และ $\psi_t(j)$ คือเส้นทางที่ให้ค่าที่ดีที่สุด สำหรับวิธีของ Viterbi Algorithm จะมีส่วนประกอบที่สำคัญอยู่ 4 ส่วนคือ

1. Initialization :

$$\delta_1(i)=\pi_i b_i(O_1), \qquad 1\le i\le N \tag{19a}$$

$$\psi_1(i)=0 \tag{19b}$$

2. Recursion :

$$\delta_t(j)=\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}]b_j(O_t), \qquad 2\le t\le T, \quad 1\le j\le N \tag{20a}$$

$$\psi_t(j)=\arg\max_{1\le i\le N}[\delta_{t-1}(i)a_{ij}], \qquad 2\le t\le T, \quad 1\le j\le N \tag{20b}$$

3. Termination :

$$P^* = \max_{1\le i\le N}[\delta_T(i)] \tag{21a}$$

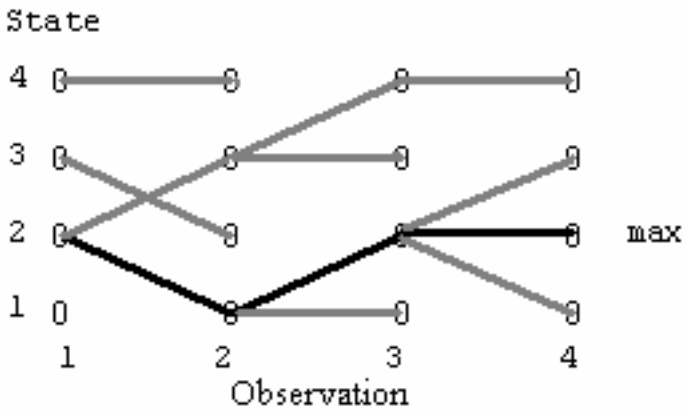
$$q_T^* = \arg\max_{1\le i\le N}[\delta_T(i)] \tag{21b}$$

4. Path (state sequence) backtracking :

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, ..., 1 \tag{22}$$

ในส่วนที่ 1 จะเป็นค่าเริ่มต้นของข้อมูลตัวแรกของทุก ๆ สถานะในโมเดล ส่วนเส้นทางเดินนั้นยังไม่มี จึงเป็นศูนย์ สำหรับส่วนที่ 2 ก็เป็นการคำนวณหาค่า $\delta_t(i)$ และ $\psi_t(i)$ ในลำดับต่าง ๆ ถัดจากข้อมูลตัวแรกไปจนถึงข้อมูลตัวสุดท้าย ในส่วนที่ 3 ก็จะเป็นส่วนที่สุดท้ายของข้อมูล ค่าที่เราจะเลือกเส้นทางของการเปลี่ยนแปลงสถานะจะเป็นค่า $\delta_T(i)$ ตัวสุดท้ายที่มีค่ามากที่สุด และจะเลือกเส้นทางเริ่มต้นจากสถานะนั้นด้วย จากนั้นในส่วนที่ 4 ก็จะเป็นการเลือกเส้นทางเดินทั้งหมดของชุดลำดับข้อมูลซึ่งจะทำการเลือกย้อนกลับไปและจะมีเพียงเส้นทางเดียวที่ให้เลือกอยู่แล้ว

สมมติให้โมเดลมีสถานะ 4 สถานะ และจำนวนข้อมูลมี 4 ตัว ดังรูปที่ 6 จะเห็นว่าเส้นทางที่มีสีเข้มเป็นเส้นทางที่ให้ค่าความเป็นไปได้มากที่สุด เราจะเลือกเส้นทางนี้เป็นเส้นทางการเปลี่ยนแปลงสถานะของโมเดลนี้



รูปที่ 6 Viterbi Algorithm

การแก้ปัญหาที่ 3

อันที่จริงแล้วการปรับปรุงค่าตัวแปรต่าง ๆ ในโมเดลนั้นเราไม่สามารถทำได้เลย ถ้าเราไม่รู้เส้นทางการเปลี่ยนแปลงของสถานะ ซึ่งเส้นทางการเปลี่ยนแปลงของสถานะที่เราหาได้จากปัญหาที่ 2 ก็เป็นเพียงเส้นทางที่เราคิดว่ามันมีความเป็นไปได้มากที่สุดเท่านั้นเอง แต่ถ้าเรามีจำนวนชุดข้อมูลที่นำมาเทรนนิ่งมากพอ เราก็สามารถปรับค่าตัวแปรต่าง ๆ ในโมเดลให้ลู่เข้าสู่ค่าที่ดีที่สุดได้ และจะทำให้ค่า $P(O|\lambda)$ มากที่สุดด้วย

