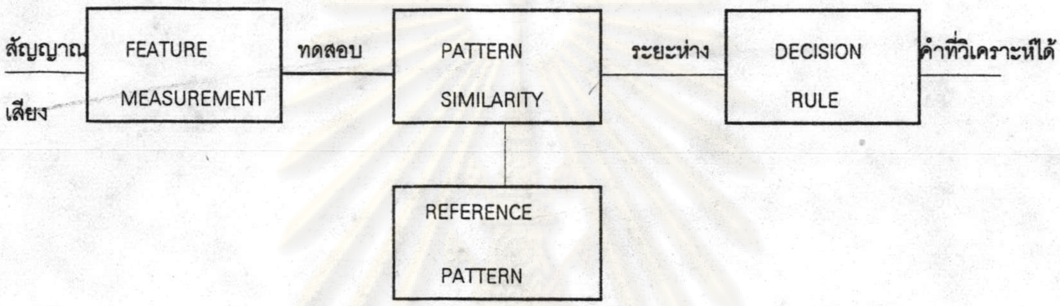
บทที่ 2

รูปแบบการรู้จำเสียง

ในรูป 2.1 แสดงโมเดลการรู้จำรูปแบบที่สำคัญของระบบการรู้จำเสียงแบบคำโดด (Rabiner and Levinson , 1981) แบ่งเป็น 3 ส่วน



รูป 2.1 โมเดลการรู้จำรูปแบบ

1. Feature measurement
2. Pattern similarity determination
3. Decision rule

โดยที่สัญญาณเข้าโมเดลเป็นคลื่นเสียง ส่วนสัญญาณออกเป็นเสียงวิเคราะห์ ที่ได้จากสัญญาณเข้า โมเดลดังกล่าวมีการใช้กันอย่างแพร่หลาย เนื่องจากเหตุผลหลายประการดังนี้

1. สามารถนำไปใช้ได้กับเสียงที่มีความแตกต่างกันของคำ, ผู้ใช้, feature รวมถึงวิธีที่ใช้ในการวิเคราะห์ และตัดสินใจ
2. ง่ายในการนำไปใช้และปรับปรุง
3. ใช้งานได้ดีในการทดสอบ

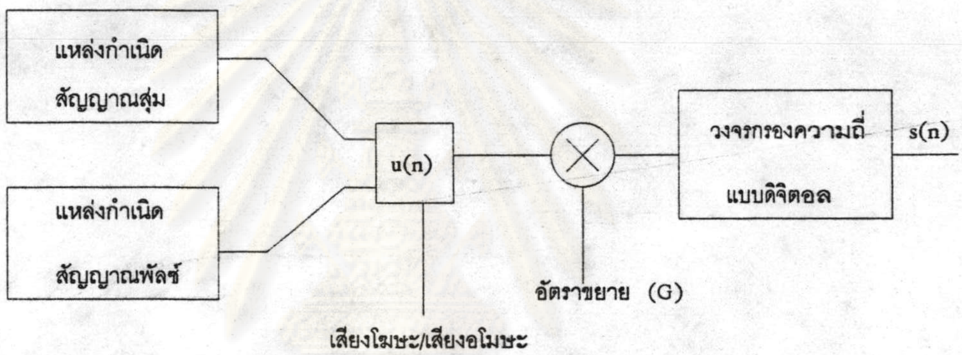
FEATURE MEASUREMENT

เป็นเทคนิคในการแปลงข้อมูลที่มีอยู่มากมายเป็นส่วนเล็ก ๆ ซึ่งส่วนเล็ก ๆ ดังกล่าวจะแสดงคุณสมบัติของคลื่นเสียงนั้น ๆ โดยวิธีที่ใช้กันอยู่หลายวิธี เช่น Filter Bank-Analysis of Speech ซึ่งใช้วิธีการป้อนสัญญาณผ่าน filter และการประมาณพันธะเชิงเส้น (Linear Predictive Coding)

1 การประมาณพันธะเชิงเส้น (Linear Predictive Coding)

เป็นขบวนการทางคณิตศาสตร์ที่ใช้ในการหาเอกลักษณ์ของ ระบบ โดยพิจารณาว่าเสียงเกิดจากผลรวมเชิงเส้น (linear combination) ของสัญญาณที่ทราบค่าแล้ว โดยใช้วิธี กำลังสองน้อยที่สุด (method of least square) ในการเลือกค่าพารามิเตอร์ของระบบ หลักการประมาณพันธะเชิงเส้นมีวิธีการใหญ่ 2 วิธี คือ วิธีการหาค่าความแปรปรวนร่วม และวิธีอัตสัมพันธ์ วิธี Linear Predictive coding สามารถแสดงคุณสมบัติได้ ใกล้เคียงกับพื้นฐานโมเดลการกำเนิดเสียงของมนุษย์ (Rabiner and Levinson, 1981) ในวิทยานิพนธ์นี้นำการประมาณพันธะเชิงเส้น โดยวิธีอัตสัมพันธ์มาหา feature ของเสียง

1.1 หลักการประมาณพันธะเชิงเส้น รูป2.2 เป็นแบบจำลองเสียงการประมวลสัญญาณเสียงพูดในการวิเคราะห์มักจะแบ่งระบบกำเนิด เสียงพูดเป็น 2 ส่วน คือ แหล่งกำเนิดสัญญาณกระตุ้น และท่อกำธรเสียง (สุเธียร เกียรติสุนทร, 2525)



รูป 2.2 แสดงแบบจำลองของระบบกำเนิดเสียงพูดแบบดิจิตอล

สามารถเขียนความสัมพันธ์ได้ดังนี้ (Sadaoki Furni, 1989)

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1-\sum_{i=1}^{m} a_i a^{-i}} \qquad (2.1)$$

สามารถแสดงความสัมพันธ์ของสัญญาณค่า X_t โดยพิจารณาจากสัญญาณก่อนหน้าจำนวน p ค่า

$$x_t + \alpha_1 x_{t-1} + \ldots.. + \alpha_p x_{t-p} = \varepsilon_t \qquad (2.2)$$

โดย $\{\varepsilon_t\}$ มีค่ากลางเป็นศูนย์ ค่า varience เป็น σ^2

สามารถคาดคะเนสัญญาณค่าถัดไปเมื่อทราบค่าก่อนหน้า p ค่า

$$\hat{x}_t = -\sum_{i=1}^{p} \alpha_i x_{t-i} \tag{2.3}$$

จากสมการ (2.2), (2.3)

$$x_t - \hat{x}_t = \varepsilon_t \tag{2.4}$$

ถ้าสมมุติ

$$F(z) = -\sum_{i=1}^{p} \alpha_i z^{-i} \tag{2.5}$$

และ $\hat{X}(z) \leftrightarrow \hat{x}_t, X(z) \leftrightarrow x_t$ Z-Transform ของสมการ (2.3) สามารถแสดงได้

$$\hat{X}(z) = F(z) X(z) \tag{2.6}$$

ทำการพิจารณาสัมประสิทธิ์ $\{\alpha_i\}$ ที่ทำให้ความคลาดเคลื่อนในการกะประมาณค่ามีค่าต่ำสุด เมื่อพิจารณาในช่วง $[t_0,t_1]$ ค่าคลาดเคลื่อนกำลังสองรวมคือ β

$$\beta = \sum_{t=t_0}^{t_1} \varepsilon_t^2$$

$$= \sum_{t=t_0}^{t_1} \left(\sum_{i=0}^{p} \alpha_i x_{t-i} \right)^2$$

$$= \sum_{t=t_0}^{t_1} \sum_{i=0}^{p} \sum_{j=0}^{p} \alpha_i \alpha_j x_{t-i} x_{t-j} \tag{2.7}$$

