บทที่ 6 การจำแนกประเภทและการทำนายข้อมูล (Classification and Prediction)

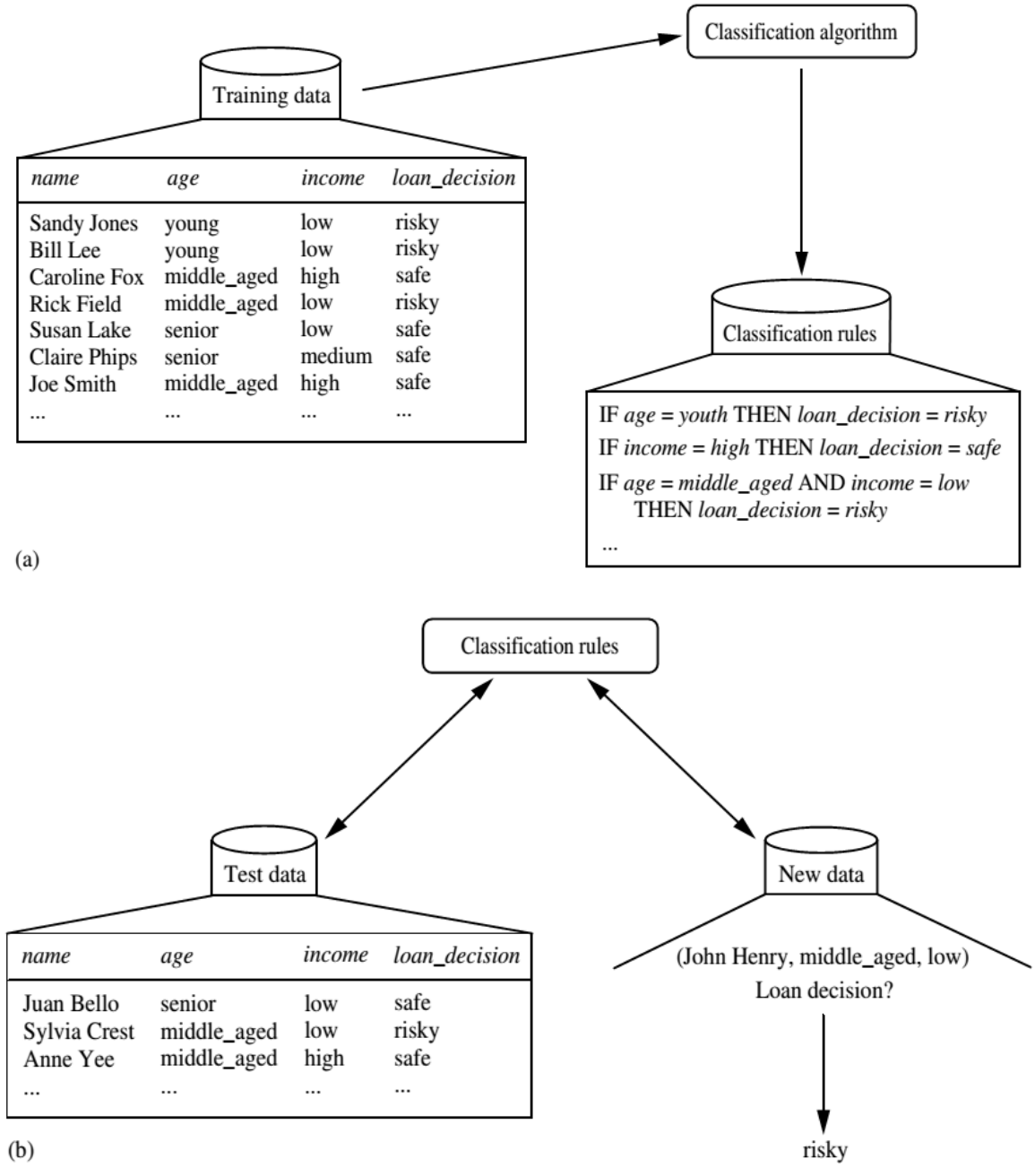
วัตถุประสงค์

ในบทนี้จะทำการศึกษาเกี่ยวกับเทคนิคต่างๆในการจำแนกประเภทของข้อมูล อาทิ เช่น การจำแนกข้อมูลด้วยการสร้างต้นไม้ตัดสินใจ (Decision tree classifier) การจำแนกข้อมูลด้วยเบย์เซียนและเบย์เซียนบีลิฟเน็ทเวิร์ค (Bayesian classifier and Bayesian belief networks) การจำแนกข้อมูลด้วยกฎ (Rule-based classifiers) การจำแนกข้อมูลด้วยโครงข่ายประสาทเทียมและการส่งค่าย้อนกลับ (Neural network and backpropagation) การจำแนกข้อมูลจากกฎความสัมพันธ์ของข้อมูล (Classification based on association rule mining) การค้นหาเพื่อนบ้านใกล้สุด *k* อันดับ (*k*-nearest-neighbor) และทำการศึกษาเกี่ยวกับการทำนายข้อมูล ที่จะประกอบไปด้วยการถดถอยเชิงเส้นตรง (linear regression) และการถดถอยที่ไม่เป็นเส้นตรง (Nonlinear regression) ท้ายสุด—เราจะทำการศึกษาเกี่ยวการวัดความถูกต้องและความผิดพลาดในการจำแนกข้อมูล รวมถึงส่วนต่อเติมหรือวิธีการเพิ่มประสิทธิภาพในแง่มุมต่างๆสำหรับการจำแนกและการทำนายข้อมูล

6.1 นิยามเบื้องต้นของการจำแนกและการทำนายข้อมูล

