6

การเรียนรู้ของเครื่อง

บทนี้กล่าวถึง*การเรียนรู้ของเครื่อง (machine learning)* ซึ่งเทคนิคการเรียนรู้ส่วนมากเป็น*การเรียนรู้เชิงอุปนัย (inductive learning)* และมีบางเทคนิคเป็น*การเรียนรู้เชิงวิเคราะห์ (analytical learning)* การเรียนรู้เชิงอุปนัยคือการเรียนรู้ที่หากฎเกณฑ์หรือความรู้ที่แฝงอยู่ใน*ชุดตัวอย่างสอน (training example set)* เพื่อเรียนรู้ให้ได้ความรู้ใหม่ที่สอดคล้องกับชุดตัวอย่างสอน ส่วนการเรียนรู้เชิงวิเคราะห์เป็นการจัดรูปแบบของความรู้ใหม่เพื่อให้ใช้งานได้อย่างมีประสิทธิภาพมากขึ้น ทำงานได้เร็วขึ้น

6.1 ขั้นตอนวิธีเชิงพันธุกรรม

ขั้นตอนวิธีเชิงพันธุกรรม – จีเอ (Genetic Algorithm – GA) [Goldberg, 1989; Mitchell, 1996] เป็นการเรียนรู้ที่จำลองการวิวัฒนาการ เราอาจมองได้ว่าจีเอเป็นกระบวนการค้นหาประเภทหนึ่งหรืออาจมองว่าจีเอเป็นการเรียนรู้ของเครื่องประเภทหนึ่งก็ได้ จีเอได้ถูกขยายขึ้นเป็น*การโปรแกรมเชิงพันธุกรรม – จีพี (Genetic Programming – GP)* [Koza, 1992] ข้อแตกต่างที่สำคัญอย่างหนึ่งระหว่างจีเอกับจีพีก็คือในจีเอสิ่งที่เรียนรู้ได้เป็นสายอักขระความยาวคงที่ (fixed-length string) ส่วนในจีพีจะได้สายอักขระความยาวแปรได้ (variable-length string) ซึ่งมักแสดงในรูปของโปรแกรมภาษา LISP

แนวคิดของจีเอมาจากทฤษฎีวิวัฒนาการของสิ่งมีชีวิต เช่นการไขว้เปลี่ยนของโครโมโซม (chromosome crossover) การกลายพันธ์ของยีน (gene mutation) การวิวัฒนาการของสิ่งมีชีวิต เป็นต้น จีเอสามารถจัดการกับปัญหาค่าดีสุดเฉพาะที่ (local optimum) ในการค้นหาได้ การค้นหาทั่วไปจะมองว่าจุดดีสุดเฉพาะที่เป็นกับดักและจะหลีกเลี่ยงกับดักโดยใช้วิธีต่างๆ เช่น การย้อนรอย (backtracking) หรือการค้นหาแบบขนาน (parallel search) โดยใช้สถานะเริ่มต้นที่ต่างๆ กัน เป็นต้น แต่เทคนิคการค้นหาด้วยจีเอจะใช้วิธีการที่แตกต่างไปดังจะกล่าวต่อไป

โครโมโซมกำหนดลักษณะพิเศษที่สืบทอดได้

เซลล์แต่ละเซลล์ในพืชชั้นสูงและสัตว์ประกอบด้วยนิวเคลียส 1 นิวเคลียส และนิวเคลียสหนึ่งๆ ประกอบด้วยโครโมโซมจำนวนมาก โครโมโซมจะอยู่กันเป็นคู่โดยได้รับมาจากพ่อและแม่อย่างละ 1 เส้น โครโมโซมแต่ละเส้นจะมียีนเป็นตัวกำหนดลักษณะพิเศษของสิ่งมีชีวิต ในขณะที่มีการจับคู่กันของโครโมโซมอาจเกิด*การไขว้เปลี่ยน (crossover)* ซึ่งเป็นการที่ยีนจากโครโมโซมพ่อแม่สลับเปลี่ยนกันทำให้เกิดโครโมโซมใหม่ขึ้น 2 คู่ และในขณะที่เซลล์แบ่งตัวจะเกิดกระบวนการ*คัดลอกโครโมโซม (chromosome copying)* ซึ่งบางครั้งจะมีการเปลี่ยนแปลงของยีนที่มาจากยีนพ่อและแม่เกิดเป็นยีนที่ไม่เคยมีมาก่อน เราเรียกการเกิดยีนลักษณะนี้ว่า *การกลายพันธุ์ (mutation)*

การไขว้เปลี่ยน

การกลายพันธุ์

ชารลส์ ดาร์วิน (Charles Darwin) ได้อธิบายการสืบทอดของสิ่งมีชีวิตด้วยกฎที่เรียกว่า *การวิวัฒนาการโดยผ่านการคัดเลือกตามธรรมชาติ (evolution through natural selection)* ไว้ว่าสิ่งมีชีวิตมีแนวโน้มที่จะสืบทอดลักษณะพิเศษให้ลูกหลานและธรรมชาติจะผลิตสิ่งมีชีวิตที่มีลักษณะพิเศษแตกต่างไปจากเดิม สิ่งมีชีวิตที่เหมาะสมที่สุด (fittest) ก็คือสิ่งมีชีวีตที่มีลักษณะพิเศษที่ธรรมชาติพอใจมากที่สุดจะมีแนวโน้มที่มีลูกหลานมากกว่าตัวที่ไม่เหมาะสม ดังนั้นประชากรจะโน้มเอียงไปทางตัวที่เหมาะสม เมื่อช่วงเวลาผ่านไปนานๆ การเปลี่ยนแปลงจะสะสมไปเรื่อยๆ และเกิดสปีชีส์ (species) ใหม่ที่เหมาะกับสภาพแวดล้อม ดังนั้นเราอาจกล่าวได้ว่าการคัดเลือกโดยธรรมชาติเกิดจากการเปลี่ยนแปลงที่เป็นผลของการไขว้เปลี่ยนและการกลายพันธุ์

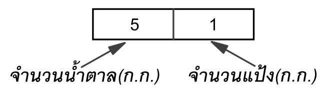
6.1.1 การออกแบบขั้นตอนวิธีเชิงพันธุกรรม

จะยกตัวอย่างปัญหาการทำคุ๊กกี้เพื่ออธิบายการออกแบบจีเอ [Winston, 1992] สมมติว่าเราต้องการหาส่วนผสมที่ดีที่สุดเพื่อทำคุ๊กกี้โดยที่คุ๊กกี้มีส่วนผสมสองอย่างคือแป้งและน้ำตาล และสมมติว่าคุณภาพของคุ๊กกี้เป็นฟังก์ชันแสดงในรูปที่ 2–8 ด้านล่างนี้

น้ำตาล \ แป้ง	1	2	3	4	5	6	7	8	9
9	1	2	3	4	5	4	3	2	1
8	2	3	4	5	6	5	4	3	2
7	3	4	5	6	7	6	5	4	3
6	4	5	6	7	8	7	6	5	4
5	5	6	7	8	9	8	7	6	5
4	4	5	6	7	8	7	6	5	4
3	3	4	5	6	7	6	5	4	3
2	2	3	4	5	6	5	4	3	2
1	1	2	3	4	5	4	3	2	1

รูปที่ 6–1 ฟังก์ชันภูเขาเรียบของคุณภาพคุ๊กกี้

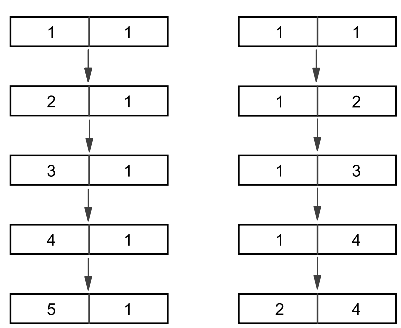
แนวตั้งและแนวนอนแสดงจำนวนกิโลกรัมของส่วนผสมน้ำตาลและแป้งตามลำดับ เช่น น้ำตาล 1 กก. กับแป้ง 1 กก. ผลิตได้คุ๊กกี้มีคุณภาพ 1 หน่วย ฟังก์ชันนี้จะมีค่าสูงสุดอยู่ที่ 5-5 (น้ำตาล 5 กก. กับแป้ง 5 กก. ผลิตได้คุ๊กกี้คุณภาพ 9 หน่วย) เราออกแบบให้แต่ละถาดของคุ๊กกี้ถูกแทนด้วยโครโมโซมเส้นหนึ่งดังแสดงในรูปที่ 1–1



รูปที่ 6–2 โครโมโซมแสดงคุ๊กกี้ถาดหนึ่ง

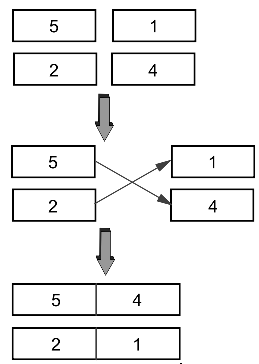
ในการออกแบบครั้งนี้กำหนดให้โครโมโซมมียีน 2 ตัว ยีนด้านซ้ายแทนจำนวนกิโลกรัมของน้ำตาลและยีนด้านขวาแทนจำนวนกิโลกรัมของแป้ง กำหนดให้ตัวเลขแสดงจำนวนกิโลกรัมของทั้งน้ำตาลและแป้งมีค่าตั้งแต่ 1 ถึง 9 โครโมโซมแทนถาดของคุ๊กกี้นี้กำหนดความเหมาะ (fitness) กับธรรมชาติของคุ๊กกี้ โครโมโซมสามารถสร้างขึ้นจากยีนน้ำตาลและแป้ง สร้างขึ้นจากการไขว้เปลี่ยนของโครโมโซมพ่อแม่คู่หนึ่ง หรือสร้างได้จากการกลายพันธ์ของยีนในโครโมโซมตัวหนึ่งที่มีอยู่ และถ้าหากเรามีโครโมโซม 1 เส้น เราสามารถตัดแบ่งเอายีนของน้ำตาลหรือยีนของแป้งได้

ในการทำจีเอครั้งนี้ กำหนดให้ประชากรรุ่นหนึ่งๆ มีโครโมโซมที่เหมือนกันเพียงเส้นเดียว เราจำลองการเกิดการกลายพันธ์ของโครโมโซมโดยการเลือกยีนตัวหนึ่งแบบสุ่มแล้วเปลี่ยนค่าของยีนโดยบวกหนึ่งหรือลบหนึ่งแบบสุ่มและยอมรับค่าที่ได้ถ้าค่านั้นอยู่ระหว่าง 1 ถึง 9 รูปที่ 6–3 แสดงวิวัฒนาการของโครโมโซมโดยการกลายพันธ์ ในรูปแสดงการกลายพันธ์สองรูปแบบซึ่งในแต่ละแบบแสดงการกลายพันธ์เมื่อผ่านไป 4 ครั้ง ในแต่ละครั้งยีนที่เลือกและค่าที่เปลี่ยนไปเกิดจากการสุ่มในครั้งนั้นๆ เราเห็นได้ว่าเมื่อผ่านการกลายพันธ์ไป 4 ครั้งโครโมโซมที่ได้มีความต่างกันค่อนข้างมาก โครโมโซมเส้นที่ดีเหมาะกับธรรมชาติก็จะถูกคัดเลือกซึ่งจะกล่าวต่อไป



รูปที่ 6–3 การจำลองการกลายพันธ์ของโครโมโซมคุ๊กกี้

เราจำลองการไขว้เปลี่ยนของโครโมโซมโดยตัดที่กึ่งกลางของโครโมโซมพ่อแม่ 2 เส้น แล้วนำแต่ละส่วนมาต่อกัน ดังรูปที่ 4–1



รูปที่ 6–4 การจำลองการไขว้เปลี่ยนของโครโมโซมคุ๊กกี้

จากรูปเราจะเห็นได้ว่าโครโมโซมพ่อแม่ 5-1 และ 2-4 ผลิตได้โครโมโซมลูกสองเส้นคือ 5-4 กับ 2-1 ในกรณีทั่วไปที่โครโมโซมประกอบด้วยยีนมากกว่า 2 ตัว การตัดและการต่อจะซับซ้อนยิ่งขึ้น

เมื่อพิจารณาปริภูมิค้นหาในบทที่ 2 จะพบว่าการกลายพันธ์มีลักษณะเทียบเคียงได้กับตัวกระทำการในปริภูมิค้นหา มีหน้าที่สร้างสถานะ (โครโมโซม) ลูกของสถานะปัจจุบัน อย่างไรก็ดีการกลายพันธ์มีความแตกต่างอยู่ที่ลักษณะสำคัญของวิธีการจีเอซึ่งใช้ความน่าจะเป็นในกระบวนการค้นหา กล่าวคือการกลายพันธ์จะสร้างโครโมโซมโดยการสุ่ม และเมื่อเราพิจารณาการไขว้เปลี่ยนจะไม่พบตัวกระทำการในปริภูมิค้นหาที่มีการทำงานในลักษณะเช่นนี้ กล่าวได้ว่าการไขว้เปลี่ยนเป็นคุณสมบัติเฉพาะของจีเอ เปรียบเสมือนการกระโดดไปยังสถานะใหม่ 2 ตัวจากสถานะพ่อแม่ 1 คู่ ซึ่งข้อดีของการไขว้เปลี่ยนจะได้กล่าวต่อไป

6.1.2 ค่าความเหมาะมาตรฐาน

ค่าความเหมาะ (fitness) ของโครโมโซมคือ*ความน่าจะเป็น*ที่โครโมโซมจะอยู่รอดในรุ่น (generation) ถัดไป ค่าความเหมาะมาตรฐานสามารถนิยามได้ดังนี้

$$f_i = \frac{q_i}{\sum_j q_j} \qquad (6.1)$$

โดยที่ f_i คือค่าความเหมาะของโครโมโซมเส้นที่ i ซึ่งมีค่าระหว่าง 0 ถึง 1 และ q_i คือคุณภาพของคุ๊กกี้ที่ถูกกำหนดโดยโครโมโซมเส้นที่ i

ตัวอย่างเช่น สมมติว่าประชากรประกอบด้วยโครโมโซม 4 เส้นคือ 1-4, 3-1, 1-2, 1-1 ค่าความเหมาะของโครโมโซมแต่ละเส้นแสดงได้ในตารางที่ 6–1

ตารางที่ 6–1 ตัวอย่างค่าความเหมาะมาตรฐานของโครโมโซมคุ๊กกี้

โครโมโซม	คุณภาพ	ค่าความเหมาะมาตรฐาน
1-4	4	0.40
3-1	3	0.30
1-2	2	0.20
1-1	1	0.10

ค่าความเหมาะมาตรฐานที่คำนวณได้ในตารางนี้ (เช่นค่าความเหมาะของโครโมโซม 1-4 จะเท่ากับ 4/(4+3+2+1)=0.40) เป็นความน่าจะเป็นที่โครโมโซมจะอยู่รอด (ถูกเลือก) ในรุ่นถัดไป ดังนั้นโครโมโซม 1-4 จะมีโอกาสอยู่รอดมากกว่าโครโมโซมเส้นอื่นๆ และมีโอกาสอยู่รอดมากกว่าโครโมโซม 1-1 ถึง 4 เท่า แต่ก็ไม่ได้หมายความว่าถ้าให้เลือกโครโมโซมได้แค่เส้นเดียวแล้วโครโมโซม 1-4 ที่มีค่าความเหมาะสูงสุดจะถูกเลือกทุกครั้งไป แต่จะขึ้นอยู่กับการสุ่มค่า แน่นอนว่า 1-4 มีโอกาสมากที่สุด และถ้าการสุ่มทำได้อย่างไม่โน้มเอียงในการสุ่ม 100 ครั้ง 1-4 น่าจะมีโอกาสถูกเลือกสัก 40 ครั้ง

6.1.3 การจำลองการคัดเลือกโดยธรรมชาติ

เราได้ออกแบบโครโมโซมสำหรับปัญหาที่เราสนใจ การกลายพันธ์ การไขว้เปลี่ยน ค่าความเหมาะแล้ว หัวข้อนี้จะกล่าวถึงการจำลองการคัดเลือกโดยธรรมชาติซึ่งสามารถทำได้โดยใช้ขั้นตอนทั่วไปดังต่อไปนี้

- กำหนดประชากรเริ่มต้น อาจมีโครโมโซม 1 เส้นหรือหลายเส้นก็ได้ เราอาจสุ่มโครโมโซมเหล่านี้หรือกำหนดขึ้นเองก็ได้
- ทำการกลายพันธ์ยีนในโครโมโซมในรุ่นปัจจุบันและผลิตลูก
- ทำการไขว้เปลี่ยนโครโมโซม (พ่อแม่) ในรุ่นปัจจุบันและผลิตลูก
- เพิ่มลูกที่เกิดใหม่ในประชากร
- สร้างประชากรรุ่นใหม่โดยเลือกโครโมโซมตามค่าความเหมาะอย่างสุ่ม

ในการแก้ปัญหาหนึ่งๆ ที่เราสนใจด้วยจีเอนั้น เราจำเป็นต้องกำหนดพารามิเตอร์ต่างๆ ในการจำลองการคัดเลือกโดยธรรมชาติ อย่างเช่นในประชากรรุ่นหนึ่งๆ ควรมีโครโมโซมจำนวนเท่าไร? ถ้าน้อยไปก็มีแนวโน้มว่าโครโมโซมในประชากรรุ่นหนึ่งๆ จะมีลักษณะ

คล้ายกันหรือเหมือนกันเกือบทั้งหมดและการทำการไขว้เปลี่ยนก็จะไม่มีผลมากนัก แต่ถ้ามากไปเราก็จะเสียเวลาคำนวณมาก อัตราการกลายพันธ์เป็นเท่าไร? ถ้าต่ำไปลักษณะใหม่จะเกิดช้า ถ้าสูงไปประชากรรุ่นใหม่จะไม่เกี่ยวเนื่องกับรุ่นเดิม จะทำการไขว้เปลี่ยนด้วยหรือไม่? ถ้าทำจะเลือกคู่ผสมอย่างไร? และการไขว้เปลี่ยนกำหนดอย่างไร? ตัดครึ่งตรงกึ่งกลางหรือสุ่มจุดตัด เป็นต้น โครโมโซมเหมือนกันจะยอมให้มีหลายเส้นหรือไม่?

ในปัญหาการหาส่วนผสมดีสุดของคุ๊กกี้นี้ เราจะจำลองการคัดเลือกโดยธรรมชาติดังนี้

- เริ่มจากโครโมโซม 1-1 เพียงเส้นเดียว
- โครโมโซมที่เหมือนกันจะมีแค่เส้นเดียวในประชากรรุ่นหนึ่งๆ
- โครโมโซม 4 เส้นหรือน้อยกว่าจะอยู่รอดไปถึงรุ่นใหม่
- สำหรับโครโมโซมแต่ละเส้นที่อยู่รอด เลือกยีนตัวหนึ่งแบบสุ่มเพื่อทำการกลายพันธ์ ถ้าโครโมโซมที่ได้จากการกลายพันธ์ยังไม่เคยมีมาเลยให้เพิ่มเข้าไปในประชากร
- ไม่ทำการไขว้เปลี่ยน
- โครโมโซมที่อยู่รอดจะแข่งขันกับโครโมโซมใหม่เพื่อกำหนดโครโมโซมที่จะอยู่ในรุ่นถัดไป โครโมโซมที่มีค่าความเหมาะสูงสุดจะถูกเลือกเสมอให้อยู่รอดไปถึงรุ่นถัดไป ส่วนเส้นที่อยู่รอดที่เหลือจะถูกเลือกจากโครโมโซมที่เหลือแบบสุ่มตามค่าความเหมาะ

จากการทดลอง 1,000 ครั้งโดยใช้การคัดเลือกโดยธรรมชาติด้านบน พบว่าส่วนผสมที่ทำให้คุณภาพของคุ๊กกี้ดีที่สุดถูกผลิตในรุ่นที่ 16 โดยเฉลี่ย และในจำนวนนี้การทดลองที่โชคดีที่สุดผลิตโครโมโซมที่ดีที่สุดในรุ่นที่ 8 ดังแสดงในตารางที่ 6–2 ด้านล่างนี้

ตารางที่ 6–2 ผลการทดลองดีสุดผลิตโครโมโซมดีสุดได้ในรุ่นที่ 8 โดยค่าความเหมาะมาตรฐาน

รุ่นที่ 0:		• 1-1 กลายพันธ์เป็น 1-2
โครโมโซม	*คุณภาพ*	
1-1	1	
รุ่นที่ 1:		• 1-2 กลายพันธุ์เป็น 1-3 และ 1-1 เป็น 1-2 ซึ่งมีอยู่แล้ว
โครโมโซม	*คุณภาพ*	
1-2	2	
1-1	1	

รุ่นที่ 2: *โครโมโซม* 1-3, 1-2, 1-1	*คุณภาพ* 3, 2, 1	• 1-3 กลายพันธุ์เป็น 1-4, 1-2 เป็น 2-2, 1-1 เป็น 2-1 โครโมโซมทั้งหมดมี 6 เส้น และ 4 เส้นถูกเลือกดังแสดงในรุ่นที่ 3 (1-4 ที่มีค่าความเหมาะสูงสุดถูกเลือกเลย ส่วนอีกสามเส้นที่เหลือได้จากการสุ่มตามค่าความเหมาะ สังเกตว่าแม้ว่า 2-2 จะมีค่าความเหมาะดีกว่า 1-2 และ 2-1 แต่ไม่ถูกเลือกในครั้งนี้)
รุ่นที่ 3: *โครโมโซม* 1-4, 1-3, 1-2, 2-1	*คุณภาพ* 4, 3, 2, 2	• การกลายพันธุ์ผลิตได้โครโมโซมใหม่ 3 เส้นดังต่อไปนี้ *โครโมโซม* / *คุณภาพ* 2-4 / 5 2-3 / 4 3-1 / 3
รุ่นที่ 4: *โครโมโซม* 2-4, 1-4, 1-3, 2-1	*คุณภาพ* 5, 4, 3, 2	• โครโมโซมทุกเส้นกลายพันธุ์และผลิตลูก
รุ่นที่ 5: *โครโมโซม* 2-5, 1-5, 2-3, 2-2	*คุณภาพ* 6, 5, 4, 3	• โครโมโซมทุกเส้นกลายพันธุ์และผลิตลูก
รุ่นที่ 6: *โครโมโซม* 3-5, 1-5, 3-2, 1-4	*คุณภาพ* 7, 5, 4, 4	• 3-5 กลายพันธุ์เป็น 4-5, 3-2 เป็น 3-1, 1-4 เป็น 1-5, 1-5 เป็น 1-4 จะเห็นได้ว่าการผลิตลูกมักมีการซ้ำซ้อนของโครโมโซม เช่นไปซ้ำเดิมกับพ่อแม่เป็นต้น

รุ่นที่ 7:		• ที่จุดนี้ 4-5 กลายพันธุ์เป็น 5-5 ซึ่งเป็นคำตอบในที่สุด
โครโมโซม	*คุณภาพ*	
4-5	8	
1-5	5	
1-4	4	
3-1	3	
รุ่นที่ 8:		
โครโมโซม	*คุณภาพ*	
5-5	9	
4-5	8	
2-5	6	
2-1	2	

6.1.4 การไขว้เปลี่ยนเพื่อเอาชนะจุดดีสุดเฉพาะที่

หัวข้อนี้เราจะดูผลของการไขว้เปลี่ยนที่มีต่อจีเอ โดยทำการทดลองเหมือนการทดลองที่แล้ว แต่เพิ่มการไขว้เปลี่ยนเพื่อสร้างโครโมโซมใหม่ด้วย การไขว้เปลี่ยนทำดังต่อไปนี้

- ทำการไขว้เปลี่ยนโดยใช้โครโมโซมที่อยู่รอดจากรุ่นที่แล้ว (อย่างมากสุด 4 เส้น)
- สำหรับโครโมโซมที่จะทำการไขว้เปลี่ยนเส้นหนึ่งๆ ให้เลือกคู่ทำการไขว้เปลี่ยนแบบสุ่ม
- สลับยีนของโครโมโซมพ่อแม่และผลิตโครโมโซมลูก 2 เส้น ถ้าโครโมโซมลูกยังไม่เคยมีมาเลยให้เพิ่มเข้าไปในประชากรเพื่อแข่งขันที่จะอยู่รอดในรุ่นถัดไป

ผลจากผลการทดลอง 1,000 ครั้ง ส่วนผสมที่ดีที่สุดถูกผลิตในรุ่นที่ 14 โดยเฉลี่ย ใช้จำนวนรุ่นน้อยกว่ากรณีไม่ใช้การไขว้เปลี่ยน 2 รุ่น แม้ว่าการไขว้เปลี่ยนจะช่วยให้เราพบส่วนผสมดีสุดโดยใช้จำนวนรุ่นน้อยกว่าเดิม แต่เราต้องใช้การคำนวณในแต่ละรุ่นมากขึ้นกว่าเดิมเนื่องจากจำนวนโครโมโซมที่มากขึ้นและการคำนวณค่าความเหมาะที่เพิ่มขึ้น ดังนั้นเวลาโดยรวมจะเพิ่มขึ้นกว่าเดิม สำหรับปัญหานี้เป็นปัญหาที่ไม่มีจุดดีสุดเฉพาะที่ มีแค่จุดดีสุดวงกว้างจุดเดียว ประสิทธิภาพของการไขว้เปลี่ยนจึงไม่เห็นอย่างชัดเจน ปัญหาที่เราจะพิจารณาต่อไปเป็นปัญหาที่มีจุดดีสุดเฉพาะที่ซึ่งจะสร้างความยากลำบากสำหรับวิธีการค้นหาทั่วไป แต่จีเอสามารถจัดการกับปัญหาลักษณะนี้ได้ ปัญหานี้เป็นการหาส่วนผสมดีสุดของคุ๊กกี้เหมือนเดิมแต่ใช้ฟังก์ชันใหม่ดังรูปที่ 6–5 ต่อไปนี้

น้ำตาล \ แป้ง	1	2	3	4	5	6	7	8	9
9	1	2	3	4	5	4	3	2	1
8	2	0	0	0	0	0	0	0	2
7	3	0	0	0	0	0	0	0	3
6	4	0	0	7	8	7	0	0	4
5	5	0	0	8	9	8	0	0	5
4	4	0	0	7	8	7	0	0	4
3	3	0	0	0	0	0	0	0	3
2	2	0	0	0	0	0	0	0	2
1	1	2	3	4	5	4	3	2	1

รูปที่ 6–5 ฟังก์ชันภูเขามีคูน้ำล้อมของคุณภาพคุ๊กกี้

เริ่มต้นจากโครโมโซม 1-1 เช่นเดิม เราพบว่าในกรณีนี้การกลายพันธ์เพียงอย่างเดียวไม่สามารถทำให้โครโมโซมในรุ่นที่อยู่ภายนอกคูน้ำ (บริเวณที่มีค่าเป็น 0) ผลิตโครโมโซมทะลุเข้าไปอยู่พื้นที่ภายในคูน้ำได้ เนื่องจากโครโมโซมตรงกลางมีค่าความเหมาะเป็น 0 ซึ่งไม่สามารถอยู่รอดในรุ่นถัดไปได้ (ค่าความเหมาะของโครโมโซมเป็น 0 ทำให้ความน่าจะเป็นที่จะอยู่รอดไม่มีเลย) อย่างไรก็ดีการไขว้เปลี่ยนที่จับคู่โครโมโซมพ่อแม่ที่เหมาะสมเช่น 1-5 และ 5-1 จะสามารถผลิตลูกที่ข้ามคูน้ำไปได้ จากการทดลอง 1,000 ครั้งพบว่าส่วนผสมที่ดีที่สุดถูกผลิตในรุ่นที่ 155 โดยเฉลี่ย!! เป็นผลที่ไม่ดี ถ้าเราคำนวณดูก็จะทราบทันทีว่าโครโมโซมที่แตกต่างกันที่เป็นไปได้ทั้งหมดมีแค่ 9 x 9 = 81 เส้นเท่านั้น ผลที่ได้คือรุ่นที่ 155 และแต่ละรุ่นมีโครโมโซมที่เราทดสอบมากกว่าหนึ่งเส้น (แม้ว่าจะมีโครโมโซมมากมายที่ซ้ำกันในรุ่นต่าง ๆ)

สาเหตุหนึ่งที่ผลไม่ดีก็เพราะว่าก่อนที่โครโมโซมจะกลายพันธุ์เป็นโครโมโซมที่อยู่บริเวณ 1-5 หรือ 5-1 นั้น โดยมากตายไปก่อนที่จะไปสู่บริเวณนั้นสำเร็จ และโอกาสที่คู่ที่เหมาะสมของโครโมโซมจะเกิดการไขว้เปลี่ยนก็มีโอกาสน้อยมาก ซึ่งที่จริงแล้วคู่ที่เหมาะสมของการไขว้เปลี่ยนมีจำนวนมากอย่างเช่น 2-6 กับ 4-2, 4-8 กับ 2-5, 6-8 กับ 2-4 เป็นต้น และโครโมโซมในคู่ทั้งหมดนี้ล้วนมีความน่าจะเป็นที่จะอยู่รอดเป็น 0 ทั้งสิ้น ที่เป็นเช่นนี้เกิดขึ้นจากฟังก์ชันความเหมาะมาตรฐานที่จะกำหนดให้โครโมโซมเหล่านี้มีความน่าจะเป็นที่จะอยู่รอดเป็น 0 หากเราปรับแก้ฟังก์ชันความเหมาะให้โครโมโซมเหล่านี้มีโอกาสอยู่รอดบ้างแม้จะน้อย ก็น่าจะช่วยให้การค้นหาส่วนผสมดีสุดทำได้ดีขึ้น ดังจะกล่าวในหัวข้อต่อไป

6.1.5 ปรับปรุงจีเอด้วยฟังก์ชันความเหมาะแบบลำดับและการใช้ความหลากหลาย

ปรับปรุงจีเอด้วยฟังก์ชันความเหมาะแบบลำดับ

ฟังก์ชันความเหมาะใหม่ที่เราจะพิจารณากันนี้เรียกว่า *ค่าความเหมาะแบบลำดับ (rank fitness)* เป็นวิธีที่ใช้ควบคุมการเลือกโครโมโซมโดยไม่สนใจคุณภาพของโครโมโซมว่ามีค่าเท่าไร จะเพียงแค่จัดลำดับเรียงโครโมโซมตามคุณภาพที่มีค่าสูงสุดจนถึงต่ำสุด จากนั้นกำหนดให้ p ค่าคงที่ค่าหนึ่งเป็นความน่าจะเป็นที่โครโมโซมลำดับที่ 1 จะถูกเลือก และเป็นความน่าจะเป็นที่โครโมโซมลำดับที่ 2 จะถูกเลือกเมื่อลำดับที่ 1 ไม่ถูกเลือก และเป็นความน่าจะเป็นที่ลำดับที่ 3 จะถูกเลือกเมื่อลำดับที่ 1 และ 2 ไม่ถูกเลือก เป็นเช่นนี้ไปจนกระทั่งถึงลำดับสุดท้ายซึ่งจะถูกเลือกเมื่อลำดับก่อนหน้ามันไม่ถูกเลือกเลย

ตัวอย่างเช่นสมมติว่า p=2/3 และโครโมโซมที่เราสนใจอยู่คือ 1-4, 3-1, 1-2, 1-1 และ 7-5 (ในกรณีของภูเขามีคูน้ำล้อม) จะได้ค่าความเหมาะของโครโมโซมดังตารางที่ 6–3 ซึ่งเปรียบเทียบค่าความเหมาะแบบลำดับกับค่าความเหมาะมาตรฐาน

ตารางที่ 6–3 เปรียบเทียบค่าความเหมาะแบบลำดับกับค่าความเหมาะมาตรฐาน

โครโมโซม	คุณภาพ	ลำดับ	ค่าความเหมาะมาตรฐาน	ค่าความเหมาะแบบลำดับ
1-4	4	1	0.40	0.667
1-3	3	2	0.30	0.222
1-2	2	3	0.20	0.074
1-1	1	4	0.10	0.025
7-5	0	5	0.00	0.012

ดังแสดงในตารางที่ 6–3 ค่าความเหมาะแบบลำดับของโครโมโซม 1-4 เท่ากับ p = 2/3 (ประมาณ 0.667) ส่วนโครโมโซม 1-3 มีค่าความน่าจะเป็นเท่ากับ p(1–p) (ความน่าจะเป็นที่ตัวเองจะถูกเลือกเมื่อโครโมโซมลำดับที่ 1 ไม่ถูกเลือก) ซึ่งมีค่าประมาณ 0.222 ส่วนลำดับที่ 3 จะถูกเลือกเมื่อเส้นที่ 1 และ 2 ไม่ถูกเลือกด้วยความน่าจะเป็นเท่ากับ $p(1-p)(1-p) \approx 0.074$ ส่วนเส้นที่ 4 ก็เท่ากับ $p(1-p)(1-p)(1-p) \approx 0.025$ และเส้นสุดท้ายมีค่าความน่าจะเป็นเท่ากับ 1 – (0.667+0.222+0.074+0.025+0.012) = 0.012

จากผลการทดลอง 1,000 ครั้งโดยใช้ค่าความเหมาะแบบลำดับและจำลองการคัดเลือกโดยธรรมชาติเหมือนเดิมทุกประการ พบว่าส่วนผสมที่ดีที่สุดถูกผลิตในรุ่นที่ 75 โดยเฉลี่ยเร็วขึ้นกว่าเดิม (ส่วนผสมดีสุดถูกผลิตในรุ่นที่ 155) ประมาณ 2 เท่า ซึ่งแสดงให้เห็นว่าค่าความเหมาะแบบลำดับดีกว่าค่าความเหมาะมาตรฐาน และจากการใช้ค่าความเหมาะแบบลำดับนี้ทำให้โครโมโซมที่อยู่ตรงกลางในคูน้ำสามารถอยู่รอดถึงรุ่นถัดไปและวิวัฒนาการเป็น

โครโมโซมที่อยู่ภายในซึ่งมีคุณภาพสูงต่อไปได้ อย่างไรก็ดีแม้ว่าค่าความเหมาะแบบลำดับจะทำให้เร็วขึ้นกว่าเดิมประมาณ 2 เท่า แต่ยังคงเป็นผลที่ไม่ดีนักดังเช่นที่ได้กล่าวแล้วว่า 75 รุ่นแต่ละรุ่นเราตรวจสอบโครโมโซมมากกว่า 1 เส้น

เพิ่มประสิทธิภาพจีเอให้สูงขึ้นโดยความหลากหลาย

หัวข้อนี้แสดงการใช้*ความหลากหลาย (diversity)* เพื่อเพิ่มประสิทธิภาพของจีเอให้สูงขึ้นอีก ซึ่งได้แนวคิดจากการวิวัฒนาการของสิ่งมีชีวิต ที่เรามักพบว่าบ่อยครั้งในธรรมชาติที่สปีชีส์ซึ่งลักษณะแตกต่างไปจากสปีชีส์ที่เหมาะกับธรรมชาติสามารถอยู่รอดได้ดี ซึ่งความหลากหลายนี้จะช่วยให้โครโมโซมที่มียีนต่างจากพวกพ้องถูกคัดเลือกได้ง่ายขึ้น

การจะนำความต่างเข้าไปช่วยเลือกโครโมโซมนั้น อย่างแรกที่ต้องทำก็คือนิยามความต่างในรูปที่วัดได้ ในที่นี้เราจะวัดความต่างของโครโมโซมเส้นหนึ่งๆ โดยคำนวณค่าของ “ผลรวมของ 1/ระยะห่างกำลังสองระหว่างโครโมโซมนั้นกับโครโมโซมอื่นที่ถูกเลือกแล้วว่าให้อยู่รอดในรุ่นถัดไป” เนื่องจากเราต้องการโครโมโซมเส้นที่ต่างจากโครโมโซมที่เหมาะกับธรรมชาติ ดังนั้นการวัดความต่างหรือความหลากหลายจึงเทียบกับโครโมโซมเส้นที่เหมาะกับธรรมชาติ ส่วนระยะห่างหมายถึงระยะห่างตามระยะยุคลิด (Euclidian distance) เช่น 5-2 กับ 1-4 มีระยะห่างกำลังสองเท่ากับ $(5-1)^2+(2-4)^2= 20$ เป็นต้น

พิจารณาโครโมโซม 5-1, 1-4, 3-1, 1-2, 1-1 และ 7-5 โครโมโซมที่มีคุณภาพสูงสุดคือ 5-1 (ซึ่งเราจะเลือกเลยให้อยู่ในรุ่นถัดไปเป็นเส้นแรก) ตารางที่ 6–4 ด้านล่างแสดงลำดับของ 5 เส้นที่เหลือโดยเรียงตามคุณภาพและผลรวม 1/ระยะห่างกำลังสองจาก 5-1

ตารางที่ 6–4 ลำดับของโครโมโซมเรียงตามลำดับความหลากหลายและลำดับคุณภาพ

โครโมโซม	คุณภาพ	$1/d^2$	ลำดับความหลากหลาย	ลำดับคุณภาพ
1-4	4	0.040	1	1
3-1	3	0.250	5	2
1-2	2	0.059	3	3
1-1	1	0.062	4	4
7-5	0	0.050	2	5

