บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้กล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องในงานวิจัย ทฤษฎีที่เกี่ยวข้อง มีเนื้อหาเกี่ยวกับทฤษฎีที่ใช้ในงานวิจัยประกอบด้วย การเรียนรู้ของเครื่อง การโปรแกรมตรรกะเชิงอุปนัย และแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก ส่วนงานวิจัยที่เกี่ยวข้องเป็นงานวิจัยด้านการโปรแกรมตรรกะเชิงอุปนัย

ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในงานวิจัยนี้ประกอบด้วย การเรียนรู้ของเครื่อง การโปรแกรมตรรกะเชิงอุปนัย วิธีการแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก และวิธีการจำแนกตามกลุ่มหลัก

1. การเรียนรู้ของเครื่อง

การศึกษาด้านการเรียนรู้ของเครื่อง [Mitchell, 1997] มีจุดมุ่งหมายเพื่อทำให้โปรแกรมคอมพิวเตอร์สามารถพัฒนาความสามารถของตัวเองโดยอัตโนมัติจากตัวอย่างที่ผู้สอนป้อนให้ ตัวอย่างเช่น โปรแกรมคอมพิวเตอร์ซึ่งสามารถเรียนรู้เพื่อเล่นหมากรุก สามารถพัฒนาความสามารถในการเล่นซึ่งวัดจากเปอร์เซ็นต์ในการชนะคู่แข่งขัน โดยใช้ตัวอย่างซึ่งได้จากการเล่นกับตัวเอง หรือการสร้างโปรแกรมที่สามารถรู้จำภาพตัวเขียนอักษรได้ ซึ่งวัดความสามารถในการรู้จำจากเปอร์เซ็นต์ของภาพตัวอักษรที่รู้จำได้ถูกต้อง โดยใช้ตัวอย่างเป็นภาพตัวอักษรซึ่งผู้สอนป้อนให้

มีการนำวิธีการเรียนรู้ของเครื่องไปใช้เป็นส่วนประกอบสำคัญในการสร้างโปรแกรมเพื่อใช้งานทั่วไป รวมทั้งมีการประยุกต์ใช้กับงานด้านต่างๆ เช่น การสร้างโปรแกรมประยุกต์เพื่อประโยชน์ในด้านธุรกิจ การอนุมัติบัตรเครดิต การค้นพบความรู้จากฐานข้อมูล (Knowledge Discovery from Database) การใช้งานทั่วไป เช่น การรู้จำเสียงพูด การรู้จำภาพ ฯลฯ ตัวอย่างของการประยุกต์ใช้ซึ่งนำวิธีการเรียนรู้ของเครื่องไปใช้งานได้อย่างดี ได้แก่ การรู้จำเสียงพูด [Waibel และคณะ, 1989; Lee, 1989] การทำนายอัตราการฟื้นตัวของผู้ป่วยโรคปอดอักเสบ [Cooper และคณะ, 1997] การสืบหากรณีที่เจ้าของบัตรเครดิตชำระเงินไม่ตรงเวลา การบังคับรถโดยอัตโนมัติบนถนนหลวง [Pomerleau, 1989] หรือการสร้างกลยุทธ์ต่างๆ ในการเล่นเกม [Tesauro, 1992; 1995] ฯลฯ ในทางทฤษฎีก็เช่นกัน มีงานวิจัยจำนวนมากได้พัฒนาวิธีการเรียนรู้ของเครื่องให้มีประสิทธิภาพมากยิ่งขึ้น ตัวอย่างของการพัฒนาเหล่านี้ ได้แก่ การหาความสัมพันธ์พื้นฐานของตัวอย่างที่ป้อนให้เพื่อทำการเรียนรู้ การหาจำนวนสมมติฐานที่เหมาะสมควรนำมาพิจารณา การ

คาดหมายความผิดพลาดของสมมติฐานต่างๆ และการสร้างแบบจำลองของการเรียนรู้ของมนุษย์และสัตว์ รวมทั้งการทำความเข้าใจความสัมพันธ์ของการเรียนรู้ของมนุษย์และสัตว์กับขั้นตอนวิธีการเรียนรู้สำหรับเครื่องคอมพิวเตอร์ [Laird และคณะ, 1986; Qin และคณะ, 1992; Chi และ Bassock, 1989]

ตัวอย่างของการนำวิธีการเรียนรู้ของเครื่องไปใช้งานได้อย่างมีประสิทธิภาพ

1.1 การเรียนรู้เพื่อการรู้จำเสียงพูด

มีการใช้วิธีการเรียนรู้ของเครื่องเป็นองค์ประกอบสำคัญ ส่งผลให้ระบบที่รู้จำเสียงพูดมีประสิทธิภาพสูงขึ้น โดยนำวิธีการเรียนรู้ของเครื่องไปใช้ในรูปแบบต่างๆ เช่น ระบบ SPHINX [Lee, 1989] เรียนรู้เพื่อการรู้จำองค์ประกอบพื้นฐานของเสียงพูด (phonemes) และคำศัพท์ (words) จากสัญญาณเสียง รวมถึงการใช้นิวรอลเน็ตเวิร์กและแบบจำลองฮิดเด็นมาร์คอฟ (Hidden Markov Model) [Waibel และคณะ, 1989; Lee, 1989] เพื่อรู้จำเสียงของผู้พูดแต่ละคน

1.2 การเรียนรู้เพื่อบังคับพาหนะโดยอัตโนมัติ

ใช้การเรียนรู้ของเครื่องเพื่อควบคุมรถยนต์ด้วยเครื่องคอมพิวเตอร์ สามารถควบคุมได้อย่างถูกต้องเมื่อขับเคลื่อนอยู่บนถนนหลายแบบ โดยเรียนรู้จากตัวอย่างที่ผู้สอนเป็นผู้ควบคุมรถ ดังตัวอย่างเช่น ระบบ ALVINN [Pomerleau, 1989] สามารถขับเคลื่อนรถยนต์ได้เองด้วยความเร็ว 70 ไมล์ต่อชั่วโมง เป็นระยะทาง 90 ไมล์ บนทางหลวง ร่วมกับรถยนต์คันอื่นซึ่งมีมนุษย์เป็นผู้บังคับ

1.3 การเรียนรู้เพื่อจำแนกโครงสร้างใหม่ทางดาราศาสตร์

มีการนำวิธีการเรียนรู้ของเครื่องไปประยุกต์ใช้เพื่อค้นหาลักษณะพื้นฐานในระบบฐานข้อมูลขนาดใหญ่ เช่น องค์การบริหารการบินและอวกาศแห่งชาติ สหรัฐอเมริกา หรือ NASA ได้ใช้ขั้นตอนวิธีการเรียนรู้ต้นไม้ตัดสินใจ (Decision Tree Learning Algorithm) เรียนรู้เพื่อจำแนกวัตถุที่อยู่บนท้องฟ้าจากภาพที่มีขนาดใหญ่มาก [Fayyad และคณะ, 1995]

1.4 การเรียนรู้เล่นเกมแบ็กแกมมอน (backgammon)

โปรแกรมคอมพิวเตอร์ที่ใช้เล่นเกมซึ่งประสบความสำเร็จมากที่สุด ได้แก่ โปรแกรมเล่นเกมแบ็กแกมมอน ซึ่งใช้วิธีการเรียนรู้ของเครื่องเป็นแนวคิดพื้นฐาน ตัวอย่างโปรแกรมเหล่านี้ได้แก่ TD-GAMMON [Tesauro 1992; 1995] เป็นโปรแกรมที่มีความสามารถสูงมากในการเล่นเกมแบ็กแกมมอน สามารถหากลยุทธ์ในการเล่นโดยการฝึกเล่นกับตัวเองจนมีความสามารถเทียบเท่ากับมนุษย์ระดับแชมเปียนโลก

2. การโปรแกรมตรรกะเชิงอุปนัย

การโปรแกรมตรรกะเชิงอุปนัยหรือไอแอลพี [Mitchell, 1997] เป็นการนำวิธีการเรียนรู้ของเครื่องมาใช้ร่วมกับวิธีการโปรแกรมเชิงตรรกะ (Logic Programming) แนวคิดหลักของวิธีการนี้ยังคงเป็นไปตามแนวทางของการเรียนรู้ของเครื่อง คือ การพยายามสร้างแนวคิดเพื่ออธิบายตัวอย่างบวกและไม่อธิบายตัวอย่างลบ เพียงแต่นำวิธีการโปรแกรมเชิงตรรกะมาใช้ในการอธิบายตัวอย่าง และกลุ่มความรู้ภูมิหลังที่ป้อนให้แก่ระบบเพื่อทำการเรียนรู้ ระบบจะเรียนรู้เพื่อสร้างแนวความคิดในรูปแบบของโปรแกรมเชิงตรรกะจากตัวอย่างและความรู้ภูมิหลัง ที่ผู้สอนป้อนให้ ซึ่งแนวความคิดที่ได้จากระบบนี้สามารถใช้นำไปใช้เพื่อทำนายตัวอย่างที่ยังไม่เคยพบได้