โดยที่ $\alpha_0 = 1$ และสมมุติ

$$c_{ij} = \sum_{t=t_0}^{t_1} x_{t-i} x_{t-j} \tag{2.8}$$

ค่า β สามารถเขียนอีกอย่างได้ว่า

$$\beta = \sum_{i=0}^{p} \sum_{j=0}^{p} \alpha_i c_{ij} \alpha_j \tag{2.9}$$

ทำการหาค่าต่ำสุดของ β โดยการหาค่าอนุพันธ์ส่วนย่อยของค่า β และให้เท่ากับศูนย์

$$\frac{\partial \beta}{\partial \alpha_j} = 2\sum_{i=0}^{p} \alpha_i c_{ij} = 0 \qquad (j = 1, 2, \ldots, p) \tag{2.10}$$

ค่าสัมประสิทธิ์ $\{\alpha_i\}$ สามารถหาได้ถ้าเราทราบค่า c_{ij} ($i = 0, 1, 2, \ldots, p$; $j = 1, 2, \ldots, p$) จากสมการ (2.8) ได้จากการพิจารณาค่า X_t จาก $t_0 - p$ จนถึง t_1 ในการหาคำตอบ

ในการหาคำตอบของ sequence N ที่ $\{X_t\} = \{X_0, X_1, \ldots, X_{N-1}\}$ มี 2 กรณีที่ต้องพิจารณาคือ วิธีการแปรปรวนร่วม และวิธีการอัตสัมพันธ์

วิธีการอัตสัมพันธ์จะทำการพิจารณา $t_0 = -\infty$ และ $t_1 = \infty$ โดยการให้ $X_t = 0$ เมื่อ $t < 0$ และ $t \geq N$ (Markel, 1972)

$$\begin{aligned} c_{ij} &= \sum_{t=-\infty}^{\infty} x_{t-i} x_{t-j} \\ &= \sum_{t=-\infty}^{\infty} x_t x_{t+|i-j|} \\ &= \sum_{t=0}^{N-1-|i-j|} x_t x_{t+|i-j|} \\ &= r_{|i-j|} \end{aligned} \tag{2.11}$$

ค่า α_i จะสามารถหาได้จาก

$$\sum_{i=0}^{p} \alpha_i r_{|i-j|} = 0 \qquad (j = 1, 2, \ldots, p) \tag{2.12}$$

$$r_\tau = \sum_{t=0}^{N-1-\tau} x_t x_{t+\tau}, (\tau \geq 0) \tag{2.13}$$

จากสมการ (2.13) เราสามารถเขียนอยู่ในรูปของเมตริกซ์ได้คือ

$$\begin{bmatrix} r_0 & r_1 \ldots\ldots\ldots\ldots r_{p-1} \\ r_1 & r_0 \ldots\ldots\ldots\ldots \\ . & \\ . & \\ . & \\ r_{p-1} & \ldots\ldots\ldots\ldots\ldots r_0 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ . \\ \alpha_p \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ . \\ . \\ . \\ r_p \end{bmatrix} \tag{2.14}$$

1.2 วิธีอัตสัมพันธ์ส่วนย่อย วิธีอัตสัมพันธ์ส่วนย่อยหรือวิธี PARCOR (Partial Autocorrelation) เป็นวิธีการหาค่าพารามิเตอร์ ในการประมาณค่าสัญญาณ ด้วยวิธีประมาณพันธะเชิงเส้นวิธีหนึ่ง โดยใช้คุณสมบัติของผลคูณภายใน (inner product) และความสัมพันธ์เชิงพิกัดฉาก (orthogonal relationship) ของฟังก์ชันพหุคูณ (polynomial function) ในการเลือกค่าพารามิเตอร์ที่ทำให้เกิดความคลาดเคลื่อนต่ำสุด

กำหนดให้ $x_m^+(n)$ เป็นความคลาดเคลื่อนในการประมาณค่าแบบก้าวหน้า (forward prediction) ของสัญญาณ { $x(n) \mid n_0 \le n \le n_1$ } โดยใช้สัมประสิทธิ์ในการกะประมาณค่า m ตัว {$a_{mi} \mid 1 \le i \le m$ }

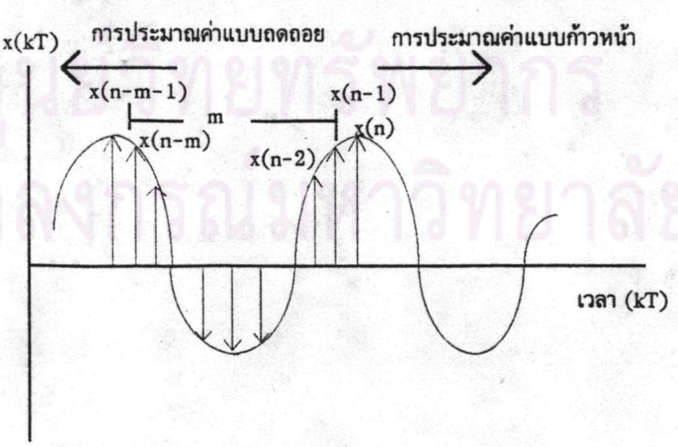
$$x_m^+(n) = x(n) - \left\{-\sum_{i=1}^{m} a_{mi}\, x(n-i)\right\} \qquad (2.15)$$

$$= \sum_{i=0}^{m} a_{mi}\, x(n-i) \qquad \text{เมื่อ} \quad a_{m0} = 1 \qquad (2.16)$$

ในทำนองเดียวกันกับความคลาดเคลื่อนในการประมาณค่าแบบถดถอย (backward prediction) $x_m^-(n)$ ของสัญญาณ { $x(n-m-1) \mid n_0 \le n \le n_1$ } โดยใช้สัมประสิทธิ์ในการกะประมาณค่า m ตัว {$b_{mi} \mid 1 \le i \le m$}

$$x_m^-(n) = x(n-m-1) - \left\{-\sum_{i=1}^{m} b_{mi}\, x(n-i)\right\} \qquad (2.17)$$

$$= \sum_{i=1}^{m+1} b_{mi}\, x(n-i) \qquad \text{เมื่อ} \quad b_{m\ m+1} = 1 \qquad (2.18)$$