$1/d^2$ แสดง 1/ระยะห่างกำลังสองระหว่างโครโมโซมที่พิจารณากับ 5-1 ตัวอย่างเช่น 1-4 กับ 5-1 มีค่าเท่ากับ $1/((5-1)^2+(1-4)^2) = 0.040$ เป็นต้น จากตารางจะพบว่าโครโมโซม 7-5 ซึ่งมีคุณภาพเป็น 0 และจะไม่เคยถูกเลือกเลยโดยค่าความเหมาะมาตรฐาน แต่เมื่อคำนวณค่าความเหมาะแบบลำดับความหลากหลายจะอยู่ในลำดับที่ 2 ซึ่งในกรณีนี้เมื่อดูจาก

รูปที่ 6–5 จะเห็นว่า 7-5 เป็นโครโมโซมที่ดีเส้นหนึ่งและมีโอกาสกลายพันธ์เข้าสู่บริเวณด้านในของคูน้ำเพื่อเป็นคำตอบต่อไป

เมื่อเราได้ลำดับความหลากหลายแล้ว เราจำเป็นต้องนำลำดับนี้ผนวกเข้าไปใช้ร่วมกับค่าความเหมาะเดิม เราไม่อาจใช้ลำดับความหลากหลายอย่างเดียวได้เพราะเป็นแค่ปัจจัยหนึ่งในการเลือกโครโมโซม ลำดับคุณภาพเดิมซึ่งค่อนข้างดีอยู่แล้วก็ไม่อาจตัดทิ้งได้ ดังนั้นวิธีผนวกลำดับความหลากหลายเข้าใช้ร่วมกับลำดับคุณภาพสามารถทำได้โดยนำลำดับทั้งสองบวกกันแล้วจัดเรียงลำดับใหม่อีกครั้ง เราเรียกลำดับที่ได้ใหม่นี้ว่า*ลำดับรวม (combined rank)* เมื่อได้ลำดับรวมซึ่งคิดทั้งคุณภาพและความหลากหลายแล้ว การเลือกกระทำได้เหมือนเดิมโดยกำหนดความน่าจะเป็นของลำดับแรกเป็น p = 2/3 (ดูตารางที่ 6–5)

ตารางที่ 6–5 ลำดับรวมที่พิจารณาทั้งคุณภาพและความหลากหลาย

โครโมโซม	ผลรวมของลำดับคุณภาพและลำดับความหลากหลาย	ลำดับผลรวม	ค่าความเหมาะ
1-4	2	1	0.667
3-1	7	4	0.025
1-2	6	2	0.222
1-1	8	5	0.012
7-5	7	3	0.074

ลำดับผลรวมในตารางได้จากการเรียงลำดับผลในสดมภ์ที่สองใหม่ ในกรณีที่มีค่าเท่ากันอย่างเช่น 3-1 กับ 7-5 มีค่าเท่ากันเท่ากับ 7 ก็ใช้การสุ่มเลือก ในที่นี้ 7-5 ถูกสุ่มให้มีลำดับผลรวมเป็นลำดับสาม จากตารางสมมติว่าเราเลือกโครโมโซมตามค่าความเหมาะได้เป็น 1-4 และเป็นเส้นที่สองต่อจาก 5-1 หลังจากนี้เราจะเลือกเส้นที่ 3 ในครั้งนี้เราต้องคำนวณหา 1/ระยะห่างกำลังสอง โดยคิดทั้ง 5-1 และ 1-4 (ดูตารางถัดไป)

ตารางที่ 6–6 การเลือกโครโมโซมเส้นที่ 3 ต่อจาก 5-1 และ 1-4

โครโมโซม	$\sum_i \frac{1}{d_i^2}$	ลำดับความหลากหลาย	ลำดับคุณภาพ	ลำดับรวม	ค่าความเหมาะ
3-1	0.327	4	1	4	0.037
1-2	0.309	3	2	3	0.074
1-1	0.173	2	3	2	0.222
7-5	0.077	1	4	1	0.667

ตัวอย่างการคำนวณค่าของ $\sum_i \frac{1}{d_i^2}$ อย่างเช่นในกรณีของโครโมโซม 3-1 จะได้ค่าเป็น $\frac{1}{(5-3)^2+(1-1)^2} + \frac{1}{(1-3)^2+(4-1)^2}$ =0.327 เป็นต้น สมมติว่าโครโมโซมที่ถูกเลือกตามค่าความเหมาะเส้นต่อไปคือ 7-5 และโครโมโซมเส้นสุดท้ายเราก็สามารถทำได้ในลักษณะเดียวกัน และเลือกได้เป็น 1-1 ดังแสดงตารางที่ 6–7 ต่อไปนี้

ตารางที่ 6–7 การเลือกโครโมโซมเส้นที่ 3 ต่อจาก 5-1, 1-4 และ 7-5

โครโมโซม	$\sum_i \frac{1}{d_i^2}$	ลำดับความหลากหลาย	ลำดับคุณภาพ	ลำดับรวม	ค่าความเหมาะ
3-1	0.358	3	1	3	0.111
1-2	0.331	2	2	2	0.222
1-1	0.190	1	3	1	0.667

ค่าความเหมาะที่คำนวณตามลำดับรวมมีความแตกต่างจากค่าความเหมาะมาตรฐานที่โครโมโซม 7-5 ซึ่งเป็นโครโมโซมที่ดีเส้นหนึ่งและไม่เคยถูกเลือกเลยด้วยค่าความเหมาะมาตรฐาน แต่สามารถจะถูกเลือกได้ด้วยค่าความเหมาะตัวใหม่นี้

จากการทดลอง 1,000 ครั้งโดยใช้ลำดับรวมด้วยค่า p = 2/3 เริ่มจากโครโมโซม 1-1 คำตอบที่ดีที่สุดถูกผลิตได้ในรุ่นที่ 15 โดยเฉลี่ย!!! เร็วกว่าลำดับคุณภาพถึง 5 เท่า นอกจากนั้นค่าความเหมาะแบบลำดับรวมนี้ไม่ได้ถูกพัฒนาขึ้นโดยเฉพาะสำหรับแก้ปัญหาภูเขามีคูน้ำล้อมอย่างเดียวเท่านั้น ยังสามารถทำงานได้ดีสำหรับปัญหาภูเขาเรียบด้วย ซึ่งดูสรุปการเปรียบเทียบค่าความเหมาะได้ในตารางที่ 6–8 ด้านล่างนี้ (ค่าในตารางได้จากการใช้การกลายพันธ์และการไขว้เปลี่ยนเหมือนกันหมด)

ตารางที่ 6–8 เปรียบเทียบค่าความเหมาะ 3 วิธี: มาตรฐาน ลำดับคุณภาพ และลำดับรวม

ฟังก์ชัน	ค่าความเหมาะมาตรฐาน	ค่าความเหมาะแบบลำดับคุณภาพ	ค่าความเหมาะแบบลำดับรวม
ภูเขาเรียบ	14	12	12
ภูเขามีคูน้ำล้อม	155	75	15

ในจำนวนการทดลอง 1,000 ครั้งโดยใช้ลำดับรวมนั้น ครั้งที่ดีที่สุดโครโมโซม 5-5 ถูกผลิตในรุ่นที่ 7 ดังแสดงในตารางที่ 6–9 ต่อไปนี้

ตารางที่ 6–9 ผลการทดลองดีสุดที่ผลิตโครโมโซมดีสุดได้ในรุ่นที่ 7 โดยลำดับรวม

รุ่นที่ 0:		• 1-1 กลายพันธ์เป็น 2-1
โครโมโซม	*คุณภาพ*	
1-1	1	
รุ่นที่ 1:		• การกลายพันธ์ผลิตได้ 3-1 ส่วนการไขว้เปลี่ยนไม่ได้ลูกตัวใหม่เพราะยีนตัวที่สองเหมือนกัน
โครโมโซม	*คุณภาพ*	
2-1	2	
1-1	1	
รุ่นที่ 2:		• การกลายพันธ์ผลิตได้ 4-1 และ 2-2 ส่วนการไขว้เปลี่ยนยังคงไม่เกิดผล
โครโมโซม	*คุณภาพ*	
3-1	3	
2-1	2	
1-1	1	
รุ่นที่ 3:		• การกลายพันธุ์ผลิตได้โครโมโซมใหม่ 3 เส้นคือ 5-1, 1-2, 2-3 ส่วนการไขว้เปลี่ยนของ 2-2 กับ 4-1 ผลิตได้ 2-1 กับ 4-2 การไขว้เปลี่ยนของคู่อื่นซ้ำกับโครโมโซมที่ผลิตได้ก่อนมัน
โครโมโซม	*คุณภาพ*	
4-1	4	
3-1	3	
1-1	1	
2-2	0	*โครโมโซม* — *คุณภาพ*
		5-1 — 5
		1-2 — 2
		2-3 — 0
		2-1 — 2
		4-2 — 0
รุ่นที่ 4:		• การกลายพันธุ์ผลิตได้ 6-1, 3-2, 2-2, 2-4 ส่วนการไขว้เปลี่ยนผลิต 2-1, 1-1, 5-2, 3-2, 5-3
โครโมโซม	*คุณภาพ*	
5-1	5	
3-1	4	*โครโมโซม* — *คุณภาพ*
1-2	2	6-1 — 4
2-3	0	3-2 — 0
		2-2 — 0

2-4	0
2-1	2
1-1	1
5-2	0
3-2	0
5-3	0

รุ่นที่ 5:		• ที่จุดนี้เกิดการไขว้เปลี่ยนของ 5-1 กับ 2-4 ได้ 5-4 ซึ่งเป็นโครโมโซมที่ดีในรุ่นหน้า
โครโมโซม	*คุณภาพ*	
5-1	5	
3-1	3	
1-2	2	
2-4	0	
รุ่นที่ 6:		• และในท้ายที่สุด 5-4 กลายพันธุ์เป็น 5-5
โครโมโซม	*คุณภาพ*	
5-4	8	
1-4	4	
3-1	3	
1-2	2	
รุ่นที่ 7:		
โครโมโซม	*คุณภาพ*	
5-5	9	
1-4	4	
1-2	2	
5-2	0	

ด้านล่างนี้แสดงการค้นหาคำตอบโดยจีเอ โดยแสดงเฉพาะโครโมโซมที่ถูกเลือกในแต่ละรุ่น

(0)

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	3	4	5	4	3	2	1

(1)

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	**2**	3	4	5	4	3	2	1

(2)

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	**2**	**3**	4	5	4	3	2	1

(3)

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	**0**	0	0	0	0	0	0	2
1	2	**3**	**4**	5	4	3	2	1

(4)

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	8	7	0	0	4
3	**0**	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	**3**	4	**5**	4	3	2	1

(5)

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	**0**	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	**3**	4	**5**	4	3	2	1

(6)

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	9	8	0	0	5
4	0	0	7	**8**	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	0	0	0	0	2
1	2	**3**	4	5	4	3	2	1

(7)

1	2	3	4	5	4	3	2	1
2	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	3
4	0	0	7	8	7	0	0	4
5	0	0	8	**9**	8	0	0	5
4	0	0	7	8	7	0	0	4
3	0	0	0	0	0	0	0	3
2	0	0	0	**0**	0	0	0	2
1	2	3	4	5	4	3	2	1

รูปที่ 6–6 การค้นหาโดยจีเอในปัญหาภูเขามีคูน้ำล้อม

จากรูปจะเห็นได้ว่าในรุ่นที่ 3 โครโมโซมที่คุณภาพเป็น 0 สามารถถูกเลือกได้โดยค่าความเหมาะแบบลำดับรวมและจะเห็นการเคลื่อนที่ของโครโมโซมจากรุ่นที่ 1 ถึง 4 ว่าโครโมโซมค่อยๆ ขยับตัวไปยังจุดสูงสุดเฉพาะที่ซึ่งมีคุณภาพเท่ากับ 5 และจะเห็นการเคลื่อนที่ของโครโมโซมที่มีคุณภาพเท่ากับ 0 ที่ค่อยๆ ขยับออกจากจุดสูงสุดเฉพาะที่ทีละ

น้อย จนกระทั่งในรุ่นที่ 5 เมื่ออยู่ในตำแหน่งที่เหมาะสมและเกิดการไขว้เปลี่ยนกับจุดสูงสุดเฉพาะที่แล้วสามารถทะลุผ่านคูน้ำเข้าไปยังภายในคูได้ แล้วเปลี่ยนเป็นจุดสูงสุดในที่สุด

จากรูปแสดงการทำงานของจีเอ เราสามารถเห็นได้ว่าการค้นหาโดยทั่วไปมักจะพยายามหลีกเลี่ยงจุดดีสุดเฉพาะที่ แต่การทำงานของจีเอใช้วิธีการที่ต่างไป โดยการผลิตโครโมโซมที่เป็นค่าดีสุดเฉพาะที่จากนั้นจึงใช้ความหลากหลายเพื่อเป็นส่วนประกอบของค่าความเหมาะแล้วผลิตโครโมโซมที่อยู่ห่างออกจากค่าดีสุดเฉพาะที่ หากมีโครโมโซมอยู่ในจุดดีสุดเฉพาะที่ทุกจุดแล้ว ก็มีโอกาสที่โครโมโซมเหล่านี้จะหาทางไปยังจุดดีสุดวงกว้าง (global optimum) ได้ในที่สุด

6.2 การเรียนรู้โดยการจำ

การเรียนรู้โดยการจำ (rote learning) เป็นการเรียนรู้แบบที่ง่ายที่สุดของกระบวนการเรียนรู้ทั้งหลาย โดยเมื่อพบความรู้หรือข้อเท็จจริงใหม่ๆ ก็เก็บไว้ในหน่วยความจำ เวลาที่ต้องการใช้ก็เพียงแค่ดึงความรู้นี้มาใช้ ถ้าเรามองว่าระบบปัญญาประดิษฐ์มีหน้าที่รับอินพุต (X1,…,Xn) แล้วทำการหาเอาต์พุต (Y1,…,Yn) = f(X1,…,Xn) โดยที่ f เป็นฟังก์ชันใดๆ ในการคำนวณเอาต์พุตหรืออาจเป็นการอนุมานหาค่าเอาต์พุตจากอินพุตก็ได้ ดังนั้นการเรียนรู้โดยการจำก็คือการเก็บคู่ลำดับ [(X1,…,Xn),(Y1,…,Yn)] ไว้ในหน่วยความจำ หลังจากนั้นเมื่อเราต้องการหา f(X1,…,Xn) ใหม่ก็ทำโดยการดึง (Y1,…,Ym) จากคู่ลำดับนี้เท่านั้นโดยไม่ต้องคำนวณหรืออนุมานซ้ำอีกครั้งซึ่งโดยมากจะเสียต้นทุนและเวลาสูง

จะเห็นได้ว่าแนวคิดนี้ง่ายแต่ไม่ได้หมายความว่าการเรียนรู้นี้จะไม่มีประสิทธิภาพ มนุษย์เราก็เรียนรู้โดยการจำด้วยเช่นกันหรือซอฟต์แวร์ในปัจจุบันหลายตัวก็สามารถจำชื่อไฟล์ที่ผู้ใช้ใช้งานครั้งล่าสุดได้และช่วยให้การเปิดไฟล์ทำได้ง่ายขึ้นมีประโยชน์ในการใช้งานจริง

ในการเรียนรู้โดยการจำนี้ สิ่งที่เราต้องพิจารณาเพิ่มเติมได้แก่ (1) การจัดการหน่วยความจำ (memory organization) ที่ต้องมีประสิทธิภาพสามารถดึงความรู้ที่เก็บไว้ได้อย่างรวดเร็ว (2) ความเสถียรภาพของสภาพแวดล้อมต้องไม่เปลี่ยนแปลงอย่างรวดเร็วจนส่งผลให้ความรู้ที่เก็บไว้ไม่ถูกต้องเมื่อเวลาเปลี่ยนไป (3) ความสมดุลย์ระหว่างการคำนวณใหม่กับการจัดเก็บ ต้องมีสมดุลย์ที่ดีไม่จัดเก็บมากไปจนทำให้การค้นคืนคู่ลำดับที่จัดเก็บมีประสิทธิภาพต่ำ ส่งผลให้ประสิทธิภาพโดยรวมลดลงเพราะเสียเวลามากไปเพื่อตรวจสอบว่าเป็นความรู้ที่อยู่ในหน่วยความจำหรือไม่ ดังนั้นควรเลือกจำเฉพาะความรู้ที่ใช้บ่อย

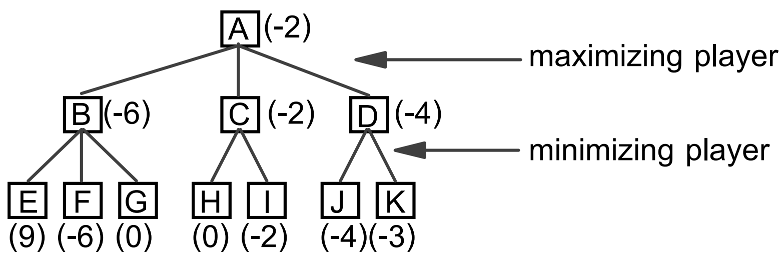
แม้ว่าแนวคิดนี้จะง่ายแต่หากใช้ได้ตรงกับงานประยุกต์หนึ่งๆ ก็จะส่งผลให้ประสิทธิภาพของระบบเพิ่มขึ้นได้อย่างดี ดังเช่นที่จะแสดงในการเรียนรู้โดยการจำของโปรแกรมเชกเกอร์ส (checkers) ของ Samuel เชกเกอร์ส[2] เป็นเกมที่เล่นให้เก่งยากและการพัฒนาโปรแกรมเชกเกอร์สให้เล่นแข่งชนะมนุษย์ก็ไม่ง่าย ตาเดินที่เป็นไปได้ทั้งหมดของเชกเกอร์สมีประมาณ 10^{40} ตาเดิน ทำให้การสร้างตาเดินทั้งหมดโดยโปรแกรมก่อนที่จะตัดสินใจว่าจะเลือกตาเดินต่อไปอย่างไรไม่สามารถทำได้อย่างรวดเร็วโดยคอมพิวเตอร์ประสิทธิภาพไม่สูงนัก ดังนั้นโปรแกรมเล่นเกมประเภทนี้จะสร้างตาเดินได้เท่าที่เวลาอำนวย เช่นถ้าต้องเดินหมากภายใน 1 นาที โปรแกรมสร้างตาเดินที่เป็นไปได้เท่าไรก็เท่านั้น แล้วเลือกจากตาเดิน

---

[2] เป็นเกมคล้ายกับหมากฮอสไทย แต่มีจำนวนเบี้ยแต่ละฝ่าย 12 ตัว

การค้นหาต้นไม้เกมน้อยสุดมากสุด

ที่สร้างได้ ลักษณะของอัลกอริทึมประเภทนี้เป็นการค้นหาแบบหนึ่งซึ่งมีผู้เล่นสองฝ่ายคือโปรแกรมกับฝ่ายตรงข้าม อัลกอริทึมที่นิยมใช้ในเกมประเภทนี้ก็คือ *การค้นหาต้นไม้เกมน้อยสุดมากสุด (minimax game-tree search)* ดังแสดงในรูปที่ 6–7



รูปที่ 6–7 ต้นไม้เกมน้อยสุดมากสุด

การค้นหาต้นไม้เกมน้อยสุดมากสุดแตกต่างจากการค้นหาในปริภูมิสถานะทั่วไป ที่ต้นไม้เกมมีผู้สร้างสถานะในต้นไม้ 2 คนคือ *ผู้เล่นฝ่ายทำมากสุด (maximizing player)* โดยทั่วไปคือโปรแกรมและ*ผู้เล่นฝ่ายทำน้อยสุด (minimizing player)* หรือฝ่ายตรงข้าม ในรูปสถานะ A เป็นสถานะเริ่มต้น (แทนการจัดเรียงตัวหมากบนกระดานหนึ่งๆ) สมมติว่า A มีสถานะลูกคือ B, C และ D การสร้างสถานะลูกทำโดยการเดินหมากทุกรูปแบบที่เป็นไปได้ และผู้เล่นที่ทำหน้าที่สร้างสถานะลูกคือผู้เล่นฝ่ายทำมากสุด จากสถานะ B, C และ D ผู้เล่นฝ่ายตรงข้ามหรือผู้เล่นฝ่ายทำน้อยสุดจะสร้างสถานะลูกทั้งหมดของ B, C และ D ได้เป็น E, F,..., K ในทางปฏิบัติโปรแกรมจะทำหน้าที่คำนวณสถานะทั้งหมดด้วยตัวเอง

ตัวเลขที่สถานะแต่ละตัวแสดงค่าความดีของสถานะนั้นๆ ค่าเหล่านี้เป็นค่าของ*ผู้เล่นฝ่ายทำมากสุด* ถ้าค่ามากแสดงว่าโอกาสชนะของผู้เล่นฝ่ายทำมากสุดมีมาก แต่ถ้าน้อยแสดงว่าผู้เล่นฝ่ายทำมากสุดมีโอกาสชนะน้อย ดังนั้นหน้าที่ของผู้เล่นฝ่ายทำมากสุดคือพยายามทำให้ตัวเลขเหล่านี้มีค่ามากโดยเลือกเส้นทางที่จะทำให้ค่าสูงสุด ตัวเลขเหล่านี้แบ่งเป็น 2 จำพวกคือ (1) ตัวเลขที่สถานะปลายต้นไม้ (ใบ) (9, -6, 0,..., -3) และ (2) ตัวเลขที่สถานะเริ่มต้นและสถานะภายในต้นไม้ เราเรียกว่าตัวเลขที่ปลายต้นไม้ว่า *ค่าประเมินสถิต (static evaluation value)* ค่าเหล่านี้เป็นค่าฮิวริสติกที่วัดค่าความดีของการจัดเรียงตัวหมากบนกระดานว่าโอกาสชนะของผู้เล่นฝ่ายทำมากสุดมีมากแค่ไหน ค่าประเมินสถิตนี้วัดจากจำนวนเบี้ยของเราว่ามากกว่าของฝ่ายตรงข้ามมากน้อยแค่ไหน จำนวนขุน (king) ของเรามีมากกว่าฝ่ายตรงข้ามแค่ไหน ตำแหน่งของตัวหมากของเราอยู่ในตำแหน่งที่ได้เปรียบฝ่ายตรงข้ามมากน้อยแค่ไหน เป็นต้น

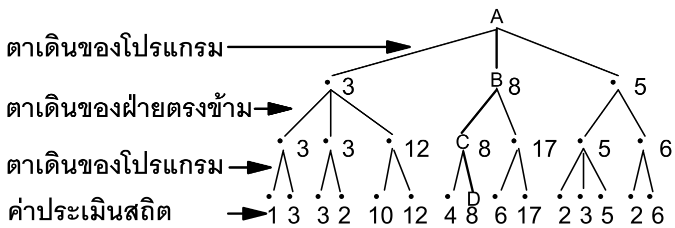
ตัวเลขที่สถานะเริ่มต้นและสถานะภายในต้นไม้เรียกว่า *ค่าแบ็คอัพ (backup value)* เป็นค่าที่ได้จากการส่งค่าประเมินสถิตจากด้านล่างย้อนกลับขึ้นไปทางด้านบนทีละระดับ ในการคำนวณค่าแบ็คอัพนั้นจะพิจารณาเป็น 2 กรณีคือ (1) กรณีที่ผู้เล่นฝ่ายทำน้อยสุดเป็นผู้สร้างสถานะลูก ค่าแบ็คอัพของสถานะพ่อแม่จะเป็นค่าต่ำสุดในจำนวนค่าทั้งหมดของสถานะลูก (2) กรณีที่ผู้เล่นฝ่ายทำมากสุดเป็นผู้สร้างสถานะลูก ค่าแบ็คอัพของสถานะพ่อแม่จะเป็นค่าสูงสุดในจำนวนค่าทั้งหมดของลูก เช่นกรณีการคำนวณค่าแบ็คอัพของสถานะ B ซึ่งเป็นกรณีที่ (1) นั้น ค่าของ B จะเท่ากับ min{9,-6,0} = -6 กรณีการคำนวณค่าแบ็คอัพของสถานะ A ซึ่งเป็นกรณีที่ (2) นั้น ค่าของ A จะเท่ากับ max{-6,-2,-4} = -2 เนื่องจากค่าในต้นไม้เป็นค่าที่แสดงโอกาสที่ผู้เล่นฝ่ายทำมากสุดมีโอกาสชนะ ดังนั้นผู้เล่นฝ่ายทำมากสุดจึงต้องพยายามทำให้ค่าที่ได้มีค่ามากสุด ส่วนผู้เล่นฝ่ายทำน้อยสุดมีหน้าที่สกัดกั้นไม่ให้ผู้เล่นฝ่ายทำมากสุดมีโอกาสชนะ ดังนั้นจึงต้องพยายามทำให้ค่าที่ได้มีค่าน้อยสุด และเป็นที่มาของชื่ออัลกอริทึมนี้

จากตัวอย่างในรูปด้านบน เมื่อผู้เล่นฝ่ายทำมากสุดจะเลือกตาเดินก็ควรเลือกเส้นทางตามค่าแบ็คอัพ กล่าวคือเมื่ออยู่ที่สถานะ A ควรเลือกตาเดินไปยังสถานะ C ซึ่งคาดว่าหลังจากนั้นฝ่ายผู้เล่นทำน้อยสุดน่าจะเดินไปยัง I สังเกตว่าในจำนวนสถานะทั้งหมดค่าที่มากสุดคือ 9 ของสถานะ E แต่อย่างไรก็ดีผู้เล่นฝ่ายทำมากสุดไม่มีโอกาสที่จะได้ค่าแบ็คอัพเป็น 9 ได้ แม้ว่าตนเองจะเดินจากสถานะ A ไปยัง B เพราะว่ามีผู้เล่นฝ่ายทำน้อยสุดพยายามขัดขวางให้ค่าที่ได้มีค่าน้อยสุด ถ้าผู้เล่นฝ่ายทำมากสุดเดินมายังสถานะ B ผู้เล่นฝ่ายทำน้อยสุดก็จะเดินไปยัง F ทำให้โอกาสชนะของผู้เล่นฝ่ายทำมากสุดเหลือ -6 (อย่าลืมว่าตัวเลขในต้นไม้นี้เป็นค่าที่แสดงโอกาสชนะของผู้เล่นฝ่ายทำมากสุดเท่านั้น) ในรูปที่ 6–7 นั้นแสดงการค้นหาที่มองล่วงหน้า 2 ก้าวเดิน (2 moves look-ahead)

ค่าแบ็คอัพที่คำนวณได้ของสถานะ A นี้จะไม่เท่ากับค่าประเมินสถิตของ A เนื่องจากว่าถ้าเราวัดค่าประเมินสถิตก็คือการคำนวณค่าฮิวริสติกของ A โดยตรงโดยดูที่ตัวหมาก ณ สถานะ A แต่ค่าแบ็คอัพของ A คือการมองล่วงหน้าต่อจากนี้อีก 2 ก้าวเดินในทุกเส้นทางแล้วคำนวณเส้นทางที่น่าจะเป็นที่สุด (เส้นทางที่ผู้เล่นทั้งสองเลือกตาเดินได้ดีที่สุด) แล้วส่งค่าประเมินสถิตที่ปลายต้นไม้ย้อนกลับมาที่สถานะ A ดังนั้นค่าแบ็คอัพจะมีความถูกต้องแม่นยำมากกว่าค่าประเมินสถิตโดยตรงของ A

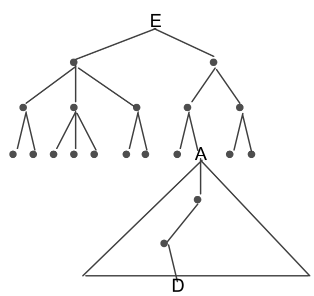
ค่าแบ็คอัพที่ได้จากการมองล่วงหน้า 2 ก้าวเดินมีความแม่นยำมากกว่าค่าประเมินสถิต ในทำนองเดียวกันค่าแบ็คอัพที่ได้จากการมองล่วงหน้า 3 ก้าวเดินก็ย่อมมีความแม่นยำมากกว่ามองล่วงหน้า 2 ก้าวเดิน ยิ่งเราเพิ่มการมองล่วงหน้าได้ลึกเท่าไร ความแม่นยำของค่าแบ็คอัพที่คำนวณได้ก็ยิ่งสูงขึ้นเท่านั้น และถ้าเราสามารถมองล่วงหน้าจนถึงสถานะที่จบ

เกม ค่าที่ได้ก็จะถูกต้องสมบูรณ์ อย่างไรก็ดีในทางปฏิบัติเราไม่สามารถมองล่วงหน้าจนจบเกมได้เนื่องจากข้อจำกัดด้านเวลา โปรแกรมจะเดินตัวหมากได้เก่งถ้าสามารถมองล่วงหน้าได้ลึกมากๆ



รูปที่ 6–8 ตัวอย่างต้นไม้เกมในกรณีมองล่วงหน้า 3 ก้าวเดิน

การเพิ่มความสามารถของโปรแกรมสามารถทำได้โดยการเพิ่มจำนวนก้าวเดินที่จะมองล่วงหน้า เพราะยิ่งมองล่วงหน้าได้ลึกค่าแบ็กอัพก็จะถูกต้องมากขึ้น และดังเช่นที่กล่าวแล้วว่าจากข้อจำกัดเรื่องเวลาที่โปรแกรมสามารถใช้ได้ การกระจายสถานะเพิ่มขึ้นจึงไม่สามารถทำได้ แต่อย่างไรก็ดีโดยการใช้การเรียนรู้โดยการจำจะสามารถเพิ่มความสามารถของโปรแกรมให้เสมือนกับว่าโปรแกรมมองล่วงหน้าได้มากขึ้น เราใช้การเรียนรู้โดยการจำเพื่อที่จะเก็บคู่ลำดับค่าแบ็คอัพของสถานะเริ่มต้น เช่นในรูปที่ 6–8 หลังจากที่เราได้ค้นหาล่วงหน้า 3 ก้าวเดินและพบว่าค่าแบ็กอัพของ A เท่ากับ 8 แล้ว เราจะจำคู่ลำดับ [A,8] ไว้ในหน่วยความจำ เมื่อ A ถูกพบอีกครั้งที่ปลายของต้นไม้เกมต้นอื่นในการเล่นครั้งใหม่ เราจะไม่ต้องหาค่าประเมินสถิตของ A แต่จะนำค่าแบ็กอัพของ A มาใช้แทน การนำเอาค่าแบ็คอัพมาใช้แทนที่จะคำนวณค่าประเมินสถิตนั้น นอกจากจะมีข้อดีที่รวดเร็วขึ้นเนื่องจากการคำนวณค่าประเมินสถิตจะใช้เวลานานกว่าแล้ว ยังส่งผลดีอีกประการที่สำคัญดังแสดงรูปที่ 6–9 ซึ่งแสดงต้นไม้เกมต้นหนึ่งที่มี E เป็นสถานะแรกและมีสถานะที่ปลายต้นไม้สถานะหนึ่งคือ A



รูปที่ 6–9 การเรียนรู้โดยการจำเพิ่มประสิทธิภาพของการค้นหา

ด้วยการใช้ค่าแบ็คอัพของ A แทนที่จะใช้ค่าประเมินสถิตก็เสมือนกับว่าที่จุด A นี้ได้รวมการค้นหาอีก 3 ก้าวเดินล่วงหน้าเข้าไว้ด้วย ดังนั้นที่ E แม้ว่าด้วยข้อจำกัดทางเวลาทำให้เราค้นหาได้เพียง 3 ก้าวเดินล่วงหน้า แต่ก็เสมือนกับว่าในเส้นทางที่รวม A จะเป็นการค้นหาล่วงหน้าถึง 6 ก้าวเดินล่วงหน้า และด้วยการจำคู่ลำดับระหว่างสถานะกับค่าแบ็คอัพไว้จำนวนมากก็จะทำให้การค้นหาเพิ่มจำนวนก้าวเดินล่วงหน้าเป็น 3, 6, 9, … ตามลำดับ ซึ่งส่งผลให้ประสิทธิภาพของโปรแกรมเพิ่มขึ้น

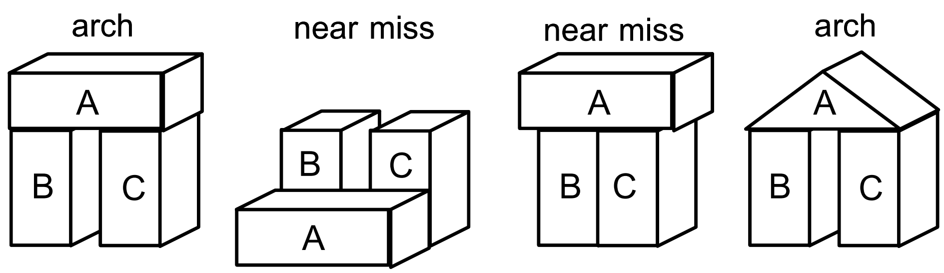
ตัวอย่างนี้แสดงให้เห็นการประยุกต์ใช้การเรียนรู้โดยการจำที่ส่งผลให้ประสิทธิภาพของงานที่กระทำดีขึ้นอย่างชัดเจน และโปรแกรมเรียนรู้การเล่นเกมเชกเกอร์ส์นี้ยังมีการจัดการหน่วยความจำอย่างประหยัดโดยจัดเก็บเฉพาะตำแหน่งตัวหมากบนกระดานของผู้เล่นฝ่ายทำมากสุดฝ่ายเดียว และเมื่อจะใช้กับผู้เล่นฝ่ายทำน้อยสุดก็สลับตำแหน่งของตัวหมากกลับด้านกันเท่านั้น นอกจากนั้นยังมีการทำดัชนีเพื่อดึงตำแหน่งตัวหมากบนกระดานให้ได้อย่างรวดเร็วโดยใช้คุณสมบัติของกระดาน เช่นจำนวนตัวหมาก การมีหรือไม่มีขุน เพื่อใช้เป็นดัชนี และยังได้จัดการปัญหาความสมดุลย์ระหว่างการจัดเก็บกับการคำนวณใหม่โดยใช้วิธีที่เรียกว่าการแทนที่ตัวที่ถูกใช้น้อยสุด (least recently used replacement) วิธีนี้พยายามจะไม่จัดเก็บคู่ลำดับให้มากมายเกินไปเพราะจะทำให้การค้นคืนคู่ลำดับใช้เวลามาก โดยกำหนดจำนวนคู่ลำดับที่จะจำเป็นค่าคงที่ค่าหนึ่ง เช่น 100,000 คู่ลำดับ จากนั้นคู่ลำดับใดที่ถูกใช้น้อยสุด (เมื่อจำไว้แล้วถูกพบในต้นไม้เกมอื่นน้อยสุด) จะถูกลบออกจากหน่วยความจำแล้วแทนที่ด้วยคู่ลำดับใหม่ตัวอื่น วิธีนี้ทำโดยกำหนดอายุให้กับคู่ลำดับแต่ละคู่และทุกครั้งที่คู่ลำดับถูกเรียกมาใช้อายุของมันจะลดลงครึ่งหนึ่ง และทุกครั้งที่มีการจำคู่ลำดับใหม่ อายุของคู่ลำดับอื่นทุกตัวในหน่วยความจำจะถูกบวกเพิ่ม 1 หน่วย จากนั้นตัวที่มีอายุมากสุดจะถูกลบออกจากหน่วยความจำ

6.3 การเรียนรู้โดยการวิเคราะห์ความแตกต่าง

การเรียนรู้โดยการวิเคราะห์ความแตกต่าง (learning by analyzing differences) ถูกพัฒนาโดย Winston ในปีคศ. 1975 [Winston, 1992] แม้ว่าจะเป็นวิธีการเรียนรู้ที่ค่อนข้างเก่ามากแล้วก็ตาม แต่ว่าแนวคิดต่างๆ สามารถนำไปใช้ในการเรียนรู้แบบใหม่ๆ ได้อย่างดี ในที่นี้จึงยกวิธีการเรียนรู้แบบนี้มาเพื่อศึกษาแนวคิดของการเรียนรู้เชิงอุปนัย การเรียนรู้โดยการวิเคราะห์ความแตกต่างนี้ใช้เรียนรู้มโนทัศน์ทางโครงสร้าง (structural concept) ในโดเมนปัญหาโลกของบล็อก เช่น arch, tent หรือ house เป็นต้น วิธีการเรียนรู้นี้จะวิเคราะห์ความแตกต่างที่ปรากฏในลำดับของตัวอย่างที่ผู้สอนป้อนให้ โดยตัวอย่างสอน (training example) มี 2 ประเภทคือ *ตัวอย่างบวก (positive example)* และ*ตัวอย่างลบ (negative example)* ตัวอย่างบวกคือตัวอย่างที่ถูกต้องของมโนทัศน์ (concept) ที่สอน เช่นจะสอน house ตัวอย่างบวกก็จะเป็นบ้านหลังที่หนึ่ง บ้านหลังที่สอง เป็นต้น ตัวอย่างลบคือตัวอย่างที่ไม่ถูกต้อง เช่นจะสอน house ตัวอย่างลบก็จะเป็นเต้นท์หลังที่หนึ่ง โรงเรียนหลังที่หนึ่งเหล่านี้เป็นต้น