ในการที่จะเข้าใจว่าการจำแนกข้อมูลคืออะไร ลองพิจารณาตัวอย่างดังต่อไปนี้—1) พนักงานสินเชื่อของธนาคารต้องการที่จะทำการวิเคราะห์ข้อมูลเพื่อที่จะทำการศึกษาว่าการกู้ยืมในครั้งหนึ่งๆ มีครั้งไหนบ้างที่ปลอดภัยและครั้งไหนบ้างที่มีความเสี่ยง 2) ผู้จัดการฝ่ายการตลาดของบริษัทขายอุปกรณ์ไฟฟ้าต้องการข้อมูลเพื่อช่วยในการคาดเดาว่า “ลูกค้ามีคุณลักษณะอย่างไรที่จะทำการซื้อคอมพิวเตอร์จากบริษัท” 3) นักวิจัยทางการแพทย์ต้องการที่จะวิเคราะห์ข้อมูลเกี่ยวกับมะเร็งเต้านมเพื่อที่จะทำการทำนายว่าผู้ป่วยควรจะได้รับการดูแลด้วยวิธีใดภายใต้วิธีการรักษาทั้ง 3 วิธีที่ได้รับความนิยมอย่างแพร่หลาย—จากตัวอย่างทั้ง 3 ข้างต้นจะต้องการการวิเคราะห์ข้อมูล ซึ่งก็คือ *การจำแนกข้อมูล (Classification)* ที่ซึ่งจะทำการสร้างโมเดลหรือตัวจำแนกข้อมูล (Classifier) เพื่อทำนายหมวดหมู่ของข้อมูล (Categories/Class) อาทิเช่น ปลอดภัย หรือ เสี่ยง ในการวิเคราะห์ข้อมูลสินเชื่อ วิธีการรักษา A หรือ B หรือ C ในการวิเคราะห์ข้อมูลของผู้ป่วยมะเร็งเต้านม เป็นต้น ในส่วนของการทำนายข้อมูล ลองพิจารณาตัวอย่างดังต่อไปนี้—สมมติว่าผู้จัดการฝ่ายการตลาดต้องการที่จะทำนายหรือคาดเดาว่าลูกค้าคนหนึ่งๆจะทำการจ่ายเงินซื้อสินค้าจากบริษัทเป็นจำนวนเท่าไหร่ การวิเคราะห์ข้อมูลลักษณะนี้จะเป็นส่วนของการทำนายข้อมูลเชิงตัวเลข (Numeric prediction)