รูป 2.3 การประมาณพันธะเชิงเส้น

ค่าความคลาดเคลื่อนกำลังสองรวม (total square error) แบบก้าวหน้า α_m และแบบถดถอย β_m เมื่อ $m = 1, 2, \ldots, M$ ในช่วง $[\, n_0 , n_1 \,]$ มีค่า

$$\alpha_m = \sum_{n=n_0}^{n_1} \left\{ x_m^{+}(n) \right\}^2 \tag{2.19}$$

$$\beta_m = \sum_{n=n_0}^{n_1} \left\{ x_m^{-}(n) \right\}^2 \tag{2.20}$$

ขั้นตอนต่อไปในการประมาณพันธะเชิงเส้น คือการเลือกค่าสัมประสิทธิ์ในการประมาณค่า $\{ a_{mi} \}$ และ $\{ b_{mi} \}$ ที่ทำให้ค่า α_m และ β_m มีค่าต่ำสุด

1.3 สัมประสิทธิ์ PARCOR การเลือกค่าพารามิเตอร์ของระบบกำเนิดเสียงโดยวิธี PARCOR คือการหาค่าของสัมประสิทธิ์ $\{ k_m | 1 \le m \le M \}$ ที่ทำให้สามารถแสดงฟังก์ชันพหุคูณ $A_m(z)$ ในรูปผลรวมเชิงเส้นของ $A_{m-1}(z)$ และ $B_{m-1}(z)$ โดยเริ่มจาก

$$A_0(z) = 1 \tag{2.21}$$

และ

$$B_0(z) = z^{-1} \tag{2.22}$$

และสำหรับค่า m ใด ๆ ที่มากกว่าศูนย์

$$A_m(z) = A_{m-1}(z) + k_m B_{m-1}(z) \tag{2.23}$$

ผลคูณภายใน

$$\left\langle A_m(z), z^{-m} \right\rangle = \left\langle A_{m-1}(z), z^{-m} \right\rangle + k_m \left\langle B_{m-1}(z), z^{-m} \right\rangle \tag{2.24}$$

จากคุณสมบัติเชิงพิกัดฉากของฟังก์ชันพหุคูณ

$$\left\langle A_m(z), z^{-m} \right\rangle = 0 \tag{2.25}$$

$$\begin{aligned} \left\langle A_{m-1}(z), z^{-m} \right\rangle &= \left\langle A_{m-1}(z), B_{m-1}(z) \right\rangle \\ &= \left\langle 1, B_{m-1}(z) \right\rangle \end{aligned} \tag{2.26}$$

และ

$$\begin{aligned} \left\langle B_{m-1}(z), z^{-m} \right\rangle &= \left\langle B_{m-1}(z), B_{m-1}(z) \right\rangle \\ &= \left\| B_{m-1}(z) \right\|^2 \\ &= \beta_{m-1} \end{aligned} \tag{2.27}$$

แทนค่าในสมการ (2.24)

$$k_m = \frac{-1}{\beta_{m-1}}\left\langle A_{m-1}(z), B_{m-1}(z)\right\rangle$$

$$= \frac{-1}{\beta_{m-1}}\left\{\sum_{n=n_0}^{n_1}\sum_{i=0}^{m-1} a_{mi}\, x(n-i)\right\}$$

$$= \left\{\sum_{n=n_0}^{n_1}\sum_{j=0}^{m} b_{mj}\, x(n-j)\right\}$$

$$= \frac{-1}{\beta_{m-1}}\left\{\sum_{n=n_0}^{n_1} x_m^{+}(n)\right\}\left\{\sum_{n=n_0}^{n_1} x_m^{-}(n)\right\} \qquad (2.28)$$

จาก

$$A_m(z) = \sum_{i=0}^{m} a_{mi}\, z^{-i} \qquad (2.29)$$

$$z^{-(m+1)} A_m\left(z^{-1}\right) = z^{-(m+1)} \sum_{i=0}^{m} a_{mi}\, z^{i}$$

$$= \sum_{i=0}^{m} a_{mi}\, z^{i-m-1} \qquad (2.30)$$

ให้ j = -(i-m-1)

$$z^{-(m+1)} A_m\left(z^{-1}\right) = \sum_{j=1}^{m+1} a_{m\ \ m+1-j}\, z^{-j} \qquad (2.31)$$

เมื่อ

$$b_{mj} = a_{m\ \ m+1+j} \qquad (2.32)$$

$$B_m(z) = z^{-(m+1)} A_m\left(z^{-1}\right) \qquad (2.33)$$

และ

$$A_m(z) = z^{-(m+1)} B_m\left(z^{-1}\right) \qquad (2.34)$$

จากสมการ (2.23)

$$z^{-(m+1)} A_m\left(z^{-1}\right) = z^{-1}\left\{z^{-m} A_{m-1}\left(z^{-1}\right) + k_m z^{-m} B_{m-1}\left(z^{-1}\right)\right\} \qquad (2.35)$$

$$B_m(z) = z^{-1}\left\{k_m A_{m-1}(z) + B_{m-1}(z)\right\} \qquad (2.36)$$

จากสมการ (2.16) และ (2.18)

$$X_m^{+}(z) = \mathrm{A_m(z)X(z)}$$

$$X_m^-(z) = B_m(z)X(z) \tag{2.37}$$

แทนค่าทั้งสองลงในสมการ (2.23) และ (2.36)

$$X_m^+(z) = X_{m-1}^+(z) + k_m X_{m-1}^-(z)$$

$$X_m^-(z) = z^{-1}\{k_m X_{m-1}^+(z) + X_{m-1}^-(z)\} \tag{2.38}$$

หรือแสดงความสัมพันธ์ในเชิงเวลา

$$x_m^+(z) = x_{m-1}^+(n) + k_m x_{m-1}^-(n)$$

$$x_m^-(z) = k_m x_{m-1}^+(n-1) + x_{m-1}^-(n-1) \tag{2.39}$$

จากความสัมพันธ์ทั้งหมดที่กล่าวมาทำให้สามารถหาค่าพารามิเตอร์ของการวิเคราะห์สัญญาณ ซึ่งต่อไปจะเรียกพารามิเตอร์ที่ได้ด้วยวิธีนี้ว่า สัมประสิทธิ์ PARCOR หรือสัมประสิทธิ์การสะท้อน (reflection coefficient)

จากสมการ (2.23) เมื่อ $A_0(z) = 1$

$$A_m(z) = 1 + \sum_{i=1}^{m} k_i B_{i-1}(z) \tag{2.40}$$

$$\left\|A_m(z) - 1\right\|^2 = \sum_{i=1}^{m} k_i^2 \beta_{i-1} \tag{2.41}$$

และ

$$\left\|A_m(z) - 1\right\|^2 = \left\|A_m(z)\right\|^2 - 2\langle A_m(z), 1\rangle + \|1\|^2$$

$$= \alpha_m - 2\alpha_m + \|1\|^2$$

$$= \|1\|^2 - \alpha_m \tag{2.42}$$

แทนค่าดังกล่าวลงในสมการ (2.41)