เนื่องจากการโปรแกรมตรรกะเชิงอุปนัยเป็นวิธีการเรียนรู้แบบหนึ่งของเครื่อง ดังนั้นจุดมุ่งหมายของการโปรแกรมตรรกะเชิงอุปนัย จึงคล้ายกับจุดมุ่งหมายของการเรียนรู้ของเครื่อง นั่นคือ การพัฒนาเครื่องมือและวิธีการเพื่อสร้างสมมติฐานจากการสังเกตหรือจากตัวอย่าง และเพื่อสังเคราะห์ความรู้ใหม่จากประสบการณ์ที่ได้รับ แต่การใช้โปรแกรมเชิงตรรกะเป็นเครื่องมือในการอธิบายสมมติฐานและตัวอย่าง ทำให้การโปรแกรมตรรกะเชิงอุปนัยมีจุดเด่นเหนือวิธีการเรียนรู้ของเครื่องแบบอื่น ดังนี้

1. การใช้โปรแกรมตรรกะเป็นรูปแบบในการอธิบายความรู้ (knowledge) เนื่องจากความรู้ส่วนใหญ่ของมนุษย์สามารถอธิบายได้ในรูปของตรรกะลำดับที่หนึ่ง แต่ในวิธีการเรียนรู้ของเครื่องส่วนใหญ่มักจะอธิบายความรู้เหล่านี้โดยการใช้ตรรกศาสตร์ประพจน์ ทำให้ไม่สามารถอธิบายความรู้ของมนุษย์ได้เท่าที่ควร นอกจากนี้ข้อได้เปรียบสำคัญอีกประการหนึ่งของการใช้โปรแกรมเชิงตรรกะ คือ การใช้โปรแกรมตรรกะเพื่ออธิบายแนวความคิดที่สังเคราะห์ได้จากตัวอย่างสามารถอธิบายแนวความคิดได้ซับซ้อนมากยิ่งขึ้น ซึ่งการอธิบายด้วยตรรกศาสตร์ประพจน์แต่เพียงอย่างเดียว จะไม่สามารถครอบคลุมแนวคิดได้ทั้งหมด
2. สามารถใช้ความรู้ภูมิหลังในกระบวนการเรียนรู้ได้ง่ายกว่าการเรียนรู้ของเครื่องแบบอื่น ที่ไม่ใช้การโปรแกรมเชิงตรรกะ

หลักการสำคัญของการโปรแกรมตรรกะเชิงอุปนัย คือ การสร้างแนวคิด หรือกฎ หรือสมมติฐาน (h) ที่สามารถอธิบายเซตของตัวอย่างบวก (E) ได้ โดยใช้ประกอบกับความรู้ภูมิหลัง (B) ดังความสัมพันธ์

$$B \wedge h \vdash E$$

จุดมุ่งหมายสำคัญของวิธีการโปรแกรมตรรกะเชิงอุปนัยคือ การสร้างสมมติฐานซึ่งเมื่อรวมกับความรู้ภูมิหลังแล้วสามารถชี้บ่งเป็นนัยถึง (imply) ตัวอย่างได้ โดยสร้างสมมติฐานจากกลุ่มตัวอย่างและความรู้ภูมิหลังที่ป้อนให้ ทฤษฎีที่เกี่ยวข้องในการสร้างสมมติฐานมีหลายวิธี เช่น วิธีการอนุมานเชิงอุปนัย (Inductive Inference) ไออาร์ (Inverse Resolution, IR) อาร์แอลจีจี (Relative Least General Generalization,

RLGG) ไออี (Inverse Entailment, IE) ฯลฯ ในที่นี้ได้นำเสนอทฤษฎีที่เกี่ยวข้องดังนี้ คือ วิธีการอนุมานเชิงอุปนัย วิธีการไออาร์ และวิธีการอาร์แอลจีจี

2.1 วิธีการอนุมานโดยอุปนัย [Mitchell, 1997]

ในการสร้างสมมติฐานขึ้นจากตัวอย่างและความรู้ภูมิหลัง จำเป็นต้องสร้างข้อกำหนดขึ้น ดังนี้ "กำหนดให้ D เป็นเซตของตัวอย่างที่ใช้สอนเพื่อเรียนรู้โดยมีรูปแบบเป็น $<x_i, f(x_i)>$ เมื่อ x_i คือ ตัวอย่างที่ i และ $f(x_i)$ เป็นค่าที่ต้องการ (target value) และ B คือความรู้ภูมิหลัง คำนิยามของการเรียนรู้คือการสร้างสมมติฐาน h ซึ่งค่า $f(x_i)$ ของแต่ละตัวอย่าง x_i แต่ละตัวสามารถชี้บ่งเป็นนัยถึงได้ด้วยสมมติฐาน h ลักษณะของ x_i และความรู้ภูมิหลัง B" ดังนิพจน์ที่ (1)

$$(\forall <x_i, f(x_i)> \in D)\ (B \wedge h \wedge x_i) \vdash f(x_i) \qquad (1)$$

นิพจน์ $X \vdash Y$ หมายความว่า "Y สามารถชี้บ่งเป็นนัยถึงได้โดยใช้ X" ซึ่งในนิพจน์ (1) หมายความว่า สำหรับทุกตัวอย่าง x_i ค่าของ $f(x_i)$ สามารถชี้บ่งเป็นนัยถึงได้โดยใช้ความรู้ภูมิหลัง B สมมติฐาน h และตัวอย่าง x_i

ตัวอย่างเช่น ในการสร้างแนวความคิดของความสัมพันธ์ระหว่างคนสองคน <U,V> เมื่อ U เป็นบุตรของ V แทนด้วยเพรดิเคต child(U,V) สมมติให้มีตัวอย่างบวกเพียงตัวอย่างเดียวคือ child(sharon,bob) โดยมีเพรดิเคตที่ใช้อธิบายตัวอย่างนี้คือ male(bob), female(sharon) และ father(bob,sharon) และมีความรู้ภูมิหลังเป็น parent(U,V) ← father(U,V) จะสามารถอธิบายโดยใช้นิพจน์ (1) ได้ดังนี้

x_i : *male(bob), female(sharon), father(bob, sharon)*

$f(x_i)$: *child(sharon, bob)*

B : $parent(U,V) \leftarrow father(U,V)$

ในกรณีนี้สามารถสร้างสมมติฐานได้หลายแบบซึ่งสามารถครอบคลุมตัวอย่างได้ ตามคำนิยามในนิพจน์ (1) $(B \wedge h \wedge x_i) \vdash f(x_i)$ ซึ่งในที่นี้จะยกตัวอย่างสมมติฐาน 2 ข้อ ดังนี้

h_1 : $child(U,V) \leftarrow father(V,U)$

h_2 : $child(U,V) \leftarrow parent(V,U)$

จะเห็นว่าเพรดิเคตเป้าหมาย *child(sharon,bob)* ถูกครอบคลุมด้วยเงื่อนไข $h_1 \wedge x_i$ โดยไม่ต้องใช้ความรู้ภูมิหลัง B แต่สำหรับในกรณีของสมมติฐาน h_2 เมื่อนำ h_2 ไปรวมกับ x_i โดยไม่ใช้ความรู้ภูมิหลัง B จะพบว่าไม่ครอบคลุมเพรดิเคต *child(sharon,bob)* จำเป็นต้องใช้ความรู้ภูมิหลัง B ประกอบด้วย จึงจะทำให้เป็นไปตามเงื่อนไข $(B \wedge h \wedge x_i) \vdash f(x_i)$ ตัวอย่างนี้แสดงให้เห็นถึงบทบาทสำคัญของความรู้ภูมิหลังในการ

ขยายเซตของสมมติฐานจากตัวอย่าง ดังเช่นในสมมติฐาน h_2 จะต้องใช้ความรู้ภูมิหลังร่วมด้วยจึงสามารถครอบคลุมตัวอย่างได้

ในการค้นหาสมมติฐานตามความสัมพันธ์ $(\forall \langle x, f(x)\rangle \in D)\ (B \wedge h \wedge x) \vdash f(x)$ มีลักษณะที่น่าสนใจหลายประการ เช่น

- ความสัมพันธ์นี้สนับสนุนคำนิยามพื้นฐานของการเรียนรู้ คือ การค้นหาแนวคิดทั่วไปซึ่งสามารถใช้อธิบายเซตของตัวอย่างที่ใช้ในการสอนได้
- เมื่อใช้ความรู้ภูมิหลังประกอบ สามารถสร้างสมมติฐานที่ครอบคลุมตัวอย่างได้มากขึ้น
- ความรู้ภูมิหลังสามารถช่วยให้กระบวนการเรียนรู้ค้นหาสมมติฐานได้ง่ายขึ้น

2.2 วิธีการไออาร์ [Mitchell, 1997]

กฎรีโซลูชัน (resolution rule) เป็นวิธีการหนึ่งที่ได้รับความนิยมในการนำมาพิจารณาตรรกะลำดับที่หนึ่ง เพื่อสร้างสมมติฐานจากตัวอย่างบวก ตัวอย่างลบ และความรู้ภูมิหลังได้ มีการนำไปใช้เป็นแนวคิดพื้นฐานในระบบ CIGOL [Muggleton และ Buntine; 1998]

