บทที่ 6

การเรียนรู้ของเครื่อง

คอมพิวเตอร์ถือได้ว่าเป็นเครื่องจักรกลที่สามารถคำนวณหรือประมวลผลต่างๆ ได้รวดเร็วกว่ามนุษย์ อย่างไรก็ตามคอมพิวเตอร์ก็ยังคงต้องรับคำสั่งหรือความรู้ต่างๆ จากมนุษย์ ดังนั้นนักคอมพิวเตอร์จึงได้มีการพยายามที่จะให้คอมพิวเตอร์มีความสามารถในการเรียนรู้ โดยปัจจุบันได้มีการพัฒนาการเรียนรู้ของเครื่องในแบบต่างๆ ขึ้นมากมาย และได้นำไปประยุกต์ใช้กับงานต่างๆ เป็นจำนวนมาก เช่น มีการนำไปใช้ในการรู้จำตัวอักษร (Boonserm Kijsirikul, Sukree Sinthupinyo and Apinya Supanwansa, 1998) โดยมีการนำรูปภาพของแต่ละตัวอักษรเข้ามาสอนให้คอมพิวเตอร์รู้จักว่าเป็นตัวอักษรใด หรือนำไปใช้ในการสร้างกฎเพื่อใช้ในการแก้ไขคำผิด (Golding and Roth,1996 ; Surapant Meknavin et.al.,1998) เป็นต้น

จากงานการตัดคำที่ผ่านมามีการนำพจนานุกรม ค่าสถิติต่างๆ ของคำเข้ามาช่วยในการประมวลผล ซึ่งสามารถเพิ่มความถูกต้องได้มากกว่าเดิม แต่อย่างไรก็ตามยังมีอีกหลายกรณีที่ไม่สามารถแก้ปัญหาด้วยวิธีการดังกล่าวได้ ดังนั้นในวิทยานิพนธ์นี้จะนำเอาคุณลักษณะ (Feature) ต่างๆ เข้ามาช่วยในการแก้ไขปัญหาความกำกวม และชื่อเฉพาะที่ไม่พบในพจนานุกรม โดยคุณลักษณะต่างๆ ที่ได้มานั้นจะนำมาจากการเรียนรู้ของเครื่อง โดยวิธีการเรียนรู้ของเครื่องที่จะนำมาศึกษาและประยุกต์ใช้คือ ริปเปอร์ และ วินโนว์

ในบทนี้จะอธิบายถึงวิธีการเรียนรู้ของเครื่องที่ชื่อว่า ริปเปอร์ และ วินโนว์ ส่วนรายละเอียดและวิธีการที่จะนำการเรียนรู้ของเครื่องทั้ง 2 แบบมาประยุกต์ใช้ในการตัดคำจะนำมากล่าวในบทถัดไป

6.1 ริปเปอร์ (RIPPER : Repeated Incremental Pruning to Produce Error Reduction)

ริปเปอร์ (Cohen, 1995) เป็นกระบวนการเรียนรู้ของเครื่องแบบหนึ่ง โดยริปเปอร์สามารถทำการเรียนรู้จากตัวอย่างที่ให้มา และจะสรุปออกมาเป็นกฎให้โดยอัตโนมัติ โดยรูปแบบของกฎที่ได้มาจะมีรูปแบบดังนี้

$$\text{if } T_1 \text{ and } T_2 \text{ and } \ldots T_n \text{ then } C$$

โดย T_i คือเงื่อนไขต่างๆ ส่วน C คือคำตอบ จากกฎที่ได้มานี้จะมีความหมายคือ ถ้าเงื่อนไข T_1, T_2, ... และ T_n เป็นจริงแล้ว ดังนั้นคำตอบก็คือ C ส่วนเงื่อนไขที่ริปเปอร์ยอมให้มีได้มีอยู่ 4 แบบคือ เท่ากับ (=) , มากกว่าเท่ากับ (≥), น้อยกว่าเท่ากับ (≤) และ การเป็นสมาชิกในเซต (∈)