$$\alpha_m = \|1\|^2 - \sum_{i=1}^{m} k_i^2 \beta_{i-1} \tag{2.43}$$

และ

$$\alpha_{m+1} = \|1\|^2 - \sum_{i=1}^{m+1} k_i^2 \beta_{i-1} \tag{2.44}$$

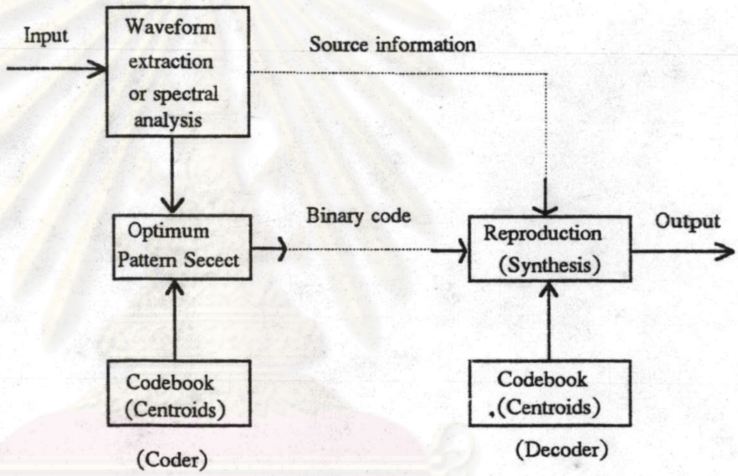
จากผลต่างของสมการทั้งสอง

$$\alpha_{m+1} = \alpha_m - k_{m+1}^2 \beta_m \tag{2.45}$$

2 Vector Quantization(VQ)

เป็นวิธีการลดไดเมนชั่น (Dimension) ของข้อมูล โดยทำการหาค่าใดค่าหนึ่งซึ่งเป็นตัวแทนของข้อมูลจำนวนมากหนึ่ง ๆ โดยเริ่มแรก วิธีนี้ใช้เพื่อการบีบอัดข้อมูล (Compression) แต่ในกรณีนี้ จะนำวิธีนี้มาใช้เพื่อทำให้ข้อมูลที่ผ่านจากการคำนวณพันธะเชิงเส้นเป็นข้อมูลที่ไม่ต่อเนื่อง (Discrete)

2.1 หลักการ Vector Quatization วิธีVector Quantization (VQ) เป็นวิธีการควอนไตซ์คลื่นเสียงหรือ spectral envelope เป็นเซ็ตแทนการ ควอนไตซ์ทีละตัว เช่น การควอนไตซ์แบบสเกลลาร์ VQ จะสามารถควอนไตซ์พารามิเตอร์ของ linear predictive coding(LPC) ได้อย่างมีประสิทธิภาพ ในรูป 2.4 เป็นขั้นตอนการทำงานของ Vector Quantization (Sadaoki Furui, 1989)



รูป 2.4 ขั้นตอนการทำงานของ VQ

ในการควอนไตซ์แบบเวกเตอร์ มีวิธีการควอนไตซ์ดังนี้คือ จะมีเวกเตอร์ code หรือ templates ที่เก็บไว้ อินพุทที่เข้ามาจะถูกทำการเปรียบเทียบกับ codebook ที่มีอยู่ โดยจะพิจารณาว่าอินพุทที่เข้ามานั้นห่างจาก codebook ใดน้อยที่สุด อินพุทดังกล่าวจะถูกแทนด้วยเวกเตอร์ code นั้น

codebook ที่ใช้จะเหมาะสมเพียงใด ขึ้นกับความคลาดเคลื่อนรวมทั้งหมดของต้นแบบที่ใช้หาเมื่อต้นแบบที่พิจารณามีจำนวนจำกัด codebook ที่มีความคลาดเคลื่อนน้อยที่สุด จะได้จากการทำซ้ำจนกว่าจะได้ค่าที่เหมาะสม โดยเหตุนี้จึงขึ้นกับ random learning และ clustering-base random learning คือการเลือกค่าขึ้นมา จำนวนหนึ่งซึ่งเท่ากับขนาดของ codebook จากข้อมูลต้นแบบ เพื่อใช้เป็นค่าเริ่มต้นในการ clustering ซึ่งเป็นวิธีการที่ใช้ใน Lloyd's algorithm (K-means algorithm) ในวิธีการนี้ข้อมูลต้นแบบที่ต้องการหาถูกแยกไปอยู่ในแต่ละกลุ่มของค่าเริ่มต้น จะทำการหาค่าเฉลี่ยของแต่ละกลุ่มเพื่อเป็นค่ากลางของกลุ่ม ทำวิธีการดังกล่าวซ้ำจนกว่าความคลาดเคลื่อนรวมจะต่ำกว่าค่า ๆหนึ่งหรือการลดลงของความคลาดเคลื่อนรวมน้อยกว่าค่า ๆหนึ่ง (ในวิทยานิพนธ์นี้ใช้ 0.00001 เนื่องจากค่าที่น้อยกว่านี้จะใช้เวลาคำนวณนานโดยให้ผล

ของระยะห่างของต้นแบบที่นำมาคำนวณหาค่าควอนไตซ์ใกล้เคียงกับ 0.00001) โดยค่ากลางดังกล่าวของกลุ่มจะถูกเก็บเป็นเวกเตอร์ code แต่แน่นอนว่าค่าความคลาดเคลื่อนรวมจะลดลงทุกครั้งที่มีการคำนวณซ้ำใหม่ จึงขึ้นกับค่าที่กำหนดว่าต้องการให้ความคลาดเคลื่อนรวมน้อยเพียงใด

2.2 การคำนวณ Vector Quantization ถ้าสมมุติ $x = [x_1, x_2, \dots, x_N]^T$ มี N มิติ เราจะทำการหาควอนไตซ์ของเวกเตอร์ x กับ codebook เราสามารถเขียนว่า y เป็นควอนไตซ์ของค่า x

$$y = q(x) \tag{2.46}$$

โดย q(.) เป็นโอเปอร์เรเตอร์ของควอนไตซ์ y ถูกเรียกว่าเอาท์พุทเวกเตอร์ของค่า x โดย y เป็นค่าใดค่าหนึ่งใน $Y = \{y_i, 1 \le i \le L\}$ โดย $y_i = [y_{i1} \; y_{i2} \; \dots \; y_{iN}]^T$ Y เป็นเซ็ตของ codebook, L เป็น ขนาดของ codebook และ $\{y_i\}$ เป็นเซ็ตของเวกเตอร์ code y_i อาจเรียกว่าเป็น code อ้างอิง และ L อาจเรียกว่าจำนวนระดับขั้น เมื่อเทียบกับสเกลาร์ควอนไตซ์ ทำการแบ่งเวกเตอร์ x ไปใน L เซล $\{C_i, 1 \le i \le L\}$ เมื่อ x อยู่ในเซล C_i

$$q(x) = y_i \qquad \text{ถ้า } x \in C_i \tag{2.47}$$

ถ้า x ถูกควอนไตซ์ได้ค่า y ค่าความคลาดเคลื่อนจากการควอนไตซ์สามารถแสดงได้โดยระยะห่างของทั้งสอง $d(x, y)$ โดยความคลาดเคลื่อนรวมคือ

$$D = \lim_{m \to \infty} \frac{1}{M} \sum_{n=1}^{M} d[x(n), y(n)] \tag{2.48}$$

การคำนวณหา codebook มีหลายวิธี แต่ที่จะนำมาใช้คือ Lloyd Agorithm (Furui, 1989)

2.3 การวัดความคลาดเคลื่อน การวัดความคลาดเคลื่อนเป็นส่วนจำเป็นและเป็นเครื่องมือที่เป็นประโยชน์ต่อการออกแบบ coding ซึ่ง ในปัจจุบันมีวิธีการหลายแบบ แต่ที่จะนำมาใช้ในวิทยานิพนธ์นี้คือ square error distortion.