กำหนดให้ L เป็นประพจน์ P และ R เป็นอนุประโยคที่เกิดจากประพจน์หลายประพจน์มารวมกัน จะได้กฎรีโซลูชัน คือ

$$\frac{\begin{array}{rcl} P & \vee & L \\ \neg L & \vee & R \end{array}}{\begin{array}{rcl} P & \vee & R \end{array}}$$

หมายความว่า สามารถสรุปอนุประโยค $P \vee R$ ได้จากอนุประโยค $P \vee L$ และอนุประโยค $\neg L \vee R$ ซึ่งสามารถนำไปใช้ในการสร้างสมมติฐานที่ครอบคลุมตัวอย่างได้ ดังนี้

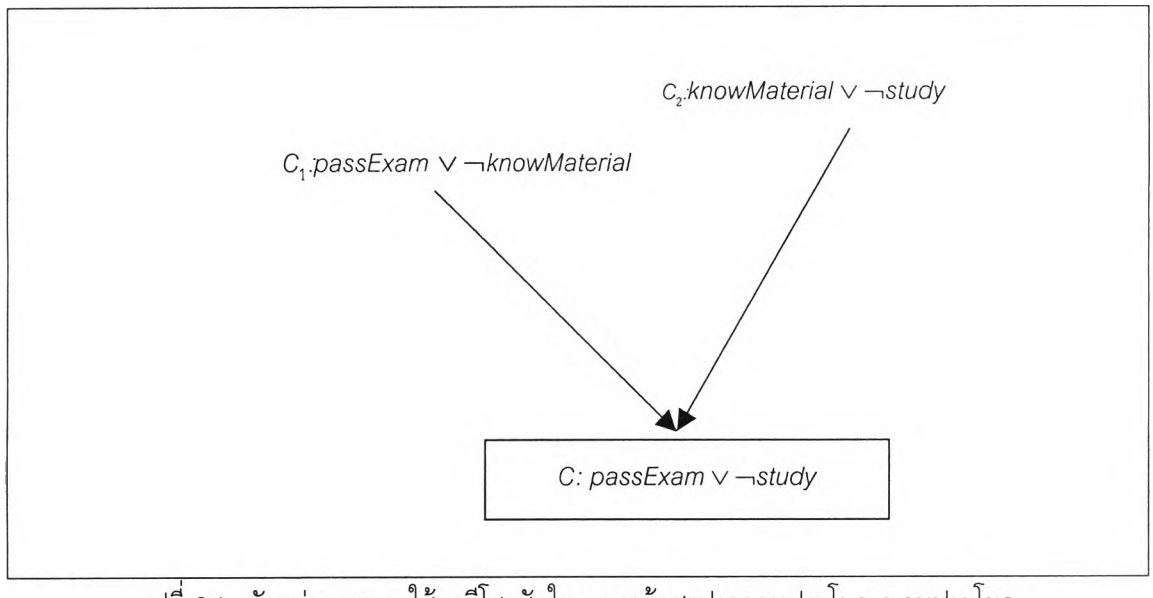
1. กำหนดให้มีอนุประโยค C_1 และ C_2 ให้หาสัญพจน์ L จากอนุประโยค C_1 ซึ่งมีสัญพจน์ $\neg L$ ปรากฏอยู่ในอนุประโยค C_2
2. สร้างอนุประโยคสรุป C โดยรวมสัญพจน์ทุกตัวจาก C_1 และ C_2 ยกเว้นสัญพจน์ L และ $\neg L$ ดังสมการ

$$C = (C_1 - \{L\}) \cup (C_2 - \{\neg L\})$$

การใช้กฎรีโซลูชันในการสรุปสมมติฐานจากตัวอย่าง โดยการใช้อนุประโยคที่มีอยู่ C_1 และ C_2 ให้กำหนดสัญพจน์ L ซึ่งเป็นสัญพจน์แบบบวก (positive literal) ในอนุประโยคหนึ่ง และเป็นสัญพจน์แบบลบ (negative literal) ในอนุประโยคที่เหลือ จะสามารถสร้างอนุประโยคสรุป C ได้ ดังรูปที่ 2.1 กำหนดให้มีอนุประโยค C_1 และ C_2 ในขั้นแรกต้องกำหนดสัญพจน์ $L = \neg knowMaterial$ ซึ่งปรากฏอยู่ในอนุประโยค

C_1 และสัญพจน์ที่ตรงข้ามคือ $\neg(\neg knowMaterial) = knowMaterial$ ซึ่งปรากฏอยู่ในอนุประโยค C_2 ดังนั้นผลสรุปของสองอนุประโยคนี้ คือ การรวมกันของสัญพจน์ $C_1 - \{L\} = passExam$ และ $C_2 - \{\neg L\} = \neg study$

เมื่อสามารถนำกฎรีโซลูชันมาใช้เพื่อหาข้อสรุปจากอนุประโยคได้ ก็เป็นการง่ายที่จะสร้างตัวดำเนินการไออาร์ $Q(C,C_i)$ ซึ่งสามารถทำการอนุมานโดยอุปนัยได้ โดยการสร้างอนุประโยคเริ่มแรก C_2 จากข้อสรุป C และอนุประโยคเริ่มแรกอีกประโยคหนึ่งคือ C_1 ตัวอย่างเช่น มีข้อสรุป $C = A \vee B$ และอนุประโยคเริ่มต้น $C_1 = B \vee D$ จุดมุ่งหมายคือ การพยายามสร้างอนุประโยค C_2 ซึ่ง $C_1 \wedge C_2 \vdash C$



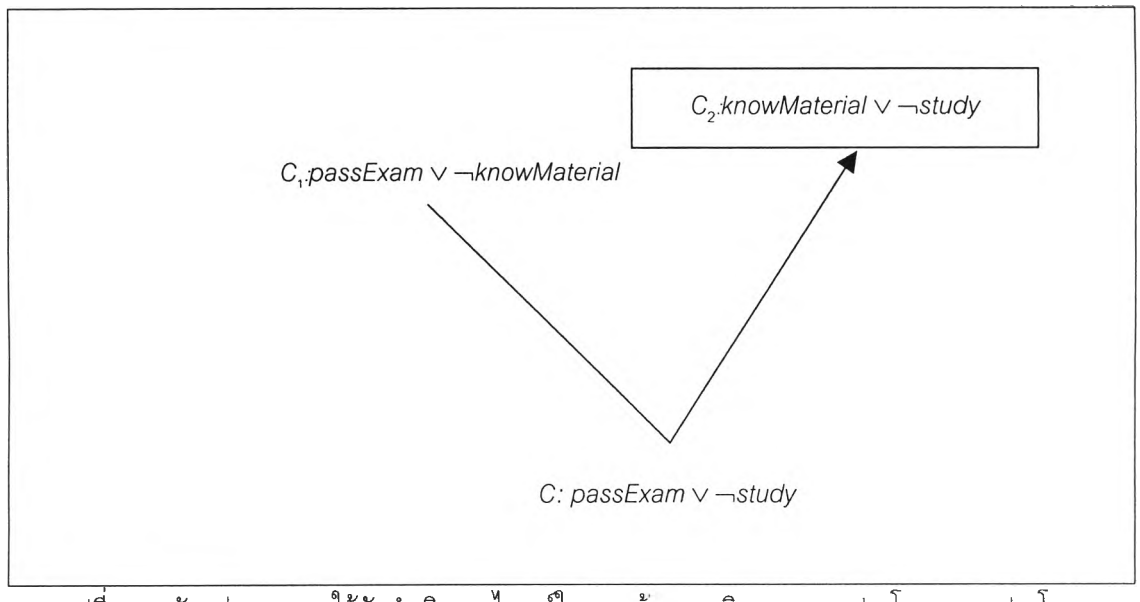
รูปที่ 2.1 ตัวอย่างของการใช้กฎรีโซลูชันในการหาข้อสรุปจากอนุประโยค 2 อนุประโยค

วิธีการ ขั้นแรก จากนิยามของกฎรีโซลูชัน สัญพจน์ใดๆ ที่ปรากฏในอนุประโยค C แต่ไม่ปรากฏในอนุประโยค C_1 จะต้องปรากฏใน C_2 ในตัวอย่างก่อนหน้านี้ ในอนุประโยค C_2 จะต้องมีอนุประโยค A ขั้นที่สอง สัญพจน์ที่ปรากฏใน C_1 แต่ไม่ปรากฏใน C จะต้องเป็นสัญพจน์ที่ถูกดึงออกโดยกฎรีโซลูชัน ดังนั้นนิเสธของสัญพจน์นี้จะต้องปรากฏใน C_2 จากตัวอย่างเดิม อนุประโยค C_2 จึงต้องมีสัญพจน์ $\neg D$ ปรากฏอยู่ด้วย สุดท้ายจะได้สัญพจน์ $C_2 = A \vee \neg D$ ซึ่งสามารถสรุปการดำเนินการไออาร์ได้ดังนี้

1. กำหนดอนุประโยค C_1 และ C ให้หาสัญพจน์ L จากอนุประโยค C_1 แต่ไม่ปรากฏอยู่ในอนุประโยค C
2. สร้างอนุประโยค C_2 ดังสมการ

$$C_2 = (C - (C_1 - \{L\})) \cup \{\neg L\}$$

ตัวอย่างในรูป 2.2 แสดงการใช้ตัวดำเนินการไออาร์ ในการสร้างสมมติฐาน $C_2 = knowMaterial \vee \neg study$ จากอนุประโยค $C = passExam \vee \neg study$ และอนุประโยค $C_1 = passExam \vee \neg knowMaterial$