ตัวอย่างกฎที่ได้จากการเรียนรู้ของริปเปอร์ ในการแก้ปัญหาความกำกวมของข้อความ "ที่อยู่" ซึ่งสามารถตัดคำได้ 2 แบบคือ "ที่อยู่" และ "ที่ อยู่" เช่น

- ถ้า pw1="ของ" แล้วให้ตัดเป็น "ที่อยู่"
- ถ้า มี ∈ pw310 และ mt2=RPRE แล้วให้ตัดเป็น ที่อยู่

โดย pw1 คือ คำถัดไปทางด้านขวามือของข้อความ "ที่อยู่" pw310 คือคำบริบทที่อยู่ด้านขวามือของข้อความ "ที่อยู่" และ mt2 คือหน้าที่คำของคำที่อยู่ถัดไปทางด้านซ้าย 2 คำของข้อความ "ที่อยู่" ส่วน RPRE คืออักษรย่อที่ใช้แสดงหน้าที่คำ ซึ่งหมายถึงคำบุพบท

6.1.1 ขั้นตอนการเรียนรู้ของริปเปอร์

การเรียนรู้ของเครื่องริปเปอร์ได้มีการปรับปรุงมากจาก การเรียนรู้ของเครื่องไอเร็ป (IREP :Incremental Reduced Error Pruning) ซึ่งพัฒนาโดย (Furnkranz and Widmer, 1994) สำหรับรายละเอียดที่ริปเปอร์ได้มีการปรับปรุงจากไอเร็ป คือ 1. ในการลดกฎ (Pruning rules) โดยริปเปอร์จะยอมให้มีการลบกฎที่มีจำนวนเงื่อนไขมากกว่า 1 เงื่อนไขได้ ในขณะที่ไอเร็ปจะอนุญาตให้ลบกฎที่มีเงื่อนไข 1 เงื่อนไขเท่านั้น และ 2. ในการหยุดเพิ่มกฎ สำหรับริปเปอร์จะหยุดเพิ่มกฎเมื่อกฎที่ได้นั้นมีข้อผิดพลาดเกินกว่า 50% และ 3. ริปเปอร์อนุญาตให้มีคุณสมบัติเพิ่มเติมจากไอเร็ปคือ อนุญาตให้มีการใส่คุณสมบัติไม่ครบ อนุญาตให้มีตัวแปรแบบตัวเลข และอนุญาตให้คำตอบที่ได้หลายคลาส (Multiple class) ซึ่งในไอเร็ปยอมให้คำตอบเป็นไปได้แค่เพียง 2 คำตอบคือจริงกับเท็จเท่านั้น

ขั้นตอนการสร้างกฎของริปเปอร์ คือเมื่อมีการนำตัวอย่างต่างๆ เข้ามาป้อนให้ริปเปอร์ ริปเปอร์จะพยายามสร้างกฎต่างๆ ที่สามารถครอบคลุมตัวอย่างที่มีอยู่ทั้งหมดให้มากที่สุด เมื่อริปเปอร์สร้างกฎทั้งหมดแล้ว ริปเปอร์ก็จะพยายามลดกฎที่เป็นกฎที่เฉพาะเจาะจงสำหรับตัวอย่างบางตัวอย่างเท่านั้น ซึ่งวิธีการนี้จะเป็นข้อดีในกรณีที่เราไม่มั่นใจว่าตัวอย่างที่เราจะนำมาให้ริปเปอร์นั้นมีความถูกต้องถึง 100% หรือกล่าวอีกนัยหนึ่งคือยอมให้มีการป้อนตัวอย่างที่ผิดได้บ้างเล็กน้อย

6.1.2 ข้อดีของริปเปอร์

1. กฎที่ได้จากการเรียนรู้ของริปเปอร์นั้นจะอยู่ในรูปแบบ ถ้า...แล้ว (if-then rule) ทำให้ง่ายต่อความเข้าใจ โดยเฉพาะอย่างยิ่งถ้าต้องการพิจารณาเข้าไปถึงลักษณะของกฎที่ได้มากับตัวอย่างที่นำไปใช้ใน