The squared error distortion:
ถ้าสัญญาณที่เข้ามามี K มิติ เราสามารถหาระยะห่างระหว่างสัญญาณเข้ากับเวกเตอร์ code

$$d(x, y) = \|x - y\|^N = \sum_{i=0}^{k-1} (x_i - y_i)^N \tag{2.49}$$

หรือ
$$d(x, y) = \|x - y\|^2 = \sum_{i=0}^{k-1} (x_i - y_i)^2 \tag{2.50}$$

วิธีนี้เป็นวิธีแบบง่าย ๆ ที่ใช้กันทั่วไป ซึ่งมีความซับซ้อนน้อย ง่ายต่อการคำนวณ เราสามารถหา signal-to-noise ratio ได้ดังนี้

$$\text{SNR} = 10 \log_{10} \frac{E\left(\|x\|^2\right)}{E\left[d(x,y)\right]} \tag{2.51}$$

2.4 การสุ่มค่าเริ่มต้น วิธีหนึ่งที่ใช้ในการออกแบบ codebook คือการเลือกค่าเริ่มต้นของ codebook ในการสุ่มตัวอย่างจาก ข้อมูลที่จะนำมาคำนวณ เราเรียก codebook ที่ได้จากการสุ่มค่าเริ่มต้นนี้ว่า random codebook ถึงแม้ว่าการสุ่มค่า เริ่มต้นจะไม่ใช่วิธีที่ดีนักแต่ codebook ที่ได้จากการสุ่มค่าเริ่มต้นให้ผลที่ยอมรับได้(makhoul, Roucos and Gish, 1985)

2.5 ขั้นตอนการทำงาน ในการออกแบบ codebook สิ่งสำคัญที่สุดคือขั้นตอนการหา codebook K-means algorithm จะต้องมี การสุ่มค่า codebook เริ่มต้น

2.5.1 การหา codebook K-means initialization วิธีการ K-means ไม่ยืนยันว่าคำตอบจะได้ค่าที่มีความคลาดเคลื่อนน้อยที่สุด จะต้องทำการสุ่มค่า codebook เริ่มต้นจากหลาย ๆ ค่า codebook ที่ได้ความคลาดเคลื่อนน้อยที่สุดจะถูกเลือก

2.5.2 การเปรียบเทียบ เมื่อมี codebook เรียบร้อย เสียงที่จะนำไปหาโมเดลโดยวิธี HMM จะต้องผ่านขั้นตอนนี้ เพื่อหา ตัวแทนของสัมประสิทธิ์การสะท้อนในรูปของควอนไตซ์

PATTERN SIMILARITY DETERMINATION

ในที่นี้หมายถึง การหาตัวแทนที่จะนำมาเปรียบเทียบกับสิ่งที่เราไม่ทราบ Hidden Markov Model เป็นวิธีการที่ใช้กันอย่างแพร่หลาย แรกเริ่มมีการเสนอวิธีนี้ในการวิเคราะห์คำต่อเนื่อง (Continuous Word) ก่อน เมื่อได้ผลดี จึงมีการนำมาใช้กับคำโดด(Isolated Word) ซึ่งได้ผลดีเช่นเดียวกัน (Rabiner, Levinson and Sondhi, 1982)

1. Hidden Markov Model (HMM)

HMM เป็นระบบ doubly stochastic HMM เป็นวิธีการใช้ทฤษฎีความน่าจะเป็น มาอธิบายการเกิดของ sequence 2 ตัว คือ state และค่าปรากฏ โดยผู้สังเกตจะเห็นเพียงเอาท์พุทของแต่ละ state (ค่าปรากฏ) แต่จะไม่ทราบแน่ชัดว่าอยู่ที่ state ใด จึงเรียกว่า Hidden Markov Model (HMM)

1.1 พารามิเตอร์ใน HMM

T = ความยาวของค่าปรากฏ

N = จำนวน state ในโมเดล

M= จำนวนค่าปรากฏ

$Q = \{q_1, q_2, \dots, q_N\}$ state

$V = \{v_1, v_2, \dots, v_N\}$ ค่าปรากฏ

$A = \{a_{ij}\}$, $a_{ij} = \Pr(q_j$ ที่เวลา $t+1/q_i$ ที่เวลา $t)$ คือ ความน่าจะเป็นของการเปลี่ยน state

$B = \{b_j(k)$, $b_j(k) = \Pr(V_k$ ที่เวลา t/q_j ที่เวลา $t)$ คือ ความน่าจะเป็นของการเกิดค่าปรากฏ

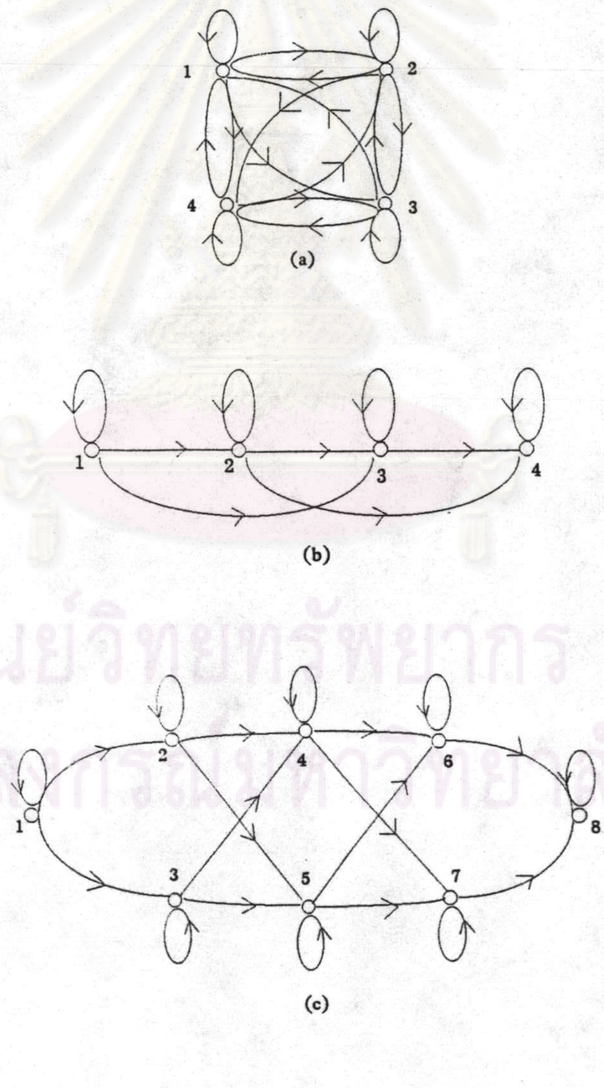
$\pi = \{ \pi_i \}$, $\pi_i =$ Pr(q_i ที่เวลา t=1) ความน่าจะเป็นที่แต่ละ state จะเป็น state เริ่มต้น

1.2 โครงสร้างของ Hidden Markov (HMM) มีอยู่ 3 ชนิดคือ

1.2.1. unconstrained model โมเดลนี้ทุก state สามารถติดต่อกับ state อื่นๆ ได้ทุก state ดังรูป 2.5(a) โดยค่า a_{ij} ทุกตัวจะมีค่า ($a_{ij} \neq 0$)