รูปที่ 2.2 ตัวอย่างของการใช้ตัวดำเนินการไออาร์ในการสร้างสมมติฐานจากอนุประโยค 2 อนุประโยค

2.3 การรีโซลูชันลำดับที่หนึ่ง (First-Order Resolution) [Mitchell, 1997]

ก่อนหน้านี้ได้นำเสนอการทำรีโซลูชันเมื่อนำไปใช้กับประพจน์ เพื่อหาอนุประโยคที่สามจากอนุประโยคสองอนุประโยคที่กำหนดให้ เมื่อนำมาประยุกต์ใช้กับตรรกะลำดับที่หนึ่ง เพื่อสร้างการรีโซลูชันลำดับที่หนึ่ง จะใช้พื้นฐานของตัวแทนรวม (unifying substitution)

กำหนดตัวแทน $\theta = \{X/bob, Y/Z\}$ หมายความว่า ใช้ค่าคงที่ *bob* แทนตัวแปร X และใช้ตัวแปร Z แทนตัวแปร Y และใช้เครื่องหมาย $W\theta$ แทนผลการแทนโดยตัวแทน θ ในนิพจน์ W ตัวอย่างเช่น ถ้า L เป็นสัญพจน์ *father(X,bill)* และ $\theta = \{X/bob, Y/Z\}$ จะได้ $W\theta$ = *father(bob,bill)*

ตัวแทนรวม θ สำหรับสัญพจน์สองสัญพจน์ L_1 และ L_2 คือ ตัวแทนรวมซึ่งเมื่อนำไปแทนตัวแปรแล้วจะทำให้สัญพจน์ $L_1\theta = L_2\theta$ ตัวอย่างเช่น ถ้า L_1 = *father*(X,Y), L_2 = *father*$(bill,Z)$, และ $\theta = \{X/bill, Z/Y\}$, แล้ว θ เป็นตัวแทนรวมสำหรับ L_1 และ L_2 เพราะ $L_1\theta = L_2\theta$ = *father*$(bill,Y)$ ซึ่งกฎรีโซลูชันลำดับที่หนึ่งจะสร้างอนุประโยคสรุป C ด้วยวิธีการดังนี้

1. หาสัญพจน์ L_1 จากอนุประโยค C_1 สัญพจน์ L_2 จากอนุประโยค C_2 และตัวแทน θ ซึ่งทำให้ $L_1\theta = L_2\theta$
2. สร้างอนุประโยคสรุปโดยรวมทุกสัญพจน์จาก $C_1\theta = C_2\theta$ ยกเว้นสัญพจน์ $L_1\theta$ และ $\neg L_2\theta$ ดังสมการ

$$C = (C_1 - \{L_1\})\theta \cup (C_2 - \{L_2\})\theta$$

2.4 วิธีการไออาร์ในกรณีลำดับที่หนึ่ง (Inverse Resolution: First-Order Case) [Mitchell, 1997]

จากหัวข้อที่แล้วในการรีโซลูชันลำดับที่หนึ่งจะได้

$$C = (C_1 - \{L_1\})\theta_1 \cup (C_2 - \{L_2\})\theta_2$$

กำหนดให้ในอนุประโยค C_2 ไม่มีสัญพจน์ที่เหมือนกับสัญพจน์ในอนุประโยค C_1 เราสามารถเปลี่ยนสมการข้างต้นเป็น

$$C - (C_1 - \{L_1\})\theta_1 = (C_2 - \{L_2\})\theta_2$$

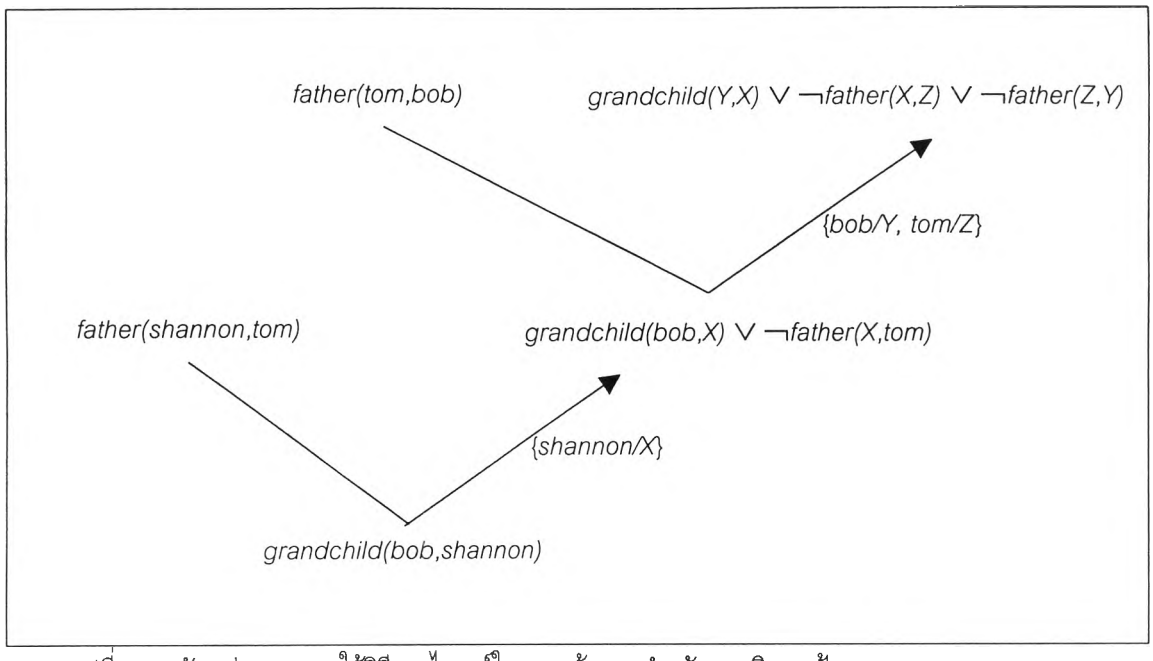
สุดท้ายเราใช้นิยามของกฎการรีโซลูชัน $L_2 = \neg L_1\theta_1\theta_2^{-1}$ เราจะสามารถสรุปอนุประโยค C_2 เป็นกฎไออาร์ได้ดังนี้

$$C_2 = (C - (C_1 - \{L_1\})\theta_1)\theta_2^{-1} \cup \{\neg L_1\theta_1\theta_2^{-1}\}$$

ตัวอย่างในรูปที่ 2.3 ต้องการเรียนรู้กฎสำหรับเพรดิเคตเป้าหมาย *grandchild(Y,X)* โดยกำหนดข้อมูลที่ใช้ทำการเรียนรู้ D = *grandchild(bob,shannon)* และข้อมูลซึ่งเป็นความรู้ภูมิหลัง B = {*father(shannon,tom)*, *father(tom,bob)*} พิจารณาส่วนล่างที่สุดของรูปที่ 2.3 ในที่นี้ให้ข้อสรุป C เป็นตัวอย่างที่ใช้ในการเรียนรู้ *grandchild(bob,shannon)* และเลือกอนุประโยค C_1 = *father(shannon,tom)* จากกลุ่มความรู้ภูมิหลังในการใช้ตัวดำเนินการไออาร์สำหรับสัญพจน์ L_1 คือ *father(shannon,tom)* เลือกตัวแทนย้อนกลับ θ_1^{-1} = {} และ θ_2^{-1} = {*shannon/X*} ในกรณีนี้เราสามารถสร้างอนุประโยค C_2 เป็นการรวมกันของอนุประโยค $(C - (C_1 - \{L_1\})\theta_1)\theta_2^{-1} = (C\theta_1)\theta_2^{-1}$ = *grandchild(bob,X)* และอนุประโยค $\{\neg L_1\theta_1\theta_2^{-1}\}$ = ¬*father(X,tom)* ดังนั้นผลลัพธ์ คือ อนุประโยค *grandchild(bob,X)* ∨ ¬*father(X,tom)* ซึ่งมีค่าเท่ากับอนุประโยค *grandchild(bob,X)* ← *father(X,tom)* ด้วยวิธีเดียวกันนี้ เราสามารถสร้างอนุประโยคสุดท้าย *grandchild(Y,X)* ← *father(X,Z)* ∧ *father(Z,Y)* ได้

2.5 วิธีการอาร์แอลจีจี [Muggleton และ Feng, 1992]

วิธีการอาร์แอลจีจี เป็นอีกวิธีการหนึ่งนอกเหนือจากวิธีการไออาร์ซึ่งมีผู้นำมาใช้ในการสร้างอนุประโยคที่สามารถครอบคลุมตัวอย่างในระบบไอแอลพี โดยใช้แนวคิดของ Plotkin [Plotkin, 1971] สำหรับวิธีการอาร์แอลจีจี คือ การแทนที่การค้นหาด้วยกระบวนการสร้างอนุโยคซึ่งครอบคลุมเซตของตัวอย่าง