ตัวอย่างบวก และ ตัวอย่างลบ

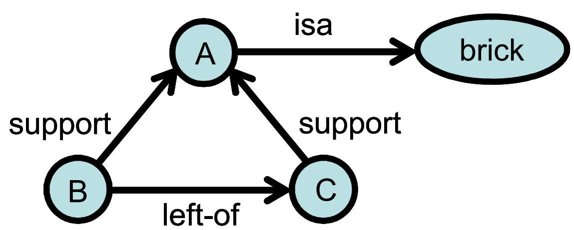
สำหรับการเรียนรู้โดยการวิเคราะห์ความแตกต่างนี้ ตัวอย่างลบที่ผู้สอนให้จะต้องเป็นตัวอย่างลบแบบที่เรียกว่า *พลาดน้อย (near miss)* กล่าวคือตัวอย่างลบแบบพลาดน้อยนี้จะต่างจากตัวอย่างบวกเพียงเล็กน้อย เช่นจะสอน house ตัวอย่างลบแบบพลาดน้อยก็จะเป็นบ้านที่ขาดประตู หรือบ้านที่ไม่มีหลังคา เป็นต้น การเรียนรู้แบบนี้ผู้สอนจะจัดเตรียมลำดับของตัวอย่างไว้ค่อนข้างดีเพื่อให้โปรแกรมเรียนรู้สามารถวิเคราะห์ความต่างของตัวอย่างบวกกับตัวอย่างลบแบบพลาดน้อย ด้านล่างนี้ยกตัวอย่างการเรียนรู้มโนทัศน์ arch ซึ่งมีลำดับของตัวอย่างที่จะสอนดังแสดงในรูปที่ 6–10



รูปที่ 6–10 ตัวอย่างบวกและตัวอย่างลบแบบพลาดน้อยของ arch

ในรูป 'arch' และ 'near miss' หมายถึงตัวอย่างบวกและตัวอย่างลบแบบพลาดน้อยตามลำดับ จากตัวอย่างที่ให้ทั้งสี่ตัวนี้ โปรแกรมจะเรียนรู้ว่าอะไรคือ arch เมื่อเราดูตัวอย่างข้างต้น เราจะพอเข้าใจได้ว่าตัวอย่างบวกตัวแรกบอกว่า arch คือสิ่งที่ประกอบด้วยอิฐ (brick) แนวตั้ง 2 ก้อนและอิฐแนวนอน 1 ก้อนที่ถูกรองรับด้วยอิฐแนวตั้ง ตัวอย่างที่สองอธิบายสิ่งที่ไม่ใช่ arch ว่าคือสิ่งที่ประกอบด้วยอิฐแนวตั้ง 2 ก้อนและอิฐแนวนอนซึ่งไม่ถูกรองรับด้วยอิฐแนวตั้ง ตัวอย่างที่ 3 และ 4 แสดงตัวอย่างของ arch และสิ่งที่ไม่ใช่ตามลำดับ โปรแกรมเรียนรู้นี้ใช้การแทนความรู้เพื่อแสดงมโนทัศน์ในรูปของ*ข่ายงานความหมาย (semantic network)* การแทนความรู้แบบนี้จะประกอบด้วยบัพ (node) และเส้นเชื่อม (link) บัพแสดงวัตถุและเส้นเชื่อมแทนความสัมพันธ์ระหว่างวัตถุในโดเมนนั้น

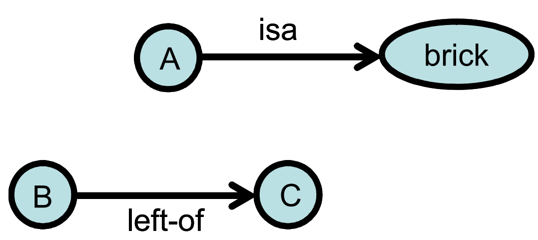
จากตัวอย่างบวกตัวที่หนึ่ง โปรแกรมจะสร้างคำอธิบายเริ่มต้น (initial description) ของมโนทัศน์ดังรูปที่ 6–11



รูปที่ 6–11 คำอธิบายเริ่มต้น

บัพ A มีเส้นเชื่อม isa แสดงความสัมพันธ์ว่า A เป็น brick และเส้นเชื่อมจาก B และ C ไป A คือ support แสดงความสัมพันธ์ว่า B และ C รองรับ A และมีเส้นเชื่อม left-of แสดงว่า B อยู่ด้านซ้ายของ C ส่วนเส้นเชื่อมอื่นๆ ที่ไม่เกี่ยวข้องโดยตรงกับ concept ขอละไว้ในที่นี้ เช่นเส้นเชื่อม isa จาก B ไปยังบัพ brick เป็นต้น

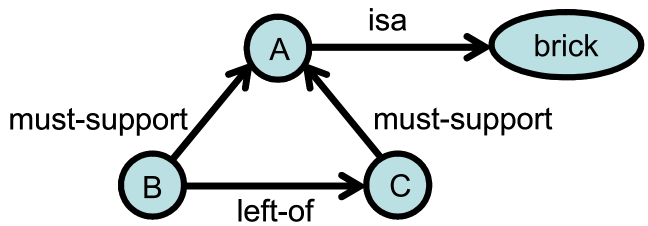
จากตัวอย่างลบแบบพลาดน้อยตัวที่สอง โปรแกรมสร้างคำอธิบายของตัวอย่างลบได้ดังรูปที่ 6–12



รูปที่ 6–12 คำอธิบายของตัวอย่างตัวที่สอง

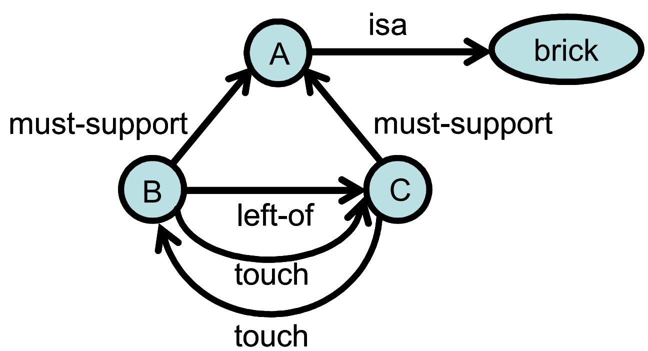
ที่จุดนี้โปรแกรมจะวิเคราะห์หาความแตกต่างของตัวอย่างที่ถูกกับที่ผิดโดยการจับคู่บัพและเส้นเชื่อม และพบว่าเส้นเชื่อม support ซึ่งต่างกันในตัวอย่างทั้งสองจำเป็นสำหรับมโนทัศน์ arch โปรแกรมจึงใส่เงื่อนไขเพิ่มเข้าไปในคำอธิบายในรูปที่ 6–11 โดยใช้เส้นเชื่อมใหม่ชื่อ must-support แทนที่เส้นเชื่อมเดิมดังแสดงในรูปที่ 6–13 เราเรียกคำอธิบายใหม่ที่ได้นี้ว่า *โมเดลระหว่างวิวัฒนาการ (evolving model)* ในกรณีนี้ตัวอย่างลบให้ข้อมูลสำหรับการใช้*ฮิวริสติกเส้นเชื่อมจำเป็น (require-link heuristic)* ที่ใส่เงื่อนไขที่มากขึ้นในเส้นเชื่อมเดิม เราเรียกการทำเช่นนี้ว่าเป็นฮิวริสติกแบบหนึ่งเนื่องจากว่าเป็นการคาดคะเนจากเหตุผลของความแตกต่างระหว่างตัวอย่างที่น่าจะเป็น แต่ก็อาจไม่ถูกต้องเสมอไปก็เป็นได้ ในกรณีนี้ตัวอย่างลบแบบพลาดน้อยเป็นตัวอย่างที่ให้ข้อมูลสำหรับ*การแจงจำเพาะของมโนทัศน์ (specialization of concept)* ซึ่งหมายถึงว่าคำอธิบายของมโนทัศน์จะถูกทำให้แคบลง มีเงื่อนไขมากขึ้น ตรงกับตัวอย่างจำนวนน้อยลง

การแจงจำเพาะของมโนทัศน์



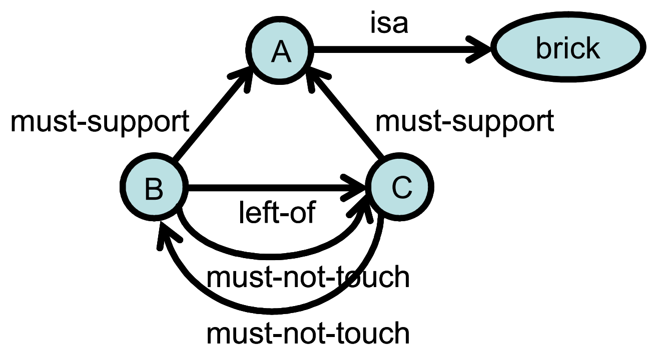
รูปที่ 6–13 โมเดลระหว่างวิวัฒนาการ

จากตัวอย่างลบตัวที่สาม โปรแกรมสร้างคำอธิบายของตัวอย่างลบได้ดังรูปที่ 6–14



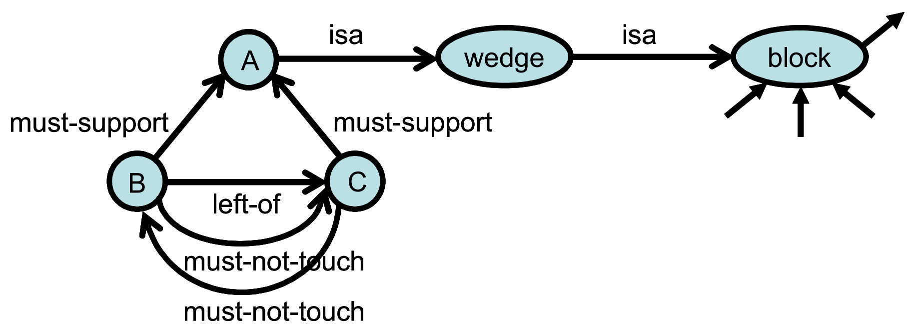
รูปที่ 6–14 คำอธิบายของตัวอย่างลบตัวที่สาม

โปรแกรมหาความแตกต่างระหว่างคำอธิบายของตัวอย่างที่สามกับโมเดล พบว่ามีเส้นเชื่อม touch อยู่ในตัวอย่างลบซึ่งไม่มีในโมเดล ดังนั้นโปรแกรมจึงเพิ่มเส้นเชื่อมเข้าไปในโมเดลและปรับโมเดลใหม่ได้ดังรูปที่ 6–15 ในกรณีนี้ตัวอย่างลบให้ข้อมูลสำหรับ*ฮิวริสติกเส้นเชื่อมห้าม (forbid-link heuristic)*



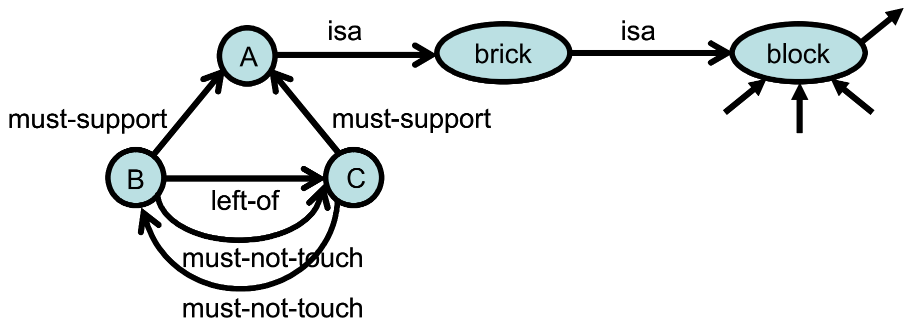
รูปที่ 6–15 โมเดลหลังรับตัวอย่างตัวที่สาม

จากตัวอย่างบวกตัวที่สี่ โปรแกรมสร้างคำอธิบายของตัวอย่างได้ดังรูปที่ 6–16



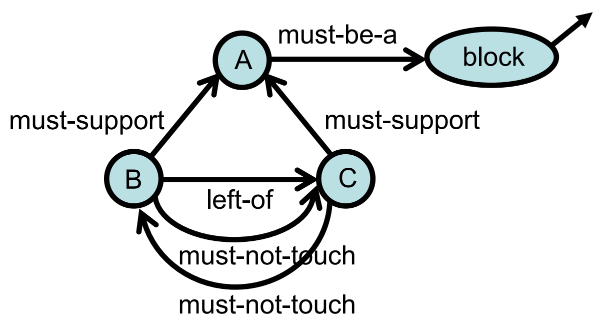
รูปที่ 6–16 คำอธิบายของตัวอย่างบวกตัวที่สี่

สมมติว่าเรามีความสัมพันธ์ของต้นไม้จำแนกประเภท (classification tree) ว่าวัตถุหนึ่งๆ จัดอยู่ในประเภทอะไรในฐานความรู้ของเราด้วย เช่นในที่นี้ wedge จัดเป็นวัตถุหนึ่งในประเภทของ block ในทำนองเดียวกัน brick ก็จัดเป็นวัตถุหนึ่งในประเภทของ block ด้วย โมเดลของเราในรูปที่ 6–15 เมื่อนำมาเขียนใหม่ให้รวมความสัมพันธ์ของต้นไม้จำแนกประเภทเข้าไปด้วยก็จะได้ดังรูปที่ 6–17



รูปที่ 6–17 โมเดลหลังรับตัวอย่างตัวที่สามที่รวมต้นไม้จำแนกประเภทด้วย

เมื่อนำโมเดลเปรียบเทียบกับคำอธิบายตัวอย่างที่สี่ข้างต้นจะพบว่ามีความแตกต่างกันที่ brick กับ wedge และทั้งคู่ต่างก็เป็นวัตถุในประเภทของ block ดังนั้นเราจึงแทนที่ brick ด้วยประเภทที่สูง (กว้าง) กว่าคือ block ได้เป็นโมเดลในรูปที่ 6–18 เราเรียกฮิวริสติกแบบนี้ว่า*ฮิวริสติกปีนต้นไม้ (climb-tree heuristic)*



รูปที่ 6–18 โมเดลเมื่อรับตัวอย่างครบทุกตัว

ในกรณีที่เราไม่มีต้นไม้จำแนกประเภท โปรแกรมจะสร้างประเภทใหม่คือ “brick-or-wedge” ขึ้นมาเพื่อใช้แทนบัพ block ในรูปที่ 6–18 เราเรียกฮิวริสติกแบบนี้ว่า*ฮิวริสติกขยายเซต (enlarge-set heuristic)* และถ้าหากว่าในกรณีที่เราไม่มีวัตถุอื่นอยู่ในโดเมนนี้อีกเลยที่นอกเหนือจาก brick และ wedge เราก็สามารถตัดเส้นเชื่อม isa ออกได้เลยเพื่อเป็นการลดเงื่อนไข และในกรณีนี้เราเรียกฮิวริสติกนี้ว่า *ฮิวริสติกตัดเส้นเชื่อม (drop-link heuristic)* ในกรณีเหล่านี้ตัวอย่างบวกทำหน้าที่สำหรับ*การวางนัยทั่วไปของมโนทัศน์ (generalization of concept)* ซึ่งหมายถึงว่าคำอธิบายของมโนทัศน์จะถูกทำให้กว้างขึ้น มีเงื่อนไขน้อยลง ตรงกับตัวอย่างจำนวนมากขึ้น

การวางนัยทั่วไปของมโนทัศน์

อัลกอริทึมของโปรแกรมเรียนรู้โดยวิเคราะห์ความแตกต่างแสดงในตารางที่ 6–10

ตารางที่ 6–10 อัลกอริทึมการเรียนรู้โดยวิเคราะห์ความแตกต่าง

```
Algorithm: Learning by Analyzing Differences

•  Near-miss is for specialize model by using
   – require-link heuristic
   – forbid-link heuristic
•  Positive example is for generalize mode by using
   – climb-tree heuristic
   – enlarge-set heuristic
   – drop-link heuristic

Speicialization algorithm
Specialization to make a model more restrictive by:
(1) Match the evolving model to the example to establish
    correspondences among parts.
(2) Determine whether there is a single, most important
    difference between the evolving model and the near
    miss.
    – If there is a single, most important difference,
      (a) If the evolving model has a link that is not in
          the near miss, use the require-link heuristic
      (b) If the near miss has a link that is not in the
          model, use the forbid-link heuristic
    – Otherwise, ignore the example.

Generalization algorithm
Generalization to make a model more permissive by:
(1) Match the evolving model to the example to establish
    correspondences among parts.
(2) For each difference, determine the difference type:
    – If a link points to a class in the evolving model
       different from the class to which the link points
       in the example,
      (a) If the classes are part of a classification
          tree, use the climb-tree heuristic
      (b) If the classes form an exhaustive set, use the
          drop-link heuristic
      (c) Otherwise, use the enlarge-set heuristic
(3) If a link is missing in the example, use the drop-
    link heuristic
(4) Otherwise, ignore the difference.
```

6.4 เวอร์ชันสเปซ

เวอร์ชันสเปซ (version space) [Mitchell, 1977] เรียนรู้คำอธิบายที่อธิบายตัวอย่างบวกและไม่อธิบายตัวอย่างลบ รูปที่ 6–19 ด้านล่างแสดงตัวอย่างของการเรียนมโนทัศน์ car ซึ่งใช้*การแทนความรู้แบบกรอบ (frame)*

Car023	
origin:	Japan
manufacturer:	Honda
color:	Blue
decade:	1970
type:	Economy

รูปที่ 6–19 ตัวอย่างบวกของมโนทัศน์ car

กรอบประกอบด้วยชื่อกรอบ ในที่นี้คือ Car023 และสล็อต (slot) ในที่นี้สล็อตมี 5 ตัวคือ origin, manufacture, color, decade และ type ซึ่งแสดงคุณสมบัติทั้งห้าอย่างของรถยนต์ สมมติว่าสล็อตแต่ละตัวมีค่าที่เป็นไปได้ตามตารางที่ 6–11 ด้านล่างนี้

ตารางที่ 6–11 ค่าที่เป็นไปได้ของสล็อตแต่ละตัว

origin	∈	{Japan, USA, Britain, Germany, Italy}
manufacturer	∈	{Honda, Toyota, Ford, Chrysler, Jaguar, BMW, Fiat}
color	∈	{Bule, Green, Red, White}
decade	∈	{1950, 1960, 1970, 1980, 1990, 2000}
type	∈	{Economy, Luxury, Sports}

การเรียนรู้โดยเวอร์ชันสเปซจะแสดงคำอธิบายมโนทัศน์ในรูปของสล็อตและค่าของสล็อต เช่นถ้าเป็นมโนทัศน์ “Japanese economy car” จะแสดงได้ดังรูปที่ 6–20 โดยที่ x1, x2 และ x3 เป็นตัวแปรสามารถถูกแทนด้วยค่าคงที่ใดๆ

origin:	Japan
manufacturer:	x1
color:	x2
decade:	x3
type:	Economy

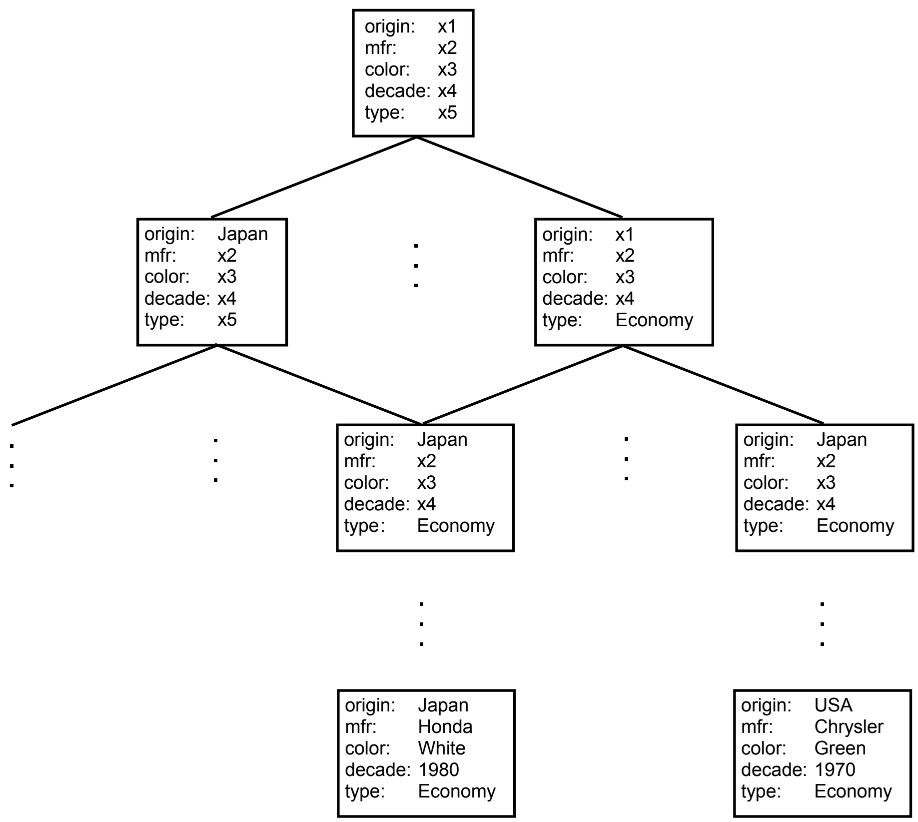
รูปที่ 6–20 มโนทัศน์ “Japanese economy car”

ปัญหาการเรียนรู้ที่เราสนใจคือ กำหนดค่าที่เป็นไปได้ของสล็อต ตัวอย่างบวกและตัวอย่างลบให้ จงหาคำอธิบายมโนทัศน์ที่*สอดคล้องกับ (consistent with)* ตัวอย่าง (อธิบายตัวอย่างบวกและไม่อธิบายตัวอย่างลบ)

สอดคล้องกับ

วิธีการเรียนรู้เวอร์ชันสเปซนี้มองว่าการเรียนรู้คือการค้นหาในปริภูมิค้นหาที่เรียกว่า *ปริภูมิมโนทัศน์ (concept space)* ซึ่งเป็นปริภูมิที่มีสมาชิกแต่ละตัวเป็นคำอธิบายในรูปของกรอบโดยที่สมาชิกเหล่านี้มีลำดับบางส่วน (partial ordering) ในลำดับนี้สมาชิกตัวที่*มีนัยทั่วไปกว่า (more general)* จะอยู่ด้านบนของสมาชิกตัวที่*จำเพาะกว่า (more specific)* ดังแสดงในรูปที่ 6–21 โดยที่ตัวอักษรเล็ก (x1, x2, x3, x4 และ x5) แสดงตัวแปรซึ่งสามารถแทนที่ด้วยค่าคงที่ได้ ส่วนตัวอักษรใหญ่และตัวเลข (เช่น Japan, Economy, 1980) แสดงค่าคงที่

มีนัยทั่วไปกว่า และ จำเพาะกว่า



รูปที่ 6–21 ปริภูมิมโนทัศน์

ตัวที่อยู่บนสุดในรูปแสดง*มโนทัศน์มีนัยทั่วไปสุด (most general concept)* ส่วนตัวที่อยู่ล่างสุดแสดง*มโนทัศน์จำเพาะสุด (most specific concept)* ซึ่งเป็นตัวอย่างหนึ่งๆ และตัวที่

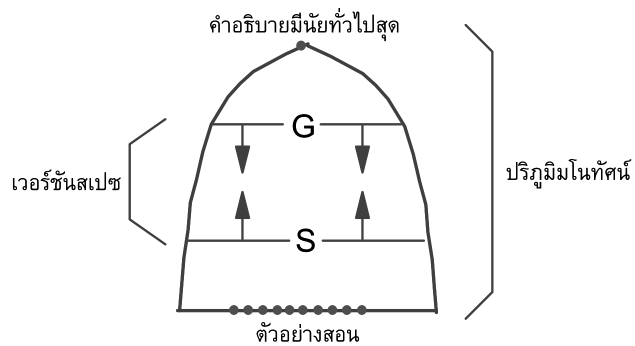
เป็น*คำอธิบายมโนทัศน์เป้าหมาย (target concept description)* จะอยู่ระหว่างบนสุดกับล่างสุด วิธีการเรียนรู้เวอร์ชันสเปซคือการสร้างเซตย่อยประกอบด้วย*สมมติฐาน (hypothesis)* ที่อยู่ในปริภูมิมโนทัศน์ที่สอดคล้องกับตัวอย่างสอน และเรียกเซตย่อยนี้ว่า*เวอร์ชันสเปซ (version space)*

เวอร์ชันสเปซที่สร้างขึ้นนี้จะต้องประกอบด้วยสมมติฐาน (คำอธิบาย) ที่สอดคล้องกับตัวอย่างที่เคยพบมาทั้งหมด วิธีการสร้างเวอร์ชันสเปซที่ทำได้วิธีหนึ่งคือการแจงสมาชิกทุกตัวในปริภูมิมโนทัศน์ แล้วตรวจสอบกับตัวอย่างสอนทุกตัวที่รับเข้ามา หากสมาชิกตัวใดไม่สอดคล้องกับตัวอย่างก็ตัดทิ้งไป คงไว้เฉพาะตัวที่สอดคล้องเท่านั้น อย่างไรก็ตามปริภูมิมโนทัศน์มีขนาดใหญ่มาก วิธีการนี้จึงไม่มีประสิทธิภาพ ตัวอย่างเช่นในกรณีของปัญหาในรูปที่ 6–21 เมื่อพิจารณาค่าที่เป็นไปได้ในสล็อตแต่ละตัวตามตารางที่ 6–11 จะเห็นว่าปริภูมิมโนทัศน์มีขนาดเท่ากับ ((5+1)(7+1)(4+1)(6+1)(3+1)) = 6,720 และในกรณีที่จำนวนสล็อตมีมากขึ้นเช่น 10 ตัว และสล็อตแต่ละตัวมีค่าที่เป็นไปได้มากขึ้นเช่น 10 ค่า จะได้ว่าปริภูมิมโนทัศน์จะยิ่งมีขนาดใหญ่ขึ้นมาก ($\approx 2.6 \times 10^{10}$)

วิธีการเรียนรู้เวอร์ชันสเปซจะใช้วิธีการแทนสเปซด้วยวิธีที่ประหยัดและมีประสิทธิภาพในการค้นหามากโดยจะใช้เซตย่อย 2 เซตเรียกว่าเซต G และเซต S

- เซต G ประกอบด้วยคำอธิบายมีนัยทั่วไปสุดที่ยังสอดคล้องกับตัวอย่างที่เคยพบมาทั้งหมด
- เซต S ประกอบด้วยคำอธิบายจำเพาะสุดที่ยังสอดคล้องกับตัวอย่างที่เคยพบมาทั้งหมด

เวอร์ชันสเปซจะอยู่ระหว่างเซต G กับ S ดังแสดงในรูปที่ 6–22



รูปที่ 6–22 เวอร์ชันสเปซ

หลักการของเวอร์ชันสเปซคือทุกครั้งที่เราได้รับตัวอย่างบวกตัวใหม่เราจะทำให้ S มีนัยทั่วไป (general) มากขึ้นและทุกครั้งที่ได้รับตัวอย่างลบเราจะทำให้ G จำเพาะ

(specific) มากขึ้น จนในที่สุด S และ G ลู่เข้าสู่ค่าเดียวที่เป็นคำอธิบายมโนทัศน์เป้าหมาย อัลกอริทึมการเรียนรู้ของเวอร์ชันสเปซเป็นดังตารางที่ 6–12 นี้

ตารางที่ 6–12 อัลกอริทึมการเรียนรู้เวอร์ชันสเปซ

```
Algorithm: Version-Space-Candidate-Elimination

1. G := {most general description}
2. S := {first positive example}
3. Accept a new example E
   IF E is positive THEN
      • Remove from G any descriptions that do not cover
        the example.
      • Update S to contain the most specific set of
        descriptions in the version space that cover the
        example and the current elements of S.
   ELSE IF E is negative THEN
      • Remove from S any descriptions that cover the
        example.
      • Update G to contain the most genereal set of
        desciptions in the version space that do not
        cover the example.
4. IF S and G are both singleton sets and S = G THEN
        Output the element
   ELSE IF S and G are both singleton sets and S<>G THEN
        examples were inconsistent
   ELSE goto 3.
```

6.4.1 ตัวอย่างการเรียนรู้มโนทัศน์ car

กำหนดเซตตัวอย่างสอนที่ประกอบด้วยตัวอย่างบวกและตัวอย่างลบดังรูปที่ 6–23 อัลกอริทึมในตารางที่ 6–12 จะเรียนรู้ดังต่อไปนี้

origin:	Japan
mfr:	Honda
color:	Blue
decade:	1980
type:	Economy

(+)

origin:	Japan
mfr:	Toyota
color:	Green
decade:	1970
type:	Sports

(–)

origin:	Japan
mfr:	Toyota
color:	Blue
decade:	1990
type:	Economy

(+)

origin:	USA
mfr:	Chrysler
color:	Red
decade:	1980
type:	Economy

(–)

origin:	Japan
mfr:	Honda
color:	White
decade:	1980
type:	Economy

(+)

รูปที่ 6–23 ตัวอย่างสอนของมโนทัศน์ car

- จากตัวอย่างบวก 3 ตัวและตัวอย่างลบ 2 ตัวตามรูปด้านบน เราเริ่มด้วยการสร้าง G และ S ตามตัวอย่างแรกได้

  G = {(x1,x2,x3,x4,x5)}

  S = {(Japan,Honda,Blue,1980,Economy)}

  โดยที่ (x1,x2,x3,x4,x5) เป็นค่าของสล็อตที่ 1, 2, 3, 4, 5 ตามลำดับ

- ตัวอย่างที่ 2 เป็นตัวอย่างลบ ดังนั้นเราทำการแจงจำเพาะของ G เพื่อไม่ให้เวอร์ชันสเปซอธิบายหรือ*คลุม (cover)* ตัวอย่างลบนี้โดยการเปลี่ยนตัวแปรให้เป็นค่าคงที่

  คลุม

  G = {(x1,Honda,x3,x4,x5), (x1,x2,Blue,x4,x5),
  (x1,x2,x3,1980,x5), (x1,x2,x3,x4,Economy)}

  S ไม่เปลี่ยนแปลง = {(Japan,Honda,Blue,1980,Economy)}

- ตัวอย่างที่ 3 เป็นบวก = (Japan,Toyota,Blue,1990,Economy) เรากำจัดคำอธิบายใน G ที่ไม่สอดคล้องกับตัวอย่างนี้

  G = {(x1,x2,Blue,x4,x5), (x1,x2,x3,x4,Economy)}

  และทำการวางนัยทั่วไปของ S ให้รวมตัวอย่างนี้

  S = {(Japan,x2,Blue,x4,Economy)}

  ที่จุดนี้เราได้เวอร์ชันสเปซที่แสดง "Japanese blue economy car", "blue car" หรือ "Economy car"

- ตัวอย่างที่ 4 เป็นลบ = (USA,Chrysler,Red,1980,Economy)

  G = {(x1,x2,Blue,x4,x5), (x1,x2,Blue,x4,Economy),
  (Japan,x2,x3,x4,Economy)}

  S = {(Japan,x2,Blue,x4,Economy)}

- ตัวอย่างที่ 5 เป็นบวก = (Japan,Honda,White,1980,Economy)

  G = {(Japan,x2,x3,x4,Economy)}

  S = {(Japan,x2,x3,x4,Economy)}

  ที่จุดนี้ ได้คำตอบ S=G แสดง "Japanese economy car"

6.4.2 ข้อจำกัดของเวอร์ชันสเปซ

ดังที่แสดงในตัวอย่างด้านบนนี้ เวอร์ชันสเปซสามารถเรียนรู้ได้จากตัวอย่างที่สอน อย่างไรก็ดีเวอร์ชันสเปซก็ยังมีข้อจำกัดดังต่อไปนี้

- อัลกอริทึมเรียนรู้นี้เป็นแบบทำน้อยสุด (least-commitment algorithm) กล่าวคือในแต่ละขั้นตอนเวอร์ชันสเปซจะถูกตัดเล็มให้เล็กลงน้อยที่สุดเท่าที่เป็นไปได้ ดังนั้นถึงแม้ว่าตัวอย่างบวกทุกตัวเป็น Japanese cars ก็ตาม อัลกอริทึมก็จะไม่ตัดความน่าจะเป็นที่มโนทัศน์อาจจะรวม car อื่นๆ ทิ้งจนกระทั่งพบตัวอย่างลบ ซึ่งหมายถึงเวอร์ชันสเปซจะเรียนรู้ไม่สำเร็จถ้าไม่มีตัวอย่างลบเลย
- กระบวนการค้นหาเป็นการค้นหาแนวกว้างแบบทั้งหมด (exhaustive breadth-first search) ซึ่งเห็นได้จากการปรับค่าของเซ็ต G ที่จะทดลองทำที่สล็อตทุกตัวให้ได้ทุกแบบที่เป็นไปได้ ดังนั้นทำให้อัลกอริทึมมีประสิทธิภาพต่ำในกรณีที่ปริภูมิใหญ่มากๆ ซึ่งอาจทำให้ดีขึ้นโดยใช้ฮิวริสติกเข้าช่วยในการค้นหาโดยลองเปลี่ยนตัวแปรเป็นค่าคงที่ในบางสล็อตที่น่าจะนำไปสู่คำตอบก่อน เป็นต้น
- เซ็ต S ประกอบด้วยสมาชิกเพียงตัวเดียวเพราะว่าตัวอย่างบวก 2 ตัวใดๆ มีการวางนัยทั่วไปเพียงหนึ่งเดียว ดังนั้นเวอร์ชันสเปซจึงไม่สามารถเรียนมโนทัศน์แบบ 'หรือ' (disjunctive concept) ซึ่งเป็นมโนทัศน์ที่อยู่ในรูปของ or เช่น "Japanese econamy car or Japanese sport car"
- ข้อจำกัดอีกอย่างของเวอร์ชันสเปซคือไม่สามารถจัดการกับตัวอย่างมีสัญญาณรบกวน (noisy example) ซึ่งเป็นตัวอย่างที่มีข้อมูลบางส่วนผิดพลาด เช่นถ้าตัวอย่างตัวที่ 3 ใน รูปที่ 6–23 (Japan Toyota Blue 1990 Economy) เราให้ประเภทผิดเป็นตัวอย่างลบ (-) อัลกอริทึมจะไม่สามารถเรียนมโนทัศน์ "Japanese economy car" ได้ถูกต้อง

6.5 การเรียนรู้ต้นไม้ตัดสินใจ

การเรียนรู้ต้นไม้ตัดสินใจ (decision tree learning) [Quinlan, 1986;Quinlan, 1993] เป็นการเรียนรู้ที่ใช้การแทนความรู้อยู่ในรูปของต้นไม้ตัดสินใจ ใช้สำหรับจำแนกประเภทของตัวอย่าง วิธีการเรียนรู้คล้ายกับการเรียนรู้เวอร์ชันสเปซโดยเริ่มจากการป้อนตัวอย่างเข้าไปในระบบ ซึ่งตัวอย่างที่ป้อนให้เป็นตัวอย่างบวกกับตัวอย่างลบก็ได้และนอกจากนั้นเรายังสามารถป้อนตัวอย่างที่มากกว่า 2 *ประเภท (class)* ได้ กล่าวคือแทนที่จะมีแต่บวกกับลบ ก็สามารถมีได้หลายประเภท เช่นในการรู้จำตัวอักษร จะมีตัวอย่างมาจากหลายประเภทที่แตกต่างกันคือประเภท 'ก', ประเภท 'ข', ประเภท 'ค', ประเภท 'ง' ฯลฯ แต่เพื่อให้ง่ายต่อการอธิบาย ตัวอย่างที่จะยกให้ดูต่อไปนี้จะมีเพียง 2 ประเภทเท่านั้น โดยเราจะใช้ปัญหาการผึ่งแดดเป็นตัวอย่างอธิบาย

ประเภท

ปัญหาการผึ่งแดด : เราไปเที่ยวที่ชายทะเลและพบว่าคนที่ไปผึ่งแดดตามชายทะเล บางคนก็จะมีผิวเปลี่ยนเป็นสีแทน แต่บางคนต้องได้รับความทรมานจากผิวไหม้ เราต้องการหาว่าอะไรคือปัจจัยที่ทำให้คนที่ไปผึ่งแดดตามชายทะเลแล้วผิวไหม้หรือไม่ไหม้ โดยข้อมูลที่สังเกตได้ประกอบด้วยความแตกต่างของสีผม น้ำหนัก ส่วนสูงของผู้ที่ไปผึ่งแดด และการใช้โลชัน ซึ่งบางคนก็ใช้โลชัน บางคนก็ไม่ใช้

สมมติว่าเราบันทึกข้อมูลของตัวอย่างสอนได้ตามตารางที่ 6–13 เพื่อใช้สร้างต้นไม้ตัดสินใจ