การจำแนกข้อมูลจะประกอบไปด้วยสองกระบวนการหลัก ดังแสดงตัวอย่างในรูปที่ 6.1 ที่เป็นการกู้-ยืมเงิน โดยจากรูปที่ 6.1(a) จะเป็นกระบวน***การสร้างตัวจำแนกข้อมูล***จากชุดของข้อมูลที่เป็นอินพุต ที่ซึ่งแต่ละเรคคอร์ดของข้อมูลที่ทำการพิจารณาจะประกอบไปด้วยเซตของแอทริบิวที่บ่งบอกถึงคุณลักษณะของบุคคลที่ทำการกู้-ยืมเงิน และหมวดหมู่ของบุคคลนั้นๆว่ามีความปลอดภัยหรือมีความเสี่ยงในการให้กู้-ยืมเงินหรือไม่ โดยกระบวนการสร้างตัวจำแนกข้อมูลมักถูกเรียกว่า ‘learning’ หรือ ‘training’ ที่เกิดจากการนำเอาขั้นตอนวิธีสำหรับการจำแนกข้อมูลมาดำเนินการกับข้อมูล ข้อมูลเรคคอร์ด X หนึ่งๆ ในชุดข้อมูลที่ทำการพิจารณา จะ



รูปที่ 6.1 ตัวอย่างกระบวนการในการจำแนกข้อมูล (a) การเรียนรู้จากข้อมูลเพื่อสร้างตัวจำแนกข้อมูล (b) การทดสอบตัวจำแนกข้อมูลเพื่อวัดความถูกต้อง

ประกอบไปด้วยเซตของแอทริบิว $X = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_n)$, n แอทริบิวที่บ่งบอกถึงคุณลักษณะต่างๆของข้อมูลเรค-คอร์ด X นอกจากนั้นเรคคอร์ด X ยังมีข้อมูลอีกหนึ่งแอทริบิวที่บ่งบอกถึงหมวดหมู่ของข้อมูล (class label attribute) โดยแอทริบิวหมวดหมู่ข้อมูลจะเป็นข้อมูลแบบไม่ต่อเนื่อง (discrete-valued) โดยชุดข้อมูลที่เป็นอินพุตสำหรับทำการสร้างตัวจำแนกข้อมูลจะถูกเรียกว่า "***ชุดข้อมูลสำหรับสอน (training data) ตัวอย่าง (samples/instances) ชุดข้อมูล (data points) or สิ่งของ (objects)***" เป็นต้น หมายเหตุ—เนื่องจากแต่ละเรคคอร์ดในชุดข้อมูลที่เป็นอินพุตจะมีแอทริบิวหมวดหมู่ข้อมูลแนบอยู่ด้วย ดังนั้น การจำแนกข้อมูลด้วยข้อมูลลักษณะนี้จะเรียกว่า *การเรียนรู้แบบมีผู้สอน (Supervised learning—คือ การสร้างตัวจำแนกข้อมูลจะถูกสอนโดยแอทริบิวหมวดหมู่ข้อมูลต่างๆที่ถูกแนบอยู่ในแต่ละเรคคอร์ดของชุดข้อมูล* โดยการเรียนรู้แบบมีผู้สอนจะแตกต่างกับการเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning หรือ clustering) ที่จะไม่ทราบถึงหมวดหมู่ของข้อมูล ตัวอย่างเช่น ในการวิเคราะห์ข้อมูลการกู้-ยืมเงินที่ไม่มีหมวดหมู่ข้อมูลที่บ่งบอกว่าการกู้ยืมครั้งหนึ่งๆมีความเสี่ยงหรือไม่ เราจะสามารถวิเคราะห์ข้อมูลได้จากการเชื่อมโยงเรคคอร์ดของการกู้-ยืมเงินที่มีลักษณะใกล้เคียงกันหรือเหมือนกันให้อยู่ในกลุ่มเดียวกัน เป็นต้น