รูปที่ 2.3 ตัวอย่างของการใช้วิธีการไออาร์ในการสร้างกฎสำหรับเพรดิเคตเป้าหมาย *grandchild(Y,X)*

ก่อนอธิบายถึงวิธีการอาร์แอลจีจี Plotkin ให้คำนิยามของคำว่า θ-*subsume* ว่า อนุประโยค C θ-*subsume* อนุประโยค D หรือ $C \preceq D$ ก็ต่อเมื่อ ถ้าสามารถหาการแทนที่ (substitution) θ ซึ่งทำให้ $C\theta \subseteq D$ จากนั้น Plotkin ได้อธิบายถึงวิธีการแอลจีจี (least general generalisation: lgg) ว่าเป็นการหาขอบเขตล่างซึ่งมีนัยทั่วไป (general) น้อยที่สุดในแลตทิซ (lattice) ที่ถูกสร้างโดยความสัมพันธ์ $\preceq$ เราสามารถหาแอลจีจีของพจน์ (term) 2 พจน์ p(g(a),a) และ p(g(b),b) ได้เป็น p(g(X),X) ซึ่งถ้าเราใช้ตัวแทน {X/a} กับพจน์ p(g(X),X) จะได้พจน์ p(g(a),a) และเมื่อใช้ตัวแทน {X/b} จะได้พจน์ p(g(b),b)

สำหรับการหาแอลจีจีของอนุประโยค 2 อนุประโยคใดๆ มีวิธีการดังนี้

1. หาแอลจีจีของสัญพจน์ในส่วนหัวของอนุประโยคเพื่อเป็นส่วนหัวของอนุประโยคใหม่
2. หาการรวม (combination) ทุกแบบที่เป็นไปได้ของแอลจีจีในส่วนคำอธิบาย (body) ของกฎเพื่อสร้างส่วนคำอธิบายของอนุประโยคใหม่
3. ถ้ามีสัญพจน์ใดปรากฏในอนุประโยคเดียวไม่ปรากฏอยู่ในอีกอนุประโยคหนึ่ง ไม่ต้องนำอนุประโยคนั้นไปไว้ในอนุประโยคใหม่

ตัวอย่างเช่น การหาแอลจีจีของอนุประโยค 2 อนุประโยค ต่อไปนี้

q(g(a)) ← p(g(a, b), r(h(b), c), r(b, d).

q(x) ← p(x, y), r(y, z), r(h(y), z), s(g(x), h(y)).

แอลจีจีของอนุประโยคทั้งสอง คือ

q(x) ← p(X, Y), r(Z, U), r(V, W), r(h(Y), U), r(Y, W).

โดยมีตัวแทนเป็น

{ X/{g(a), x}, Y/{b, y}, Z/{h(b), y}, U/{c, z}, V/{b, h(y)}, W/{d, z} }

เมื่อสามารถหาแอลจีจีของสองอนุประโยคได้แล้ว เราก็สามารถทำอาร์แอลจีจีได้ โดยกำหนดให้ $\mathcal{K}$ เป็นเซตของความรู้ภูมิหลัง e_1 และ e_2 เป็นตัวอย่างบวก ($e_1, e_2 \in \mathcal{E}^+$) ซึ่ง $\mathcal{K} \not\vdash e_1$ และ $\mathcal{K} \not\vdash e_2$ อนุประโยค C เป็นแอลจีจีของตัวอย่าง e_1 และ e_2 สัมพันธ์กับ $\mathcal{K}$ ก็ต่อเมื่อ C เป็นอนุประโยคที่มีนัยทั่วไปน้อยที่สุดภายในขอบเขตแห่งแลตทิซของ *θ-subsumption* ซึ่ง $\mathcal{K} \wedge C \vdash e_1 \wedge e_2$ โดยที่อนุประโยค C ถูกใช้เพียงครั้งเดียวในการหา e_1 และ e_2 วิธีการสร้างอาร์แอลจีจีสามารถพิสูจน์ได้ดังนี้

$$
\begin{aligned}
\mathcal{K} \wedge C &\vdash e_1 \\
C &\vdash \mathcal{K} \rightarrow e_1 \\
&\vdash C \rightarrow (\mathcal{K} \rightarrow e_1) \\
&\vdash C \rightarrow (\neg \mathcal{K} \vee e_1) \\
&\vdash C \rightarrow ((\neg (a_1 \wedge a_2 \wedge \ldots) \vee e_1) \qquad (2) \\
&\vdash C \rightarrow ((\neg a_1 \vee \neg a_2 \vee \ldots) \vee e_1)
\end{aligned}
$$

ในนิพจน์ที่ (2) เป็นการแทน $\mathcal{K}$ ด้วยกลุ่มความรู้ภูมิหลัง $a_1 \wedge a_2 \wedge \ldots$ ซึ่งสามารถแทนได้หลายรูปแบบด้วยกัน ในขณะเดียวกันเราสามารถหา e_2 ได้เช่นเดียวกับ e_1 จะได้

$$\vdash C \rightarrow ((\neg a_1 \vee \neg a_2 \vee \ldots) \vee e_2)$$

ถ้ากำหนดให้ C_1 และ C_2 เป็น $((\neg a_1 \vee \neg a_2 \vee \ldots) \vee e_1)$ และ $\vdash C \rightarrow ((\neg a_1 \vee \neg a_2 \vee \ldots) \vee e_2)$ ตามลำดับ จะได้

$$\vdash C \rightarrow \mathrm{lgg}(C_1, C_2)$$

ตัวอย่างเช่น หากมีกลุ่มความรู้ภูมิหลังเป็นดังนี้

```
partition(X,[],[],[]).
partition(X,[Head|Tail],[Head|Sublist1],Sublist2) ←
        lte(Head,X),
        partition(X,Tail,Sublist1,Sublist2).
partition(X,[Head|Tail],Sublist1,[Head|Sublist2) ←
        gt(Head,X),
        partition(X,Tail,Sublist1,Sublist2).
append([],List,List).
append([Head|Tail],List,[Head|Rest]) ←
        append(Tail,List,Rest).
```

lte(0,0).

lte(0,1).

...

gt(1,0).

gt(2,0).

gt(2,1).

...

สมมติให้ e_1 = qsort([1],[1]) และ e_2 = qsort([2,4,3,1,0],[0,1,2,3,4])

C_1 = qsort([1],[1]) ←

append([],[1],[1]),

append([0,1],[2,3,4],[0,1,2,3,4]), ...

partition(1,[],[],[]),

partition(2,[4,3,1,0],[1,0],[4,3]), ...

qsort([],[]),

qsort([1,0],[0,1]),

qsort([4,3],[3,4]), ...

C_2 = qsort([2,4,3,1,0],[0,1,2,3,4]) ←

append([],[1],[1]),

append([0,1],[2,3,4],[0,1,2,3,4]), ...

partition(1,[],[],[]),

partition(2,[4,3,1,0],[1,0],[4,3]), ...

qsort([],[]),

qsort([1,0],[0,1]),

qsort([4,3],[3,4]), ...

จะได้ C = lgg(C_1, C_2)

= qsort([A|B],[C|D]) ←

append(E,[A|F],[C|D]),

append([],[1],[1]),

append(G,[H|I],[J|K]),

append([0,1],[2,3,4],[0,1,2,3,4]),

partition(A,B,L,M),

```
partition(1,[],[],[]),
partition(H,N,O,P),
partition(2,[4,3,1,0],[1,0],[4,3]),
qsort(L,E),
qsort([],[]),
qsort(O,G),
qsort([1,0],[0,1]),
qsort(M,F),
qsort(P,I),
qsort([4,3],[3,4]) ...
```

3. แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก (Backpropagation Neural Network)

วิธีการเรียนรู้โดยใช้นิวรอลเน็ตเวิร์ก สามารถเรียนรู้ได้ในปัญหาหลายแบบ ทั้งปัญหาที่เป็นฟังก์ชันจำนวนจริง (real-valued) ฟังก์ชันไม่ต่อเนื่อง (discrete-valued) รวมทั้งฟังก์ชันที่เป็นค่าเวกเตอร์ (vector-valued) จากตัวอย่างที่ผู้สอนป้อนให้

แนวคิดในการสร้างนิวรอลเน็ตเวิร์กเกิดจากระบบสมองของมนุษย์ โดยจำลองนิวรอน (neuron) และการเชื่อมต่อของนิวรอนภายในสมอง เป็นองค์ประกอบพื้นฐานและการเชื่อมต่อขององค์ประกอบเหล่านั้นตามลำดับ องค์ประกอบเหล่านี้สามารถรับอินพุตเป็นค่าจำนวนจริง ซึ่งบางครั้งอินพุตเหล่านี้อาจเป็นเอาต์พุตจากนิวรอนอื่นก็ได้ จากนั้นจึงสร้างเอาต์พุตเพื่อส่งออกไปยังนิวรอนอื่นๆ อีก