ตารางที่ 6–13 ตัวอย่างสอนที่สังเกตได้

ประเภท

คุณสมบัติ

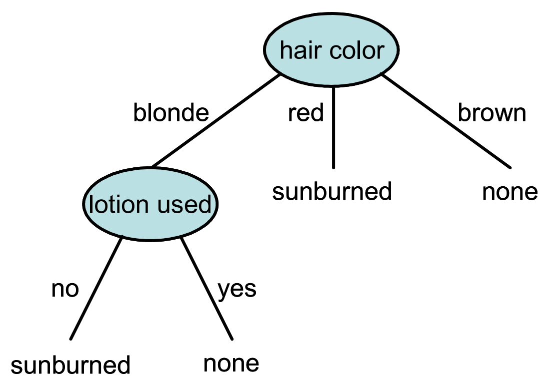
ค่า

Name	Hair	Height	Weight	Lotion	Result
Sarah	blonde	average	light	no	sunburned
Dana	blonde	tall	average	yes	none
Alex	brown	short	average	yes	none
Annie	blonde	short	average	no	sunburned
Emily	red	average	heavy	no	sunburned
Pete	brown	tall	heavy	no	none
John	brown	average	heavy	no	none
Katie	blonde	short	light	Yes	none

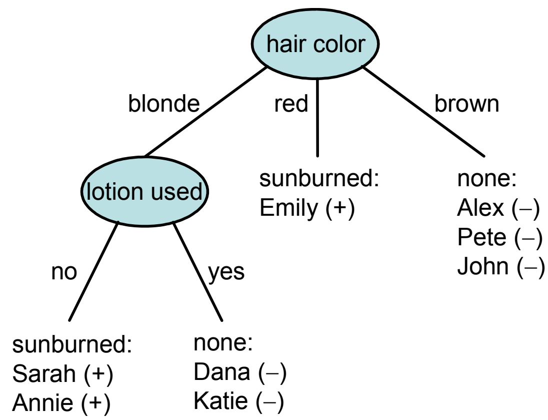
แถวแรกสุดในตารางแสดง*คุณสมบัติ (attribute)* ของข้อมูลซึ่งประกอบด้วยชื่อ (Name) สีผม (Hair) ส่วนสูง (Height) น้ำหนัก (Weight) และการใช้โลชัน (Lotion) ส่วนสดมภ์สุดท้ายแทนประเภทของตัวอย่าง คุณสมบัติ Name ไว้สำหรับอ้างอิงตัวอย่างและไม่มีผลต่อการจำแนกข้อมูล เราจึงจะไม่ใช้ Name ในการเรียนรู้ด้านล่างนี้ แต่ละแถวในตารางนอกเหนือจากแถวแรกแทนตัวอย่างหนึ่งตัว เช่นแถวที่สองแสดงตัวอย่างของคนที่ชื่อ Sarah ซึ่งมีสีผม ส่วนสูง น้ำหนัก และการใช้โลชัน เป็น blond, average, light และ no ตามลำดับ ตัวอย่างนี้อยู่ในประเภท sunburned เป็นต้น

เมื่อเราได้ข้อมูลตัวอย่างทั้ง 8 ตัวแล้ว สิ่งที่เราต้องการทำก็คือทำการวางนัยทั่วไปของตัวอย่างเพื่อสร้างเป็นโมเดลสำหรับทำนายประเภทของข้อมูลของคนอื่นที่ไม่ได้บันทึกไว้ วิธีที่ง่ายสุดก็คือการเรียนรู้โดยการจำ และเมื่อมีตัวอย่างในอนาคตที่เรายังไม่ทราบประเภทและถ้าต้องการทำนาย เราก็นำตัวอย่างนั้นมาเปรียบเทียบกับตัวอย่างสอนในตาราง ถ้าตัวอย่างที่นำมาเปรียบเทียบมีคุณสมบัติตรงกับข้อมูลในตาราง เราก็นำประเภทของตัวอย่างสอนที่ตรงกันทำนายให้กับตัวอย่างนั้น อย่างไรก็ดีวิธีการนี้ทำงานได้ไม่ดีนักเนื่องจากว่าโอกาสที่เราจะพบตัวอย่างทดสอบที่ตรงกับตัวอย่างสอนมีน้อย สมมติว่าสีผมมีค่าที่เป็นไปได้ทั้งหมด 3 ค่าคือ bronde, brown, red ส่วนสูงมีได้ 3 ค่าคือ tall, average, short น้ำหนักมีได้ 3 ค่าคือ heavy, average, light และการใช้โลชันมีได้ 2 ค่าคือ yes, no เราจะพบว่าความน่าจะเป็นที่ตัวอย่างทดสอบจะตรงกับตัวอย่างสอนมีค่าเท่ากับ 8/(3x3x3x2) = 15% (สมมติว่าความน่าจะเป็นที่ค่าแต่ละค่าสำหรับคุณสมบัติหนึ่งๆ มีความน่าจะเป็นที่จะเกิดขึ้นเท่ากัน)

การเรียนรู้ต้นไม้ตัดสินใจจะทำการวางนัยทั่วไปของข้อมูลโดยสร้างเป็นโมเดลอยู่ในรูปต้นไม้ตัดสินใจ ตัวอย่างของต้นไม้ตัดสินใจแสดงในรูปที่ 6–24



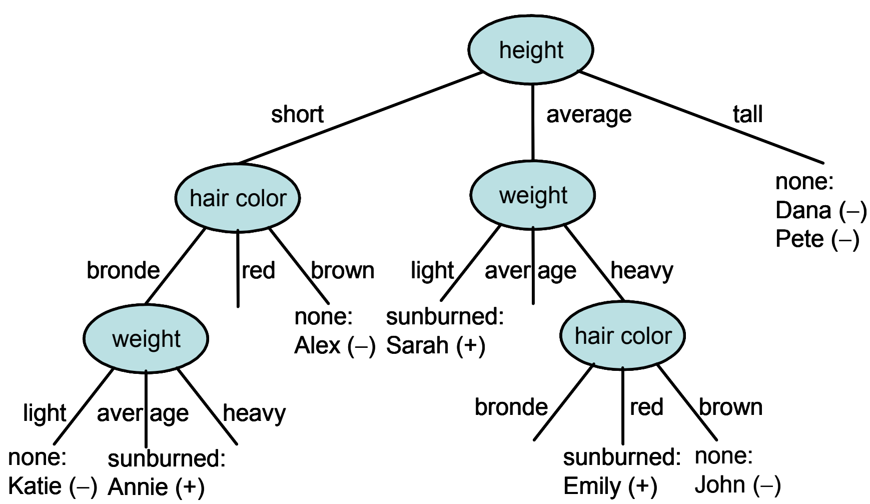
รูปที่ 6–24 ตัวอย่างของต้นไม้ตัดสินใจ



รูปที่ 6–26 ต้นไม้ตัดสินใจที่สอดคล้องกับตัวอย่างสอน

ต้นไม้ตัดสินใจในรูปที่ 6–26 ด้านบนนี้สอดคล้องกับตัวอย่างสอนทุกตัว หมายความว่า ถ้านำตัวอย่างสอนมาตรวจสอบด้วยต้นไม้ตัดสินใจ ต้นไม้จะทำนายประเภทได้ถูกต้องทุกตัว การตรวจสอบทำโดยดูว่าตัวอย่างมี hair colr เป็นค่าอะไร ถ้าเป็น brown จะทำนายประเภทเป็น none ถ้าเป็น red จะทำนายประเภทเป็น sunburned แต่ถ้าเป็น bronde จะดู lotion used ด้วยว่าถ้าเป็น no แสดงว่าประเภทเป็น sunburned แต่ถ้าเป็น yes แสดงว่าประเภทเป็น none

โดยทั่วไปต้นไม้ตัดสินใจที่สอดคล้องกับตัวอย่างสอนมีได้มากกว่า 1 ต้น เช่น ต้นไม้ในรูปที่ 6–27 ก็เป็นต้นไม้อีกต้นหนึ่งที่สอดคล้องกับตัวอย่าง



รูปที่ 6–27 ต้นไม้ตัดสินใจอีกต้นหนึ่งที่มีความซับซ้อนมากกว่าต้นแรก

เมื่อเราพิจารณาต้นไม้ต้นแรกในรูปที่ 6–26 และต้นที่สองในรูปที่ 6–27 เราพบว่าต้นไม้ต้นแรกน่าจะถูกต้องมากกว่าต้นที่สอง เนื่องจากว่าในต้นแรกนั้นใช้คุณสมบัติสีผมและการใช้โลชันในการจำแนกข้อมูล ซึ่งน่าจะเป็นไปได้เพราะสีผมมีความสัมพันธ์อย่างมากกับความแข็งแรงของผิวเรา คนที่มีผมสีน้ำตาลน่าจะมีผิวที่แข็งแรงไปผึ่งแดดแล้วมักจะไม่เป็นอะไร ส่วนผมสีแดงมีผิวบอบบาง และผมสีบรอนซ์มีผิวปานกลางซึ่งจะขึ้นกับการใช้โลชันหรือไม่ใช้ ถ้าใช้ไปผึ่งแดดก็จะไม่เป็นอะไร ถ้าไม่ใช้ไปผึ่งแดดแล้วผิวจะไหม้ ส่วนต้นไม้ต้นที่สองเราไม่สามารถอธิบายได้ว่าทำไมส่วนสูงที่ใช้เป็นบัพรากหรือน้ำหนักที่บัพในระดับถัดมาจึงมีความสำคัญต่อการที่ผิวจะไหม้หรือไม่ไหม้

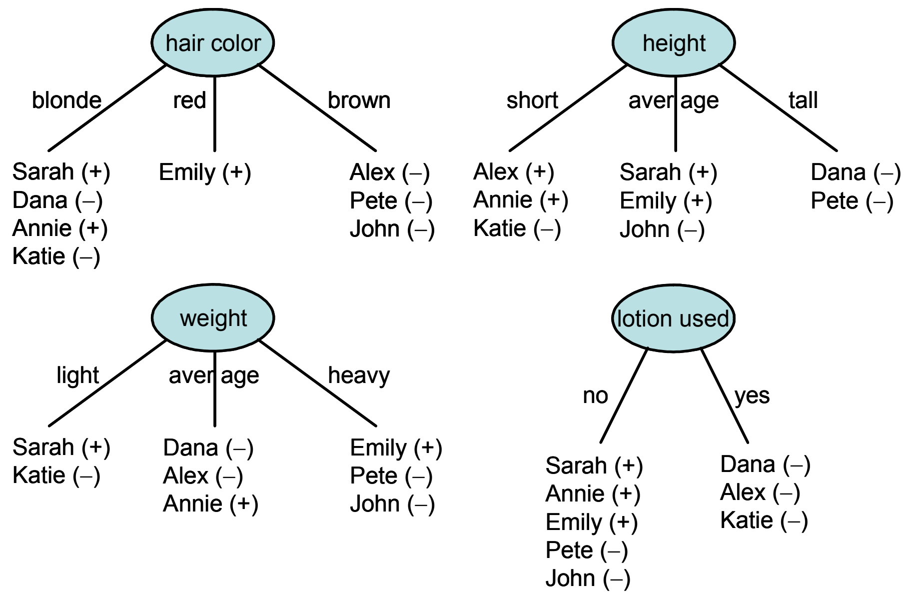
ความแตกต่างที่เห็นได้เด่นชัดอีกประการของต้นไม้ทั้งสองคือจำนวนบัพภายในต้นไม้ จะเห็นได้ว่าจำนวนบัพของต้นไม้ต้นที่สองมีจำนวนมากกว่า หรือกล่าวอีกนัยหนึ่งคือต้นไม้ต้นที่สองมีความซับซ้อนมากกว่า หรือกล่าวได้ว่าต้นไม้ต้นแรกมีขนาดเล็กกว่าต้นไม้ต้นที่สองโดยที่ขนาดวัดจากจำนวนบัพภายในต้นไม้

มีดโกนของอ็อคแคม

ในการเรียนรู้ของเครื่องนั้น เรามีฮิวริสติกตัวหนึ่งที่นิยมใช้กันและพบว่าทำงานได้อย่างดีในหลายกรณีเรียกว่า *มีดโกนของอ็อคแคม (occam's razor)* เมื่อเรานำมีดโกนของอ็อคแคมมาใช้ในการเลือกต้นไม้ตัดสินใจ เราก็จะได้ว่า “ต้นไม้ตัดสินใจขนาดเล็กที่สุดที่สอดคล้องกับตัวอย่างสอนคือต้นไม้ตัดสินใจที่ดีที่สุด” อย่างไรก็ดีถ้าเราจะหาต้นไม้ตัดสินใจที่มีขนาดเล็กที่สุดที่สอดคล้องกับตัวอย่างสอนก็ไม่สามารถทำได้โดยง่าย เราต้องสร้างต้นไม้ตัดสินใจจำนวนมาก โดยเริ่มจากต้นไม้ที่มีจำนวนบัพ 1 บัพทุกต้นที่เป็นไปได้แล้วดูว่ามีต้นไหนหรือไม่ที่สอดคล้องกับตัวอย่างสอน ถ้าไม่มีก็เพิ่มจำนวนบัพเป็น 2 บัพ ทำอย่างนี้ไปจนกระทั่งพบต้นไม้ตัดสินใจที่สอดคล้องกับตัวอย่าง เราพบว่าวิธีการนี้จะมีจำนวนต้นไม้ที่ต้องสร้างเป็นฟังก์ชันเลขยกกำลังของจำนวนคุณสมบัติซึ่งไม่เหมาะกับการใช้งานจริง

6.5.1 ฟังก์ชันเกนสำหรับการเลือกบัพทดสอบ

ส่วนนี้จะกล่าวถึงวิธีการเลือกบัพเพื่อสร้างต้นไม้ตัดสินใจโดยใช้หลักการว่า เนื่องจากจุดมุ่งหมายของการสร้างต้นไม้คือเพื่อจำแนกประเภทของข้อมูลเพื่อให้ตัวอย่างในแต่ละบัพใบอยู่ในประเภทเดียวกันทั้งหมด ดังนั้นบัพที่ดีควรเป็นบัพที่แยกตัวอย่างออกเป็นเซตย่อยตามกิ่งของบัพนั้นและเซตย่อยในแต่ละกิ่งประกอบด้วยสมาชิกที่ส่วนใหญ่เป็นประเภทเดียวกันมากที่สุด ตัวอย่างในรูปที่ 6–28 แสดงผลของบัพทดสอบแต่ละบัพ



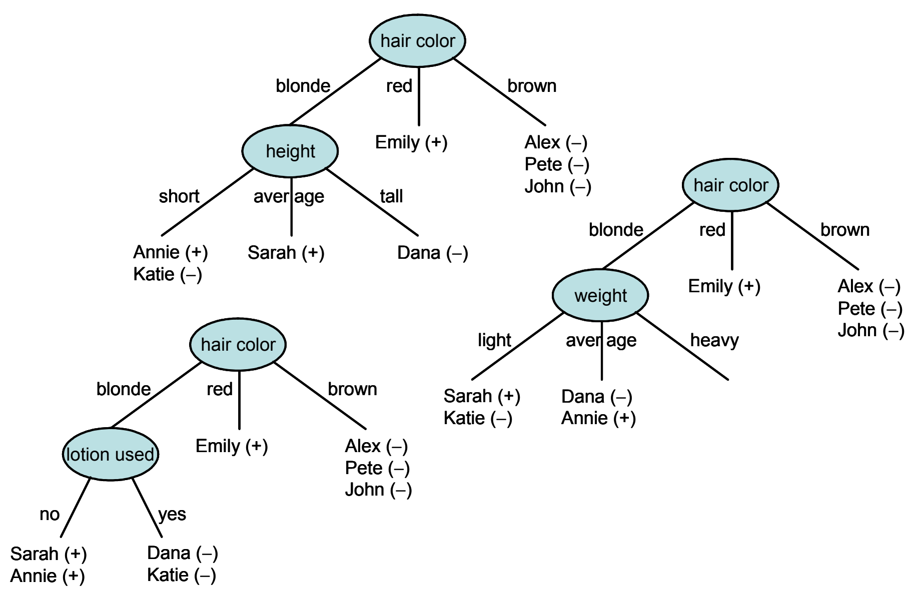
รูปที่ 6–28 ผลของบัพทดสอบแต่ละบัพในการแยกตัวอย่างเพื่อเลือกบัพราก

ดังแสดงในรูปด้านบน บัพแต่ละบัพแยกตัวอย่างได้ดีต่างกันดังนี้

- ในกรณีของบัพทดสอบเป็น hair color สามารถแยกตัวอย่างเป็น 3 เซตย่อย เซตย่อยแรก (blonde) มีตัวอย่างของ 2 ประเภทปนกันอยู่ ส่วนเซตย่อยที่ 2 (red) และ 3 (brown) มีตัวอย่างของประเภท sunburned และ none อยู่อย่างเดียวตามลำดับ ซึ่งกรณีนี้ hair color แยกตัวอย่างได้ดีเมื่อเทียบกับบัพอื่นด้านล่างนี้
- ในกรณีของบัพทดสอบเป็น height สามารถแยกตัวอย่างเป็น 3 เซตย่อย เซตย่อยแรก (short) และเซตย่อยที่ 2 (average) มีตัวอย่าง 2 ประเภทปนกันอยู่ในแต่ละเซต ส่วนเซตย่อยที่ 3 (tall) มีตัวอย่างของ none อยู่อย่างเดียว จะเห็นว่ากรณีนี้แยกตัวอย่างไม่ดีเท่ากรณีของ hair color
- ในกรณีของบัพทดสอบเป็น weight สามารถแยกตัวอย่างเป็น 3 เซตย่อย เซตย่อยทั้งสามเซต (light, average, heavy) ต่างก็มีตัวอย่าง 2 ประเภทปนกันอยู่ ซึ่งกรณีนี้เป็นกรณีที่แย่ที่สุด
- ในกรณีของบัพทดสอบเป็น lotion used สามารถแยกตัวอย่างเป็น 2 เซตย่อย เซตย่อยแรก (no) มีตัวอย่างของ 2 ประเภทปนกัน ส่วนเซตย่อยที่ 2 (yes) มีตัวอย่างของ none อยู่อย่างเดียว และในเซตย่อยแรกสมาชิกส่วนใหญ่ของเซตนี้เป็น sunburned (+) เกือบทั้งหมด ซึ่งเมื่อเทียบกับกรณีแรกของบัพ hair color ถือได้ว่ามีความสามารถในการแยกตัวอย่างได้ใกล้เคียงกัน

ดังจะเห็นได้ในตัวอย่างด้านบนนี้ เราพอจะเปรียบเทียบได้ว่าบัพหนึ่งๆ มีความสามารถในการแยกตัวอย่างดีกว่าบัพอีกบัพหนึ่งหรือไม่ แต่ในบางกรณีเช่น hair color กับ lotion used เราอาจบอกความแตกต่างไม่ได้ ดังนั้นเราจำเป็นต้องหาการวัดที่สามารถบอกความต่างได้อย่างชัดเจนโดยการนิยามฟังก์ชันเพื่อวัดประสิทธิภาพของบัพออกเป็นค่าที่วัดได้อย่างละเอียด ซึ่งจะกล่าวต่อไป

ณ จุดนี้ เพื่อให้เข้าใจถึงการสร้างต้นไม้ตัดสินใจจะขอสมมติว่าเรามีฟังก์ชันนั้นอยู่ และสมมติว่าระหว่างบัพ hair color กับ lotion used ค่าฟังก์ชันของ hair color ดีกว่า และได้รับเลือกเป็นบัพราก ในขั้นตอนต่อไปก็คือเราต้องพิจารณาต่อว่าในแต่ละกิ่งของบัพราก มีกิ่งใดหรือไม่ที่ยังมีตัวอย่างจากหลายประเภทปะปนกันอยู่ ถ้ามีเราต้องเพิ่มบัพของคุณสมบัติอื่นเพื่อช่วยแยกตัวอย่างที่ยังปะปนกันอยู่ต่อไป ในกรณีของบัพ hair color ในรูปที่ 6–28 กิ่ง blonde เท่านั้นที่ยังมีตัวอย่างจากหลายประเภทปนกัน เราจึงจำเป็นต้องเพิ่มบัพต่อไปโดยทดลองเพิ่มคุณสมบัติที่เหลือทั้งสาม (height, weight และ lotion used) ผลที่ได้แสดงในรูปที่ 6–29



รูปที่ 6–29 ผลของบัพทดสอบแต่ละบัพในการแยกตัวอย่างเพื่อเลือกบัพต่อจากกิ่ง blonde

จากรูปด้านบนจะเห็นได้ว่าบัพ lotion used เป็นบัพที่แยกตัวอย่างออกเป็นเซตย่อยโดยที่แต่ละเซตย่อยมีสมาชิกอยู่ในประเภทเดียวกัน ดังนั้นบัพ lotion used ถูกเลือกในขั้นตอนนี้

6.5.2 ฟังก์ชันเกน

ฟังก์ชันเกน

ทฤษฎีสารสนเทศ และ เอ็นโทรปี

ฟังก์ชันที่ใช้วัดความสามารถในการแยกตัวอย่างของบัพทดสอบที่มีประสิทธิภาพมากฟังก์ชันหนึ่งคือ *ฟังก์ชันเกน (Gain function)* ฟังก์ชันเกนนี้ใช้ในการตัดสินใจเลือกคุณสมบัติที่จะใช้เป็นรากหรือบัพในต้นไม้โดยการคำนวณค่าเกนของคุณสมบัติแต่ละตัวเมื่อทดลองใช้คุณสมบัตินั้นแบ่งตัวอย่าง แล้วเลือกคุณสมบัติที่มีค่าเกนสูงที่สุดมาเป็นรากหรือบัพ ค่าเกนนี้คำนวณได้โดยใช้ความรู้จาก*ทฤษฎีสารสนเทศ (information theory)* ซึ่งมีสาระสำคัญคือค่าสารสนเทศหรือของข้อมูลขึ้นอยู่กับค่าความน่าจะเป็นของข้อมูลซึ่งสามารถวัดอยู่ในรูปของบิต (bits) จากสูตร

$$\text{ค่าสารสนเทศของข้อมูล} = -\log_2(\text{ความน่าจะเป็นของข้อมูล}) \tag{6.2}$$

เอ็นโทรปี

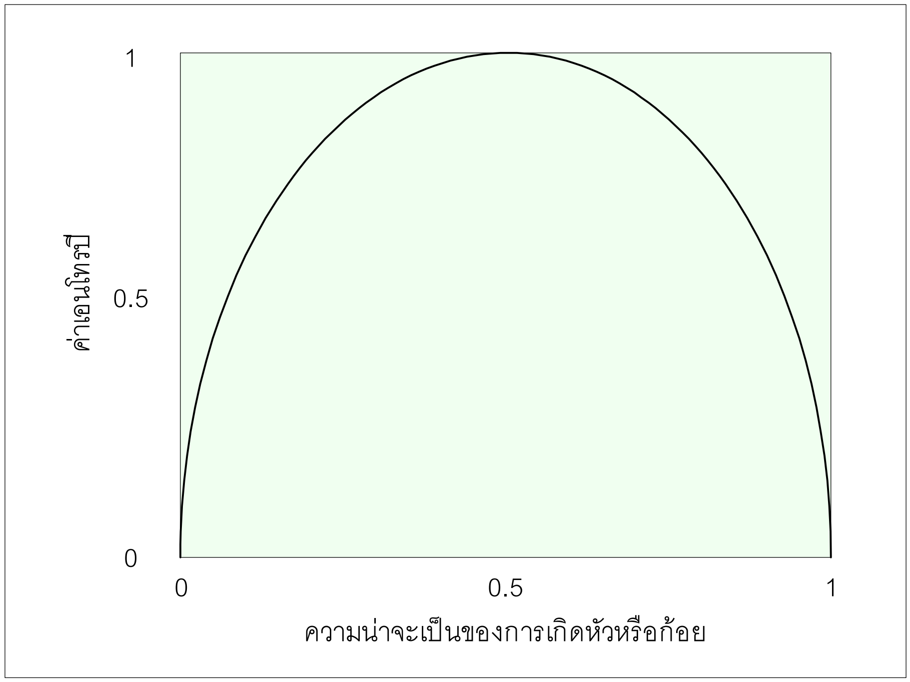
ถ้าให้ชุดของข้อมูล M ประกอบด้วยค่าที่เป็นไปได้ คือ $\{m_1,m_2,\ldots,m_n\}$ และให้ความน่าจะเป็นที่จะเกิดค่า m_i มีค่าเท่ากับ $P(m_i)$ จะได้ว่าค่า*เอนโทรปี (entropy)* ของ M ซึ่งใช้วัดค่าสารสนเทศโดยเฉลี่ยเพื่อระบุประเภทของข้อมูลสามารถเขียนแทนด้วย $I(M)$ คำนวณได้จากสูตร

$$I(M) = \sum_i^n -P(m_i)\log_2 P(m_i) \tag{6.3}$$

ตัวอย่างเช่นในการโยนหัวโยนก้อย ชุดข้อมูล M จะประกอบด้วยค่าที่เป็นไปได้คือ {หัว, ก้อย} และถ้าให้ความน่าจะเป็นที่ออกหัวเท่ากับ P(หัว) และความน่าจะเป็นที่ออกก้อยเท่ากับ P(ก้อย) ดังนั้นค่าเอนโทรปีของการโยนหัวโยนก้อย จะคำนวณได้จากสูตร

$$I(\text{การโยนหัวโยนก้อย}) = -P(\text{หัว})\log_2(P(\text{หัว})) - P(\text{ก้อย})\log_2(P(\text{ก้อย})) \tag{6.4}$$

เมื่อความน่าจะเป็นของการเกิดหัวหรือก้อยมีค่าต่าง ๆ กันจะสามารถคำนวณค่าเอนโทรปีของการโยนหัวโยนก้อยได้ต่าง ๆ กันดังรูปที่ 6–30 จะเห็นได้ว่าเมื่อออกหัวหมดหรือก้อยหมด ค่าเอนโทรปีจะเป็น 0 และค่าเอนโทรปีจะค่อย ๆ เพิ่มขึ้นจนสูงที่สุดเมื่อความน่าจะเป็นของการเกิดหัวเท่ากับความน่าจะเป็นของการเกิดก้อย แสดงให้เห็นว่าค่าเอนโทรปีที่น้อยจะบ่งบอกว่าข้อมูลชุดนั้นมีความแตกต่างกันน้อยหรือเกือบจะเป็นพวกเดียวกัน แต่ถ้าค่าเอนโทรปีสูงจะบ่งบอกว่าข้อมูลชุดนั้นมีความแตกต่างกันมากหรือประกอบด้วยตัวอย่างหลายพวกที่มีจำนวนใกล้เคียงกัน



รูปที่ 6–30 ค่าเอนโทรปีของการโยนหัวโยนก้อย

ในการเลือกคุณสมบัติที่จะมาเป็นบัพรากจะอาศัยค่าเกน ซึ่งคำนวณจากค่าเอนโทรปีทั้งหมดของชุดข้อมูลนั้นลบด้วยค่าเอนโทรปีหลังจากเลือกคุณสมบัติใดคุณสมบัติหนึ่งเป็นราก ค่าเอนโทรปีหลังจากแบ่งตามคุณสมบัติที่เลือกแล้วคำนวณได้จาก ค่าผลรวมของผลคูณระหว่างค่าเอนโทรปีของแต่ละบัพกับอัตราส่วนของตัวอย่างในแต่ละกิ่งต่อตัวอย่างทั้งหมดที่บัพนั้นๆ

ถ้าให้ข้อมูลสอนคือ T และคุณสมบัติที่เป็นบัพคือ X และมีค่าทั้งหมดที่เป็นไปได้ n ค่า บัพปัจจุบันจะแบ่งตัวอย่าง T ออกตามกิ่งเป็น $\{t_1, t_2, \ldots, t_n\}$ ตามค่าที่เป็นไปได้ของ X ดังนั้นจึงสามารถคำนวณค่าเอนโทรปีหลังจากแบ่งตามคุณสมบัติ X ดังนี้

$$I_x(T) = \sum_{i=1}^{n} \frac{|t_i|}{|T|} I(t_i) \qquad (6.5)$$

ค่าเกนของคุณสมบัติ X ที่ใช้แบ่งข้อมูลที่บัพหนึ่งๆ สามารถคำนวณได้จากการลบค่าเอนโทรปีทั้งหมดที่บัพนี้กับค่าเอนโทรปีที่ได้หลังจากแบ่งด้วยคุณสมบัติ X ดังนี้

$$Gain(X) = I(T) - I_x(T) \qquad (6.6)$$

พิจารณาคุณสมบัติ hair color ซึ่งแบ่งแยกข้อมูลได้ดังรูปที่ 6–25 ในกรณีที่ใช้ hair color เป็นบัพราก เราคำนวณหาค่าเกนได้ดังนี้

$$Gain(hair\,color) = \left[-\left(\frac{3}{8}\right)\log_2\left(\frac{3}{8}\right)-\left(\frac{5}{8}\right)\log_2\left(\frac{5}{8}\right)\right]-$$
$$\left[\frac{4}{8}\left(-\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right)-\left(\frac{2}{4}\right)\log_2\left(\frac{2}{4}\right)\right)+\frac{1}{8}\left(-\left(\frac{1}{1}\right)\log_2\left(\frac{1}{1}\right)\right)+\frac{3}{8}\left(-\left(\frac{3}{3}\right)\log_2\left(\frac{3}{3}\right)\right)\right]$$
$$= 0.45$$

ในทำนองเดียวกัน คุณสมบัติอื่นจะมีค่ามาตรฐานเกนเป็นดังต่อไปนี้

Gain (height) = 0.26 Gain (weight) = 0.01 Gain (lotion) = 0.34

จึงเลือกคุณสมบัติ hair color มาเป็นบัพแรกของต้นไม้ตัดสินใจ แต่คุณสมบัตินี้เพียงอย่างเดียวไม่สามารถแยกตัวอย่างบวกและลบออกจากกันได้ในกิ่งของค่าคุณสมบัติ blonde จึงต้องพิจารณาคุณสมบัติอื่นเพื่อแบ่งแยกข้อมูลที่ตกลงมายังกิ่งนี้ (ดูรูปที่ 6–29 ประกอบ) โดยค่าฟังก์ชันเกนของคุณสมบัติแต่ละตัวมีค่าดังนี้

Gain (height) = 0.5 Gain (weight) = 0.0 Gain (lotion) = 1.0

เราใช้คุณสมบัติ lotion ซึ่งมีค่าเกนมากสุดมาแบ่งแยกข้อมูลต่อไป ซึ่งพบว่าเมื่อแบ่งแยกแล้วข้อมูลที่ผ่านการแบ่งแยกมีกลุ่มเดียวกัน จึงได้ต้นไม้ตัดสินใจดังรูปที่ 6–24

6.5.3 การเปลี่ยนต้นไม้เป็นกฎ

ระบบปัญญาประดิษฐ์ส่วนใหญ่ใช้การแทนความรู้ในรูปของกฎ ดังนั้นเมื่อเราสร้างต้นไม้ตัดสินใจแล้วเราสามารถเปลี่ยนต้นไม้ให้อยู่ในรูปของกฎเพื่อใช้กับในกรณีที่ระบบของเราใช้การแทนความรู้ของกฎเป็นหลัก วิธีการแปลงต้นไม้เป็นกฎ "IF THEN" ทำได้โดยแสดงทุกเส้นทางเริ่มต้นจากบัพรากไปยังบัพใบและทุกครั้งที่พบบัพทดสอบก็ให้เพิ่มบัพทดสอบกับค่าของการทดสอบไว้ในส่วนของ IF และเมื่อพบบัพใบก็ให้ใส่ประเภทไว้ในส่วนของ THEN จากต้นไม้รูปที่ 6–24 เราเปลี่ยนเป็นกฎได้ดังนี้

(1) IF the person's hair color is blonde AND
 the person uses lotion
 THEN nothing happens

(2) IF the person's hair color is blonde AND
 the person uses no lotion
 THEN the person turns red

(3) IF the person's hair color is red

THEN the person turns red

(4) IF the person's hair color is brown

THEN nothing happens

ตารางด้านล่างแสดงการเปรียบเทียบการใช้เครื่องมือการเรียนรู้ (learning tools) และไม่ใช้เครื่องมือในการพัฒนาระบบผู้เชี่ยวชาญ (expert system) GASOIL และ BMT เป็นระบบผู้เชี่ยวชาญที่สร้างโดยเครื่องมือการเรียนรู้แบบต้นไม้ตัดสินใจซึ่งพัฒนาโดยใช้แนวคิดของการเรียนรู้ต้นไม้ตัดสินใจ

ตารางที่ 6–14 เปรียบเทียบระบบผู้เชี่ยวชาญที่ใช้และไม่ใช้การเรียนรู้ของเครื่องเพื่อช่วยในการพัฒนาระบบ

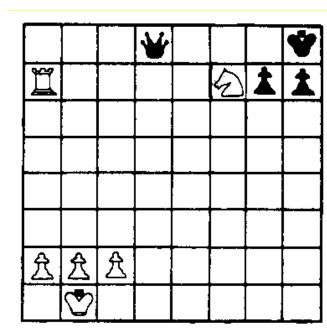
	Application	No. of Rules	Develop (Man Ys)	Maintain (Man Ys)	Learning Tools
MYCIN	Medical Diagnosis	400	100	N/A	N/A
XCON	VAX computer configuration	8,000	180	30	N/A
GASOIL	Hydrocarbon separtation system configuration	2,800	1	0.1	ExpertEase and Extran7
BMT	Configuration of fire-protection equipment in buildings	30,000	9	2.0	1st Class and Rulemaster

จากตารางจะเห็นได้ว่าระบบผู้เชี่ยวชาญที่ไม่ใช้เครื่องมือการเรียนรู้ (MYCIN และ XCON) ใช้แรงงานในการพัฒนาและดูแลระบบมากกว่าระบบผู้เชี่ยวชาญที่ใช้เครื่องมือเรียนรู้ (GASOIL และ BMT) หลายเท่าเมื่อเทียบโดยจำนวนกฎที่ใช้ในระบบ ตัวอย่างนี้แสดงให้เห็นถึงประโยชน์ของการเรียนรู้ของเครื่องได้อย่างชัดเจน

6.6 การเรียนรู้โดยการอธิบาย

การเรียนรู้โดยการอธิบาย — อีบีแอล (Explanation Based Learning — EBL) [DeJong & Mooney, 1986;Mitchell, et al., 1986] เป็นการเรียนรู้ที่มีลักษณะเด่นคือสามารถเรียนรู้ได้จากตัวอย่างบวกเพียงอย่างเดียวไม่จำเป็นต้องใช้ตัวอย่างลบ และจำนวนตัวอย่างบวกที่ใช้ก็ใช้เพียงตัวเดียวก็สามารถทำการเรียนรู้ได้ โดยมีแนวคิดว่าการเรียนรู้สามารถทำได้โดยการให้ความรู้พื้นฐานของโดเมนที่เกี่ยวข้อง จากนั้นจะให้ตัวอย่างบวกที่เป็นตัวอย่างของมโนทัศน์ที่จะสอน กระบวนการเรียนรู้ก็คือการใช้ความรู้ในโดเมนนั้นมาอธิบายให้ได้ว่าทำไมตัวอย่างที่สอนจึงเป็นตัวอย่างของมโนทัศน์แล้วจึงทำการวางนัยทั่วไปให้ครอบคลุมกรณีอื่นๆ

ยกตัวอย่างการเรียนรู้มโนทัศน์ fork ในการเล่นหมากรุกสากล (chess) โดยให้ตัวอย่างบวกของ fork ดังด้านล่างนี้



รูปที่ 6–31 ตัวอย่างบวกของ fork

ตัวอย่างด้านบนนี้แสดงสถานการณ์ที่ “ม้าขาวโจมตีคิงดำและควีนดำพร้อมกัน” ในกรณีนี้ฝ่ายดำต้องยอมเสียควีน ไม่เช่นนั้นจะแพ้ จากตัวอย่างบวกตัวเดียวด้านบน อีบีแอลจะเรียนได้กฎดังนี้ “ถ้าตัวหมาก x โจมตีคิงกับตัวหมาก y ของฝ่ายตรงข้ามพร้อมกันแล้ว ฝ่ายตรงข้ามจะเสีย y” ซึ่งวิธีการเรียนรู้กฎจะกล่าวต่อไป กฎที่เรียนรู้ได้นี้สามารถใช้กับสถานการณ์ณ์อื่นๆ นอกเหนือจากตัวอย่างสอนอีกด้วย กล่าวคือ x ไม่จำเป็นต้องเป็นม้า หรือ y ไม่จำเป็นต้องเป็นควีน นอกจากนั้นตำแหน่งของตัวหมากอื่นๆ ที่ไม่เกี่ยวข้องกับมโนทัศน์นี้ก็จะไม่ปรากฏในกฎ หมายความว่าตำแหน่งของตัวหมากอื่นๆ จะอยู่ที่ใดก็ได้ตราบเท่าที่ตัวหมากเรากำลังโจมตีคิงและตัวหมากอีกตัวของฝ่ายตรงข้ามพร้อมกัน

จะเห็นได้ว่าประสิทธิภาพของอีบีแอลสูงมากเพราะใช้ตัวอย่างแค่ตัวเดียวก็สามารถทำการวางนัยทั่วไปได้ สาเหตุที่สามารถทำได้เช่นนี้เนื่องจากว่าในอีบีแอลนี้เราต้องให้ความรู้ใน