1.2.2. constrained serial model ในโมเดลนี้เมื่อผ่าน state ใดไปแล้วจะไม่มีการย้อนกลับมาที่ state นั้น อีก ดังรูป 2.5(b)

1.2.3. constrained parallel model ในโมเดลนี้จะมีคุณสมบัติเช่นเดียวกับ constrained serial model แต่มีความซับซ้อนกว่า ดังรูป 2.5(c)



รูป 2.5 โครงสร้างแบบต่างๆ

1.3 ปัญหาของ HMM มีปัญหา 3 อย่างซึ่งต้องใช้ algorithm ต่างๆ ในการคำนวณ (Fabiner, and Jung, 1986)

ปัญหา 1: ถ้ามี sequence ของค่าปรากฏ $O = O_1, O_2, \ldots, O_T$ และมีโมเดล $\lambda = (A, B, \pi)$ เราจะคำนวณหา $\Pr(O/\lambda)$ ของ sequence ค่าปรากฏได้อย่างไร

ปัญหา 2: ถ้ามี sequence ของค่าปรากฏ $O = O_1, O_2, \ldots, O_T$ เราจะหา sequence ของ state ที่ $I = i_1, i_2, \ldots, i_T$ ที่ให้ค่าปรากฏนั้นได้อย่างไร

ปัญหา 3: จะหาโมเดล $\lambda = (A, B, \pi)$ ที่ให้ค่า $\Pr(O/\lambda)$ มากที่สุดอย่างไร

ปัญหา 1: เมื่อให้โมเดลและ sequence ค่าปรากฏ เราจะคำนวณความน่าจะเป็นที่โมเดลนั้นให้ค่าปรากฏที่กำหนดมา หรือ อาจกล่าวว่า เมื่อมีโมเดลและ sequence ของค่าปรากฏ เราจะหาว่าโมเดลนั้นมีโอกาสให้ค่า sequence ค่าปรากฏนั้นด้วยค่าเท่าไร หรือโมเดลนั้นเหมาะสมกับค่าปรากฏมากหรือน้อย

ปัญหา 2: เป็นส่วนที่เราพยายามจะระบุ state ให้กับ sequence ของค่าปรากฏ ซึ่งทราบอยู่แต่แรกแล้วว่า state เป็นค่าที่ไม่แน่ชัด มีหลายวิธีในการคำนวณ

ปัญหา 3: พยายามหาโมเดลที่เหมาะสมจาก sequence ค่าปรากฏ เราเรียกขั้นตอนนี้ว่า การหาโมเดลของเสียง เพื่อสร้างรูปแบบโมเดลที่เหมาะสมให้กับเสียงนั้น

ถ้าเราจะพิจารณาหาโมเดล λ เราจะต้องกำหนดค่า N-state สำหรับแต่ละคำเราใช้ Vector Quantization (VQ) มาช่วย เราทำการแทนสัญญาณเสียงด้วยค่าใดค่าหนึ่งใน codebook ของ VQ จาก codebook ที่มีขนาด M เริ่มต้นการหาโมเดลของเสียง จาก sequence ค่าปรากฏจำนวนหนึ่งของแต่ละคำ โดยค่าหนึ่งๆ จะมีผู้พูดหลายคน ใช้ปัญหา 3 ในการพิจารณา หลังจากนั้นเราจะระบุ state ให้กับ sequence ค่าปรากฏ โดยปัญหา 2 จนขั้นตอนสุดท้ายเราจะหาว่า sequence ใดๆ เมื่อนำมาเปรียบเทียบกับโมเดลที่หาได้ มีความเป็นไปได้ที่โมเดลจะให้ค่าปรากฏดังกล่าวมากน้อยเพียงไร ใช้ ปัญหา 1

1.4 การคำนวณของ HMM

ปัญหา 1:

ในการคำนวณว่าโมเดล λ ให้ความน่าจะเป็นที่จะได้ sequence ค่าปรากฏ O มากน้อยเพียงใด โดยมีการระบุ state ต่างๆ ให้กับ sequence ค่าปรากฏยาว T $I = i_1 i_2 \ldots i_T$ ความน่าจะเป็นของค่าปรากฏ O คือ $\Pr(o/l,\lambda)$

$$\Pr(o/l,\lambda) = b_{i_1}(O_1)b_{i_2}(O_2)\ldots b_{i_T}(O_T) \quad (2.52)$$

$$\Pr(l/\lambda) = \pi_{i_1} a_{i_1 i_2} a_{i_2 i_3} \ldots a_{i_{T-1} i_T} \quad (2.53)$$

หรืออาจเขียนได้ว่า

$$\Pr(O,l/\lambda) = \Pr(O/l,\lambda)\Pr(l/\lambda) \quad (2.54)$$

โดยที่ความน่าจะเป็นของ O คือผลรวมของ joint probability ของทุก state ที่เป็นไปได้

$$\Pr(O/\lambda) = \sum_{all\ l} \Pr(O/l,\lambda)\Pr(l/\lambda)$$

$$= \sum_{i_1,i_2,\ldots,i_T} \pi_{i_1} b_{i_1}(O_1) a_{i_1 i_2} b_{i_2}(O_2) \ldots a_{i_{T-1} i_T} b_{i_T}(O_T) \quad (2.55)$$

ถ้าเรากำหนด forward variable $\alpha_t(i)$

$$\alpha_t(i) = \Pr(O_1, O_2, \ldots, O_t, i_t = q_i / \lambda) \quad (2.56)$$

เราจะได้ว่า

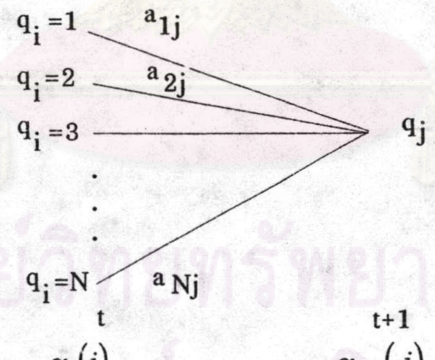
$$\alpha_1(i) = \pi_i b_i(O_1) \quad , 1 \le i \le N \quad (2.57)$$

และที่เวลา t ต่างๆ ดังในรูป 2.6

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \quad , t=1, 2, \ldots, T-1 \quad , 1 \le j \le N \quad (2.58)$$

ทำให้สมการ (2.55) เขียนเป็น

$$\Pr(O/\lambda) = \sum_{i=1}^{N} \alpha_T(i) \quad (2.59)$$



รูป 2.6 แสดงการพิจารณา forward variable

ถ้ากำหนด backward variable $\beta_t(i)$

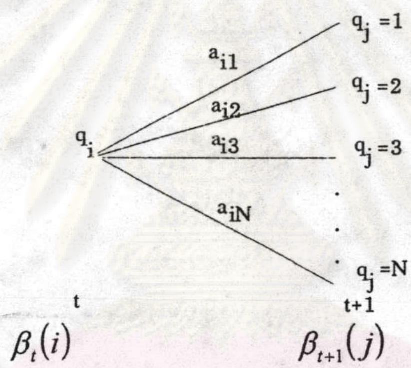
$$\beta_t(i) = \Pr(O_{t+1}, O_{t+2}, \ldots, O_T / i_t = q_i, \lambda) \tag{2.60}$$