องค์ประกอบพื้นฐานของนิวรอลเน็ตเวิร์กมีได้หลายแบบ เช่น เพอร์เซปตรอน (perceptron) ไบโพลาร์ (bipolar) และ องค์ประกอบซิกมอยด์ (sigmoid unit) ฯลฯ เพอร์เซปตรอนทำหน้าที่รับค่าเวกเตอร์ซึ่งประกอบด้วยอินพุตจำนวนจริงแล้วคำนวณผลรวมเชิงเส้น (linear combination) ของอินพุตเหล่านี้ จากนั้นให้เอาต์พุตเป็น 1 ถ้าผลรวมนั้นมากกว่าค่าที่กำหนดไว้ และให้เอาต์พุตเป็น -1 ในกรณีที่ได้ผลรวมไม่เกินค่าที่กำหนด เราสามารถมองเพอร์เซปตรอนเป็นดิซิชันเซอร์เฟสที่มีลักษณะเป็นระนาบเกิน (hyperplane decision surface) ใน n มิติได้ โดยเพอร์เซปตรอนให้เอาต์พุตเป็น 1 สำหรับตัวอย่างซึ่งอยู่บนด้านหนึ่ง และให้เอาต์พุตเป็น -1 เมื่อตัวอย่างนั้นอยู่ที่ด้านตรงข้าม เซตของตัวอย่างบางเซตสามารถแยกตัวอย่างบวกและตัวอย่างลบออกจากกันโดยระนาบเกินใดๆ ได้ เราเรียกเซตของตัวอย่างที่มีลักษณะแบบนี้ว่าเซตตัวอย่างที่สามารถแยกได้แบบเชิงเส้น (linearly separable sets of examples)

โดยปกติแล้วเพอร์เซปตรอนเดี่ยวสามารถใช้แสดงฟังก์ชันบูลีน (Boolean Function) ได้เป็นจำนวนมาก โดยกำหนดให้ค่าของบูลีนที่เป็นจริงแทนด้วย 1 และค่าที่เป็นเท็จแทนด้วยค่า -1 ตัวอย่างเช่น ฟังก์ชัน AND, OR, NOR, NAND ฯลฯ แต่ก็มีบางฟังก์ชันบูลีนที่ไม่สามารถแทนด้วยเพอร์เซปตรอนเดี่ยวได้ จำเป็น

ต้องใช้เพอร์เซปตรอนมาต่อกันให้มีลักษณะเป็นหลายชั้น เพื่อให้สามารถแยกตัวอย่างเหล่านี้ได้ จึงมีการสร้างนิวรอลเน็ตเวิร์กแบบหลายชั้นขึ้น

แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กแบบหลายชั้น (Multilayer Backpropagation Neural Network) เป็นขั้นตอนวิธีแบบหนึ่งของนิวรอลเน็ตเวิร์ก มีความสามารถในการสร้างดิซิชันเซอร์เฟสแบบไม่เป็นเชิงเส้น (nonlinear decision surface) ตัวอย่างของการนำนิวรอลเน็ตเวิร์กแบบหลายชั้นไปใช้งานได้แก่ การจำแนกเสียงสระ 10 เสียง [Huang และคณะ, 1988] ทั้งหมดอยู่ในรูปแบบของคำศัพท์ "h_d" (เช่น "hid", "had", "head", "hood", ฯลฯ) มีสัญญาณอินพุตเป็นตัวเลขพารามิเตอร์ 2 ตัว จากการวิเคราะห์สเปกตรัม (spectral analysis) ของเสียง ทำให้สามารถสร้างดิซิชันเซอร์เฟสได้ใน 2 มิติ ซึ่งสามารถแบ่งแยกตัวอย่างได้ดีกว่าดิซิชันเซอร์เฟสแบบเชิงเส้น (linear decision surface)

แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กต่างจากเพอร์เซปตรอนที่องค์ประกอบย่อยของนิวรอลเน็ตเวิร์ก ซึ่งเป็นองค์ประกอบที่สามารถทำงานกับฟังก์ชันแบบไม่เชิงเส้น (nonlinear function) โดยใช้องค์ประกอบซิกมอยด์ดังรูปที่ 2.4

องค์ประกอบซิกมอยด์จะทำงานโดยหาผลคูณเวกเตอร์ของอินพุตเวกเตอร์กับค่าน้ำหนักของเส้นเชื่อมระหว่างอินพุตนิวรอนกับนิวรอนในชั้นถัดไป จากนั้นนำผลที่ได้ผ่านตัวกรอง (threshold) เอาต์พุตขององค์ประกอบซิกมอยด์จะเป็นฟังก์ชันต่อเนื่อง (Continuous Function) ที่มีค่าเข้าใกล้ 0 หรือ 1 ในขณะที่เอาต์พุตของเพอร์เซปตรอนธรรมดาจะให้ค่าเป็นค่าไม่ต่อเนื่อง ฟังก์ชันซิกมอยด์มีคำนิยามเป็นดังนี้

$$o = \sigma(\vec{w}.\ \vec{x})$$

โดย

$$\sigma(y) = \frac{1}{1+e^{-y}}$$

เมื่อ

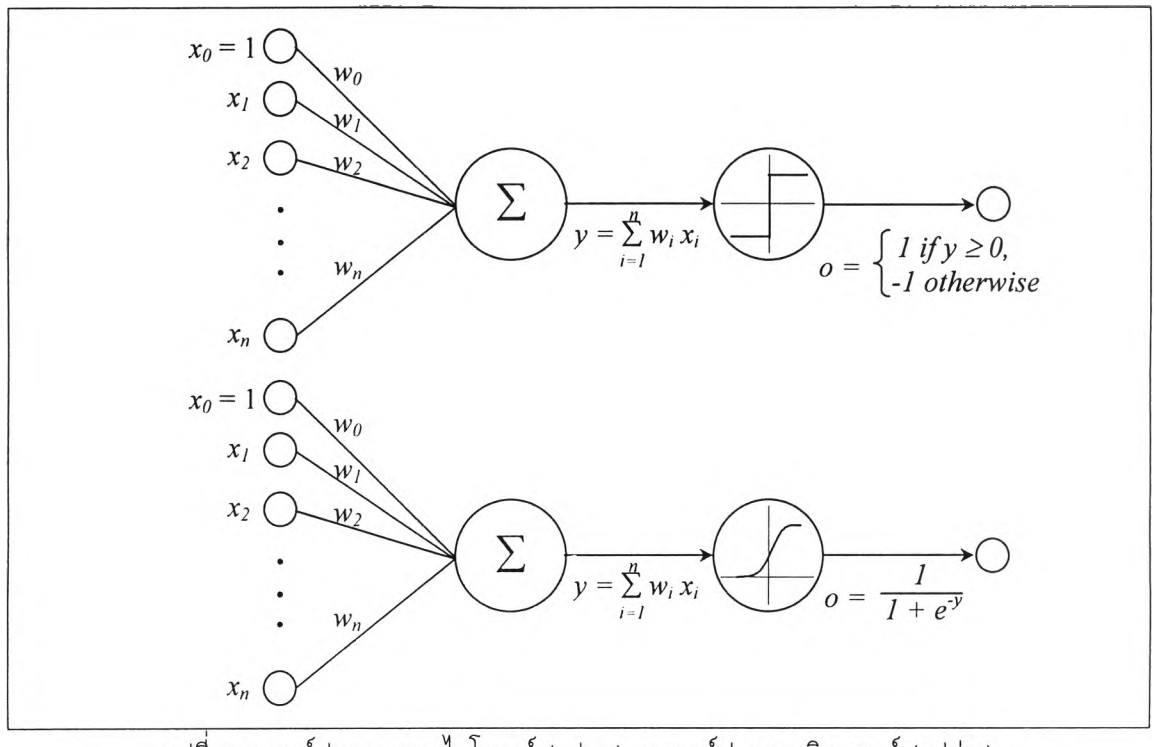
o - เอาต์พุต

$\vec{x}$ - อินพุต

$\vec{w}$ - ค่าน้ำหนักของอินพุตนั้นๆ

σ - ฟังก์ชันซิกมอยด์ (sigmoid function) ซึ่งให้ค่าเอาต์พุตระหว่าง 0 และ 1

ขั้นตอนวิธีแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก [Mitchell, 1992] ทำการเรียนรู้เพื่อปรับค่าน้ำหนักสำหรับนิวรอลเน็ตเวิร์กแบบหลายชั้น โดยใช้วิธีการเกรเดียนต์เดสเซนต์ (gradient descent) เพื่อลดกำลังสองของค่าผิดพลาด (squared error) ระหว่างเอาต์พุตที่ได้จากเน็ตเวิร์กและค่าเป้าหมายของอินพุตเหล่านี้ให้น้อยที่สุด โดยมีขั้นตอนวิธีสำหรับการปรับค่าน้ำหนักในแบบแบ็กพรอพาเกชันดังนี้



รูปที่ 2.4 องค์ประกอบของไบโพลาร์ (รูปบน) และองค์ประกอบซิกมอยด์ (รูปล่าง)

กำหนดให้ตัวอย่างที่ใช้ในการเรียนรู้แต่ละตัวอย่างอยู่ในรูป $(\vec{x}, \vec{t})$ เมื่อ $\vec{x}$ เป็นอินพุตเวกเตอร์ของนิวรอลเน็ตเวิร์ก และ $\vec{t}$ เป็นเวกเตอร์เป้าหมายของนิวรอลเน็ตเวิร์ก η เป็นค่าอัตราการเรียนรู้ (learning rate)