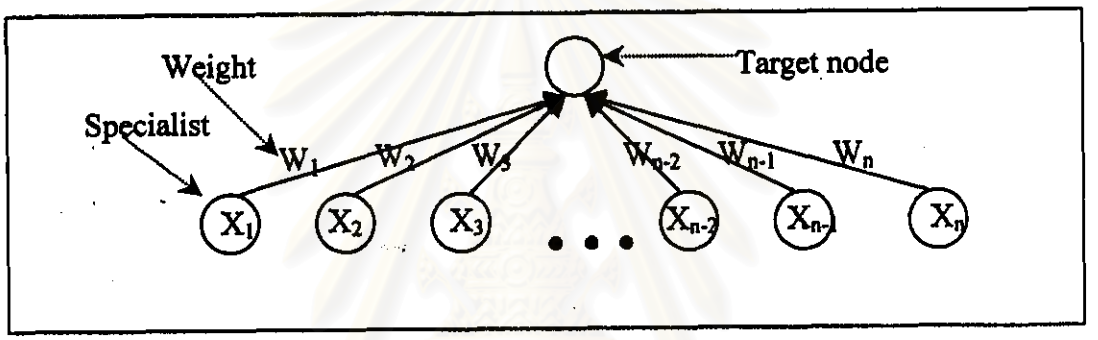
การเรียนรู้ ในกรณีนี้การใช้ริปเปอร์จะสามารถดีกว่าการนำกระบวนการเรียนรู้ของเครื่องแบบโครงข่ายประสาทเทียม (Neural network) หรือ การใช้ต้นไม้ตัดสินใจ (Decision tree)

2. การทำงานของริปเปอร์สามารถจะทำงานได้รวดเร็วกว่าขั้นตอนวิธีการเรียนรู้แบบกฎ (Rule learning algorithm) แบบอื่นๆ ในกรณีที่ตัวอย่างที่นำมาใช้ในการเรียนรู้มีจำนวนมากๆ

3. ริปเปอร์สามารถให้ผู้ใช้กำหนดเงื่อนไขสำหรับรูปแบบของกฎที่ได้ออกมา ทำให้กฎที่ได้ออกมานั้นจะมีความถูกต้องมายิ่งขึ้น

4. ริปเปอร์อนุญาตให้มีการใช้คุณสมบัติที่เป็นเซต (Set-valued attribute)ได้ ซึ่งในกรณีนี้ช่วยให้ลักษณะกฎที่ได้ออกมานั้นมีรูปแบบกะทัดรัด

6.2 วินโนว์ (Winnow)



รูปที่ 6-1 โครงข่ายวินโนว์ (Winnow Network)

วินโนว์ (Littlestone, 1988; Golding and Roth, 1996; Blum, 1997) เป็นวิธีการเรียนรู้ของเครื่องวิธีหนึ่ง โดยวินโนว์จะเป็นการเรียนรู้โดยพิจารณาจากตัวอย่าง (Learning by examples) ที่ป้อนเข้ามา ซึ่งลักษณะของวินโนว์จะคล้ายกับโครงข่ายประสาทเทียม (Neural Network) ซึ่งจะแสดงในรูปที่ 6-1 วินโนว์จะประกอบด้วยโหนดต่างๆ ที่มีการเชื่อมไปสู่โหนดเป้าหมาย (Target node) โดยโหนดต่างๆ ซึ่งเราจะเรียกว่าผู้เชี่ยวชาญ (Specialist) โดยแต่ละผู้เชี่ยวชาญ จะทำการตรวจสอบค่าของคุณสมบัติ (attribute = value) จำนวน 2 คู่เท่านั้น เมื่อแต่ละผู้เชี่ยวชาญทำการตรวจสอบแล้วก็จะทำการทำนายผลลัพธ์ โดยแต่ละผู้เชี่ยวชาญ จะมีน้ำหนัก (Weight) หรือความน่าเชื่อถือไม่เท่ากัน เมื่อผู้เชี่ยวชาญได้ทำนายผลลัพธ์หมดแล้ว วินโนว์จะทำการรวมคะแนนโดยพิจารณาจากค่าความน่าเชื่อถือของแต่ละผู้เชี่ยวชาญแล้วค่อยทำนายออกมาเป็นคำตอบ