โดย $$\beta_T(i) = 1 \qquad , 1 \le i \le N \tag{2.61}$$

และที่เวลา t ใดๆ ดังรูป 2.7

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j\left(O_{t+1}(j)\right) \qquad , t = T-1, T-2, \ldots, 1$$

$$, 1 \le i \le N \tag{2.62}$$



รูป 2.7 แสดงการพิจารณา backward variable

ปัญหา 2:

เมื่อกำหนด

$$\gamma_t(i) = \Pr(i_t = q_i / O, \lambda) \tag{2.63}$$

ความน่าจะเป็นที่ state q_i ที่เวลา t ให้ sequence ค่าปรากฏด้วยโมเดล λ และ $\gamma_t(i)$ สามารถเขียนในรูป α และ β ได้

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\Pr(O/\lambda)} \tag{2.64}$$

โดย $\alpha_t(i)$ เริ่มจาก $O_1, O_2, \ldots, O_t$ จนถึง state q_i ที่เวลา t ส่วน $\beta_t(i)$ เริ่มจาก $O_{t+1}, \ldots, O_T$ จนถึง q_i ที่เวลา t

ค่า $\gamma_t(i)$ เป็น conditional probability จึงเขียนได้ว่า

$$\sum_{i=1}^{N} \gamma_t(i) = 1 \tag{2.65}$$

ปัญหา 3:

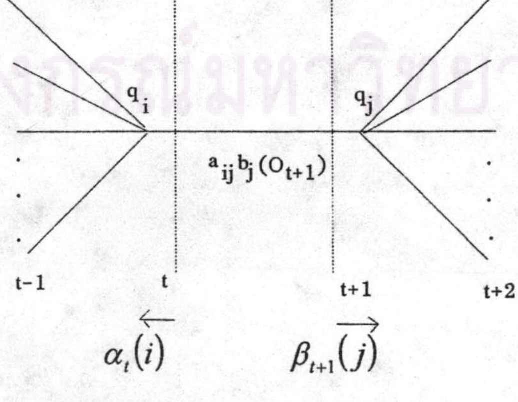
เมื่อมี sequence ค่าปรากฏจำนวนหนึ่ง ทำการหาโมเดลที่จะให้ผลตามค่าปรากฏเหล่านี้ $\lambda = (A, B, \pi)$ เราจะใช้การทำซ้ำ โดยใช้วิธี Baum-Welch หรือ gradient techniques
ถ้ากำหนด

$$\xi_t(i,j) = \Pr(i_t = q_i, i_{t+1} = q_j / O, \lambda) \tag{2.66}$$

ความน่าจะเป็นที่ state q_i ที่เวลา t ทำการเปลี่ยนไปที่ state q_j ที่เวลา t+1 จากรูป 2.8 เราสามารถเขียน $\xi_t(i,j)$ ได้

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\Pr(O/\lambda)} \tag{2.67}$$

$\alpha_t(i)$ เริ่มจากเวลา t=1 ที่ค่าปรากฏแรก จนถึง state q_i ที่เวลา t ค่า $a_{ij}b_j(O_{t+1})$ เป็นการเปลี่ยน state q_i ที่เวลา t ไปเป็น q_j ที่เวลา t+1 และให้ค่าปรากฏ O_{t+1}



รูป 2.8 การพิจารณา forward และ backward

เราสามารถเขียนความสัมพันธ์ของ $\xi_t(i,j)\ x \xleftrightarrow{1-1} \xi$ กับ $\gamma_t(i)$ ได้ดังนี้

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) \tag{2.68}$$

ใช้ Baum-Welch reestimation ในการประมาณค่า π , A, B ตามสมการ (2.69-2.71)

$$\overline{\pi}_i = \gamma_1(i) \quad , 1 \le i \le N \tag{2.69}$$

$$\overline{a}_{ij} = \sum_{t=1}^{T-1} \xi_t(i,j) / \sum_{t=1}^{T-1} \gamma_t(i) \tag{2.70}$$

$$\overline{b}_j(k) = \sum_{\substack{t=1 \\ O_t=k}}^{T} \gamma_t(j) / \sum_{t=1}^{T} \gamma_t(j) \tag{2.71}$$

DECISION RULE

การจะใช้วิธีใดในการตัดสินใจ จะต้องคำนึงถึงวิธีการที่ใช้ใน pattern similarity determination ต้องมีความสอดคล้องกัน

1 Viterbi Algorithm (VA)

ถูกเสนอใช้ในปี 1967 ในการ decode convolution codes ต่อมามีการนำไปใช้ในทางอื่นๆ เช่น VA ที่ใช้กับ state ของระบบ stochastic โดยใช้ recusive method ในการหาค่าที่เหมาะสม วิธีนี้จะทำให้การคำนวณรวดเร็วยิ่งขึ้น เพราะพิจารณาเฉพาะ path ที่มีค่าความน่าจะเป็นสูงสุด (Forney, 1973)

1.1 ทฤษฎี VA นำ VA มาใช้กับระบบ Markov state x_k ที่เวลา k เป็นหนึ่งในจำนวนทั้งหมด M state $1 \le m \le M$, {1, 2, ... , M} โดยเราสมมุติว่าระบบจะเริ่มที่เวลา t=0 จนถึง t= K และจะเริ่มที่ state x_0 จนถึง state x_K โดยสามารถเขียน state sequence ได้เป็น $X = \{x_0, x_1, \ldots, x_K\}$

ในระบบของ Markov ซึ่งจะอยู่ในรูปความน่าจะเป็น $P(x_{k+1}/x_0, x_1, \ldots, x_k)$ ใน state x_{k+1} ที่เวลา k+1 จะขึ้นกับ state x_k ที่เวลา k เท่านั้น

$$P(x_{k+1}/x_0, x_1, \ldots, x_k) = P(x_{k+1}/x_k) \tag{2.72}$$

เราสามารถเขียน transition ξ_k ที่เวลา k ในรูปของ state

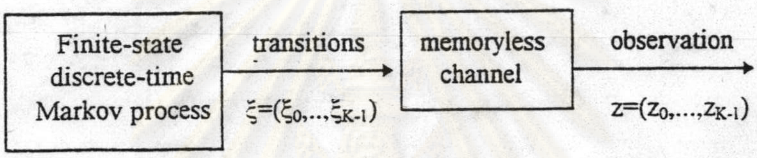
$$\xi_k \,\underline{\Delta}\, (x_{k+1}, x_k) \tag{2.73}$$

โดย Ξ คือเซ็ตของ transitions $\xi_k = (x_{k+1}, x_k)$ ซึ่ง $P(x_{k+1}/x_k) \neq 0$ และ $|\Xi| \leq M^2$ ความสัมพันธ์ ของ state sequence x และ transition sequence $\xi = \xi_0, \ldots, \xi_{K-1}$ เป็นแบบ 1:1 $x \xleftrightarrow{1-1} \xi$