สำหรับวิธีการปรับค่าตัวแปรต่าง ๆ ในโมเดลนี้ เรากำหนดให้มีจำนวนชุดข้อมูล R ชุด $(O_1O_2...O_R)$ มาทำการโปรเซสในโมเดลและหาเส้นทางการเปลี่ยนแปลงของสถานะ จากนั้นเราจะทำการปรับค่าตัวแปรต่าง ๆ คือค่า $\pi, a_{ij}, b_j(k)$ ดังนี้

$$\overline{\pi} = \frac{\text{จำนวนครั้งที่อยู่ในสถานะ } S_i \text{ ณ ข้อมูลตัวแรกของแต่ละชุด}}{\text{จำนวนชุดข้อมูลทั้งหมด}}$$

$$\overline{a_{ij}} = \frac{\text{จำนวนครั้งที่มีการเปลี่ยนแปลงสถานะจาก } S_i \text{ ไปยัง } S_j}{\text{จำนวนครั้งที่มีการเปลี่ยนแปลงสถานะจาก } S_i \text{ ทั้งหมด}}$$

$$\overline{b_j}(k) = \frac{\text{จำนวนครั้งที่อยู่ในสถานะ j และข้อมูลเป็น } v_k}{\text{จำนวนครั้งที่อยู่ในสถานะ j ทั้งหมด}}$$

จากนี้เราก็จะได้โมเดลที่มีค่าตัวแปรเหล่านี้ใหม่ ถ้าเราทำการเทรนนิ่งไปเรื่อย ๆ โดยใช้โมเดลที่ทำการปรับค่าตัวแปรเรียบร้อยแล้ว ค่าตัวแปรเหล่านี้จะถูกปรับปรุงจนลู่เข้าค่าใดค่าหนึ่ง ซึ่งค่านั้นเองจะเป็นค่าที่ทำให้โมเดลนี้มีความสมบูรณ์มากที่สุด และจะทำให้ค่า $P(O \mid \lambda)$ มากที่สุดด้วย

7. สรุป

จากเนื้อหาที่กล่าวมาทั้งหมดได้แสดงให้เห็นถึงองค์ประกอบต่าง ๆ ใน hidden Markov model และการคำนวณหาค่าองค์ประกอบเหล่านั้น อันที่จริงแล้วชุดลำดับข้อมูลที่กล่าวในรายงานนี้เป็นชุดข้อมูลที่มีลักษณะเป็น discrete ทั้งหมด สำหรับชุดข้อมูลที่มีลักษณะเป็น continuous ก็จะมีลักษณะเดียวกัน เพียงแค่ค่าความน่าจะเป็นของข้อมูลในสถานะ $b_i(k)$ จะมีลักษณะเป็น statistical function ค่าความน่าจะเป็นหาได้จากการกระจายของข้อมูลในแบบต่าง ๆ ซึ่งแล้วแต่ว่าข้อมูลนั้นจะเป็นข้อมูลที่มีการกระจายลักษณะใด หลังจากนั้นก็ใช้ความรู้ทางคณิตศาสตร์มาแก้ปัญหาของ function สำหรับหลักการของ hidden Markov model ก็ยังเหมือนเดิม

เมื่อเราได้ทำความเข้าใจถึงทฤษฎี hidden Markov model อย่างแท้จริงจะพบว่าสัญญาณแต่ละสัญญาณจะมีลักษณะเฉพาะไม่เหมือนกัน เพราะฉะนั้นหนึ่งลักษณะเฉพาะของสัญญาณเราจะสามารถสร้างโมเดลได้หนึ่งโมเดล เช่น สัญญาณเสียงของคำว่า “ฉัน” ก็จะเป็นหนึ่งโมเดล “ไป” ก็จะเป็นอีกหนึ่งโมเดล และถ้าเราใช้แค่เสียงคน ๆ เดียวมาทำการเทรนนิ่ง โมเดลจะรู้จำได้ถึงลักษณะเฉพาะของเสียงคน ๆ นั้นเท่านั้น ถ้านำเสียงของคนหลาย ๆ คนมาทำการเทรนนิ่งคำว่า “ฉัน” โมเดลก็จะรู้จำได้ถึงลักษณะเสียงของคำว่า “ฉัน” มากขึ้น ทำให้มีโอกาสที่คนทั่วไปสามารถใช้โมเดลนี้ได้ เราเรียกว่า Independent Speech Recognition

การนำทฤษฎี hidden Markov model ไปประยุกต์ใช้นั้น จะแบ่งเป็น 3 ลักษณะคือ ระบบการคาดเดา (prediction system) ระบบการรู้จำของสัญญาณ (recognition system) และระบบการหาเอกลักษณ์ (identification system) ซึ่งส่วนใหญ่นิยมนำไปใช้ในการรู้จำของสัญญาณ ตัวอย่างเช่น Speech Recognition, Image Processing หรือการ Identify บุคคลไม่ว่าจะเป็นเสียง ภาพหรือรูปลักษณ์ต่าง ๆ

8. เอกสารอ้างอิง

[1] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state Markov chains,” Ann. Math. Stat., vol. 37, pp. 1554-1563, 1966.