ตัวอย่างโครงข่ายวินโนว์ที่นำมาใช้ในการแก้ปัญหาความกำกวมของข้อความ "ที่อยู่" เช่น

- ผู้เชี่ยวชาญ X_1 ในรูป 6.1 จะพิจารณาเฉพาะตัวอย่างที่นำเข้ามาที่มี pw1=ของ และ pt1=RPRE แล้วทำการทำนายว่า จะต้องตัดคำได้ว่า "ที่อยู่" ด้วยค่าความน่าเชื่อถือ 20.2

- ผู้เชี่ยวชาญ X_2 ในรูป 6.1 จะพิจารณาเฉพาะตัวอย่างที่นำเข้ามาที่มีคำว่า "มี"∈pw310 และ pt1=RPRE แล้วทำการทำนายว่า จะต้องตัดคำได้ว่า "ที่ อยู่" ด้วยค่าความน่าเชื่อถือ 25.25 เป็นต้น

ดังนั้นเมื่อตัวอย่างที่นำเข้ามาป้อนให้กับวินโนว์ ผู้เชี่ยวชาญแต่ละตัวจะทำการตรวจสอบว่าคุณสมบัติและค่านั้นตรงกับค่าที่ผู้เชี่ยวชาญตัวนั้นรับผิดชอบหรือไม่ ในกรณีที่ตรงก็ทำการทำนายผลลัพธ์ออกมา แต่ถ้าไม่ตรงจะไม่ทำการทำนาย เมื่อทำนายครบทุกตัวแล้ว วินโนว์จะทำการรวมคะแนนแล้วทำนายผลลัพธ์ออกมา

6.2.1 หลักการทำงานของวินโนว์

1. กำหนดค่าเริ่มต้นให้กับความน่าเชื่อถือของผู้เชี่ยวชาญแต่ละตัวให้เป็น 1
2. นำตัวอย่างที่ทราบคำตอบแล้วส่งให้กับวินโนว์
3. ผู้เชี่ยวชาญแต่ละตัว จะทำการตรวจสอบคุณสมบัติที่ตนเองรับผิดชอบและทำการทำนายโดยพิจารณาจากข้อมูลที่ผ่านมา
4. วินโนว์จะทำการรวมคะแนนของผู้เชี่ยวชาญแต่ละตัว โดยพิจารณาจากค่าความน่าเชื่อถือด้วย
5. ในกรณีที่วินโนว์มีการทำนายผิด วินโนว์จะทำการปรับค่าความน่าเชื่อถือของผู้เชี่ยวชาญ
    5.1 สำหรับผู้เชี่ยวชาญที่ตอบผิด ค่าความน่าเชื่อถือจะถูกลดลงครึ่งหนึ่ง
    5.2 สำหรับผู้เชี่ยวชาญที่ตอบถูก ค่าความน่าเชื่อถือจะถูกเพิ่มขึ้น โดยคูณด้วย 1.5

หลังจากวินโนว์ได้ทำการเรียนรู้จากตัวอย่างที่ป้อนให้ทั้งหมดแล้ว เราก็จะได้โครงข่ายวินโนว์ (Winnow Network) และเราจะนำโครงข่ายนี้มาใช้ในการแก้ปัญหา

6.2.2 ข้อดีของวินโนว์

1. สามารถทำการเรียนรู้ได้เร็วเนื่องจากลักษณะของวินโนว์เป็นการเรียนรู้แบบค่อยๆ เรียนรู้ (Incremental algorithm)
2. สามารถจัดการกับคุณลักษณะต่างๆ ที่ไม่เกี่ยวข้องได้ดี (Littlestone, 1998)