โดเมนกับระบบเรียนรู้แบบนี้ด้วย ความรู้ในโดเมนของหมากรุกสากลก็อย่างเช่นกฎการเล่นหมากรุก ตัวหมากแต่ละตัวเดินอย่างไร ม้า คิง ควิน เดินอย่างไร การกินกันเกิดขึ้นได้เมื่อไร เกมจบเมื่อไร เป็นต้น ซึ่งกฎเหล่านี้เราสามารถให้ได้ไม่ยากนักเพราะมีเขียนไว้ในหนังสืออธิบายวิธีเล่นหมากรุกอยู่แล้ว อย่างไรก็ดีแม้ว่าเราจะให้ความรู้ในโดเมนแล้วก็ไม่ได้หมายความว่าเราไม่ต้องสอนอีบีแอล เปรียบเสมือนการเรียนรู้ของนักเรียนมัธยม แม้ว่าเราจะยกทฤษฎีเกี่ยวกับการเท่ากันของสามเหลี่ยมไปครบทุกทฤษฎีบท ก็ไม่ได้หมายความว่านักเรียนจะพิสูจน์การเท่ากันของสามเหลี่ยมสองรูปใดๆ ได้ทันที ครูก็ยังคงต้องยกตัวอย่างการพิสูจน์แสดงสามเหลี่ยม 2 รูปคู่หนึ่งๆ แล้วอธิบายว่าต้องใช้ทฤษฎีบทใดบ้างเพื่อการพิสูจน์และทำไมทฤษฎีบทเหล่านี้จึงพิสูจน์การเท่ากันของสามเหลี่ยมที่ยกตัวอย่างให้ดูได้ ซึ่งจะช่วยให้นักเรียนเข้าใจได้ดีขึ้น และเมื่อทำโจทย์การพิสูจน์สามเหลี่ยม 2 รูปที่ใช้ทฤษฎีบทซึ่งเหมือนกับครูยกตัวอย่างก็จะทำโจทย์ได้

กระบวนการเรียนรู้ของอีบีแอลประกอบด้วย 2 ขั้นตอนหลักคือ

- ใช้ความรู้ในโดเมนอธิบายให้ได้ว่าทำไมตัวอย่างจึงเป็นตัวอย่างของมโนทัศน์ในรูปของกฎ
- ทำการวางนัยทั่วไปของกฎที่ได้เพื่อให้ใช้กับกรณีอื่นได้

อินพุตและเอาต์พุตของอีบีแอลเป็นดังตารางที่ 6–15 ต่อไปนี้

ตารางที่ 6–15 อินพุตและเอาต์พุตของอีบีแอล

อินพุต: • ตัวอย่างสอน (training example) – ตัวอย่างบวกของมโนทัศน์ที่จะสอน เช่นในกรณีของ fork ตัวอย่างสอนคือตำแหน่งตัวหมากบนกระดานที่เกิด fork • มโนทัศน์เป้าหมาย (goal concept) – มโนทัศน์ที่จะสอนเช่นมโนทัศน์ fork • เกณฑ์ดำเนินการ (operational criterion) – คำอธิบายที่สามารถนำไปใช้ได้ทันที เช่นในกรณีของ fork นั้น เพรดิเคต attack-both(WKn,BK,BQ) ไม่สามารถนำไปใช้ได้ทันทีต้องแสดงในรูปของตำแหน่งตัวหมากบนกระดาน เช่น position(WKn,f7), position(BK,h8), position(BQ,d8) เป็นต้น • ความรู้ในโดเมน (domain knowledge) – กฎต่างๆ ที่ใช้แสดงความสัมพันธ์ของวัตถุและการกระทำต่างๆ ในโดเมนนั้น เช่น กฎการเล่นหมากรุกสากล เป็นต้น เอาต์พุต: • การวางนัยทั่วไปของตัวอย่างสอนซึ่งเพียงพอสำหรับอธิบายมโนทัศน์เป้าหมายและสอดคล้องกับเกณฑ์ดำเนินการ

ตัวอย่างการเรียนรู้มโนทัศน์ cup

ตัวอย่างการเรียนรู้ที่จะยกมานี้เป็นตัวอย่างการเรียนรู้มโนทัศน์ cup (อะไรคือถ้วย) โดยมีอินพุตที่ให้ดังนี้

- ตัวอย่างบวก:

  owner(object23,ralph), has-part(object23,concavity12),
  isa(concavity12,concavity), is(concavity12,upward-pointing),
  has-part(object23,handle16), isa(handle16,handle), is(object23,light),
  color(object23,brown), has-part(object23,bottom19), is(bottom19,bottom),
  is(bottom19,flat), …

- ความรู้ในโดเมน:

  liftable(X), stable(X), open-vessel(X) → cup(X)
  is(X,light), has-part(X,Y), isa(Y,handle) → liftable(X)
  small(X), made-from(X,Y), low-density(Y) → liftable(X)
  has-part(X,Y), isa(Y,bottom), is(Y,flat) → stable(X)
  has-part(X,Y), isa(Y,concavity), is(Y,upward-pointing) → open-vessel(X)

- มโนทัศน์เป้าหมาย: cup(X)

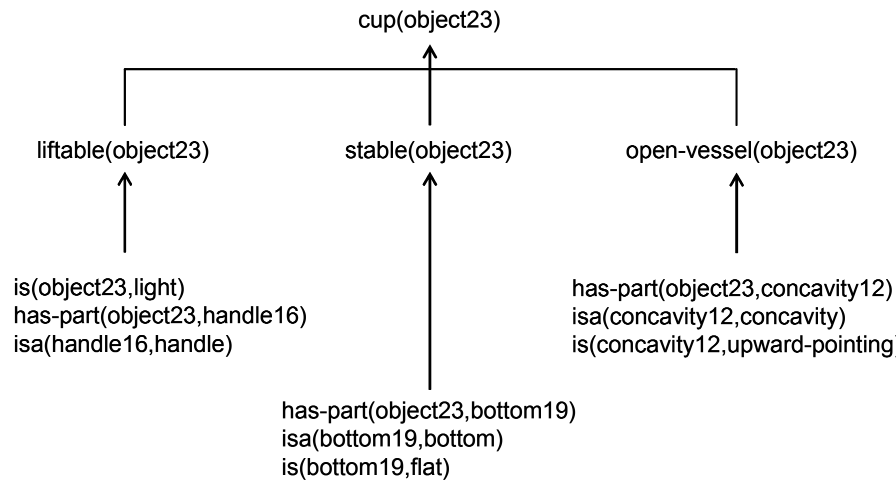
  X เป็นถ้วยก็ต่อเมื่อ X ยกได้ (liftable) เสถียร (stable) และเป็นภาชนะเปิด (open-vessel)

- เกณฑ์ดำเนินการ: สิ่งที่แสดงลักษณะต่างๆ ของถ้วย เช่น isa, has-part, color, owner เป็นต้น

ขั้นตอนการเรียนรู้ประกอบด้วย 2 ขั้นตอนดังนี้

ต้นไม้พิสูจน์

(1) ใช้ความรู้ในโดเมนอธิบายว่าทำไม object23 จึงเป็น cup โดยการสร้าง*ต้นไม้พิสูจน์ (proof tree)* ของ object23 ดังแสดงในรูปที่ 6–32



รูปที่ 6–32 ต้นไม้พิสูจน์ของตัวอย่างบวก cup

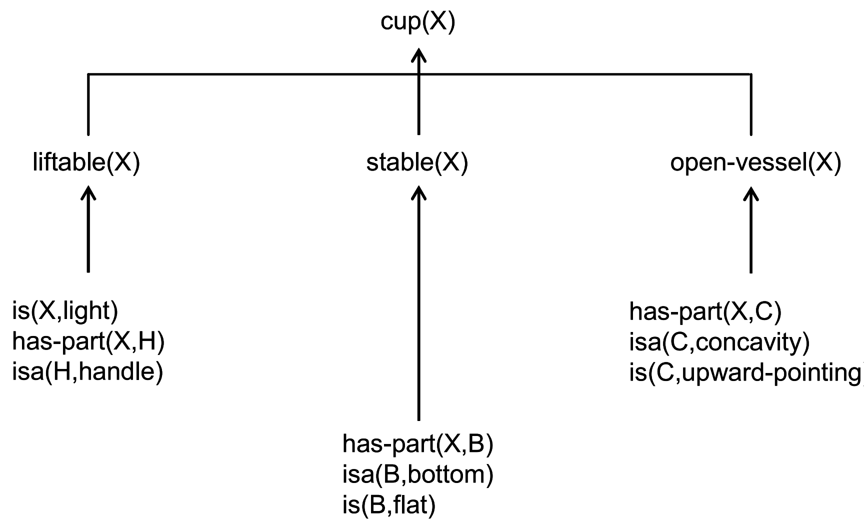
ต้นไม้พิสูจน์แสดงว่า object23 เป็น cup โดยมีคุณสมบัติ 3 อย่างคือยกได้ เสถียร และเป็นภาชนะเปิด เมื่อเราสังเกตความรู้ในโดเมนเรื่องถ้วยจะพบว่าการยกได้ของถ้วยมีกฎ 2 ข้อที่ใช้อธิบายได้และ object23 ตรงกับกฎข้อแรกของการยกได้ กล่าวคือเป็นถ้วยที่มีหูหิ้วและเบา นอกจากนั้นในต้นไม้พิสูจน์นี้จะไม่มีเพรดิเคตที่ไม่เกี่ยวข้องกับการเป็นถ้วย อย่างเช่น owner, color เป็นต้น ซึ่งสิ่งนี้เป็นการทำวางนัยทั่วไปแบบหนึ่งที่ตัดเงื่อนไขไม่จำเป็นทิ้งไป ดังนั้น ณ จุดนี้ถ้าเราสร้างกฎขึ้นเพื่ออธิบายการเป็นถ้วยของ object23 ก็จะได้กฎดังนี้

is(object23,light), has-part(object23,handle16), isa(handle16,handle),
has-part(object23,bottom19), isa(bottom19,bottom), is(bottom19,flat),
has-part(object23,concavity12), isa(concavity12,concavity),
is(concavity12,upward-pointing) $\rightarrow$ cup(object23)

อย่างไรก็ดีแม้ว่ากฎนี้จะไม่มีเพรดิเคตที่ไม่เกี่ยวข้อง แต่ว่ากฎนี้ยังคงอธิบายได้เฉพาะ object23 เท่านั้น เราจำเป็นต้องทำการวางนัยทั่วไปเพิ่มเติมขึ้นเพื่อให้ใช้กับถ้วยที่มีคุณสมบัติเหมือนกับ object23 ได้

(2) การวางนัยทั่วไปและดึงเพรดิเคตที่อยู่ในเกณฑ์ดำเนินการมาสร้างกฎ ขั้นตอนนี้ทำการวางนัยทั่วไปโดยทำตามความรู้ในโดเมน กล่าวคือถ้าอาร์กิวเมนต์ของเพรดิเคตในต้นไม้พิสูจน์ที่ตรงกันกับอาร์กิวเมนต์ของเพรดิเคตของความรู้ในโดเมนเป็นตัวแปรก็เปลี่ยนอาร์กิวเมนต์ที่เป็นค่าคงที่ให้เป็นตัวแปร แต่ถ้าอาร์กิวเมนต์ของเพรดิเคตที่

ตรงกันกับความรู้ในโดเมนเป็นค่าคงที่ก็ไม่ต้องเปลี่ยน เช่น is(object23,light) เปลี่ยนเป็น is(X,light) เป็นต้น ผลที่ได้แสดงในรูปที่ 6–33



รูปที่ 6–33 การวางนัยทั่วไปของตัวอย่างบวก cup

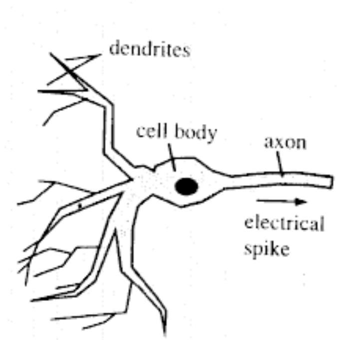
ดังนั้นเราจะได้กฎดังนี้

is(X,light), has-part(X,H), isa(H,handle), has-part(X,B), isa(B,bottom), is(B,flat), has-part(X,C), isa(C,concavity), is(C,upward-pointing) → cup(X)

การเรียนรู้อีบีแอลนี้เป็นการเรียนรู้ประเภทที่เรียกว่า*การเรียนรู้เชิงวิเคราะห์ (analytical learning)* กล่าวคือการเรียนรู้ประเภทนี้จะเป็นการจัดความรู้ (ความรู้ในโดเมน) ในรูปแบบใหม่ให้ใช้งานได้อย่างมีประสิทธิภาพ ดังจะเห็นได้ว่ากฎที่ได้โดยอีบีแอลประกอบด้วยเพรดิเคตที่อยู่ในเกณฑ์ดำเนินการเท่านั้น ซึ่งเพรดิเคตเหล่านี้จะใช้งานได้อย่างมีประสิทธิภาพสามารถจับคู่ (match) กับข้อมูลในตัวอย่างที่สอนแล้วทราบทันทีว่าตรงกันหรือไม่ ต่างกับความรู้ในโดเมนเดิมที่ประกอบด้วยเพรดิเคตบางตัว เช่น liftable ที่ต้องการการอธิบายโดยการพิสูจน์ต่อว่าเพรดิเคตนี้ตรงกับตัวอย่างหรือไม่

6.7 ข่ายงานประสาทเทียม

ข่ายงานประสาทเทียม (Artificial Neural Network) เป็นการจำลองการทำงานบางส่วนของสมองมนุษย์ เซลล์ประสาท (neuron) ในสมองของคนเราประกอบด้วยนิวเคลียส (nucleus) ตัวเซลล์ (cell body) ใยประสาทนำเข้า (dendrite) แกนประสาทนำออก (axon) แสดงในรูปที่ 6–34

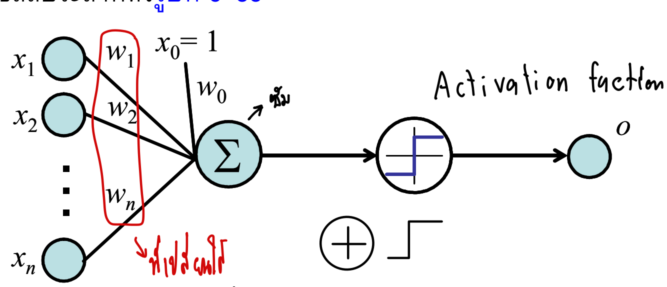


รูปที่ 6–34 เซลล์ประสาท

เดนไดรท์ทำหน้าที่รับสัญญาณไฟฟ้าเคมีซึ่งส่งมาจากเซลล์ประสาทใกล้เคียง เซลล์ประสาทตัวหนึ่งๆ จะเชื่อมต่อกับเซลล์ตัวอื่นๆ ประมาณ 10,000 ตัว เมื่อสัญญาณไฟฟ้าเคมีที่รับเข้ามาเกินค่าค่าหนึ่ง เซลล์จะถูกกระตุ้นและส่งสัญญาณไปทางแกนประสาทนำออกไปยังเซลล์อื่นๆ ต่อไป ประมาณกันว่าสมองของคนเรามีเซลล์ประสาทอยู่ทั้งสิ้นประมาณ 10^{11} ตัว

6.7.1 เพอร์เซปตรอน

เพอร์เซปตรอน (perceptron) เป็นข่ายงานประสาทเทียมแบบง่ายมีหน่วยเดียวที่จำลองลักษณะของเซลล์ประสาทดังรูปที่ 6–35

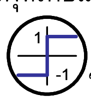


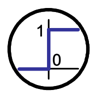
รูปที่ 6–35 เปอร์เซปตรอน

$x_0 w_0 + x_1 w_1 + x_2 w_2 + \ldots + x_n w_n$

เพอร์เซปตรอนรับอินพุตเป็นเวกเตอร์จำนวนจริงแล้วคำนวณหาผลรวมเชิงเส้น (linear combination) แบบถ่วงน้ำหนักของอินพุต ($x_1, x_2, \ldots, x_n$) โดยที่ค่า $w_1, w_2, \ldots, w_n$ ในรูปเป็นค่าน้ำหนักของอินพุตและให้เอาต์พุต (o) เป็น 1 ถ้าผลรวมที่ได้มีค่าเกินค่าขีดแบ่ง (θ) และเป็น -1 ถ้าไม่เกิน ส่วน w_0 ในรูปเป็นค่าลบของค่าขีดแบ่งดังจะได้อธิบายต่อไป และ x_0 เป็นอินพุตเทียมกำหนดให้มีค่าเป็น 1 เสมอ

ฟังก์ชันกระตุ้น



ในรูปแสดง*ฟังก์ชันกระตุ้น (activation function)* ชนิดที่เรียกว่า*ฟังก์ชันสองขั้ว (bipolar function)* ซึ่งแสดงผลของเอาต์พุตเป็น 1 กับ -1 ฟังก์ชันกระตุ้นอื่นๆ ที่นิยมใช้ก็อย่างเช่น *ฟังก์ชันไบนารี (binary function)* ซึ่งแสดงผลของเอาต์พุตเป็น 1 กับ 0 และเขียนแทนด้วยรูป 

เราสามารถแสดงเอาต์พุต (o) ในรูปของฟังก์ชันของอินพุต ($x_1, x_2, \ldots, x_n$) ได้ดังนี้

$$o(x_1, x_2, \ldots, x_n) = \begin{cases} 1 \;\; if \;\; w_1x_1 + w_2x_2 + \cdots + w_nx_n > \theta \\ -1 \;\; if \;\; w_1x_1 + w_2x_2 + \cdots + w_nx_n < \theta \end{cases} \tag{6.7}$$

เอาต์พุตเป็นฟังก์ชันของอินพุตในรูปของผลรวมเชิงเส้นแบบถ่วงน้ำหนัก น้ำหนักจะเป็นตัวกำหนดว่าในจำนวนอินพุตนั้น อินพุต (x_i) ตัวใดมีความสำคัญต่อการกำหนดค่าเอาต์พุต ตัวที่มีความสำคัญมากจะมีค่าสัมบูรณ์ของน้ำหนักมาก ส่วนตัวที่มีความสำคัญน้อยจะมีค่าใกล้ศูนย์ ในกรณีที่ผลรวมเท่ากับค่าขีดแบ่งค่าเอาต์พุตไม่นิยาม (จะเป็น 1 หรือ -1 ก็ได้)

จากฟังก์ชันในสูตรที่ (6.7) เราจัดรูปใหม่โดยย้าย θ ไปรวมกับผลรวมเชิงเส้นแล้วแทน $-\theta$ ด้วย w_0 เราจะได้ฟังก์ชันของเอาต์พุตดังด้านล่างนี้

$$o(x_1, x_2, \ldots, x_n) = \begin{cases} 1 \;\; if \;\; w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n > 0 \\ -1 \;\; if \;\; w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n < 0 \end{cases} \tag{6.8}$$

กำหนดให้ $g(\vec{x}) = \sum_{i=0}^{n} w_i x_i = \vec{w} \cdot \vec{x}$ โดยที่ $\vec{x}$ แทนเวกเตอร์อินพุต เราสามารถเขียนฟังก์ชันของเอาต์พุตได้ใหม่ดังนี้

$$o(x_1, x_2, \ldots, x_n) = \begin{cases} 1 \;\; if \;\; g(\vec{x}) > 0 \\ -1 \;\; if \;\; g(\vec{x}) < 0 \end{cases} \tag{6.9}$$

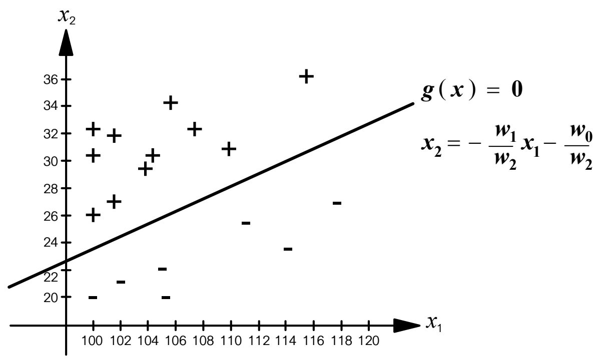
สมมติว่าเรามีอินพุตสองตัวคือ x_1 และ x_2 ซึ่งแสดงค่าส่วนสูงและน้ำหนักของเด็กนักเรียนประถมและหลังจากที่แพทย์ตรวจร่างกายของเด็กโดยละเอียดแล้วได้จำแนกนักเรียน

ออกเป็นสองกลุ่มคือเด็กอ้วนและเด็กไม่อ้วน เราให้เอาต์พุตเป็นค่าที่แสดงเด็กอ้วนแทนด้วย +1 กับไม่อ้วนแทนด้วย -1 ดังตารางที่ 6–16

ตารางที่ 6–16 ข้อมูลเด็กอ้วนและเด็กไม่อ้วน

เด็กคนที่	ส่วนสูง (ซม.)	น้ำหนัก (กก.)	อ้วน/ไม่อ้วน
1	100.0	20.0	-1
2	100.0	26.0	1
3	100.0	30.4	1
4	100.0	32.4	1
5	101.6	27.0	1
6	101.6	32.0	1
7	102.0	21.0	-1
8	103.6	29.6	1
9	104.4	30.4	1
10	104.9	22.0	-1
11	105.2	20.0	-1
12	105.6	34.4	1
13	107.2	32.4	1
14	109.9	34.9	1
15	111.0	25.4	-1
16	114.2	23.5	-1
17	115.5	36.3	1
18	117.8	26.9	-1

ในกรณีที่มีอินพุต 2 ตัว (ไม่รวม x_0) เราจะได้ $g(\vec{x}) = w_0 + w_1x_1 + w_2x_2$ ซึ่งถ้าเราให้ $g(\vec{x}) = 0$ จะได้ว่า $w_0 + w_1x_1 + w_2x_2 = 0$ ซึ่งแทนสมการเส้นตรงในระนาบสองมิติ x_1, x_2 สมการนี้มีจุดตัดแกนอยู่ที่ $-\frac{w_0}{w_2}$ และมีความชันเท่ากับ $-\frac{w_1}{w_2}$ เมื่อนำสมการนี้ไปวาดในระนาบสองมิติร่วมกับตัวอย่างสอนในตารางที่ 6–16 โดยกำหนดค่า w_0, w_1, w_2 ที่เหมาะสมจะได้ดังรูปที่ 6–36



รูปที่ 6–36 สมการเส้นตรงสร้างโดยเพอร์เซปตรอน

เครื่องหมาย + และ – ในรูปแทนตัวอย่างบวก (เด็กอ้วน) และตัวอย่างลบ (เด็กไม่อ้วน) ตามลำดับ ดังจะเห็นได้ในรูปว่าเส้นตรงนี้เมื่อกำหนดจุดตัดแกนและความชันที่เหมาะสมซึ่งกำหนดโดย w_0, w_1, w_2 เส้นตรงนี้จะแบ่งตัวอย่างออกเป็นสองกลุ่มซึ่งอยู่คนละด้านของเส้นตรง และเมื่อมีข้อมูลส่วนสูงและน้ำหนักของเด็กคนอื่นที่เราต้องการทำนายว่าจะเป็นเด็กอ้วนหรือไม่ ก็ใช้เส้นตรงนี้โดยดูว่าข้อมูลใหม่นี้อยู่ด้านใดของเส้นตรง ถ้าด้านบนก็ทำนายว่าเป็นเด็กอ้วน (+) ถ้าด้านล่างก็ทำนายว่าเด็กไม่อ้วน (–)

ระนาบตัดสินใจหลายมิติ

ตัวอย่างด้านบนแสดงกรณีของอินพุตในสองมิติ จะเห็นได้ว่าเพอร์เซปตรอนจะเป็นเส้นตรง ในกรณีที่อินพุตมากกว่าสองมิติเพอร์เซปตรอนจะเป็น*ระนาบตัดสินใจหลายมิติ (hyperplane decision surface)* ปัญหาการเรียนรู้เพอร์เซปตรอนก็คือการหาค่าเวกเตอร์น้ำหนัก ($\vec{w}$) ที่เหมาะสมในการจำแนกประเภทของข้อมูลสอนเพื่อให้เพอร์เซปตรอนแสดงเอาต์พุตได้ตรงกับค่าที่สอน *กฎการเรียนรู้เพอร์เซปตรอน (perceptron learning rule)* ใช้สำหรับสอนเพอร์เซปตรอนโดยจะหาค่าเวกเตอร์น้ำหนักดังแสดงในตารางที่ 6–17

อัลกอริทึมเริ่มต้นจากสุ่มค่าเวกเตอร์น้ำหนัก ซึ่งโดยมากค่าที่สุ่มมานี้จะไม่ได้ระนาบหลายมิติที่แบ่งตัวอย่างได้ถูกต้องทุกตัวดังนั้นจึงต้องมีการแก้ไขน้ำหนักโดยเทียบเพอร์เซปตรอนกับตัวอย่างที่สอน หมายถึงว่าเมื่อเราป้อนตัวอย่างสอนเข้าไปในเพอร์เซปตรอน เราจะคำนวณค่าเอาต์พุตได้ นำค่าเอาต์พุตที่คำนวณได้โดยเพอร์เซปตรอนเทียบกับเอาต์พุตเป้าหมาย ถ้าตรงกันแสดงว่าจำแนกตัวอย่างได้ถูกต้อง ไม่ต้องปรับน้ำหนักสำหรับตัวอย่างนั้น แต่ถ้าไม่ตรงกันก็จะทำการปรับน้ำหนักตามสมการในอัลกอริทึม ส่วนอัตราการเรียนรู้เป็นตัวเลขบวกจำนวนน้อยๆ เช่น 0.01, 0.005 เป็นต้น อัตราการเรียนรู้นี้จะส่งผลต่อการลู่เข้าของเพอร์เซปตรอน ถ้าอัตราการเรียนรู้มีค่ามากเพอร์เซปตรอนก็จะเรียนรู้ได้เร็ว แต่ก็อาจเรียนรู้ไม่สำเร็จเนื่องจากการปรับค่ามีความหยาบเกินไป อัตราการเรียนรู้ที่มีค่าน้อยก็จะทำให้การปรับน้ำหนักทำได้อย่างละเอียดแต่ก็อาจเสียเวลาในการเรียนรู้นาน

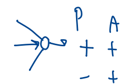
Stopping candidate

1 ครบ Epoch

2. Weight ไม่เปลี่ยน

ตารางที่ 6–17 อัลกอริทึมกฎการเรียนรู้เพอร์เซปตรอน

Algorithm: Perceptron-Learning-Rule



1. Initialize weights w_i of the perceptron.
2. **UNTIL** the termination condition is met **DO**
   2.1 **FOR EACH** training example **DO**
      - Input the example and compute the output.
      - Change the weights if the output from the perceptron is not equal to the target output using the following rule.

new old

$$w_i \leftarrow w_i + \Delta w_i$$

learning rate

$$\Delta w_i \leftarrow \alpha(t - o)x_i$$

Actr Predr

where t, o and α are the target output, the output from the perceptron and the learning rate, respectively.

การปรับน้ำหนักตามกฎการเรียนรู้เพอร์เซปตรอนโดยใช้อัตราการเรียนรู้ที่มีค่าน้อยเพียงพอ จะได้ระนาบหลายมิติที่จะลู่เข้าสู่ระนาบหนึ่งที่สามารถแบ่งข้อมูลออกเป็นสองส่วน (ในกรณีที่ข้อมูลสามารถแบ่งได้) เพื่ออธิบายผลที่เกิดจากการปรับค่าน้ำหนัก เราจะลองพิจารณาพฤติกรรมของกฎการเรียนรู้นี้ดูว่าทำไมการปรับน้ำหนักเช่นนี้จึงลู่เข้าสู่ระนาบที่แบ่งข้อมูลได้อย่างถูกต้อง

- พิจารณากรณีแรกที่เพอร์เซปตรอนแยกตัวอย่างสอนตัวหนึ่งที่รับเข้ามาได้ถูกต้อง กรณีนี้จะพบว่า (t-o) จะมีค่าเป็น 0 ดังนั้น Δw_i ไม่เปลี่ยนแปลงเพราะ $\Delta w_i = \alpha(\text{t-o})x_i$
- พิจารณาในกรณีที่เพอร์เซปตรอนให้เอาต์พุตเป็น –1 แต่เอาต์พุตเป้าหมายหรือค่าที่แท้จริงเท่ากับ 1 ในกรณีนี้หมายความว่าค่าที่เราต้องการคือ 1 แต่ค่าน้ำหนักไม่เหมาะสม ดังนั้นเพื่อที่จะทำให้เพอร์เซปตรอนให้เอาต์พุตเป็น 1 น้ำหนักต้องถูกปรับให้สามารถเพิ่มค่าของ $\vec{w}\cdot\vec{x}$ ในกรณีนี้หมายความว่าผลรวมเชิงเส้นน้อยเกินไปและน้อยกว่า 0 จึงได้เอาต์พุตเป็น -1 ดังนั้นสิ่งที่เราต้องการคือการเพิ่มค่าผลรวมเชิงเส้นเพราะถ้าเราเพิ่มค่าได้เรื่อยๆ จนมากกว่า 0 เพอร์เซปตรอนจะให้เอาต์พุตเป็น 1 ซึ่งตรงกับที่เราต้องการ พิจารณาดูดังต่อไปนี้ว่าการปรับค่าโดยกฎเรียนรู้ทำให้ผลรวมเชิงเส้นเพิ่มขึ้นได้อย่างไร กรณีนี้เราจะได้ว่า (t-o) เท่ากับ (1-(-1)) มีค่าเป็น 2 และลองพิจารณาค่าของอินพุต x_i แยกกรณีดังนี้

- ถ้า $x_i > 0$ จะได้ว่า Δw_i มากกว่า 0 เพราะว่า $\Delta w_i \leftarrow \alpha(t-o)x_i$ และ α มากกว่า 0, (t-o) = 2 และ $x_i > 0$ จากสมการการปรับน้ำหนัก $w_i \leftarrow w_i + \Delta w_i$ เมื่อ Δw_i มากกว่า 0 จะทำให้ w_i มีค่าเพิ่มขึ้นและ $\sum w_i x_i$ ก็จะมีค่าเพิ่มขึ้น เมื่อผลรวมมีค่ามากขึ้นแสดงว่าการปรับไปในทิศทางที่ถูกต้องคือเมื่อปรับไปจนกระทั่งได้ผลรวมมากกว่า 0 จะทำให้เพอร์เซปตรอนเอาต์พุตได้ถูกต้องยิ่งขึ้น
- ถ้า $x_i < 0$ เราจะได้ว่า $\alpha(t-o)x_i$ จะมีค่าน้อยกว่า 0 แสดงว่า w_i ตัวที่คูณกับ x_i ที่น้อยกว่า 0 จะลดลงทำให้ $\sum w_i x_i$ เพิ่มขึ้นเหมือนเดิม เพราะ x_i เป็นค่าลบและ w_i มีค่าลดลง ในที่สุดก็จะทำให้เพอร์เซปตรอนให้เอาต์พุตได้ถูกต้องยิ่งขึ้น

- ในกรณีที่เพอร์เซปตรอนให้เอาต์พุตเป็น 1 แต่เอาต์พุตเป้าหมายหรือค่าที่แท้จริงเท่ากับ -1 จะได้ว่า w_i ของ x_i ที่เป็นค่าบวกจะลดลง ส่วน w_i ของ x_i ที่เป็นค่าลบจะเพิ่มขึ้นและทำให้การปรับเป็นไปในทิศทางที่ถูกต้องเช่นเดียวกับในกรณีแรก

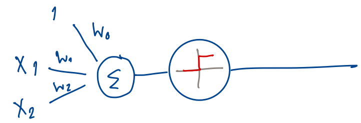
6.7.2 ตัวอย่างการเรียนฟังก์ชัน AND และ XOR ด้วยกฎเรียนรู้เพอร์เซปตรอน

พิจารณาตัวอย่างการเรียนรู้ของเพอร์เซปตรอนโดยจะให้เรียนรู้ฟังก์ชัน 2 ฟังก์ชัน ฟังก์ชันแรกคือฟังก์ชัน AND แสดงในตารางที่ 6–18 ในกรณีนี้เราใช้ฟังก์ชันไบนารีเป็นฟังก์ชันกระตุ้น

ตารางที่ 6–18 ฟังก์ชัน AND(x1,x2)

x_1	x_2	เอาต์พุตเป้าหมาย
0	0	0
0	1	0
1	0	0
1	1	1

ฟังก์ชัน AND ตามตารางด้านบนนี้จะให้ค่าที่เป็นจริงก็ต่อเมื่อ x1 และ x2 เป็นจริงทั้งคู่ (ดูที่สดมภ์เอาต์พุตเป้าหมาย) ผลการใช้กฎการเรียนรู้เพอร์เซปตรอนกับฟังก์ชัน AND แสดงในตารางที่ 6–19



ตารางที่ 6–19 ผลการเรียนรู้ฟังก์ชัน AND โดยกฎการเรียนรู้เพอร์เซปตรอน

-0.9+1.1+0

$\boxed{0.2}$ 1

1 0

$x_1 x_0$

Epoch

3Epoch จบ

Perceptron Learning Example - Function AND											
		Bias Input x0=+1					Alpha =	0.5			
Input	Input				Net Sum	Target	Actual	Alpha*	Weight Values		
x1	x2	1.0*w0	x1*w1	x2*w2	Input	Output	Output	Error	w0	w1	w2
								Error $\alpha(t-o)x_i$	0.1	0.1	0.1
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	0.10	0.10
0	1	-0.40	0.00	0.10	-0.30	0	0	0.00	-0.40	0.10	0.10
1	0	-0.40	0.10	0.00	-0.30	0	0	0.00	-0.40	0.10	0.10
1	1	-0.40	0.10	0.10	-0.20	1	0	0.50	0.10	0.60	0.60
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	0.60	0.60
0	1	-0.40	0.00	0.60	0.20	0	1	-0.50	-0.90	0.60	0.10
1	0	-0.90	0.60	0.00	-0.30	0	0	0.00	-0.90	0.60	0.10
1	1	-0.90	0.60	0.10	-0.20	1	0	0.50	-0.40	1.10	0.60
0	0	-0.40	0.00	0.00	-0.40	0	0	0.00	-0.40	1.10	0.60
0	1	-0.40	0.00	0.60	0.20	0	1	-0.50	-0.90	1.10	0.10
1	0	-0.90	1.10	0.00	0.20	0	1	-0.50	-1.40	0.60	0.10
1	1	-1.40	0.60	0.10	-0.70	1	0	0.50	-0.90	1.10	0.60
0	0	-0.90	0.00	0.00	-0.90	0	0	0.00	-0.90	1.10	0.60
0	1	-0.90	0.00	0.60	-0.30	0	0	0.00	-0.90	1.10	0.60
1	0	-0.90	1.10	0.00	0.20	0	1	-0.50	-1.40	0.60	0.60
1	1	-1.40	0.60	0.60	-0.20	1	0	0.50	-0.90	1.10	1.10
0	0	-0.90	0.00	0.00	-0.90	0	0	0.00	-0.90	1.10	1.10
0	1	-0.90	0.00	1.10	0.20	0	1	-0.50	-1.40	1.10	0.60
1	0	-1.40	1.10	0.00	-0.30	0	0	0.00	-1.40	1.10	0.60
1	1	-1.40	1.10	0.60	0.30	1	1	0.00	-1.40	1.10	0.60
0	0	-1.40	0.00	0.00	-1.40	0	0	0.00	-1.40	1.10	0.60
0	1	-1.40	0.00	0.60	-0.80	0	0	0.00	-1.40	1.10	0.60
1	0	-1.40	1.10	0.00	-0.30	0	0	0.00	-1.40	1.10	0.60
1	1	-1.40	1.10	0.60	0.30	1	1	0.00	-1.40	1.10	0.60

ขั้นตอนแรกเริ่มจากการสุ่มค่า w_0 จนถึง w_2 ในที่นี้กำหนดให้เป็น 0.1 ทั้งสามตัว จากนั้นก็เริ่มป้อนตัวอย่างเข้าไป (ทีละแถว) ตัวอย่างแรกได้ผลรวมเชิงเส้น (Net Sum) เป็น 0.10 ซึ่งมากกว่า 0 ดังนั้นเปอร์เซปตรอนจะให้เอาต์พุตจริง (Actual Output) ออกมาเป็น 1 ซึ่งผิดเพราะเอาต์พุตเป้าหมาย (Target Output) จะต้องได้เป็น 0 ทำให้อัตราการเรียนรู้คูณค่าผิดพลาด (Alpha x Error) ได้ –0.50 หลังจากนี้ก็นำไปปรับน้ำหนักตาม $w_i \leftarrow w_i + \Delta w_i$ และ $\Delta w_i \leftarrow \alpha(t-o)x_i$ ดังนั้นจะได้เป็น $w_0 \leftarrow w_0 + \alpha(t-o)x_0$ = w_0 + 0.50(-1) x 1 = 0.10 + (-0.5) = -0.4 ต่อไปก็ปรับค่า w_1 ในทำนองเดียวกัน $w_1 \leftarrow w_1 + \alpha(t-o)x_1$ = w_1 + 0.50(-1) x 0 ดังนั้น w_1 จะเท่ากับ 0.10 คือไม่เปลี่ยนแปลง เช่นเดียวกับ w_2 ที่ไม่เปลี่ยนแปลง จะเห็นได้ว่าแม้มีค่าผิดพลาดแต่ไม่มีการปรับค่า w_1 และ w_2 เนื่องจากอินพุตที่ใส่เข้าไปเป็น 0 ทำ