อินพุตขององค์ประกอบ j ซึ่งมาจากองค์ประกอบ i แทนด้วย x_{ji} และค่าน้ำหนักขององค์ประกอบ j ซึ่งมาจากองค์ประกอบ i แทนด้วย w_{ji}

1. สร้างนิวรอลเน็ตเวิร์กตามโครงสร้างที่ต้องการ กำหนดจำนวนนิวรอนของแต่ละชั้น
2. กำหนดค่าน้ำหนักเริ่มต้นแบบสุ่มให้มีค่าน้อยๆ (เช่น ระหว่าง -0.05 ถึง 0.05)
3. ปรับค่าน้ำหนักด้วยขั้นตอนวิธีดังนี้
   - สำหรับแต่ละอินพุตเวกเตอร์ $(\vec{x}, \vec{t})$ ในเซตตัวอย่างที่ใช้เรียนรู้
     1. ใช้ตัวอย่าง x เป็นอินพุต คำนวณหาค่า o_u ของทุกๆ นิวรอนในนิวรอลเน็ตเวิร์ก
     2. คำนวณค่าความคลาดเคลื่อน δ_k ของทุกนิวรอน k ในชั้นเอาต์พุต

        $$\delta_k \leftarrow o_k(1 - o_k)(t_k - o_k)$$

     3. คำนวณค่าความคลาดเคลื่อน δ_h ของทุกนิวรอน h ในชั้นซ่อน

        $$\delta_h \leftarrow o_h(1 - o_h) \sum_{k \in outputs} w_{kh}\delta_k$$

     4. ปรับค่าน้ำหนักของเส้นเชื่อม w_{ji}

        $$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

        เมื่อ

        $$\Delta w_{ji} = \eta \delta_j x_{ji}$$

4. วิธีการจำแนกตามกลุ่มหลัก (Majority Class Method)

ในการทดลองเพื่อเปรียบเทียบขั้นตอนวิธี CN2 ของ Clark และ Niblett (1989) มีการนำวิธีการจำแนกตามกลุ่มหลักมาทดลองเปรียบเทียบกับวิธีการอื่น ในการจำแนกตัวอย่างของวิธีการจำแนกตามกลุ่มหลักนี้ ระบบจะจำแนกตัวอย่างทดสอบให้อยู่ในกลุ่มซึ่งปรากฏมากที่สุดในกลุ่มตัวอย่างที่ใช้เรียนรู้ โดยไม่สนใจค่าคุณลักษณะของตัวอย่างนั้นๆ เลย ผลการทดลองที่ได้พบว่า ด้วยวิธีการง่ายๆ นี้ สามารถจำแนกตัวอย่างในกลุ่มปัญหามะเร็งเต้านม (Breast Cancer) ได้เป็นอย่างดี เมื่อเทียบกับขั้นตอนวิธีอื่นที่นำมาทดลองเปรียบเทียบ

งานวิจัยที่เกี่ยวข้อง

งานวิจัยอื่นที่เกี่ยวข้องกับงานวิจัยนี้เป็นงานวิจัยทางด้านการโปรแกรมตรรกะเชิงอุปนัยเป็นหลัก ซึ่งมีการวิจัยเป็น 2 แนวทาง คือ การนำไปประยุกต์ใช้กับงานด้านต่างๆ และการปรับปรุงเพื่อเพิ่มประสิทธิภาพของการโปรแกรมตรรกะเชิงอุปนัย ซึ่งงานวิจัยที่เกี่ยวข้องมีดังนี้

1. Ivan Bratko และ Stephen H. Muggleton (1995)

เอกสารอ้างอิงนี้เป็นบทความที่กล่าวถึงวิธีการโปรแกรมตรรกะเชิงอุปนัยหรือไอแอลพี ข้อได้เปรียบของวิธีการไอแอลพีกับวิธีการเรียนรู้ของเครื่องแบบอื่น แนวคิดพื้นฐานของระบบไอแอลพี ความสำคัญของความรู้ภูมิหลัง และการนำไปประยุกต์ใช้ในงานด้านต่างๆ เช่น การวิเคราะห์ไฟไนต์เอลิเมนต์ (Finite Element) การวิเคราะห์ความสามารถก่อกลายพันธุ์ของโมเลกุล การจำแนกคุณภาพทางด้านชีววิทยาของน้ำในแม่น้ำ การสร้างแบบจำลองของโมเลกุลทางชีวภาพ (Biomolecular modeling) การปรับปรุงคุณภาพของข้อมูลในการออกแบบโปรแกรม ฯลฯ

2. Bojan Dolsak และ Stephen Muggleton (1992)

ผู้วิจัยทั้งสองคนนี้ได้นำวิธีการไอแอลพีไปใช้ในงานด้านไฟไนต์เอลิเมนต์ จุดมุ่งหมายของการทดลองคือ การสร้างกฎเพื่อทำนายจำนวนเอลิเมนต์ที่เหมาะสมของโครงสร้างแต่ละแบบ เนื่องจากโดยปกติแล้วจะต้องมีการแทนโครงสร้างที่ต้องการวิเคราะห์ด้วยองค์ประกอบย่อยๆ ก่อน จากนั้นจึงวิเคราะห์ความเค้น (stress) ของโครงสร้างนั้นๆ Dolsak และ Muggleton ใช้ระบบไอแอลพี GOLEM ในการทดลองครั้งนั้น โดยใช้กลุ่มความรู้ภูมิหลังเป็นลักษณะต่างๆ ของเส้นเชื่อม ประกอบด้วยชนิดของเส้นเชื่อม เช่น `important`, `not_important`, `circuit`, `half_circuit` และ `short_for_hole` ฯลฯ เงื่อนไขขอบเขต (boundary conditions) เช่น `free` และ `one_side_fixed` ฯลฯ ชนิดของโหลด (load) เช่น `not_loaded`, `one_side_loaded` และ `cont_loaded` ฯลฯ ในการทดลองประกอบด้วยตัวอย่างบวก 75 ตัวอย่าง ตัวอย่างลบ 618 ตัวอย่าง และกลุ่มความรู้ภูมิหลัง 588 ข้อ จากนั้นทำการทดลอง 2 ครั้ง ในแต่ละครั้งทำการสุ่มเพื่อแบ่งตัวอย่างบวกมาใช้เป็นตัวอย่างที่ใช้ทดสอบ 10% ของตัวอย่างบวกทั้งหมด ผลการทดลองเฉลี่ยของทั้งสองครั้งได้เปอร์เซ็นต์ความถูกต้องเท่ากับ 79%

3. Johannes Furnkranz (1994)

นักวิจัยผู้นี้ได้สร้างระบบไอแอลพีชื่อ FOSSIL สามารถใช้กับปัญหาที่มีสัญญาณรบกวนได้ โดยใช้ค่าฮิวริสติกซึ่งวัดจากค่าสหสัมพันธ์ทางสถิติ (statistical correlation) ซึ่งเป็นวิธีการที่แตกต่างจากระบบโดยทั่วไปที่ใช้แนวคิดจากขั้นตอนวิธีการเรียนรู้ต้นไม้ตัดสินใจ (decision tree learning algorithm) มาจัดการกับสัญญาณรบกวนในเซตของตัวอย่างที่ใช้เรียนรู้ ทำการทดลองเปรียบเทียบกับระบบ FOIL [Quilan, 1990] โดยสุ่มเลือกตัวอย่างที่ใช้เรียนรู้ตามระดับสัญญาณรบกวน เปลี่ยนกลุ่มจากตัวอย่างบวกเป็นตัวอย่างลบ หรือเปลี่ยนจากตัวอย่างลบเป็นตัวอย่างบวก กลุ่มตัวอย่างที่ใช้ในการทดลอง คือ กลุ่มตัวอย่างการวิเคราะห์ตำแหน่งของตัวหมากรุกว่ามีการวางผิดตำแหน่งหรือไม่ (King-Rook-King illegal) เมื่อมีตัวหมากเหลืออยู่บนกระดาน 3 ตัว คือ ขุน (king) ฝ่ายขาว ขุนฝ่ายดำ และเรือ (rook) ฝ่ายขาว ผลการทดลองปรากฏว่า ระบบ FOSSIL ให้เปอร์เซ็นต์ความถูกต้องสูงกว่าระบบ FOIL ในระดับสัญญาณรบกวน 6 ระดับจากทั้งหมด 8 ระดับ

4. William W. Cohen (1995)

นักวิจัยผู้นี้ได้ทดลองโดยใช้ระบบการเรียนรู้ในลำดับที่หนึ่ง (first order learning system) ชื่อ FOIL ในงานด้านการจำแนกประเภทข้อความ (Text Categorization) เพื่อเปรียบเทียบวิธีการโปรแกรมตรรกะเชิงอุปนัย และวิธีการแบบตรรกศาสตร์ประพจน์ ในการทดลองมีการใช้ความรู้ภูมิหลังเกี่ยวกับตำแหน่งและความสัมพันธ์ของตำแหน่งของคำศัพท์ ผลการทดลองพบว่า ในการจำแนกข้อความจำนวน 51,991 ข้อความ ออกเป็น 10 หมวดหมู่ การเรียนรู้โดยใช้วิธีการโปรแกรมตรรกะเชิงอุปนัยให้ผลถูกต้องกว่าแบบตรรกศาสตร์ประพจน์ 2 หมวดหมู่ และ ทั้งสองแบบให้ผลการทดลองใกล้เคียงกัน 8 หมวดหมู่