[2] L. E. Baum and J. A. Egon, “An inequality with applications to statistical estimation for probabilistic functions of a Markov process and to a model for ecology,” Bull. Amer. Meteorol. Soc., vol. 73, pp. 360-363, 1967.

[3] L. E. Baum and G. R. Sell, “Growth functions for transformations on manifolds,” Pac. J. Math., vol. 27, no. 2, pp. 211-227, 1968.

[4] L. E. Baum, T. Perie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains,” Ann. Math. Stat., vol. 41, no. 1, pp. 164-171, 1970.

[5] L. E. Baum, “An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes,” Inequalities, vol. 3, pp. 1-8, 1972.

[6] J. K. Baker, “The dragon system--An overview,” IEEE Trans. Acoust. Speech Signal Processing, vol. ASSP-23, no. 1, pp. 24-29, Feb. 1975.

[7] F. Jelinek, “A fast sequential decoding algorithm using a stack,” IBM J. Res. Develop., vol. 13, pp. 675-685, 1969.

[8] L. R. Bahl and F. Jelinek, “Decoking for channels with insertions, deletions, and substitutions with applications to speech recognition,” IEEE Trans. Informat. Theory, vol. IT-21, pp. 404-411, 1975.

[9] F. Jelinek, L. R. Bahl, and R. L. Mercer, “Design of a linguistic statistical decoder for the recognition of continuous speech,” IEEE Trans. Informat. Theory, vol. IT-21, pp. 250-256, 1975.

[10] F. Jelinek, “Continuous speech recognition by statistical methods,” Proc. IEEE, vol. 64, pp. 532-536, Apr. 1976.

[11] R. Bakis, “Continuous speech word recognition via centi-second acoustic states,” in Proc. ASA Meeting (Washington, DC), Apr. 1976.

[12] F. Jelinek, L. R. Bahl, and R. L. Mercer, “Continuous speech recognition: Statistical methods,” in Handbook of Statistics, II, P.R. Krishnaiad, Ed. Amsterdam, The Netherlands: North-Holland, 1982 .

[13] L. R. Bahl, F. Jelinek, and R. L. Mercer, “A maximum likelihood approach to continuous speech recognition,” IEEE Trans. Pattern Anal. Machine Intell., vol. PAMI-5, pp. 179-190, 1983.

[14] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, “An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition,” Bell Syst. Tech. J., vol. 62, no. 4, pp. 1035-1074, Apr. 1983.

[15] B. H. Juang, “On the hidden Markov model and dynamic time warping for speech recognition—A unified view,” AT&T Tech. J., vol. 63, no. 7, pp. 1213-1243, Sept. 1984.

[16] L.R. Rabiner and B. H. Juang, “An introduction to hidden Markov models,” IEEE ASSP Mag., vol. 3, no. 1, pp. 4-16, 1986.

[17] J. S. Bridle, “Stochastic models and template matching: Some important relationships between two apparently different techniques for automatic speech recognition,” in Proc. Inst. Of Acoustics, Autum Conf., pp. 1-8, Nov. 1984.

[18] J. Makhoul, S. Roucos, and H. Gish, “Vector quantization in speech coding,” Proc. IEEE, vol. 73, no. 11, pp. 1551-1588, Nov. 1985.

[19] S. E. Levinson, “Structural methods in automatic speech recognition,” Proc. IEEE, vol. 73, no. 11, pp. 1625-1650, Nov. 1985.

[20] A. W. Drake, “Discrete--state Markov processes,” Chapter 5 in Fundamentals of Applied Probability Theory. New York, NY: McGraw-Hill, 1967.

[21] A. J. Viterbi, “Error bounds for convolutional codes and an asymptotically optimal decoding algorithm,” IEEE Trans. Informat. Theory, vol. IT-13, pp. 260-269, Apr. 1967.

[22] G. D. Forney, “The Viterbi algorithm,” Proc. IEEE, vol. 61, pp. 268-278, Mar. 1973.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” J. Roy. Stat. Soc., vol. 39, no. 1, pp. 1-38, 1977.

[24] Lawrence R. Rabiner and B. Juang. An introduction to hidden Markov models. IEEE ASSP Magazine. 1986.

9. กิตติกรรมประกาศ

บทความนี้เป็นส่วนหนึ่งของรายวิชา 2102790 ELEC ENG SEMINAR ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ภาคการศึกษาปลาย ปีการศึกษา 2545