ให้ผลคูณเป็น 0 จึงไม่ได้ปรับ และเป็นข้อเสียของฟังก์ชันกระตุ้นแบบไบนารีซึ่งถ้าผลออกมาเป็น 0 จะไม่มีการปรับค่าให้ (ถ้าเราเปลี่ยน 0 เป็น –1 การปรับค่าจะดีขึ้น w_i จะถูกปรับทันทีตั้งแต่รอบแรก)

ตัวอย่างที่สองจนถึงตัวอย่างที่สี่ก็ทำเช่นเดียวกัน และเมื่อทำครบ 1 รอบการสอน (epoch) แล้วจะต้องทำการสอนซ้ำด้วยข้อมูลชุดเดิม นี่คือวิธีการสอนของข่ายงานประสาทเทียมซึ่งต่างจากวิธีอื่นๆ ที่ต้องใช้ข้อมูลชุดเดิมสอนซ้ำไปจนกระทั่งค่าผิดพลาดลดลงจนถึงจุดที่เราต้องการ ในที่นี้คือ 0 เนื่องจากเราต้องการให้มีการแบ่งข้อมูลอย่างเด็ดขาด และสมการเส้นตรงที่ได้จะมีค่า w_0 = -1.40, w_1 = 1.10 และ w_2 = 0.60

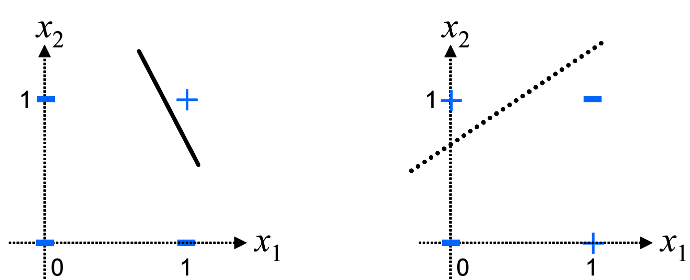
ฟังก์ชันที่สองที่จะทดลองเรียนรู้ด้วยกฎการเรียนรู้เพอร์เซปตรอนคือฟังก์ชัน XOR แสดงในตารางที่ 6–20

ตารางที่ 6–20 ฟังก์ชัน XOR(x1,x2)

x_1	x_2	เอาต์พุตเป้าหมาย
0	0	0
0	1	1
1	0	1
1	1	0

ฟังก์ชัน XOR ตามตารางด้านบนนี้จะให้ค่าที่เป็นจริงก็ต่อเมื่อ x_1 หรือ x_2 ตัวใดตัวหนึ่งเพียงตัวเดียวเป็นจริง (ดูที่สดมภ์เอาต์พุตเป้าหมาย) ผลการใช้กฎการเรียนรู้เพอร์เซปตรอนกับฟังก์ชัน XOR แสดงในตารางที่ 6–21

ในกรณีของฟังก์ชัน XOR นี้พบว่าค่าผิดพลาดไม่ลดลง และค่าน้ำหนักจะแกว่งไปมาโดยไม่ลู่เข้าแม้ว่าจะสอนต่อจากนี้ไปอีกกี่รอบการสอนก็ตาม จึงสรุปว่าฟังก์ชัน XOR เรียนไม่สำเร็จด้วยเพอร์เซปตรอน เมื่อเรานำฟังก์ชัน AND และ XOR ไปวาดกราฟในสองมิติจะได้กราฟดังรูปที่ 6–37



(ก) ฟังก์ชันแยกเชิงเส้นได้ (AND) (ข) ฟังก์ชันแยกเชิงเส้นไม่ได้ (XOR)

รูปที่ 6–37 ฟังก์ชันแยกเชิงเส้นได้และไม่ได้

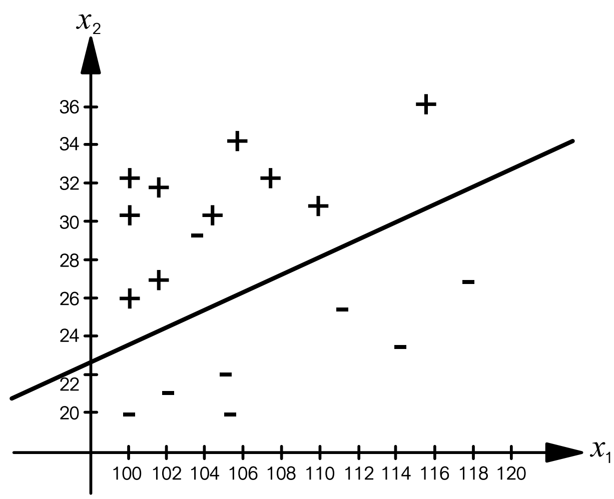
ตารางที่ 6–21 ผลการเรียนรู้ฟังก์ชัน XOR โดยกฎการเรียนรู้เพอร์เซปตรอน

Perceptron Learning Example XOR											
		Bias Input X0=+1					Alpha =	0.5			
Input	Input				Net Sum	Target	Actual	Alpha*	Weight Values		
x1	x2	1.0*w0	x1*w1	x2*w2	Input	Output	Output	Error	w0	w1	w2
									0.1	0.1	0.1
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	0.10	0.10
0	1	-0.40	0.00	0.10	-0.30	1	0	0.50	0.10	0.10	0.60
1	0	0.10	0.10	0.00	0.20	1	1	0.00	0.10	0.10	0.60
1	1	0.10	0.10	0.60	0.80	0	1	-0.50	-0.40	-0.40	0.10
0	0	-0.40	0.00	0.00	-0.40	0	0	0.00	-0.40	-0.40	0.10
0	1	-0.40	0.00	0.10	-0.30	1	0	0.50	0.10	-0.40	0.60
1	0	0.10	-0.40	0.00	-0.30	1	0	0.50	0.60	0.10	0.60
1	1	0.60	0.10	0.60	1.30	0	1	-0.50	0.10	-0.40	0.10
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	-0.40	0.10
0	1	-0.40	0.00	0.10	-0.30	1	0	0.50	0.10	-0.40	0.60
1	0	0.10	-0.40	0.00	-0.30	1	0	0.50	0.60	0.10	0.60
1	1	0.60	0.10	0.60	1.30	0	1	-0.50	0.10	-0.40	0.10
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	-0.40	0.10
0	1	-0.40	0.00	0.10	-0.30	1	0	0.50	0.10	-0.40	0.60
1	0	0.10	-0.40	0.00	-0.30	1	0	0.50	0.60	0.10	0.60
1	1	0.60	0.10	0.60	1.30	0	1	-0.50	0.10	-0.40	0.10
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	-0.40	0.10
0	1	-0.40	0.00	0.10	-0.30	1	0	0.50	0.10	-0.40	0.60
1	0	0.10	-0.40	0.00	-0.30	1	0	0.50	0.60	0.10	0.60
1	1	0.60	0.10	0.60	1.30	0	1	-0.50	0.10	-0.40	0.10
0	0	0.10	0.00	0.00	0.10	0	1	-0.50	-0.40	-0.40	0.10
0	1	-0.40	0.00	0.10	-0.30	1	0	0.50	0.10	-0.40	0.60
1	0	0.10	-0.40	0.00	-0.30	1	0	0.50	0.60	0.10	0.60
1	1	0.60	0.10	0.60	1.30	0	1	-0.50	0.10	-0.40	0.10

ฟังก์ชันแยกเชิงเส้นไม่ได้

จากรูปจะเห็นได้ว่าฟังก์ชัน AND เป็นฟังก์ชันที่แยก (ระหว่างตัวอย่างบวกกับตัวอย่างลบ) ได้ด้วยเส้นตรง ส่วนฟังก์ชัน XOR เราไม่สามารถหา*เส้นตรง*ที่มาแบ่งตัวอย่างบวกและลบออกจากกัน (ไม่สามารถลากเส้นตรงให้ตัวอย่างบวกและลบให้อยู่คนละด้านของเส้น) ตัวอย่างการเรียนรู้ฟังก์ชัน XOR ข้างต้นได้แสดงให้เห็นว่า เพอร์เซปตรอนเรียนรู้บางฟังก์ชันไม่ได้ ฟังก์ชันเหล่านี้เรียกว่า *ฟังก์ชันแยกเชิงเส้นไม่ได้ (linearly non-separable function)* ส่วนฟังก์ชันที่แยกได้เรียกว่า *ฟังก์ชันแยกเชิงเส้นได้ (linearly separable function)* ซึ่งเป็นข้อจำกัดของเพอร์เซปตรอน เมื่อเราย้อนกลับไปดูตารางที่ 6–21 จะพบว่านอกจากการเรียนรู้ฟังก์ชัน XOR ไม่สำเร็จแล้ว การเรียนรู้ก็จะไม่ลู่เข้าสู่เส้นตรงใดเส้นตรงหนึ่งอีกด้วย ดังจะเห็นได้จากการที่เวกเตอร์น้ำหนักจะแกว่งไปมา การไม่ลู่เข้าก่อให้เกิดปัญหาในการเรียนรู้เพราะเราจะไม่รู้ว่าเมื่อไรจะหยุดอัลกอริทึม พิจารณาตัวอย่างใน

รูปที่ 6–38 ซึ่งมีตัวอย่างสอนเหมือนกับในรูปที่ 6–36 ยกเว้นว่าตัวอย่างตัวที่แปดในตารางที่ 6–16 มีการบันทึกค่าของประเภทผิดจาก 1 เป็น -1 ทำให้เกิดตัวอย่างลบปะปนไปในกลุ่มของตัวอย่างบวกดังแสดงในรูป ในกรณีเช่นนี้เส้นตรงที่ดีที่สุดก็ยังคงเป็นเส้นตรงเดิมเหมือนกับในรูปที่ 6–36 แต่ว่ากฎการเรียนรู้เพอร์เซปตรอนจะไม่ให้คำตอบเป็นเส้นตรงนี้ เนื่องจากอัลกอริทึมไม่ลู่เข้าสู่เส้นตรงเดียว แต่จะแกว่งไปมา

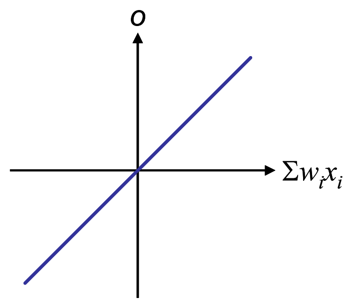


รูปที่ 6–38 เส้นตรงที่ให้ค่าผิดพลาดน้อยสุด

กฎเดลต้า (delta rule) เป็นกฎการเรียนรู้สำหรับหาค่าเวกเตอร์น้ำหนักของเพอร์เซปตรอนอีกกฎหนึ่งและมีข้อดีที่การเรียนรู้จะลู่เข้าสู่ระนาบหลายมิติที่ให้ค่าผิดพลาดน้อยสุด แม้ว่าตัวอย่างจะเป็นฟังก์ชันแบบแยกเชิงเส้นไม่ได้ กฎนี้ใช้หลักการของ*การเคลื่อนลงตามความชัน (gradient descent)* เพื่อหาคำตอบจากปริภูมิของเวกเตอร์น้ำหนักที่เป็นไปได้ ซึ่งกฎนี้เป็นพื้นฐานของอัลกอริทึมการแพร่กระจายย้อนกลับ (back-propagation) ดังจะกล่าวต่อไป

กฎเดลต้านี้จะหาเวกเตอร์น้ำหนักที่ให้ค่าผิดพลาดของตัวอย่างสอนน้อยสุดโดยใช้การหาอนุพันธ์ทางคณิตศาสตร์ ซึ่งในการหาค่าน้อยสุดด้วยอนุพันธ์นั้นจำเป็นต้องใช้ฟังก์ชันกระตุ้นที่หาอนุพันธ์ได้ ฟังก์ชันที่เราเคยใช้ก่อนหน้านี้เช่นฟังก์ชันสองขั้วและฟังก์ชันไบนารีเป็นฟังก์ชันที่หาอนุพันธ์ไม่ได้ในบางจุด ดังนั้นในกฎเดลต้านี้เราจะใช้ฟังก์ชันกระตุ้นแบบ*ฟังก์ชันเชิงเส้น (linear function)* ดังแสดงในรูปที่ 6–39 ซึ่งค่าเอาต์พุต (o) แสดงโดย $o(\vec{x}) = \vec{w} \cdot \vec{x} = \sum w_i x_i$ กล่าวคือเอาต์พุตจะเท่ากับผลรวมเชิงเส้น แม้ว่าตัวอย่างสอนจะแบ่งเป็นสองกลุ่ม (เช่น 1 กับ -1) ก็ไม่เกิดปัญหาเมื่อเราใช้ฟังก์ชันเชิงเส้นโดยจะทำนาย

ประเภทของตัวอย่างได้โดยดูที่เครื่องหมาย เช่นถ้าฟังก์ชันเชิงเส้นให้เอาต์พุตเป็น -0.21 ก็ให้ทำนายประเภทเป็น -1 เป็นต้น



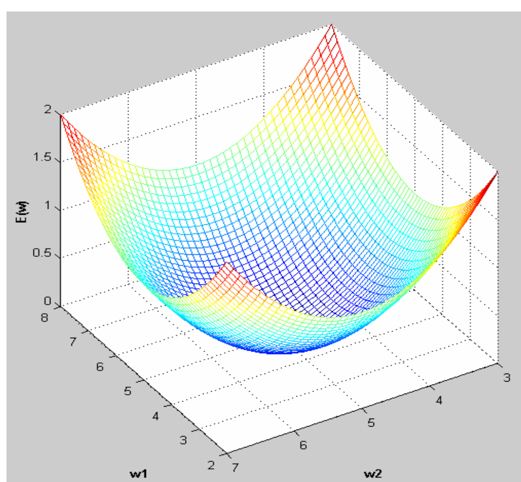
รูปที่ 6–39 ฟังก์ชันเชิงเส้น

ดังที่กล่าวข้างต้น กฎเดลต้าจะหาค่าเวกเตอร์น้ำหนักที่ให้ค่าผิดพลาดต่ำสุด ดังนั้นเรานิยามฟังก์ชันค่าผิดพลาดการสอน (training error function) $E(\vec{w})$ ดังนี้

$$E(\vec{w}) = \frac{1}{2}\sum_{d \in D}(t_d - o_d)^2 \tag{6.10}$$

โดยที่ D เป็นเซตของตัวอย่างสอน t_d เป็นเอาต์พุตเป้าหมายของตัวอย่าง d และ o_d เป็นเอาต์พุตของเพอร์เซปตรอนสำหรับตัวอย่าง d

ฟังก์ชันค่าผิดพลาดการสอน $E(\vec{w})$ เป็นฟังก์ชันของ $\vec{w}$ จะมีค่า $\vec{w}$ บางตัวที่ทำให้ฟังก์ชันมีค่าต่ำสุด และพบว่าจะมี $\vec{w}$ เช่นนั้นแค่ตัวเดียวเพราะ $E(\vec{w})$ เป็นฟังก์ชันพาราโบล่าของ $\vec{w}$ ในกรณีที่ $\vec{w}$ ประกอบด้วยน้ำหนัก 2 ค่าคือ w_1 และ w_2 เราจะได้ฟังก์ชันดังรูปที่ 6–40



รูปที่ 6–40 ฟังก์ชันค่าผิดพลาดการสอน $E(\vec{w})$

คุณสมบัติของฟังก์ชันพาราโบล่าคือจะมีค่าต่ำสุดเพียงค่าเดียว ในการหาค่าต่ำสุดเราสามารถทำได้โดยกำหนดจุด (w_1,w_2) เริ่มต้น สมมติว่าเป็น (w_{10},w_{20}) จากนั้นหาเวกเตอร์สัมผัสพาราโบล่า ณ ตำแหน่ง $E(w_{10},w_{20})$ แล้วเราจะวิ่งลงตามความชันของเวกเตอร์ที่สัมผัสกับผิวค่าผิดพลาด (error surface) ถ้าชันมากก็ปรับค่าเวกเตอร์น้ำหนักมาก ถ้าชันน้อยก็ปรับค่าน้อยจนกระทั่งมาถึงจุดต่ำสุด ซึ่ง ณ จุดนี้ความชันจะเท่ากับศูนย์และไม่ต้องปรับค่าเวกเตอร์น้ำหนักอีกต่อไป ดังนั้นการใช้หลักการนี้ต้องการการหาอนุพันธ์ของผิวค่าผิดพลาด ซึ่งจะได้เป็นความชันของผิวสัมผัสกับผิวค่าผิดพลาด $E(\vec{w})$ นี้ (เขียนแทนด้วย $\nabla E(\vec{w})$) ดังแสดงต่อไปนี้

$$\nabla E(\vec{w}) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \ldots, \frac{\partial E}{\partial w_n} \right] \qquad (6.11)$$

เนื่องจากเวกเตอร์สัมผัสนี้มีทิศในแนวขึ้น แต่เราต้องการวิ่งลงดังนั้นเวกเตอร์ในแนวลงจึงเป็น $-\nabla E(\vec{w})$ เราจะได้ว่ากฎการปรับค่าเวกเตอร์น้ำหนักเป็น $\vec{w} \leftarrow \vec{w} + \Delta\vec{w}$ โดยที่ $\Delta\vec{w} = -\eta \nabla E(\vec{w})$ และ η คืออัตราการเรียนรู้เป็นค่าคงที่ตัวเลขบวก กฎเดลต้านี้สามารถเขียนให้อยู่ในรูปของสมาชิกแต่ละตัวของเวกเตอร์น้ำหนักได้เป็น $w_i \leftarrow w_i + \Delta w_i$ โดยที่ $\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$

$\frac{\partial E}{\partial w_i}$ สามารถคำนวณได้ดังต่อไปนี้

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} 2(t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) \\
&= \sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (t_d - \vec{w} \cdot \vec{x}_d) \\
\frac{\partial E}{\partial w_i} &= \sum_{d \in D} (t_d - o_d)(-x_{id})
\end{aligned}$$

โดยที่ $-x_{id}$ คือสมาชิก x_i ของตัวอย่าง d

$$\therefore \Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{id} \qquad (6.12)$$

อัลกอริทึมของกฎเดลต้าเป็นดังตารางที่ 6–22 ต่อไปนี้

ตารางที่ 6–22 อัลกอริทึมกฎเดลต้า

Algorithm: Delta-Rule(training-examples, η**)**

Each training example is a pair $<\vec{x},t>$, where $\vec{x}$ is the vector of input values, and t is the target output value. η is the learing rate.

1. Initialize each w_i to some small random value.
2. **UNTIL** the termination condition is met **DO**
   - 2.1 Initialize each Δw_i to zero.
   - 2.2 **FOR EACH** $<\vec{x},t>$ in training-examples **Do**
     - Input the instance $\vec{x}$ to the unit and compute the output o.
     - **FOR EACH** linear unit weight w_i **DO**

$$\Delta w_i \leftarrow \Delta w_i + \eta(t-o)x_i$$

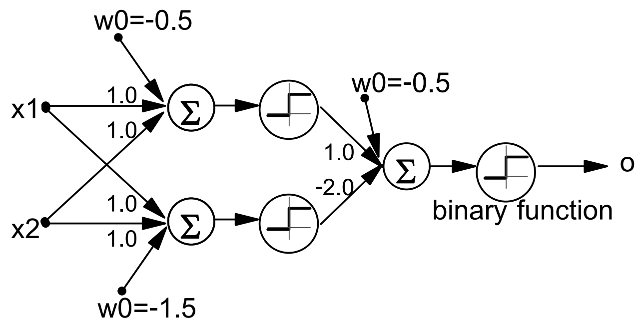
   - 2.3 **FOR EACH** linear weight w_i **DO**

$$w_i \leftarrow w_i + \Delta w_i$$

อัลกอริทึมกฎเดลต้าข้างต้นนี้จะหาค่าเวกเตอร์น้ำหนักที่ให้ค่าความผิดพลาดน้อยสุด ซึ่งมีข้อดีที่อัลกอริทึมจะลู่เข้า อย่างไรก็ดีฟังก์ชันแยกเชิงเส้นไม่ได้ที่เรียนรู้ไม่ได้ด้วยกฎเรียนรู้เพอร์เซปตรอนก็ไม่สามารถแยกได้อย่างถูกต้องสมบูรณ์ด้วยกฎเดลต้าเช่นกัน ในหัวข้อต่อไปจะกล่าวถึงข่ายงานหลายชั้นที่สามารถแยกฟังก์ชันประเภทนี้ได้

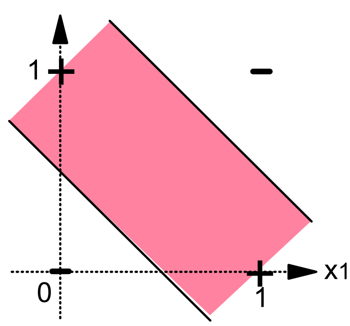
6.7.3 ข่ายงานหลายชั้นและการแพร่กระจายย้อนกลับ

จากข้างต้นจะเห็นว่าเพอร์เซปตรอนสามารถเรียนรู้ฟังก์ชันแยกได้เชิงเส้นเท่านั้น ในส่วนนี้จะอธิบายการนำเพอร์เซปตรอนหลายๆ ตัวมาเชื่อมต่อกัน เพื่อสร้างเป็นข่ายงานประสาทหลายชั้น (multilayer neural network) ที่สามารถแสดงผิวตัดสินใจไม่เชิงเส้น (non-linear decision surface) เพื่อให้เห็นถึงประสิทธิภาพของข่ายงานหลายชั้น จะยกตัวอย่างการต่อเพอร์เซปตรอน 3 ตัวเข้าด้วยกันเพื่อเรียนรู้ฟังก์ชัน XOR ดังแสดงในรูปที่ 6–41



รูปที่ 6–41 ข่ายงานหลายชั้นสามารถเรียนรู้ฟังก์ชัน XOR

รูปที่ 6–41 แสดงการเชื่อมต่อเพอร์เซปตรอน 3 ตัวเข้าด้วยกัน เพอร์เซปตรอนสองตัวแรกรับอินพุตโดยตรงส่วนเพอร์เซปตรอนตัวที่สามรับอินพุตจากเอาต์พุตของเพอร์เซปตรอนสองตัวแรก จะเห็นได้ว่าเพอร์เซปตรอนตัวแรกที่อยู่ด้านซ้ายบนของรูปนั้นแทนฟังก์ชันเชิงเส้น x_1+x_2 = 0.5 ส่วนเพอร์เซปตรอนตัวที่สองที่อยู่ด้านซ้ายล่างของรูปนั้นแทนฟังก์ชันเชิงเส้น x_1+x_2 = 1.5 ฟังก์ชันเชิงเส้นทั้งสองมีความชันเท่ากันเท่ากับ -1 แต่มีจุดตัดแกนต่างกันดังแสดงในรูปที่ 6–42 ส่วนเพอร์เซปตรอนตัวที่สามทำหน้าที่รวมผลลัพธ์จากเพอร์เซปตรอนสองตัวแรก และโดยการกำหนดเวกเตอร์น้ำหนักที่เหมาะสมของเพอร์เซปตรอนตัวที่สามทำให้ได้ผิวตัดสินใจที่อยู่ระหว่างเส้นตรงทั้งสองเป็นตัวอย่างบวกและที่อยู่ด้านนอกเป็นตัวอย่างลบ

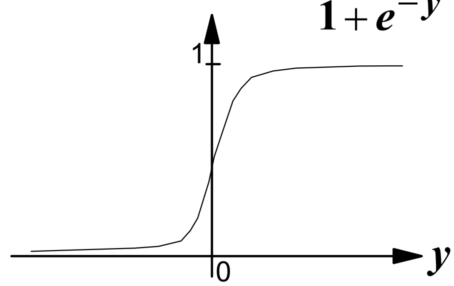


x1	x2	x1 XOR x2
0	0	0
0	1	1
1	0	1
1	1	0

รูปที่ 6–42 ผิวตัดสินใจของข่ายงานในรูปที่ 6–41

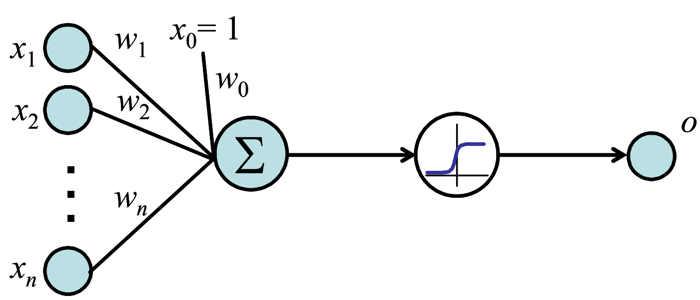
ในการเชื่อมต่อครั้งนี้ใช้ฟังก์ชันกระตุ้นแบบไบนารีเพื่อให้ง่ายต่อการทำความเข้าใจ แต่การคำนวณหากฎเรียนรู้สำหรับข่ายงานหลายชั้นต้องใช้ฟังก์ชันกระตุ้นที่หาอนุพันธ์ได้ ดังนั้นเราจะไม่ใช้ฟังก์ชันไบนารีกับข่ายงานหลายชั้น แต่จะใช้ฟังก์ชันเชิงเส้นหรืออาจใช้ฟังก์ชันซิกมอยด์ (sigmoid function) ดังแสดงในรูปที่ 6–43

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



รูปที่ 6–43 ฟังก์ชันซิกมอยด์

เพอร์เซปตรอนที่ใช้ฟังก์ชันกระตุ้นเป็นฟังก์ชันซิกมอยด์แสดงในรูปที่ 6–44

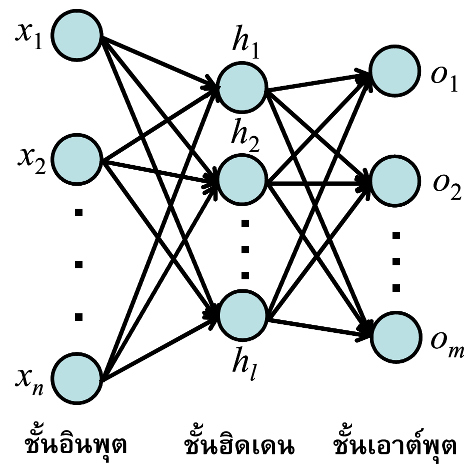


รูปที่ 6–44 เพอร์เซปตรอนที่ใช้ฟังก์ชันซิกมอยด์

คุณสมบัติหนึ่งของฟังก์ชันซิกมอยด์ก็คือสามารถแสดงอนุพันธ์ของฟังก์ชันในรูปของเอาต์พุตได้อย่างง่าย กล่าวคือ

$$\frac{d\sigma(y)}{dy} = \sigma(y)(1-\sigma(y)) \tag{6.13}$$

อัลกอริทึมการแพร่กระจายย้อนกลับ (backpropagation algorithm) [Rumelhart & McClelland, 1986] เรียนรู้ค่าเวกเตอร์น้ำหนักสำหรับข่ายงานป้อนไปหน้าแบบหลายชั้น (multilayer feedforward network) โดยใช้การเคลื่อนลงตามความชันเพื่อหาค่าต่ำสุดของค่าผิดพลาดระหว่างเอาต์พุตของข่ายงานกับเอาต์พุตเป้าหมาย ตัวอย่างของข่ายงานป้อนไปหน้าแบบหลายชั้นแสดงในรูปที่ 6–45



รูปที่ 6–45 ตัวอย่างข่ายงานป้อนไปหน้าแบบหลายชั้น

ตัวอย่างในรูปด้านบนแสดงข่ายงานป้อนไปหน้าแบบหลายชั้นซึ่งประกอบด้วยชั้นอินพุต ชั้นฮิดเดนหรือชั้นซ่อน และชั้นเอาต์พุต ในรูปแสดงชั้นฮิดเดนเพียงชั้นเดียวแต่อาจมีมากกว่าหนึ่งชั้นก็ได้ เส้นเชื่อมจะเชื่อมต่อเป็นชั้นๆ ไม่ข้ามชั้นจากชั้นอินพุตไปชั้นฮิดเดน ถ้ามีชั้นฮิดเดนมากกว่าหนึ่งชั้นก็เชื่อมต่อกันไป และสุดท้ายจากชั้นฮิดเดนไปชั้นเอาต์พุต ข่ายงานป้อนไปหน้าแบบหลายชั้นนี้จะไม่มีเส้นเชื่อมย้อนกลับจะมีแต่เส้นเชื่อมไปข้างหน้าอย่างเดียวเช่นไม่มีเส้นเชื่อมจากบัพในชั้นเอาต์พุตส่งกลับมายังบัพในชั้นฮิดเดนหรือชั้นอินพุต เป็นต้น

ในการปรับค่าเวกเตอร์น้ำหนักโดยอัลกอริทึมการแพร่กระจายย้อนกลับนั้น เราต้องนิยามค่าผิดพลาดการสอนสำหรับข่ายงาน $E(\vec{w})$ จากนั้นจะหาค่าเวกเตอร์น้ำหนักที่ให้ค่าผิดพลาดต่ำสุด นิยามค่าผิดพลาดดังนี้

$$E(\vec{w}) = \frac{1}{2}\sum_{d\in D}\sum_{k\in outputs}(t_{kd} - o_{kd})^2 \qquad (6.14)$$

โดยที่ *outputs* คือเซตของบัพเอาต์พุตในข่ายงาน t_{kd} และ o_{kd} เป็นค่าเอาต์พุตเป้าหมายและเอาต์พุตที่ได้จากข่ายงานตามลำดับของบัพเอาต์พุตที่ k ของตัวอย่างตัวที่ d อัลกอริทึมการแพร่กระจายย้อนกลับจะค้นหาเวกเตอร์น้ำหนักที่ให้ค่าผิดพลาดต่ำสุด แต่ในกรณีของข่ายงานป้อนไปหน้าแบบหลายชั้นนี้ค่าต่ำสุดมักมีมากกว่าหนึ่งจุด ดังนั้นคำตอบของการแพร่กระจายย้อนกลับจึงเป็นค่าต่ำสุดเฉพาะที่ อัลกอริทึมแสดงในตารางที่ 6–23

ตารางที่ 6–23 อัลกอริทึมการแพร่กระจายย้อนกลับ

```
Algorithm: Backpropagation(training-examples,h,n_in,n_out,n_hidden)
```

Each training example is a pair $<\vec{x},\vec{t}>$,where $\vec{x}$ is the input vector, $\vec{t}$ is the target output vector, η is the learning rate. n_{in},n_{out}, n_{hidden} are the number of network inputs, units in the hidden layer, output units, respectively. The input from unit i into unit j and the weight from unit i to unit j are denoted x_{ji} and w_{ji}

1. Initialize all network weights to small random numbers (e.g., [-0.05..0.05])
2. **UNTIL** the termination condition is met **DO**

   2.1 **FOR EACH** $<\vec{x},\vec{t}>$ in training-examples **DO**

   /*Propagate input forward through the network*/

   - Input the instance $\vec{x}$ to the network, compute the output o_u of every unit u.

   /*Propagate errors backward through the network*/

   - For each network output unit k, calculate its error term δ_k

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

   - For each hidden unit h, calculate its error term δ_h

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

   - Update each network weight w_{ji} : $w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$

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

6.8 การเรียนรู้แบบเบส์

การเรียนรู้แบบเบส์ (Bayesian learning) เป็นวิธีการเรียนรู้ที่ใช้ทฤษฎีความน่าจะเป็นซึ่งมีพื้นฐานมาจาก*ทฤษฎีของเบส์ (Bayes theorem)* เข้ามาช่วยในการเรียนรู้ จุดมุ่งหมายก็เพื่อต้องการสร้างโมเดลที่อยู่ในรูปของความน่าจะเป็น ซึ่งเป็นค่าที่บันทึกได้จากการสังเกต จากนั้นนำโมเดลมาหาว่าสมมติฐานใดถูกต้องที่สุดโดยใช้ความน่าจะเป็นเข้ามาช่วย ข้อดีก็คือเราสามารถใช้ข้อมูลและ*ความรู้ก่อนหน้า (prior knowledge)* เข้ามาช่วยในการเรียนรู้ได้ด้วย ความรู้ก่อนหน้าหมายถึงความรู้ที่เรามีเกี่ยวกับสมมติฐานแต่ละตัวก่อนที่เราจะเก็บข้อมูล เมื่อใช้งานเราจะนำความน่าจะเป็นของข้อมูลที่เก็บได้มาปรับสมมติฐานซ้ำอีกครั้ง ซึ่งพบว่าวิธีนี้ให้ประสิทธิภาพในการเรียนรู้ได้ดีไม่ด้อยกว่าวิธีการเรียนรู้ประเภทอื่น

6.8.1 ทฤษฎีของเบส์

กำหนดให้ A และ B เป็นเหตุการณ์ใดๆ ความน่าจะเป็นของ A *เมื่อรู้* B (ความน่าจะเป็นที่จะเกิดเหตุการณ์ A โดยมีเงื่อนไขว่าเหตุการณ์ B ได้เกิดขึ้นแล้ว) เขียนแทนด้วย P(A|B) สามารถคำนวณได้ด้วยทฤษฎีของเบส์ดังนี้

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{6.15}$$

ความน่าจะเป็นก่อน และ ความน่าจะเป็นภายหลัง

กล่าวคือความน่าจะเป็นของ A *เมื่อรู้ B (โดยมีเงื่อนไขว่า B เกิดขึ้นแล้ว)* สามารถคำนวณได้จากผลคูณของความน่าจะเป็นของ B เมื่อรู้ A กับความน่าจะเป็นของ A หารด้วยความน่าจะเป็นของ B เราเรียก P(A) ว่าเป็น*ความน่าจะเป็นก่อน (prior probability)* และเรียก P(A|B) ว่าเป็น*ความน่าจะเป็นภายหลัง (posterior probability)* ความน่าจะเป็นก่อนเป็นค่าที่ได้จากข้อมูลเบื้องต้น ส่วนความน่าจะเป็นภายหลังเป็นค่าความน่าจะเป็นก่อนที่ถูกปรับด้วยข้อมูลที่เพิ่มขึ้น

ในกรณีของการเรียนรู้ของเครื่องนั้น สิ่งที่เราสนใจก็คือเมื่อเรามีชุดข้อมูลหรือเซตของตัวอย่างสอน D เราต้องการหาค่าความน่าจะเป็นที่สมมติฐาน (h) ที่เราสนใจว่ามีโอกาสจะเกิดขึ้นเท่าไร เราก็สามารถใช้ทฤษฎีของเบส์ในการคำนวณได้ดังนี้

$$P(h \mid D) = \frac{P(D \mid h)P(h)}{P(D)} \tag{6.16}$$

โดยที่ P(h) คือความน่าจะเป็นก่อนซึ่งเป็นความน่าจะเป็นที่สมมติฐาน h จะเป็นจริงโดยที่เรายังไม่ได้ดูข้อมูลตัวอย่างสอน ส่วน P(h|D) เป็นความน่าจะเป็นภายหลังซึ่งเป็นความน่าจะเป็นที่สมมติฐาน h จะเป็นจริงโดยมีเงื่อนไขว่า D เป็นจริง (เราเห็นข้อมูลตัวอย่างสอน D แล้ว) ในการเรียนรู้ของเครื่อง เราต้องการคำนวณความน่าจะเป็นภายหลังนี้ ซึ่งมักจะหาไม่ได้โดยตรง แต่ถ้าเราใช้ทฤษฎีของเบส์ดังข้างต้นความน่าจะเป็นนี้จะคำนวณได้ง่ายขึ้นโดยใช้นิพจน์ทางด้านขวามือของสูตรที่ (6.16)

ยกตัวอย่างเช่นถ้าเรามีต้นไม้ตัดสินใจหลายๆ ต้นและอยากทราบว่าแต่ละต้นมีโอกาสเกิดขึ้นหรือมีความถูกต้องเท่าไร ก็คือเราต้องการหา P(h|D) นั่นเอง โดยที่ h แทนต้นไม้ตัดสินใจต้นหนึ่งที่เรากำลังพิจารณา เราอาจจะมีความเชื่อว่าต้นไม้ต้นเล็กมีโอกาสที่จะเป็นจริงมากกว่าต้นใหญ่ (คล้ายกับกฎของอ็อกแคม) นั่นคือเรามีความน่าจะเป็นก่อน P(h) ที่ต้นไม้จะเป็นจริงโดยยังไม่ได้ดูตัวอย่างสอน ซึ่งจะให้ค่าความน่าจะเป็นของต้นไม้ต้นเล็กมีค่ามากกว่าของต้นไม้ต้นใหญ่ เมื่อเรารับตัวอย่างสอนแล้วนำมาปรับค่าความน่าจะเป็นก่อน ได้เป็นความน่าจะเป็นภายหลัง ส่วน P(D|h) เป็นความน่าจะเป็นที่ D จะเป็นจริงเมื่อรู้ว่า h เป็นจริง ความน่าจะเป็นค่านี้สามารถวัดได้โดยนำตัวอย่างสอนมาตรวจสอบกับต้นไม้ h ว่าในจำนวนตัวอย่างสอนทั้งหมดนั้นมีอัตราส่วนของตัวอย่างที่ตรงหรือสอดคล้องกับต้นไม้เท่าไร ส่วน P(D) เป็นความน่าจะเป็นที่เซตตัวอย่างสอนจะเป็นจริง ซึ่งในการหา h ที่ดีที่สุดนั้นโดยมากเรามักละค่านี้ได้โดยไม่ต้องนำมาคำนวณดังจะกล่าวต่อไป ดังนั้นจะเห็นได้ว่าการใช้ทฤษฎีของเบส์สามารถใช้คำนวณความน่าจะเป็นของสมมติฐานแต่ละตัว เมื่อรู้ว่าเซตตัวอย่างสอนเป็นจริงซึ่งจะช่วยให้เราเลือกสมมติฐานที่ดีที่สุดได้

สมมติฐานภายหลังมากสุด

เราเรียกสมมติฐานที่ดีที่สุดว่า *สมมติฐานภายหลังมากสุด – เอ็มเอพี (Maximum A Posterior hypothesis – MAP)* ซึ่งนิยามให้เป็นดังนี้

$$
\begin{aligned}
h_{MAP} &= \arg\max_{h\in H} P(h\,|\,D) \\
&= \arg\max_{h\in H} \frac{P(D\,|\,h)P(h)}{P(D)}
\end{aligned} \tag{6.17}
$$

$$
h_{MAP} = \arg\max_{h\in H} P(D\,|\,h)P(h) \tag{6.18}
$$

โดยที่ H เป็นปริภูมิของสมมติฐานทั้งหมด arg max f(x) เป็นฟังก์ชันที่คืนค่า x ที่ทำให้ f(x) สูงสุด สมการที่ (6.17) ได้จากการใช้ทฤษฎีของเบส์และเนื่องจากว่าสำหรับ $h \in H$ ทุกตัวมี

ค่า P(D) เท่ากันหมด ดังนั้นเราจึงสามารถละ P(D) ได้และได้สมการที่ (6.18) กล่าวคือ h ที่ดีที่สุดตามเอ็มเอพีคือ h ที่ทำให้ค่า P(D|h)P(h) มีค่าสูงสุด

เทคนิคการเรียนรู้ของเครื่องหลายวิธีไม่ได้หาค่า h_{MAP} แต่มักหา h_{ML} (Maximum Likelihood hypothesis) ดังในสมการที่ (6.19) ด้านล่างนี้ ซึ่งหมายถึงสมมติฐานที่ตรงหรือสอดคล้องกับข้อมูลสอนมากสุดจะเป็นสมมติฐานที่ดีสุดโดยไม่ได้พิจารณาความน่าจะเป็นก่อน

$$h_{ML} = \arg\max_{h \in H} P(D \mid h) \tag{6.19}$$

ยกตัวอย่างการใช้ทฤษฎีของเบส์เพื่อเลือกสมมติฐานที่น่าจะเป็นที่สุด สมมติว่าคนไข้คนหนึ่งไปตรวจหามะเร็งและผลการตรวจเป็นบวก อย่างไรก็ดีเรามีค่าสถิติว่าผลการตรวจเมื่อเป็นบวกจะให้ความถูกต้อง 98% ของกรณีที่มีโรคนั้นอยู่จริง และผลการตรวจเมื่อเป็นลบจะให้ความถูกต้อง 97% ของกรณีที่ไม่มีโรคนั้น นอกจากนั้นเรายังมีสถิติของการเป็นโรคมะเร็งว่า 0.008 ของประชากรทั้งหมดเป็นโรคมะเร็ง คำถามคือว่าคนไข้คนนี้มีโอกาสเป็นมะเร็งหรือไม่เป็นมะเร็งมากกว่ากัน?