5. Ashwin Srinivsan, Stephen H. Muggleton, R. D. King และ M. J. E. Stenberg (1996)

คณะนักวิจัยกลุ่มนี้ได้นำระบบไอแอลพี PROGOL มาใช้ในงานวิเคราะห์ความสามารถก่อกลายพันธุ์ในโมเลกุลขนาดเล็ก โดยใช้ข้อมูลของอะตอม พันธะ และลักษณะต่างๆ ของโมเลกุลเป็นความรู้ภูมิหลัง เช่น `anthracene`, `phenanthrene`, `ball3`, `ring_size_6`, `carbon_5_aromatic_ring` และ `benzene` ฯลฯ มีตัวอย่างบวกจำนวน 125 ตัวอย่าง ตัวอย่างลบ 63 ตัวอย่าง รวมตัวอย่างทั้งหมด 188 ตัวอย่าง ผลการทดลองได้เปอร์เซ็นต์ความถูกต้องเท่ากับ 80% จากตัวอย่างทั้งหมด และเมื่อทำการทดสอบโดยแบ่งตัวอย่างที่ใช้เรียนรู้จำนวน 30% หรือ 56 ตัวอย่าง เป็นกลุ่มตัวอย่างที่ใช้ทดสอบ ให้เปอร์เซ็นต์ความถูกต้องเท่ากับ 71%

6. Hendrik Blockeel และ Luc De Raedt (1997)

นักวิจัยทั้งสองนี้ได้พัฒนาระบบ TILDE (Top-down Induction of Logical Decision Tree) โดยใช้วิธีการสร้างต้นไม้ตัดสินใจเพื่อให้สามารถทำงานกับปัญหาที่อยู่ในรูปตรรกะลำดับที่หนึ่ง ปกติแล้ว

ขั้นตอนวิธีต้นไม้ตัดสินใจมักจะถูกนำไปใช้กับปัญหาที่มีลักษณะเป็นค่าคุณลักษณะ หรืออยู่ในรูปตรรกศาสตร์ประพจน์มากกว่าปัญหาที่อยู่ในรูปแบบตรรกะลำดับที่หนึ่ง กลุ่มตัวอย่างที่ใช้ทดลองประกอบด้วยกลุ่มตัวอย่างการวิเคราะห์ความสามารถก่อกลายพันธุ์ ทำการทดลองเปรียบเทียบกับระบบ PROGOL, FOIL และ Indigo [Geibel และคณะ, 1996] ผลการทดลองปรากฏว่า TILDE ให้เปอร์เซ็นต์ความถูกต้องใกล้เคียงกับ PROGOL และ Indigo แต่สูงกว่า FOIL ในการวิเคราะห์ความสามารถก่อกลายพันธุ์ การวิเคราะห์ไฟไนต์เอลิเมนต์ ทำการทดลองเทียบกับ FOIL, Indigo, FFOIL [Quinlan, 1996] และ FORS [Karalic, 1995] TILDE ให้เปอร์เซ็นต์ความถูกต้องต่ำกว่า Indigo และ FFOIL แต่สูงกว่า FOIL และ FORS

7. Mary E. Califf และ Raymond J. Mooney (1997)

นักวิจัยทั้งสองนี้ได้นำวิธีการโปรแกรมตรรกะเชิงอุปนัยมาประยุกต์ใช้กับงานด้านการดึงข้อมูลสำคัญจากข้อความภาษาธรรมชาติ (natural language information extraction) ในการทดลองได้ใช้ตัวอย่างในขั้นตอนการเรียนรู้เป็นประกาศรับสมัครงานคู่กับข้อมูลสำคัญที่มีอยู่ในประกาศนั้น เช่น ลักษณะงาน เงินเดือน สถานที่ บริษัทนายจ้าง ฯลฯ ซึ่งผู้สอนเป็นผู้ป้อนข้อมูลเหล่านี้ให้ จากนั้นจึงทดสอบโดยให้ระบบดึงลักษณะต่างๆ เหล่านี้จากข้อความในประกาศ ผลการทดลองปรากฏว่า ได้ค่าความแม่นยำ (precision) เฉลี่ย 83.7 เปอร์เซ็นต์ และค่าเรียกคืน (recall) เฉลี่ย 53.1 เปอร์เซ็นต์

8. Saso Dzeroski, Nico Jacobs, Martin Molina และ Carlos Moure (1998)

คณะนักวิจัยกลุ่มนี้ได้เรียนรู้เพื่อหาลักษณะของทางแยกซึ่งทำให้เกิดอุบัติเหตุทางรถยนต์ โดยใช้ข้อมูลจากเครื่องตรวจจับสัญญาณซึ่งติดตั้งในบริเวณทางแยกต่างๆ โดยตัวอย่างที่ใช้ในการเรียนรู้ประกอบด้วยความเร็วของรถยนต์ ความหนาแน่นของรถยนต์ ลักษณะของถนน ทำการทดลองโดยเปรียบเทียบผลระหว่างวิธีการโปรแกรมตรรกะเชิงอุปนัย 3 วิธี กับวิธีการเรียนรู้แบบตรรกศาสตร์ประพจน์ ผลการทดลองปรากฏว่า วิธีการโปรแกรมตรรกะเชิงอุปนัยทั้งสามวิธีให้เปอร์เซ็นต์ความถูกต้องสูงกว่าวิธีการเรียนรู้แบบตรรกศาสตร์ประพจน์

9. Sam Roberts, Wimi Van Laer, Nico Jacobs, Stephen Muggleton และ Jeremy Broughton (1998)

คณะนักวิจัยกลุ่มนี้ได้ทดลองเปรียบเทียบวิธีการโปรแกรมตรรกะเชิงอุปนัย และการเรียนรู้แบบตรรกศาสตร์ประพจน์ เรียนรู้ข้อมูลตัวอย่างอุบัติเหตุทางรถยนต์ของวัยรุ่นชายที่อยู่ในรูปแบบตรรกศาสตร์ประพจน์ จากข้อมูลอุบัติเหตุทางรถยนต์ที่เกิดขึ้นทั้งหมด ผลการทดลองปรากฏว่า เปอร์เซ็นต์ความถูกต้องของทุกวิธีการต่ำกว่าเปอร์เซ็นต์ความถูกต้องของวิธีการเลือกตามกลุ่มหลัก เพราะว่า จำนวนตัวอย่างบวกมีน้อยกว่าจำนวนตัวอย่างลบมาก จึงได้ทดลองโดยปรับจำนวนตัวอย่างที่ใช้ในการเรียนรู้ให้จำนวนตัวอย่างบวกเท่ากับตัวอย่างลบ ทำให้เปอร์เซ็นต์ความถูกต้องของทุกวิธีสูงกว่าเปอร์เซ็นต์ความถูกต้องของวิธีการเลือกโดยคลาสที่มีตัวอย่างมากกว่า และเปอร์เซ็นต์ความถูกต้องของวิธีการโปรแกรมตรรกะเชิงอุปนัยและวิธีการเรียนรู้แบบตรรกศาสตร์ประพจน์มีค่าใกล้เคียงกัน

10. Peter Flach and Nicolas Lachiche (1999)

นักวิจัยทั้งสองนี้ได้นำวิธีการจำแนกแบบเบย์ (Bayesian Classification) มาใช้กับปัญหาที่มีลักษณะเป็นตรรกะลำดับที่หนึ่ง ซึ่งจากเดิมวิธีการจำแนกแบบเบย์จะถูกนำไปใช้ในปัญหาที่มีลักษณะเป็นแบบตรรกศาสตร์ประพจน์ ทำการสร้างระบบ 1BC ขึ้นเพื่อให้สามารถใช้วิธีการของเบย์มาจำแนกตัวอย่างที่มีลักษณะเป็นตรรกะลำดับที่หนึ่ง โดยแบ่งลักษณะของเพรดิเคตออกเป็น 2 ประเภท คือ เพรดิเคตโครงสร้าง (structural predicate) และเพรดิเคตลักษณะ (property predicate) โดยให้ผู้ใช้กำหนดว่าเพรดิเคตใดจัดอยู่ในประเภทใด จากนั้นระบบจึงสร้างลักษณะสำคัญขึ้นจากเพรดิเคตทั้งสองประเภท แล้วจึงใช้วิธีการนาอีฟเบย์ (Naive Bayes) เข้ามาใช้เรียนรู้และจำแนกตัวอย่าง ทำการทดลองกับกลุ่มปัญหาการวิเคราะห์ความสามารถก่อกลายพันธุ์ การวิเคราะห์ไฟไนต์เอลิเมนต์ และการวิเคราะห์ตำแหน่งของหมากรุก ผลการทดลองให้เปอร์เซ็นต์ความถูกต้อง 80.9%, 66.2% และ 88.3% ในปัญหาการวิเคราะห์ความสามารถก่อกลายพันธุ์ การวิเคราะห์ไฟไนต์เอลิเมนต์ และการวิเคราะห์ตำแหน่งหมากรุก ตามลำดับ