เราสามารถเขียน sequence ของค่าปรากฏ z_k ที่ขึ้นกับค่าความน่าจะเป็นของ transition ξ_k ที่เวลา k

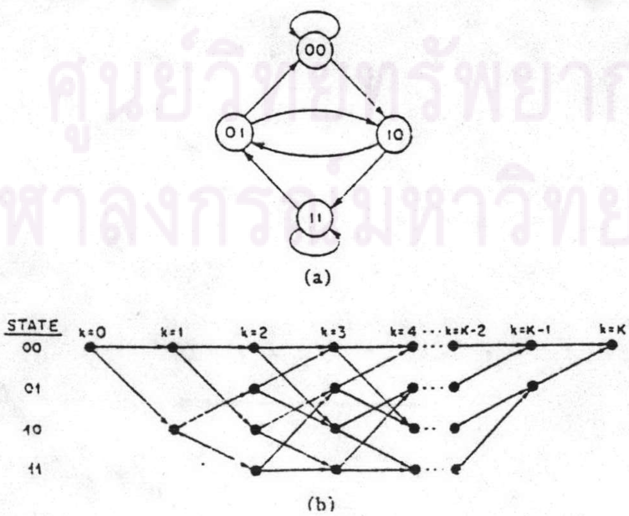
$$P(z/x) = P(z/\xi) = \prod_{k=0}^{K-1} P(z_k / \xi_k) \qquad (2.74)$$

จากรูป 2.9 จะเห็นว่า z จะเป็นค่าเอาท์พุทที่ออกจาก channel โดยอินพุคือ sequence ของ ξ



รูป 2.9 โมเดลของ Viterbi

1.2 Algorithm จากรูป 2.10(a) แสดงถึง state diagram ของระบบ Markov ที่พิจารณาในช่วงเวลาที่ไม่ต่อเนื่อง ซึ่งมี 4 state โดย node แสดงถึง state เส้นเชื่อมแสดงถึง transition



รูป 2.10 โครงสร้าง state และ trellis ของ Viterbi

ในรูป 2.10(b) เป็น trellis โดย node แสดง state และเส้นเชื่อมแสดงการเปลี่ยนจาก state หนึ่งไปอีก state หนึ่ง ในเวลาถัดไป โดยเราจะทราบ state เริ่มต้นและ state สิ้นสุด x_0 และ x_K

เมื่อให้ sequence ของค่าปรากฏ z เราจะหา state sequence ที่ให้ค่า P(x/z) มากที่สุด หรือ P(x,z) = P(x/z)P(z) มีค่ามากที่สุด โดยให้ระยะทาง -lnP(x,z) สั้นที่สุด ซึ่งความสัมพันธ์ path และ sequence เป็นแบบ 1:1

$$P(x,z) = P(x)P(z/x)$$

$$= \prod_{k=0}^{K-1} P(x_{k+1} / x_k) \prod_{k=0}^{K-1} P(z_k / x_{k+1}, x_k) \quad (2.75)$$

โดยระยะทางหาได้จาก

$$\lambda(\xi_k) \triangleq -\ln P(x_{k+1}, x_k) - \ln P(z_k / \xi_k) \quad (2.76)$$

ระยะทางทั้งหมดของ path

$$-\ln P(x,z) = \sum_{k=0}^{K-1} \lambda(\xi_k) \quad (2.77)$$

การหาระยะทางที่สั้นที่สุด ถูกเสนอโดย Minty ในปี 1957 แต่วิธีที่เสนอนี้ไม่เหมาะสมจะคำนวณโดยคอมพิวเตอร์ แต่วิธีการหาระยะทางที่สั้นที่สุดจำเป็นสำหรับ VA

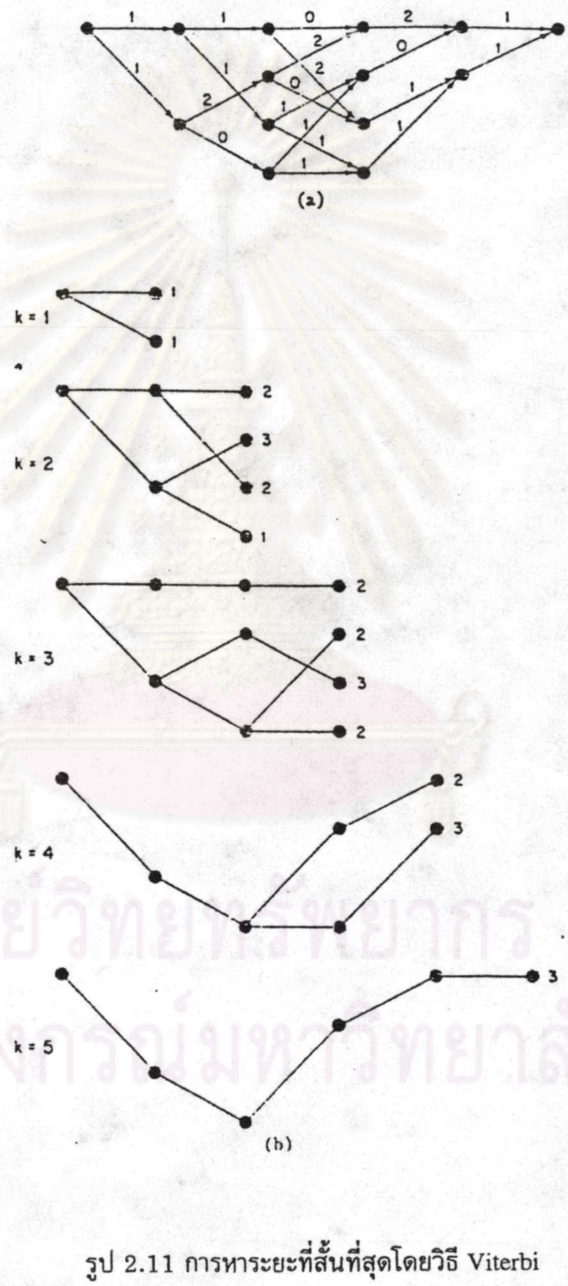
ถ้าเรากำหนด x_0^k เป็น state sequence ตั้งแต่ x_0 จนถึง x_k ซึ่งมีหลายเส้นทางที่เป็นไปได้ ซึ่งสามารถเขียนได้เป็น

$$\lambda(x_0^k) = \sum_{i=0}^{K-1} \lambda(\xi_i) \quad (2.78)$$

โดยระยะทางที่สั้นที่สุดเรียกว่า survivor โดยปลายทางคือ node x_k เขียนได้เป็น $\hat{x}(x_k)$ ถ้า k>0 ที่ค่า k ต่างๆ เขียนทางที่จะเป็นไปได้ทั้งหมด แล้วเลือกทางที่สั้นที่สุด เมื่อค่า k เพิ่มขึ้น เลือกทางที่สั้นที่สุดสำหรับ k นั้นๆอีก จนกระทั่งถึง k=K จะได้ระยะทางรวมสั้นที่สุด ดังรูป 2.11(a),(b)

หาระยะทางโดย

$$\Gamma(x_k) \triangleq \lambda[\hat{x}(x_k)] \quad (2.79)$$



รูป 2.11 การหาระยะที่สั้นที่สุดโดยวิธี Viterbi