เราใช้ทฤษฎีของเบส์สำหรับปัญหานี้ โดยกำหนดให้ H = {cancer, ∼cancer} กล่าวคือมีสมมติฐานที่เป็นไปได้สองข้อคือคนไข้คนนี้เป็นมะเร็งกับไม่เป็นมะเร็ง เซตตัวอย่างสอนหรือข้อมูลของเราคือผลการตรวจเป็นบวก แทนด้วย + ดังนั้นเราแทนค่า H, h, D ในสมการที่ (6.18) จะได้ว่า

$$h_{MAP} = \arg\max_{h \in \{cancer, \sim cancer\}} P(+ \mid h)P(h) \tag{6.20}$$

จากข้อมูลทางสถิติทำให้ได้ว่า

P(cancer) = 0.008 P(∼cancer) = 0.992

P(+|cancer) = 0.98 P(+|∼cancer) = 0.03

ดังนั้นเราจะได้ว่าในกรณีของ

h=cancer ได้ด้านขวามือของสมการที่ (6.20) เป็น 0.98 × 0.008 = 0.00784

h=∼cancer ได้ด้านขวามือของสมการที่ (6.20) เป็น 0.03 × 0.992 = 0.02976

เพราะฉะนั้นเราสรุปได้ว่า h_{MAP} = ∼cancer กล่าวคือมีโอกาสไม่เป็นมะเร็งมากกว่า

6.8.2 สูตรพื้นฐานของความน่าจะเป็น

สูตรพื้นฐานเกี่ยวกับความน่าจะเป็น ที่จะใช้บ่อยครั้งในการเรียนรู้แบบเบส์มีดังต่อไปนี้

1. กฎผลคูณ (product rule): ความน่าจะเป็น $P(A \wedge B)$ ที่สองเหตุการณ์ A และ B จะเกิดพร้อมกัน (หรือเขียนย่อเป็น P(A,B)) มีค่าเท่ากับ

$$P(A \wedge B) = P(A|B)P(B) = P(B|A)P(A)$$

2. กฎผลรวม (sum rule): ความน่าจะเป็น $P(A \vee B)$ ที่เหตุการณ์ A หรือ B เหตุการณ์ใดเหตุการณ์หนึ่งจะเกิดหรือเกิดพร้อมกันมีค่าเท่ากับ

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

3. ทฤษฎีความน่าจะเป็นทั้งหมด (theorem of total probability) ถ้าเหตุการณ์ $A_1,\ldots,A_n$ ไม่เกิดร่วมกันและ $\sum_{i=1}^{n} P(A_i) = 1$ แล้ว ความน่าจะเป็น P(B) มีค่าเท่ากับ

$$P(B) = \sum_{i=1}^{n} P(B \mid A_i)P(A_i)$$

4. กฎลูกโซ่ (chain rule): $A_1,\ldots,A_n$ เป็นเหตุการณ์ n เหตุการณ์จะได้ว่าความน่าจะเป็นร่วม $P(A_1,\ldots,A_n)$ มีค่าเท่ากับ

$$P(A_1, A_2, \cdots, A_n) = \sum_{i=1}^{n} P(A_i \mid A_{i-1}, \cdots, A_1)$$

6.8.3 การจำแนกประเภทที่น่าจะเป็นที่สุดสำหรับตัวอย่าง

ดังที่กล่าวข้างต้น ในกรณีที่กำหนดให้เราใช้สมมติฐานได้*เพียงข้อเดียว*ในการจำแนกประเภทของตัวอย่าง จะได้ว่า h_{MAP} เป็นสมมติฐานที่ดีที่สุด แต่การจำแนกประเภทของตัวอย่างด้วย h_{MAP} ไม่ใช่*การจำแนกประเภทที่น่าจะเป็นที่สุด (most probable classification)* สำหรับตัวอย่างนั้น ในบางกรณีที่เราสามารถใช้สมมติฐานหลายข้อเราสามารถจำแนกประเภทของตัวอย่างได้ดีกว่าการใช้ h_{MAP} ตัวเดียว

สมมติว่าเรามีสมมติฐาน 3 ข้อ แต่ละข้อมีค่าความน่าจะเป็นภายหลังดังต่อไปนี้

$$P(h_1|D) = 0.4 \quad P(h_2|D) = 0.3 \quad P(h_3|D) = 0.3$$

และเมื่อให้ตัวอย่าง x ผลการจำแนกประเภทของสมมติฐานเป็นดังนี้

$$h_1(x) = + \qquad h_2(x) = - \qquad h_3(x) = -$$

ในกรณีนี้เราควรจะจำแนกประเภทของ x เป็นบวกหรือลบ? ซึ่งถ้าใช้ h_{MAP} ก็จะได้ว่า h_1 เป็นสมมติฐานที่ดีที่สุดเนื่องจาก h_1 มีค่าความน่าจะเป็นภายหลังมากที่สุด แต่เมื่อพิจารณาดูสมมติฐานอื่นในปริภูมิของสมมติฐาน เราพบว่า h_{MAP} ให้คำตอบเป็น + เพียงตัวเดียว แต่สมมติฐานอีกสองตัวให้คำตอบเป็น – เราจะได้ว่าการจำแนกประเภทที่น่าจะเป็นที่สุดในแบบของเบส์มีสูตรการคำนวณดังนี้

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) P(h_i \mid D) \tag{6.21}$$

โดยที่ V เป็นเซตของค่า (ประเภท) ของตัวอย่าง H เป็นปริภูมิของสมมติฐาน ในตัวอย่างด้านบนเราจะได้ว่า

$P(h_1 \mid D) = 0.4$	$P(- \mid h_1) = 0.0$	$P(+ \mid h_1) = 1.0$
$P(h_2 \mid D) = 0.3$	$P(- \mid h_1) = 1.0$	$P(+ \mid h_1) = 0.0$
$P(h_3 \mid D) = 0.3$	$P(- \mid h_1) = 1.0$	$P(+ \mid h_1) = 0.0$

ทำให้ได้ค่าความน่าจะเป็นของประเภท + และ – ดังนี้

$$\sum_{h_i \in H} P(+ \mid h_i) P(h_i \mid D) = 0.4$$
$$\sum_{h_i \in H} P(- \mid h_i) P(h_i \mid D) = 0.6$$

ดังนั้น

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j \mid h_i) P(h_i \mid D) = -$$

6.8.4 ตัวจำแนกประเภทเบส์อย่างง่าย

ตัวจำแนกประเภทเบส์อย่างง่าย (naive Bayes classifier) เป็นตัวจำแนกประเภทแบบหนึ่งที่ใช้งานได้ดี เหมาะกับกรณีของเซตตัวอย่างมีจำนวนมากและคุณสมบัติ (attribute) ของตัวอย่างไม่ขึ้นต่อกัน มีการนำตัวจำแนกประเภทเบส์อย่างง่ายไปประยุกต์ใช้งานในด้านการจำแนกประเภทข้อความ (text classification) การวินิจฉัย (diagnosis) และพบว่าใช้งานได้ดีไม่ต่างจากการจำแนกประเภทวิธีการอื่น เช่นการเรียนรู้ต้นไม้ตัดสินใจ ข่ายงานประสาท เป็นต้น

สมมติให้ $A_1, A_2, \cdots, A_n$ เป็นคุณสมบัติของตัวอย่าง เราจะได้ว่าค่า (ประเภท) ที่น่าจะเป็นที่สุดของตัวอย่าง x คือ

$$\begin{aligned} v_{MAP} &= \underset{v_j \in V}{\arg\max}\, P(v_j \mid a_1, a_2, \cdots, a_n) \\ &= \underset{v_j \in V}{\arg\max} \frac{P(a_1, a_2, \cdots, a_n \mid v_j) P(v_j)}{P(a_1, a_2, \cdots, a_n)} \end{aligned} \tag{6.22}$$

$$v_{MAP} = \underset{v_j \in V}{\arg\max}\, P(a_1, a_2, \cdots, a_n \mid v_j) P(v_j) \tag{6.23}$$

โดยที่ a_i ในสมการเป็นค่าของคุณสมบัติ A_i V เป็นเซตของประเภทหรือค่าที่เป็นไปได้ของ x สมการที่ (6.23) แสดงการหาประเภทที่ดีสุดของตัวอย่าง $\boldsymbol{x}$ แต่เราจะพบว่าสมการนี้ใช้งานไม่ได้อย่างมีประสิทธิภาพ เนื่องจากว่าการคำนวณค่าของ $P(a_1, a_2, \cdots, a_n \mid v_j)$ ทำได้ยากลำบากมากเพื่อให้ได้ค่าที่น่าเชื่อถือในเชิงสถิติ ที่เป็นเช่นนี้เพราะว่าถ้าให้คุณสมบัติ A_i แต่ละตัวของตัวอย่างมีค่าที่เป็นไปได้ 10 ค่า และคุณสมบัติทั้งหมดมี 10 ตัว เราจะได้ว่ามีลำดับ a_1, $a_2, \cdots,$ a_n ที่เป็นไปได้ทั้งสิ้นเท่ากับ 10^{10} รูปแบบ ซึ่งหมายถึงว่าเราต้องหาตัวอย่างทั้งสิ้น 10^{10} ตัว จึงจะมีโอกาสพบรูปแบบหนึ่งๆ ของ a_1, $a_2, \cdots,$ a_n สักหนึ่งครั้งโดยประมาณ ดังนั้นถ้าต้องการให้ค่า $P(a_1, a_2, \cdots, a_n \mid v_j)$ มีความน่าเชื่อถือเชิงสถิติ เราต้องการตัวอย่างมากกว่า 10^{10} ตัวหลายเท่า ซึ่งการที่จะหาตัวอย่างจำนวนมากขนาดนั้นแทบจะทำไม่ได้จริงในทางปฏิบัติ เราจึงต้องการโมเดลที่จะคำนวณ $P(a_1, a_2, \cdots, a_n \mid v_j)$ ให้ได้ในเชิงปฏิบัติ

สมมติฐานของตัวจำแนกประเภทเบส์อย่างง่ายคือ เรากำหนดให้คุณสมบัติแต่ละตัวไม่ขึ้น (เป็นอิสระ) กับคุณสมบัติอื่นๆ ซึ่งทำให้เราสามารถเขียนแทน $P(a_1, a_2, \cdots, a_n \mid v_j)$ ด้วยผลคูณของค่าความน่าจะเป็นด้านล่างนี้ที่หาค่าได้ง่ายขึ้น

$$P(a_1, a_2, \cdots, a_n \mid v_j) = \prod_{i=1}^{n} P(a_i \mid v_j) \tag{6.24}$$

โดยที่ $\prod$ หมายถึงการนำค่า $P(a_i \mid v_j)$ ทั้งหมดมาคูณกัน สูตรนี้ถ้าใช้กฎลูกโซ่มาคำนวณค่าความน่าจะเป็นที่ด้านซ้ายของสูตรจะได้เท่ากับ $P(a_1 \mid v_j) \times P(a_2 \mid a_1, v_j) \times P(a_3 \mid a_2, a_1, v_j) \times \cdots \times P(a_n \mid a_{n\text{-}1}, a_{n\text{-}2}, \ldots, a_1, v_j)$ ดังนั้นค่าความน่าจะเป็นทางด้านซ้ายของสมการจะเท่ากับผลคูณค่าความน่าจะเป็นทางด้านขวาก็ต่อเมื่อคุณสมบัติ a_1, $a_2, \cdots,$ a_n ไม่ขึ้นต่อกัน เช่นสีผมไม่ขึ้นกับส่วนสูง ฯลฯ แต่ในความเป็นจริงแล้วคุณสมบัติส่วนใหญ่มักจะมีความสัมพันธ์กัน เช่นส่วนสูงกับน้ำหนัก เพราะถ้าตัวสูงน้ำหนักก็จะมากตามไปด้วย แต่อย่างไรก็ตามการใช้*สมมติฐานความไม่ขึ้นต่อกัน (conditional independence assumption)* นี้จะช่วยให้เราคำนวณค่าความน่าจะเป็นในสูตรที่ (6.24) ได้ง่ายขึ้น เพราะค่าความน่าจะเป็นของ a_i เมื่อรู้ v_j หาได้ง่ายกว่า เช่นถ้าจะหาคนผมสีน้ำตาล ส่วนสูงมาก น้ำหนักมากและไม่ใช้โลชันไปผึ่งแดดแล้วผิวจะไหม้หรือไม่ เมื่อเอาไปหาดูในฐานข้อมูลอาจจะมีโอกาส

สมมติฐานความไม่ขึ้นต่อกัน

พบข้อมูลที่มีค่าครบทั้ง 4 ค่านี้น้อยมากๆ หรือต้องใช้จำนวนตัวอย่างมากมายมหาศาลถึงจะพบข้อมูลที่มีค่าครบตรงที่ต้องการ แต่ถ้าเราแยกคุณสมบัติออกจากกันเช่นหาคนผมสีน้ำตาลที่เป็นตัวอย่างบวก หรือหาคนไม่ใช้โลชันที่เป็นตัวอย่างบวก ทำให้ใช้ตัวอย่างไม่มากและได้คำตอบ ถึงแม้ว่าคำตอบที่ได้อาจจะไม่ถูกต้องสมบูรณ์แต่ก็พบว่าทำงานได้ดีในทางปฏิบัติ

ดังนั้นเราจะได้ว่าตัวจำแนกประเภทแบบเบส์อย่างง่ายคือ

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \times \prod_{i=1}^{n} P(a_i \mid v_j) \tag{6.25}$$

จากสมการด้านบนนี้เราจะได้อัลกอริทึมการเรียนรู้เบส์อย่างง่ายดังตารางที่ 6–24

ตารางที่ 6–24 อัลกอริทึมการเรียนรู้เบส์อย่างง่าย

Algorithm: Naïve-Bayes

- **Naive_Bayes_Learn(examples)**
  FOR EACH target value v DO
  $\bar{P}(v_j) \leftarrow \text{estimate } P(v_j)$
  FOR EACH attribute value a of each attribute DO
  $\bar{P}(a_i \mid v_j) \leftarrow \text{estimate } P(a_i \mid v_j)$
- **Classify_New_Example(x)**
  $$v_{NB} = \arg\max_{v_j \in V} \bar{P}(v_j) \times \prod_{i=1}^{n} \bar{P}(a_i \mid v_j)$$

ยกตัวอย่างการใช้อัลกอริทึมการเรียนรู้เบส์อย่างง่าย โดยใช้ชุดตัวอย่างสอนในตารางที่ 6–25 ต่อไปนี้

ตารางที่ 6–25 ตัวอย่างสอนสำหรับการเรียนรู้เบส์อย่างง่าย (เหมือนกับตารางที่ 6–13)

class

attribute →

value

Name	Hair	Height	Weight	Lotion	Result
Sarah	blonde	average	light	no	Sunburned
Dana	blonde	tall	average	yes	none
Alex	brown	short	average	yes	none
Annie	blonde	short	average	no	sunburned
Emily	red	average	heavy	no	sunburned
Pete	brown	tall	heavy	no	none
John	brown	average	heavy	no	none
Katie	blonde	short	light	yes	none

สมมติว่าตัวอย่างที่ต้องการจำแนกประเภทคือ

Name	Hair	Height	Weight	Lotion	Result
Judy	blonde	average	heavy	no	?

คำนวณ $v_{NB} = \arg\max_{v_j \in V} P(v_j) \times \prod_{i=1}^{n} P(a_i \mid v_j)$ โดย V = {+,–} เราจะได้ดังต่อไปนี้

กรณี v_j = + ได้ว่า

$$P(+)P(blonde|+)P(average|+)P(heavy|+)P(no|+) = \frac{3}{8} \times \frac{2}{3} \times \frac{2}{3} \times \frac{1}{3} \times \frac{3}{3} = \frac{1}{18}$$

ส่วนกรณี v_j = – ได้ว่า

$$P(-)P(blonde|-)P(average|-)P(heavy|-)P(no|-) = \frac{5}{8} \times \frac{2}{5} \times \frac{1}{5} \times \frac{2}{5} \times \frac{2}{5} = \frac{1}{125}$$

ดังนั้นได้ v_{NB} = +

การเรียนรู้เพื่อจำแนกประเภทข้อความโดยเบส์อย่างง่าย

ในการเรียนรู้เพื่อจำแนกประเภทข้อความโดยใช้เบส์อย่างง่ายนี้ สมมติว่าเรามีข้อความที่เราสนใจกับไม่สนใจ เมื่อทำการเรียนรู้แล้วเราต้องการทำนายว่าเอกสารหนึ่งๆ จะเป็นเอกสารที่เราสนใจหรือไม่ สามารถนำประยุกต์ใช้งานเช่นการกรองข่าวสารเลือกเฉพาะข่าวที่สนใจเป็นต้น

ก่อนอื่นเราให้เอกสารหนึ่งๆ คือตัวอย่างหนึ่งตัว และเราแทนเอกสารแต่ละฉบับด้วยเวกเตอร์ของคำโดยใช้คำที่ปรากฏในเอกสารเป็นคุณสมบัติของเอกสาร กล่าวคือคำที่หนึ่งในเอกสารเป็นคุณสมบัติตัวที่หนึ่ง คำที่สองในเอกสารเป็นคุณสมบัติตัวที่สอง ตามลำดับ ดังนั้นจะได้ว่า a_1 คือคำที่หนึ่ง a_2 คือคำที่สองตามลำดับ จากนั้นก็ทำการเรียนรู้โดยใช้ตัวอย่างสอนเพื่อประมาณค่าความน่าจะเป็นต่อไปนี้คือ

1. P(+) 2. P(–) 3. P(doc|+) 4. P(doc|–)

จากสมมติฐานเรื่องความไม่ขึ้นต่อกันของคุณสมบัติของเบส์อย่างง่ายทำให้เราได้ว่า

$$P(doc \mid v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k \mid v_j) \quad (6.26)$$

เมื่อ a_i คือคุณสมบัติตัวที่ i ส่วนค่าของมันคือ w_k (คำที่ k ในรายการของคำที่เรามีอยู่) $P(a_i = w_k \mid v_j)$ คือความน่าจะเป็นที่คำในตำแหน่งที่ i เป็น w_k เมื่อรู้ v_j แต่พบว่าสูตรนี้ก็ยังนำไปคำนวณยากเนื่องจากเหตุผลในทำนองเดียวกันกับสมมติฐานความไม่ขึ้นต่อกันข้างต้น

จึงสร้างสมมติฐานเพิ่มเติมดังสมการที่ (6.27) เพื่อให้การคำนวณทำได้มีประสิทธิภาพในทางปฏิบัติ

$$P(a_i = w_k \mid v_j) = P(a_m = w_k \mid v_j), \forall i, m \qquad (6.27)$$

หมายความว่าโอกาสที่เราจะเห็นคำที่หนึ่งไปปรากฏที่ตำแหน่งใดๆ มีค่าเท่ากันหมด ทำให้การคำนวณง่ายขึ้นเพราะไม่ต้องสนใจว่าคำหนึ่งๆ จะไปปรากฏในตำแหน่งใด หรือคำแต่ละคำจะไม่ขึ้นกับตำแหน่ง อัลกอริทึมสำหรับการจำแนกประเภทข้อความโดยใช้การเรียนรู้เบส์อย่างง่ายเป็นดังตารางที่ 6–26 ต่อไปนี้

ตารางที่ 6–26 อัลกอริทึมการเรียนรู้เบส์อย่างง่ายสำหรับจำแนกประเภทข้อความ

Algorithm: Learn_naive_Bayes_text(*Examples*,V)

1. Collect all words and other tokens that occur in *Examples*.
   - *Vocabulary* ← all distinct words and other tokens in *Examples*.
2. Calculate the required $P(v_j)$ and $P(w_k|v_j)$:
   - **FOR EACH** target value v_j in V **DO**
     - *docs*$_j$ ← subset of *Examples* for which the target value is v_j
     - $P(v_j) = \dfrac{|docs_j|}{Examples}$
     - *Text*$_j$ ← a single document created by concatenating all members of *docs*$_j$
     - *n* ← total number of words in *Text*$_j$ (counting duplicate words multiple times)
     - FOR EACH word w_k in *Vocabulary* **DO**
       - *n* ← number of times word w_k occurs in *Text*$_j$
       - $P(w_k \mid v_j) = \dfrac{n_k + 1}{n + |Vocabulary|}$

Algorithm: Classify_naive_Bayes_text(*Doc*)

- positions ← all word positions in *Doc* that contain tokens found in *Vocabulary*
- **Return** v_{NB}

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \times \prod_{i \in positions} P(a_i \mid v_j)$$

6.8.5 ข่ายงานความเชื่อเบส์

ข่ายงานความเชื่อเบส์ (Bayesian belief network) หรือเรียกโดยย่อว่า*ข่ายงานเบส์ (Bayes net)* เป็นวิธีการเรียนรู้ที่ลดข้อจำกัดของการเรียนรู้เบส์อย่างง่ายในสมมติฐานของความไม่ขึ้นต่อกันระหว่างคุณสมบัติ ในวิธีการเรียนรู้เบส์อย่างง่ายในหัวข้อที่แล้วจะตั้งสมมติฐานว่าคุณสมบัติใดๆ ไม่ขึ้นต่อกัน แต่ในความเป็นจริงเราพบว่าคุณสมบัติบางตัวจะขึ้นต่อกันบ้าง และควรที่จะนำความขึ้นต่อกันนี้เข้ามาใส่ไว้ในโมเดลด้วย เราจึงใช้ข่ายงานความเชื่อเบส์ในการอธิบายความไม่ขึ้นต่อกันอย่างมีเงื่อนไข (condition independent) ระหว่างตัวแปร (ในบริบทของข่ายงานความเชื่อเบส์นิยมใช้คำว่า 'ตัวแปร' (variable) แทนคำว่า 'คุณสมบัติ') และในโมเดลนี้เราสามารถใช้ (1) ความรู้ก่อน (prior knowledge) เกี่ยวกับความ(ไม่)ขึ้นต่อกันระหว่างตัวแปร ร่วมกับ (2) ตัวอย่างสอน เพื่อทำให้กระบวนการเรียนรู้มีประสิทธิภาพ โดยเราสามารถใส่ความรู้ก่อนในข่ายงานความเชื่อเบส์ให้อยู่ในรูปของโครงสร้างข่ายงานและตารางความน่าจะเป็นมีเงื่อนไข ดังจะกล่าวต่อไป

ก่อนอื่นเรานิยามความไม่ขึ้นต่อกันอย่างมีเงื่อนไขดังนี้

ความไม่ขึ้นต่อกันอย่างมีเงื่อนไข

นิยามที่ 5.1 ความไม่ขึ้นต่อกันอย่างมีเงื่อนไข

X ไม่ขึ้นกับ Y อย่างมีเงื่อนไขเมื่อรู้ Z ถ้าความน่าจะเป็นของ X ไม่ขึ้นกับค่าของ Y เมื่อรู้ค่าของ Z นั่นคือ

$$(\forall x_i, y_j, z_k)\ P(X=x_i \mid Y=y_j, Z=z_k) = P(X=x_i \mid Z=z_k)$$

หรือในรูปง่าย

$$P(X \mid Y, Z) = P(X \mid Z) \qquad \square$$

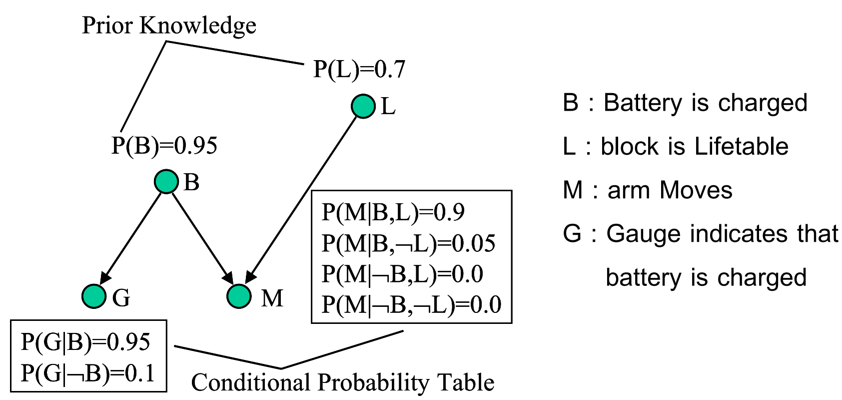
นิยามด้านบนนี้หมายความว่าสำหรับ x_i, y_j, z_k ใดๆ ความน่าจะเป็นที่ X จะมีค่าเป็น x_i (X เป็นตัวแปรส่วน x_i คือค่าของมัน) เมื่อรู้ว่า Y มีค่าเป็น y_j และ $\boldsymbol{Z}$ มีค่าเป็น z_k จะมีค่าเท่ากับความน่าจะเป็นของ X จะมีค่าเป็น x_i เมื่อรู้ว่า $\boldsymbol{Z}$ มีค่าเป็น z_k ในกรณีที่ความน่าจะเป็นทั้งสองเท่ากันเช่นนี้ เราเรียกว่าค่าของ X ไม่ขึ้นกับค่าของ Y อย่างมีเงื่อนไขเมื่อรู้ค่าของ $\boldsymbol{Z}$ เราจึงสามารถตัด Y ทิ้งไปได้

ตัวอย่างเช่นฟ้าร้องไม่ขึ้นกับฝนตกถ้ารู้ว่าฟ้าแลบ หรือเขียนได้เป็น

P (Thunder | Rain, Lighting) = P (Thunder | Lightning)

ดังนั้นถ้ามีฟ้าแลบสามารถบอกได้เลยว่าจะต้องได้ยินเสียงฟ้าร้องด้วยความน่าจะเป็นเท่าไรโดยไม่ต้องสนใจว่าเกิดฝนตกหรือไม่

จากความไม่ขึ้นต่อกันอย่างมีเงื่อนไขข้างต้น เราสร้างข่ายงานของเบส์ได้ดังตัวอย่างใน รูปที่ 6–46 ต่อไปนี้



รูปที่ 6–46 ตัวอย่างของข่ายงานเบส์

ตารางความน่าจะเป็นมีเงื่อนไข - ซีพีที

จากรูปจะเห็นได้ว่าข่ายงานประกอบด้วยบัพหลายบัพ บัพแต่ละบัพหมายถึงคุณสมบัติของข้อมูลหรือตัวแปร และบัพแต่ละบัพจะมี*ตารางความน่าจะเป็นมีเงื่อนไข – ซีพีที (conditional probability table – CPT)* ติดอยู่ด้วย ข่ายงานเบส์นี้แสดงในรูปของกราฟมีทิศทางซึ่งสามารถบอกได้ว่ามีตัวแปรใดบ้างที่ขึ้นกับตัวแปรอื่น และตัวแปรตัวใดบ้างที่ไม่ขึ้นกับตัวอื่น ตัวอย่างเช่นบัพ M ขึ้นกับบัพ B และบัพ L หรือถ้ามองเป็นลักษณะความสัมพันธ์ของบัพพ่อแม่กับบัพลูกจะเห็นว่าบัพพ่อแม่ของ M คือ B และ L ส่วนบัพพ่อแม่ของ G คือ B และสามารถบอกต่อได้ว่าบัพ G จะไม่ขึ้นกับบัพ L ถ้ารู้ B และได้ว่า G ไม่ขึ้นกับ M เมื่อรู้ B (สมมติว่าตัวแปรทั้งสี่คือ G, M, B และ L เป็นตัวแปรแบบบูล และเขียนแทนค่าของตัวแปรอย่างง่ายโดยใช้ตัวแปรนั้นแทนค่าจริงและใส่เครื่องหมาย ¬ แทนค่าเท็จ เช่น G แทนค่าตัวแปร G เป็นจริง ส่วน ¬G แทนค่าตัวแปร G เป็นเท็จ)

บัพใดๆ จะไม่ขึ้นกับบัพอื่นถ้ารู้บัพพ่อแม่โดยตรงของมัน จึงได้ว่า G จะไม่ขึ้นกับบัพอื่นถ้ารู้บัพ B ส่วน L ไม่มีบัพพ่อแม่ แสดงว่า L ไม่ขึ้นกับบัพอื่นๆ เช่นเดียวกับบัพ B ก็ไม่ขึ้นกับบัพอื่น ส่วน M ขึ้นกับ B และ L

จากข่ายงานเบส์ข้างต้น สมมติว่าเรากำลังจะเขียนข่ายงานที่อธิบายหุ่นยนต์ตัวหนึ่งที่กำลังจะย้ายของในโดเมนโลกของบล็อก หุ่นยนต์ตัวนี้จะชาร์จแบตเตอรีและมีเกจ (G) คอยวัดว่าขณะนี้แบตเตอรีเหลืออยู่หรือไม่ หุ่นยนต์ทำงานด้วยการเคลื่อนแขนไปยกบล็อก เมื่อเราจำลองเหตุการณ์นี้ในข่ายงานเบส์จะได้ว่าแบตเตอรี (B) จะส่งผลต่อเกจ G นอกจากนั้นยังส่งผลต่อ M (การเคลื่อนแขนของหุ่นยนต์) และเราได้ใส่ความรู้ก่อนหน้าเข้าไปในรูปของ

ตารางความน่าจะเป็นมีเงื่อนไขว่า 70% ของบล็อกทั้งหมดสามารถยกได้ (P(L)=0.7) และในเวลา 100 ชั่วโมงมี 95 ชั่วโมงที่แบตเตอรีมีไฟ (P(B)=0.95)

เมื่อดูที่ซีพีทีของ G พบว่า ถ้าแบตเตอรีมีไฟ เกจซึ่งมีความบกพร่องอยู่บ้างนี้จะแสดงผลว่ามีไฟด้วยความน่าจะเป็นเท่ากับ 0.95 (P(G|B)=0.95) และถ้าไฟหมดแต่เกจยังแสดงว่ามีไฟด้วยความน่าจะเป็นเท่ากับ 0.1 (P(G|¬B)=0.1)

ในตารางซีพีทีของบัพ M นั้น ตัวแรก P(M|B,L)=0.9 หมายความว่าหุ่นยนต์จะเคลื่อนแขนถ้าแบตเตอรีมีไฟและบล็อกสามารถยกได้ และถ้ามีไฟแต่บล็อกไม่สามารถยกได้แขนจะเคลื่อนด้วยความน่าจะเป็น 0.05 (P(M|B,¬L)=0.05) ถ้าไม่มีไฟและบล็อกสามารถยกได้หุ่นยนต์ก็จะไม่เคลื่อนแขน (P(M|¬B, L)=0.0) และสุดท้ายถ้าบล็อกยกไม่ได้และไฟไม่มีแขนก็จะไม่เคลื่อนเช่นกัน (P(M|¬B, ¬L)=0.0)

ทั้งหมดนี้คือความน่าจะเป็นทั้งหมดที่เราป้อนให้ระบบในรูปของซีพีที ผู้ที่ป้อนข้อมูลคือผู้เชี่ยวชาญที่ทำงานเกี่ยวกับหุ่นยนต์ เมื่อเราทราบค่าต่างๆ ทั้งหมดเราก็สามารถที่จะคำนวณความน่าจะเป็นต่างๆ ที่จะเกิดขึ้นภายในระบบนี้ได้เช่น ถ้าต้องการคำนวณหาว่าความน่าจะเป็นที่ แบตเตอรีมีไฟ บล็อกสามารถยกได้ เกจขึ้นและหุ่นยนต์เคลื่อนแขน ทั้งสี่เหตุการณ์เกิดขึ้นพร้อมกันว่ามีค่าเท่าไรก็สามารถคำนวณได้จากข่ายงานเบส์นี้

ความน่าจะเป็นร่วม (Joint probability) ระหว่างตัวแปรคือความน่าจะเป็นที่ตัวแปรหลายตัวจะมีค่าตามที่กำหนด เช่น P(Battery, Liftable, Gauge, Move) เป็นต้น เราเขียนความน่าจะเป็นร่วมให้อยู่ในรูปทั่วไปได้เป็น

$$P(y_1,\cdots,y_n)=\prod_{i=1}^{n}P(y_i\mid Parents(Y_i)) \tag{6.28}$$

โดยที่ Parents (Y_i) หมายถึง บัพพ่อแม่โดยตรงของบัพ Y_i ถ้าเราต้องการจะหาความน่าจะเป็นที่ $y_1,\ldots,y_n$ เกิดขึ้นพร้อมกันสามารถคำนวณได้จากความน่าจะเป็นของ y_1 คูณกับความน่าจะเป็นของ y_2 คูณไปเรื่อยๆ จนถึง y_n แต่ต้องดูว่าบัพแต่ละบัพขึ้นกับบัพพ่อแม่ตัวใดบ้าง เช่น y_1ขึ้นกับบัพใด y_2 ขึ้นกับบัพใด เป็นต้น ยกตัวอย่างเช่นจากรูปที่ 6–46

$$\begin{aligned}P(G,M,B,L) &= P(G|B,M,L)P(M|B,L)P(B|L)P(L)\\ &= P(G|B)P(M|B,L)P(B)P(L)\\ &= (0.95)(0.9)(0.95)(0.7)\\ &= 0.57\end{aligned}$$

สังเกตได้ว่าบรรทัดแรกใช้กฎลูกโซ่กระจาย P(G,M,B,L) ออกมาเป็นด้านขวามือ และเมื่อกระจายแล้วจะเห็นว่าตัวแปรบางตัวไม่ขึ้นกับตัวอื่น เช่นเราสังเกตได้ว่า G จะขึ้นกับ B ตัวเดียวไม่ขึ้นกับ M หรือ L ดังนั้น P (G|B,M,L) จึงลดรูปลงมาเหลือเป็น P(G|B) เท่านั้น และ B ไม่ขึ้นกับ L ดังนั้น P(B|L) จึงเหลือแค่ P(B) พอลดรูปครบทุกตัวก็นำค่ามาใส่ไว้ในสมการแล้วหาผลลัพธ์ออกมา จะสังเกตได้ว่าเมื่อลดรูปลงมาแล้วบัพที่เราสนใจจะขึ้นกับพ่อแม่ของมันเท่านั้นเช่น P(G|B,M,L) ก็จะเหลือ P(G|B) หรือหาความน่าจะเป็นของ G เมื่อรู้ B กรณีตัวอย่างที่ยกมาเป็นกรณีง่ายๆ เพราะเรารู้ค่าความน่าจะเป็นครบทั้งสี่ตัวแล้ว แต่ในบางกรณีเช่นเราทราบค่าของตัวแปรเพียงแค่ 2 ตัว หรือ 3 ตัว ไม่ใช่ทั้งหมดก็สามารถใช้เทคนิคในการอนุมานของข่ายงานเบส์เพื่อหาความน่าจะเป็นร่วมได้เช่นกัน ดังจะได้อธิบายด้านล่างนี้ ซึ่งเป็นเทคนิคการอนุมานที่ใช้ทั่วไปสำหรับข่ายงานเบส์ 3 เทคนิคเพื่อหาความน่าจะเป็นของตัวแปรที่เราสนใจ

1. *การอนุมานจากเหตุ (causal reasoning):* เมื่อเราทราบเหตุ เราสามารถหาได้ว่าผลจะเกิดขึ้นด้วยความน่าจะเป็นเท่าไร เช่น P(M|L) หรือความน่าจะเป็นที่แขนจะเคลื่อนเมื่อรู้ว่าบล็อกสามารถยกได้ (บล็อกยกได้เป็นสาเหตุหนึ่งของการที่หุ่นยนต์จะเคลื่อนแขน) แต่เราไม่ทราบค่า B (ไม่ทราบว่าขณะนี้แบตเตอรีมีไฟหรือไม่) ถ้าเราย้อนกลับไปดูในข่ายงานเบส์ในรูปที่ 6–46 จะเห็นว่าเราไม่สามารถคำนวณ P(M|L) ได้โดยตรง เพราะไม่มีค่าบอกไว้ในตาราง ในตารางมีค่าที่ใกล้เคียงที่สุดคือ P(M|B,L) ดังนั้นเราต้องพยายามกระจาย P(M|L) ให้อยู่ในรูปที่เกี่ยวข้อง ในที่นี้จะใช้ทฤษฎีความน่าจะเป็นทั้งหมดที่กล่าวว่า ถ้าเหตุการณ์ $A_1,\ldots, A_n$ ไม่เกิดร่วมกันและ $\sum P(A_i)=1$ แล้ว $P(B) = \sum P(B|A_i)P(A_i) = \sum P(B,A_i)$ เราสามารถนำ $A_1,\ldots,A_n$ กระจายเข้ามาได้ ดังนั้นเราสามารถกระจาย P(M|L) ให้อยู่ในรูปของผลรวมของความน่าจะเป็นร่วมระหว่าง M กับบัพพ่อแม่อื่นนอกจาก L (ซึ่งก็คือ B ที่เป็นบัพพ่อแม่ของ M ด้วย) ได้เป็น

$$P(M|L) = P(M, B|L) + P(M, \neg B|L)$$

เมื่อกระจายแล้วก็ยังพบว่าเรายังไม่ทราบค่าของ P(M,B|L) อยู่ สิ่งที่เรารู้คือ P(M|B,L) เราจึงต้องใช้กฎลูกโซ่กระจายแต่ละตัวได้เป็น

$$\begin{aligned} P(M|L) &= P(M|B, L)P(B|L) + P(M|\neg B, L)P(\neg B|L) \\ &= P(M|B, L)P(B) + P(M|\neg B,L)P(\neg B) \\ &= (0.9)(0.95) + (0.0)(0.05) \\ &= 0.855 \end{aligned}$$

2. *การอนุมานจากผล (diagnosis reasoning):* ข้อนี้จะตรงข้ามกับข้อแรก กล่าวคือเราทราบผลแล้วแต่อยากทราบว่าสาเหตุจะเกิดขึ้นด้วยความน่าจะเป็นเท่าไร เช่นต้องการคำนวณ P(¬L|¬M) หรือความน่าจะเป็นที่บล็อกยกไม่ได้เมื่อรู้ว่าแขนไม่ได้เคลื่อนและหาไม่ได้โดยตรง ในกรณีนี้เราใช้ทฤษฎีของเบส์ดังที่กล่าวในตอนแรกว่า $P(A|B)=\dfrac{P(B|A)P(A)}{P(B)}$ ดังนั้น $P(\neg L|\neg M)=\dfrac{P(\neg M|\neg L)P(\neg L)}{P(\neg M)}$

ในส่วนของ P(¬M|¬L) สามารถคำนวณได้โดยใช้การอนุมานจากเหตุในข้อที่แล้ว (ลองคำนวณดู) จะได้ P(¬M|¬L) = 0.9525 ส่วน P(¬L)=0.3 ดังนั้นจะได้ว่า $P(\neg L|\neg M)=\dfrac{0.9525\times 0.3}{P(\neg M)}$ และพบยังมี P(¬M) ที่ยังไม่ทราบค่าอีก ซึ่งการหาค่าโดยตรงค่อนข้างยุ่ง เราจึงไปหาค่า P(L|¬M) เนื่องจาก P(¬L|¬M)+ P(L|¬M) = 1

ดังนั้นเราจะได้ว่า $P(L|\neg M)=\dfrac{P(\neg M|L)P(L)}{P(\neg M)}=\dfrac{0.145\times 0.7}{P(\neg M)}=\dfrac{0.1015}{P(\neg M)}$

จาก P(¬L|¬M) + P(L|¬M) =1 ซึ่งจะทำให้เราหาค่าของ P(¬M) ได้แล้วก็นำไปแทนค่าได้ P(¬L|¬M) = 0.88632

3. *การอธิบายลดความเป็นไปได้ (explaining away):* เป็นการทำการอนุมานจากเหตุภายในการอนุมานจากผล เป็นการผสมระหว่างวิธีการทั้งสองแบบข้างต้น เช่นถ้าเราทราบ ¬M (แขนไม่เคลื่อน) เราสามารถคำนวณ ¬L หรือความน่าจะเป็นที่บล็อกไม่สามารถยกได้ แต่ถ้าเรารู้ ¬B แล้ว ¬L ควรจะมีค่าความน่าจะเป็นน้อยลง ในกรณีนี้เรียกว่า ¬B อธิบาย ¬M ทำให้ ¬L มีความเป็นไปได้น้อยลง

$$
\begin{aligned}
P(\neg L|\neg B,\neg M) &= \frac{P(\neg B,\neg M|\neg L)P(\neg L)}{P(\neg B,\neg M)} \\
&= \frac{P(\neg M|\neg B,\neg L)P(\neg B|\neg L)P(\neg L)}{P(\neg B,\neg M)} \\
&= \frac{P(\neg M|\neg B,\neg L)P(\neg B)P(\neg L)}{P(\neg B,\neg M)}
\end{aligned}
$$

หลังคำนวณ P(¬B,¬M) เราจะได้ P(¬L|¬B,¬M) = 0.03

6.8.6 การเรียนรู้ข่ายงานเบส์

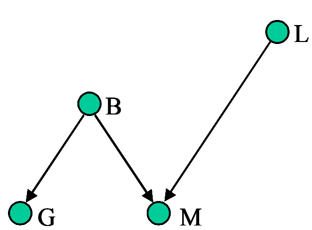
การเรียนรู้ข่ายงานเบส์คือการหาโครงสร้างข่ายงานและ/หรือซีพีทีที่สอดคล้องกับตัวอย่างสอนมากที่สุด ปัญหาการเรียนรู้ข่ายงานเบส์แบ่งออกเป็นกรณีดังต่อไปนี้

1. โครงสร้างไม่รู้ (structure unknown)
2. โครงสร้างรู้ (structure known)
   - 2.1 ข้อมูลมีค่าครบ (no missing value data)
   - 2.2 ข้อมูลมีค่าหาย (missing value data)

กรณีที่ 1 เป็นกรณียากที่สุด เพราะเราไม่รู้ว่าโครงสร้างของข่ายงานเบส์มีรูปร่างเป็นอย่างไร มีการเชื่อมต่อระหว่างบัพอย่างไร และแน่นอนว่าเราไม่รู้ค่าในซีพีทีอีกด้วย ดังนั้นการเรียนรู้ต้องคำนวณหาทั้งโครงสร้างข่ายงานและซีพีที ส่วนกรณีที่สองเป็นกรณีที่รู้โครงสร้างแล้ว ซึ่งบ่อยครั้งผู้เขียนข่ายงานเบส์เป็นผู้เชี่ยวชาญในปัญหานั้นสามารถบอกโครงสร้างได้อย่างชัดเจน รู้ความสัมพันธ์ระหว่างตัวแปรในปัญหานั้นแต่อาจไม่รู้ค่าที่ถูกต้องและแม่นยำในตารางซีพีที ดังนั้นกรณีนี้การเรียนรู้เป็นการหาค่าในซีพีทีโดยอาศัยตัวอย่างสอน กรณีที่สองนี้ยังแบ่งเป็นกรณีย่อยอีกสองกรณีคือ กรณีที่ข้อมูลหรือตัวอย่างสอนทุกตัวมีค่าครบถ้วน กับอีกกรณีที่ตัวอย่างสอนบางตัวหรือทุกตัวมีค่าบางส่วนหายไป เช่นไม่มีค่าของคุณสมบัติบางตัว เป็นต้น กรณีที่ 2.1 เป็นกรณีที่ง่ายสุดสามารถทำการเรียนรู้ได้ในลักษณะเดียวกับการเรียนรู้ของตัวจำแนกประเภทเบส์อย่างง่าย โดยนับจำนวนครั้งที่เกิดขึ้นของข้อมูลเพื่อไปคำนวณซีพีทีของแต่ละบัพว่ามีค่าเท่าไรดังจะแสดงต่อไปนี้ ส่วนกรณีที่ 1 ไม่ขออธิบายในที่นี้

การเรียนรู้ข่ายงานเบส์ในกรณีที่รู้โครงสร้างและข้อมูลครบ

ดูตัวอย่างต่อไปนี้ เรานับความถี่ของการเกิดค่าต่างๆ ของ G, M, B, L ว่าเกิดขึ้นกี่ครั้งได้ดังรูปที่ 6–47 โดยที่สมมติว่าโครงสร้างถูกกำหนดแล้วดังรูปที่ 6–47



G	M	B	L	จำนวนตัวอย่าง
True	True	True	True	54
True	True	True	False	1
True	False	True	True	7
True	False	True	False	27
False	True	True	True	3
False	False	True	False	2
False	False	False	True	4
False	False	False	False	2
				100

รูปที่ 6–47 ตัวอย่างสอนสำหรับเรียนรู้ซีพีทีในกรณีข้อมูลครบ

จาก $P(Vi=vi|Parents(Vi)=\mathbf{Pi}) = \dfrac{\text{จำนวนตัวอย่างที่มี } Vi = vi}{\text{จำนวนตัวอย่างที่มี } Parents(Vi)=\mathbf{Pi}}$

ดังนั้นจะได้ค่าความน่าจะเป็นต่างๆ ดังนี้

P(B=true) = (54+1+7+27+3+2)/100 = 0.94

คือนับจำนวนตัวอย่างที่ B เป็นจริงในตารางหารด้วยจำนวนตัวอย่างทั้งหมด ค่าความน่าจะเป็นอื่นๆ ก็คำนวณในทำนองเดียวกัน

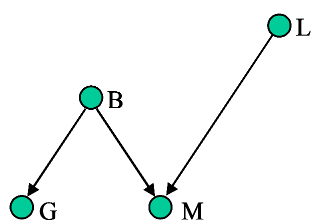
P(L=true) = (54+7+3+4)/100 = 0.68

P(M|B,L) เท่ากับอัตราส่วนที่ M=true เมื่อ B=true, L=false เท่ากับ 1/(1+27+2) =0.03

ด้วยวิธีนี้เราสามารถนำไปคำนวณหาความน่าจะเป็นของบัพ G ได้เช่นเดียวกัน

การเรียนรู้ข่ายงานเบส์ในกรณีที่โครงสร้างรู้และข้อมูลมีค่าหาย

กรณีต่อไปที่จะพิจารณาก็คือกรณีที่ข้อมูลบางตัวมีค่าบางค่าหายไปดังแสดงในรูปที่ 6–48 โดยที่สมมติว่ารู้โครงสร้างของข่ายงานเบส์แล้ว



G	M	B	L	จำนวนตัวอย่าง
True	True	True	True	54
True	True	True	False	1
*	*	True	True	7
True	False	True	False	27
False	True	*	True	3
False	False	True	False	2
False	False	False	True	4
False	False	False	False	2
				100

รูปที่ 6–48 ตัวอย่างสอนสำหรับเรียนรู้ซีพีทีในกรณีข้อมูลมีค่าหาย

'*' ในตารางหมายถึงค่าหายไป พิจารณาแถวที่ห้าของข้อมูลในรูปซึ่งเป็นกรณีของตัวอย่าง 3 ตัวที่มีค่า G=false, M=true, L=true ในกรณีนี้เราไม่รู้ค่าของ B แต่อาจคำนวณ P(B|¬G,M,L) หรือ P(¬B|¬G,M,L) ได้ถ้าหากเรารู้ซีพีที (แต่เรายังไม่รู้) สมมติว่าเรารู้ซีพีทีซึ่งจะทำให้เราหาความน่าจะเป็นที่ B จะเป็นจริง (หรือเท็จ) ของตัวอย่างทั้ง 3 ตัวได้ จากนั้นเราจะแทนที่ตัวอย่างทั้งสามนี้ด้วยตัวอย่างมีน้ำหนัก (weighted example) 2 ตัว ดังนี้

- ตัวแรกคือตัวอย่างที่ B=true มีน้ำหนักเท่ากับ P(B|¬G,M,L)
- ตัวที่สองคือตัวอย่างที่ B=false มีน้ำหนักเท่ากับ P(¬B|¬G,M,L)

ในทำนองเดียวกัน กรณีของตัวอย่าง 7 ตัวในแถวที่สองที่มีค่า B=true, L=true ส่วน G และ M ไม่รู้ค่านั้น เราสามารถแทนที่ตัวอย่างทั้งเจ็ดตัวด้วยตัวอย่างมีน้ำหนัก 4 ตัว ดังนี้

- ตัวอย่างที่ 1 คือตัวอย่างที่ G=true, M=true มีน้ำหนักเท่ากับ P(G,M|B,L)
- ตัวอย่างที่ 2 คือตัวอย่างที่ G=true, M=false มีน้ำหนักเท่ากับ P(G,¬M|B,L)
- ตัวอย่างที่ 3 คือตัวอย่างที่ G=false, M=true มีน้ำหนักเท่ากับ P(¬G,M|B,L)
- ตัวอย่างที่ 4 คือตัวอย่างที่ G=false, M=false มีน้ำหนักเท่ากับ P(¬G,¬M|B,L)

ดังที่ได้กล่าวข้างต้นว่าเราสามารถหาค่าน้ำหนักทั้งสองค่าของตัวอย่าง 3 ตัวด้านบนกับค่าน้ำหนักทั้งสี่ค่าของตัวอย่าง 7 ตัวนี้ได้ถ้าเรารู้ค่าความน่าจะเป็นในซีพีที จากนั้นเราจะใช้ตัวอย่างมีน้ำหนักเหล่านี้ร่วมกับตัวอย่างที่เหลือในรูปที่ 6–48 เพื่อคำนวณซีพีทีซึ่งเป็นสิ่งที่เราต้องการเรียน (ตัวอย่างที่ไม่รู้ค่าถูกแทนที่ด้วยตัวอย่างมีน้ำหนัก) แต่อย่างไรก็ดีเราจะทำเช่นนี้ได้โดยมีเงื่อนไขว่าเราต้องรู้ค่าในซีพีทีก่อน ซึ่งเรายังไม่รู้

วิธีการทำก็คือเราจะสมมติค่าความน่าจะเป็นในซีพีทีโดยสุ่มค่าเริ่มต้นเข้าไปในซีพีที ซึ่งก็จะเสมือนว่าเรามีค่าในซีพีทีแล้ว และเราจะสามารถหาน้ำหนักของตัวอย่างไม่ทราบค่าได้ทุกตัว ก็จะทำให้เซตตัวอย่างเดิมที่มีตัวอย่างไม่รู้ค่าเป็นเซตตัวอย่างที่เรารู้ค่าทุกตัว การเรียนรู้ก็จะเหมือนกับกรณีที่ตัวอย่างมีข้อมูลครบ แน่นอนว่าการคำนวณค่าน้ำหนักจะไม่ได้ค่าน้ำหนักที่ถูกต้องเพราะว่าเราสุ่มซีพีทีเริ่มต้นที่ไม่ใช่ซีพีทีที่ถูก แต่เนื่องจากว่าเมื่อเราได้น้ำหนักแล้วนำตัวอย่างไปรวมกับตัวอย่างที่เหลือที่เป็นตัวอย่างมีข้อมูลครบ ก็จะทำให้การประมาณค่าซีพีทีครั้งใหม่มีความถูกต้องเพิ่มขึ้นกว่าซีพีทีเริ่มต้น เพราะว่าตัวอย่างส่วนใหญ่ของเราเป็นตัวอย่างที่ถูกต้อง จะมีตัวอย่างมีน้ำหนักเท่านั้นที่ไม่ถูกต้องสมบูรณ์ แสดงว่าการปรับค่าซีพีทีทำให้ได้ซีพีทีใหม่ที่ดีขึ้น และถ้าเราทำซ้ำกระบวนการเดิมด้วยซีพีทีที่ดีขึ้นก็จะทำให้การหาค่าน้ำหนักมีความแม่นยำยิ่งขึ้น และส่งผลให้การปรับซีพีทีในรอบต่อไปดีขึ้นอีก เมื่อวนซ้ำไปเรื่อย ๆ ก็จะได้ซีพีทีที่ดีขึ้นเรื่อย ๆ จนกระทั่งซีพีทีไม่เปลี่ยนแปลง เราก็หยุดกระบวนการเรียนรู้ได้ อัลกอริทึมการเรียนรู้แบบนี้เรียกว่า *อัลกอริทึมอีเอ็ม (EM – expectation maximization algorithm)* [Dempster, et al., 1977; McLachlan & Krishman, 1996] ซึ่งแสดงในตารางที่ 6–27 ต่อไปนี้

ตารางที่ 6–27 อัลกอริทึมอีเอ็มสำหรับคำนวณค่าน้ำหนักของตัวอย่างไม่รู้ค่า

```
Algorithm: EM
1. Initialize all entries in all CPTs to some random
   values.
2. UNTIL the termination condition is met DO
   2.1 Use the CPTs to calculate weights of the weighted
       examples.
   2.2 Use the calculated weighted to estimate new CPTs.
```

อัลกอริทึมอีเอ็มนี้โดยทั่วไปจะใช้เวลาในการลู่เข้าไม่มาก ดังจะได้แสดงในตัวอย่างการเรียนรู้ซีพีทีของตัวอย่างสอนในรูปที่ 6–48 ดังนี้

(1) สุ่มค่าสำหรับตารางซีพีที

- $P(L) = 0.5$ $(P(\neg L) = 1 - P(L))$
- $P(B) = 0.5$ $(P(\neg B) = 1 - P(B))$
- $P(M|B,L) = 0.5$ $(P(\neg M|B,L) = 1 - P(M|B,L))$
  $P(M|B,\neg L) = 0.5$ $(P(\neg M|B,\neg L) = 1 - P(M|B,\neg L))$
  $P(M|\neg B,L) = 0.5$ $(P(\neg M|\neg B,L) = 1 - P(M|\neg B,L))$
  $P(M|\neg B,\neg L) = 0.5$ $(P(\neg M|\neg B,\neg L) = 1 - P(M|\neg B,\neg L))$
- $P(G|B) = 0.5$ $(P(\neg G|B) = 1 - P(G|B))$
  $P(G|\neg B) = 0.5$ $(P(\neg G|\neg B) = 1 - P(G|\neg B))$

(2) ใช้ซีพีทีที่สุ่มมาได้ในการหาน้ำหนักของตัวอย่างไม่รู้ค่า

ตัวอย่างไม่รู้ค่าคือ

G	M	B	L	จำนวนตัวอย่าง
*	*	True	True	7
False	True	*	True	3

ในกรณีของ 7 ตัวอย่างแรกเราต้องการหา $P(G,M|B,L)$, $P(G,\neg M|B,L)$, $P(\neg G,M|B,L)$ และ $P(\neg G,\neg M|B,L)$

- $P(G,M|B,L) = P(G|B)\times P(M|B,L) = 0.5\times 0.5$
- $P(G,\neg M|B,L) = P(G|B)\times P(\neg M|B,L) = 0.5\times 0.5$
- $P(\neg G,M|B,L) = P(\neg G|B)\times P(M|B,L) = 0.5\times 0.5$
- $P(\neg G,\neg M|B,L) = P(\neg G|B)\times P(\neg M|B,L) = 0.5\times 0.5$

ดังนั้นสำหรับตัวอย่าง 7 ตัวแรก เราสามารถใส่น้ำหนักให้เป็นตัวอย่างมีน้ำหนักดังต่อไปนี้

G	M	B	L	จำนวนตัวอย่าง
True	True	True	True	7×0.5×0.5=1.75
True	False	True	True	7×0.5×0.5=1.75
False	True	True	True	7×0.5×0.5=1.75
False	False	True	True	7×0.5×0.5=1.75

ในกรณีของ 3 ตัวอย่าง

G	M	B	L	จำนวนตัวอย่าง
False	True	*	True	3

เราต้องหา P(B|¬G,M,L) และ P(¬B|¬G,M,L) ซึ่งทำได้ดังนี้

- $P(B|\neg G,M,L) = \dfrac{P(B,\neg G,M,L)}{P(\neg G,M,L)}$

$$= \frac{P(\neg G|B,M,L)P(M|B,L)P(B|L)P(L)}{P(\neg G,M,L,B) + P(\neg G,M,L,\neg B)}$$

$$= \frac{P(\neg G|B,M,L)P(M|B,L)P(B|L)P(L)}{P(\neg G|B,M,L)P(M|B,L)P(B|L)P(L) + P(\neg G|\neg B,M,L)P(M|\neg B,L)P(\neg B|L)P(L)}$$

$$= \frac{P(\neg G|B)P(M|B,L)P(B)}{P(\neg G|B)P(M|B,L)P(B) + P(\neg G|\neg B)P(M|\neg B,L)P(\neg B)}$$

ดังนั้น $P(B|\neg G,M,L) = \dfrac{0.5\times0.5\times0.5}{0.5\times0.5\times0.5 + 0.5\times0.5\times0.5} = 0.5$

$P(\neg B|\neg G,M,L) = 0.5$

ดังนั้นสำหรับตัวอย่าง 3 ตัว เราสามารถใส่น้ำหนักให้เป็นตัวอย่างมีน้ำหนักดังต่อไปนี้

G	M	B	L	จำนวนตัวอย่าง
False	True	True	True	$3\times0.5=1.5$
False	True	False	True	$3\times0.5=1.5$

จะได้ว่าตัวอย่างทั้งหมดเป็นดังนี้

G	M	B	L	จำนวนตัวอย่าง
True	True	True	True	54
True	True	True	False	1
True	True	True	True	1.75
True	False	True	True	1.75
False	True	True	True	1.75
False	False	True	True	1.75
True	False	True	False	27
False	True	True	True	1.5
False	True	False	True	1.5
False	False	True	False	2
False	False	False	True	4
False	False	False	False	2

(3) ใช้ตัวอย่างมีน้ำหนักที่คำนวณได้ เพื่อประมาณซีพีทีใหม่

- P(L) = 68/100 = 0.680 (P(¬L) = 1 – P(L))
- P(B) = 92.5/100 = 0.925 (P(¬B) = 1 – P(B))
- P(M|B,L) = 59/62.5 = 0.944 (P(¬M|B,L) = 1 – P(M|B,L))
  P(M|B,¬L) = 1/30 = 0.033 (P(¬M|B,¬L) = 1 – P(M|B,¬L))
  P(M|¬B,L) = 1.5/5.5 = 0.273 (P(¬M|¬B,L) = 1 – P(M|¬B,L))
  P(M|¬B,¬L) = 0/2 = 0.000 (P(¬M|¬B,¬L) = 1 – P(M|¬B,¬L))
- P(G|B) = 85.5/92.5 = 0.924 (P(¬G|B) = 1 – P(G|B))
  P(G|¬B) = 0/7.5 = 0.000 (P(¬G|¬B) = 1 – P(G|¬B))

(2) ใช้ซีพีทีเพื่อคำนวณน้ำหนักของตัวอย่างไม่รู้ค่าใหม่

ในกรณีของ 7 ตัวอย่างแรก

- $P(G,M|B,L) = P(G|B) \times P(M|B,L) = 0.924 \times 0.944 = 0.872$
- $P(G,\neg M|B,L) = P(G|B) \times P(\neg M|B,L) = 0.924 \times 0.056 = 0.052$
- $P(\neg G,M|B,L) = P(\neg G|B) \times P(M|B,L) = 0.076 \times 0.944 = 0.072$
- $P(\neg G,\neg M|B,L) = P(\neg G|B) \times P(\neg M|B,L) = 0.076 \times 0.056 = 0.004$

ได้ตัวอย่างมีน้ำหนักเป็น

G	M	B	L	จำนวนตัวอย่าง
True	True	True	True	7×0.872=6.11
True	False	True	True	7×0.052=0.36
False	True	True	True	7×0.072=0.50
False	False	True	True	7×0.004=0.03

ในกรณีของ 3 ตัวอย่าง

- $P(B|\neg G,M,L) = \dfrac{0.076 \times 0.944 \times 0.925}{0.076 \times 0.944 \times 0.925 + 1.000 \times 0.273 \times 0.075} = 0.764$
- $P(\neg B|\neg G,M,L) = 1 - P(B|\neg G,M,L) = 0.236$

ได้ตัวอย่างมีน้ำหนักเป็น

G	M	B	L	จำนวนตัวอย่าง
False	True	True	True	3× 0.764=2.29
False	True	False	True	3× 0.236=0.71

เมื่อทำซ้ำขั้นตอน (2), (3) จนครบ 20 รอบซีพีทีลู่เข้าดังนี้ (ค่าความน่าจะเป็นทุกตัวในทุกตารางซีพีทีมีค่าเปลี่ยนแปลงน้อยกว่า 0.001)

- $P(L) = 0.680$
- $P(B) = 0.940$
- $P(M|B,L) = 1.000$

  $P(M|B,\neg L) = 0.033$

  $P(M|\neg B,L) = 0.005$

  $P(M|\neg B,\neg L) = 0.000$
- $P(G|B) = 0.943$
- $P(G|\neg B) = 0.000$

เอกสารอ่านเพิ่มเติมและแบบฝึกหัด

หนังสือของ Mitchell [Mitchell, 1997] ได้ถูกใช้เป็นตำราเรียนของวิชาการเรียนรู้ของเครื่องในมหาวิทยาลัยจำนวนมาก มีคำอธิบายครอบคลุมเทคนิคของการเรียนรู้ของเครื่องไว้ค่อนข้างครบถ้วน แต่ถ้าต้องการศึกษาอย่างละเอียดเฉพาะเทคนิคหนึ่งๆ เช่นถ้าเกี่ยวกับอัลกอริทึมเชิงพันธุกรรมแนะนำให้ดูหนังสือของ Mitchell [Mitchell, 1996] และ Goldberg [Goldberg, 1989] ถ้าเป็นการเรียนรู้ต้นไม้ตัดสินใจให้ดูหนังสือของ Quinlan [Quinlan, 1993] ถ้าเกี่ยวกับข่ายงานประสาทเทียมก็แนะนำให้อ่านหนังสือ [Hassoun, 1995] ส่วนหนังสือของ Pearl [Pearl, 1988] เป็นหนังสือเกี่ยวกับข่ายงานเบส์ที่น่าศึกษาอย่างยิ่ง

บรรณานุกรม

Dejong, G. and Mooney, R. (1986) Explanation-based learning: An alternative view. *Machine Learning*, 1(2), 145-176.

Dempster, A. P., Laird, N. M. and Rubin D. B. (1977) Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, Series B, 39 (1), 1-38.

Goldberg, D. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.Hassoun, M. H. (1995) *Fundamentals of Artificial Neural Networks.* The MIT Press.

Koza, J. (1992) *Genetic Programming: On the Programming of Computrs by Means of Natural Selection.* The MIT Press.

McLachlan, G. J. and Krishman, T. (1996) *The EM Algorithm and Extensions.* Wiley Interscience.

Mitchell, M. (1996) *An Introduction to Genetic Algorithms.* The MIT Press.

Mitchell, T. (1977) Version space: A candidate elimination approach to rule learning. *In Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-77).*

Mitchell, T., Keller, R. and Kedar-Cabelli, S. (1986) Explanation-based generalization: A unifying view, *Machine Learning*, 1(1), 47-80.

Mitchell, T. (1997) *Machine Learning,* McGraw-Hill.

Pearl, J. (1998) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann.

Quinlan, J. R. (1986) Induction of decision trees. *Machine Learning*, 1(1), 81-106.

Quinlan, J. R. (1993) *C4.5: Programs for Machine Learning.* Morgan Kaufmann.

Rumelhart, D. E., and McClelland, J. L. (1986) Parallel Distributed Processing: Exploration in the Microstructure of Cognition (Vol. 1&2) The MIT Press.Winston, P. H. (1992) *Artificial Intelligence.* Thrid Edition. Addison Wesley.

แบบฝึกหัด

1. กำหนดให้ความรู้ในโดเมนและตัวอย่างสอนเป็นดังต่อไปนี้

   ความรู้ในโดเมน:

   a(X,X,Y), b(red,Z) → c(W,X,Y,Z)
   d(Z,Z), d(Y,X), e(X,Y) → a(X,Y,Z)
   f(Y,X) → b(X,Y)
   g(X,X) → d(X,Y)

   ตัวอย่างสอน:

e(eyes,eyes)	e(eyes,ears)
e(eyes,nose)	f(fire,red)
f(tree,green)	f(snow,white)
g(2,2)	g(2,1)
g(3,2)	g(eyes,eyes)
g(eyes,ears)	g(eyes,nose)

   - จงแสดงให้เห็นว่า c(white,eyes,2,fire) เป็นตัวอย่างที่ถูกโดยใช้ต้นไม้พิสูจน์
   - กำหนดให้เกณฑ์ดำเนินการประกอบด้วยเพรดิเคต 3 ตัวคือ e, f และ g จงเขียนกฎที่เรียนได้จากตัวอย่างด้านบน

2. ในการเรียนมโนทัศน์ของ “EnjoySport” เราสังเกตว่าเพื่อนของเราคนหนึ่งจะสนุกกับการเล่นกีฬาทางน้ำหรือไม่ โดยได้พิจารณาถึงปัจจัย 6 อย่างคือ Sky (ท้องฟ้า), AirTemp (อุณหภูมิอากาศ), Humidity (ความชื้น), Wind (ลม), Water (น้ำ), Forecast (คำพยากรณ์) และได้บันทึกตัวอย่างบวก (3 ตัว) และตัวอย่างลบ (1 ตัว) ดังแสดงด้านล่าง

(Sunny, Warm, Normal, Strong, Warm, Same)	+
(Sunny, Warm, High, Strong, Warm, Same)	+
(Rainy, Cold, High, Strong, Warm, Change)	–
(Sunny, Warm, High, Strong, Cool, Change)	+

   หมายเหตุ: ตัวอย่างแต่ละตัวแสดงอยู่ในรูป (x1,x2,x3,x4,x5,x6) โดยที่ x1 เป็นค่าของ Sky, x2 เป็นค่าของ AirTemp, x3 เป็นค่าของ Humidity, x4 เป็นค่าของ Wind, x5 เป็นค่าของ Water, x6 เป็นค่าของ Forcast และเครื่องหมายบวกแสดงตัวอย่างบวก เครื่องหมายลบแสดงตัวอย่างลบ กำหนดภาษาที่ใช้แสดงเป็นดังต่อไปนี้

- ค่าของ Sky ที่เป็นไปได้คือ Sunny, Cloudy, Rainy
- ค่าของ AirTemp ที่เป็นไปได้คือ Warm, Cold
- ค่าของ Humidity ที่เป็นไปได้คือ Normal, High
- ค่าของ Wind ที่เป็นไปได้คือ Strong, Weak
- ค่าของ Water ที่เป็นไปได้คือ Warm, Cool
- ค่าของ Forecast ที่เป็นไปได้คือ Same, Change

จงตอบคำถามต่อไปนี้

- ปริภูมิมโนทัศน์ในกรณีนี้มีขนาดเท่าไร
- จงแสดงเซต S และ G เมื่อรับตัวอย่างเข้าไปทีละตัวตามลำดับ
- เมื่อรับตัวอย่างทั้ง 4 ตัวเข้าไปหมดแล้ว เวอร์ชันสเปซจะประกอบด้วยสมมติฐานที่เป็นไปได้ทั้งหมดกี่ตัว อะไรบ้าง (สมมติฐานทั้งหมดที่อยู่ระหว่าง S และ G (รวม S และ G ด้วย))

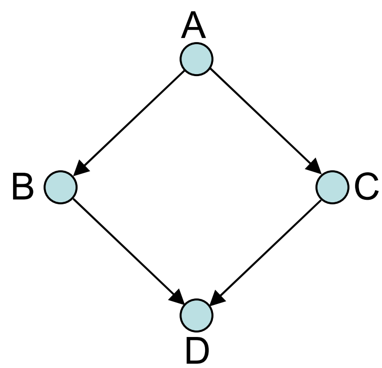
3. ตารางด้านล่างนี้เป็นข้อมูลของผู้ที่มาขอทำบัตรเครดิตจากธนาคารแห่งหนึ่ง ข้อมูลของคนหนึ่งๆ ประกอบด้วย account, employed, cash ธนาคารใช้ข้อมูลเหล่านี้สำหรับกำหนดว่าจะทำบัตรให้หรือไม่ ถ้าทำให้จะมีประเภท (class) เป็น accept ถ้าไม่ทำให้จะมีประเภทเป็น reject จงสร้างต้นไม้ตัดสินใจเพื่อจำแนกประเภทข้อมูลของประเภททั้งสองนี้

Attribute			
account	employed	cash	class
bank	yes	3000	accept
bank	no	3000	accept
bank	no	40000	accept
none	yes	40000	accept
none	yes	3000	reject
none	no	40000	reject
none	no	3000	reject
other	yes	3000	reject
other	no	3000	reject
other	no	40000	accept

4. คณะกรรมการสอบคัดเลือกนิสิตที่สมัครเรียนต่อปริญญาโทของภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ต้องการทราบว่าผู้ที่สอบผ่านจริงๆ แล้วมีคุณสมบัติดีจริงหรือไม่ จึงได้สร้างข่ายงานเบส์ดังรูปต่อไปนี้

$P(A)=1/2$

$P(B|A) = 1$
$P(B|\neg A) = 1/2$



$P(C|A) = 1$
$P(C|\neg A) = 1/2$

$P(D|B,C) = 1$
$P(D|B,\neg C) = 1/2$
$P(D| \neg B,C) = 1/2$
$P(D| \neg B,\neg C) = 0$

A = applicant is qualified.
B = applicant has high grade point average.
C = applicant has excellent recommendations.
D = applicant is admitted.

กำหนดให้ตัวแปร A,B,C,D เป็นตัวแปรแบบบูลคือมีค่าได้ 2 ค่าคือจริงกับเท็จ และการเขียนค่าตัวแปรในข่ายงานเป็นการเขียนแบบย่อ กล่าวคือ

X แทนตัวแปร X มีค่าความจริงเป็นจริง

$\neg$ X แทนตัวแปร X มีค่าความจริงเป็นเท็จ

เช่น $P(D|\neg B,C)$ คือความน่าจะเป็นที่ D มีค่าเป็นจริงเมื่อรู้ว่า B มีค่าเป็นเท็จและ C มีค่าเป็นจริง เป็นต้น

จงแสดงวิธีการคำนวณหาค่า $P(A|D)$ ว่ามีค่าเท่ากับเท่าไร

5. พิจารณาเพอร์เซปตรอนยูนิตเดี่ยวที่มีอินพุต 3 บัพคือ x, y, z และมีเอาต์พุต 1 บัพคือ f ถ้าให้ตัวอย่าง 4 ตัวต่อไปนี้ จงแสดงให้เห็นว่าเพอร์เซปตรอนนี้จะเรียนรู้ได้สำเร็จหรือไม่

อินพุต			เอาต์พุต
x	y	z	f
0	1	0	0
1	0	0	1
1	1	1	0
0	0	1	1