ขั้นตอนที่สองของการจำแนกข้อมูล (ดังแสดงในรูปที่ 6.1(b)) จะเป็นการเรียกใช้ตัวจำแนกข้อมูลที่สร้างขึ้นจากขั้นตอนที่หนึ่งเพื่อทำการจำแนกข้อมูล โดยในตอนเริ่มต้น ตัวจำแนกข้อมูลจะถูกทดสอบและประเมินค่าความถูกต้อง (ถ้าเราใช้ชุดข้อมูลสำหรับสอนในการทดสอบตัวจำแนกข้อมูลจะทำให้ความถูกต้องจะมีค่าค่อนข้างสูง เนื่องจากตัวจำแนกข้อมูลที่สร้างขึ้นจะเหมาะกับข้อมูลชุดนั้นเป็นอย่างมาก (overfit) แต่ถ้าเราใช้ชุดข้อมูลที่แตกต่างออกไปในการทดสอบ (test set) โดยชุดข้อมูลที่ใช้จะต้องมีแอทริบิวหมวดหมู่ข้อมูลแนบอยู่ด้วย จะทำให้เราทราบค่าความถูกต้องของตัวจำแนกข้อมูลได้) โดยค่าความถูกต้องของตัวจำแนกข้อมูลที่ถูกสร้างขึ้นจะเป็นเปอร์เซ็นต์ของตัวจำแนกข้อมูลที่สามารถจำแนกข้อมูลได้อย่างถูกต้อง (ตัวจำแนกข้อมูลบ่งบอกถึงหมวดหมู่ข้อมูลได้เหมือนกับหมวดหมู่ข้อมูลที่ถูกแนบมากับข้อมูลเรคคอร์ดหนึ่งๆ)

เมื่อค่าความถูกต้องของตัวจำแนกข้อมูลมีค่าที่น่าพึงพอใจหรือยอมรับได้ เราจะใช้ตัวจำแนกข้อมูลในการจำแนกหรือบ่งบอกถึงหมวดหมู่ข้อมูลที่เข้ามาใหม่ที่ซึ่งเราไม่ทราบหมวดหมู่ข้อมูลมาก่อน (ข้อมูลที่เข้ามาใหม่จะถูกเรียกว่า 'unknown' หรือ 'previously unseen' data) ตัวอย่างเช่น ตัวจำแนกข้อมูลที่ถูกสร้างขึ้นในรูปที่ 6.1(a) จะถูกใช้เพื่อตัดสินใจการให้กู้-ยืมเงินของเอกสารที่ยื่นเข้ามาใหม่ว่าจะให้กู้-ยืมหรือไม่ เป็นต้น

6.2 สิ่งที่ต้องคำนึงถึงในการจำแนกและการทำนายข้อมูล

ในการจำแนกและทำนายข้อมูลครั้งหนึ่งๆจะมีหลายปัจจัยที่เราต้องพิจารณาและคำนึงถึง แต่อย่างไรก็ตาม โดยทั่วไปเราจะต้องพิจารณา 2 ปัจจัยหลัก ดังนี้

6.2.1 การจัดเตรียมข้อมูลสำหรับการจำแนกและการทำนายข้อมูล

- **การทำความสะอาดข้อมูล (Data cleansing)**—จะเกี่ยวข้องกับการประมวลผลข้อมูลเบื้องต้นที่จะลบหรือลดข้อมูลที่มีสิ่งรบกวน (noise) ด้วยการประยุกต์ใช้วิธีการปรับเรียบข้อมูลแบบต่างๆ และจัดการกับการขาดหายไปของข้อมูลด้วยการแทนค่าของข้อมูลที่ขาดหายไปด้วยค่าของข้อมูลที่ปรากฏบ่อยที่สุดหรือทำการแทนด้วยค่าของข้อมูลที่มีค่าเชิงสถิติสูงที่สุด เป็นต้น
- **ความเกี่ยวเนื่องของข้อมูล (Relevance analysis)**—จะทำการตรวจสอบข้อมูลแอทริบิวต่างๆว่ามีความเกี่ยวเนื่องหรือซ้ำซ้อนกันมากน้อยเพียงใด ซึ่งโดยปกติของชุดข้อมูลจะมีแอทริบิวที่ซ้ำซ้อนกัน ดังนั้น เพื่อที่จะหลีกเลี่ยงความซ้ำซ้อนดังกล่าว เราสามารถประยุกต์ใช้การวิเคราะห์สหสัมพันธ์ (correlation analysis) เพื่อทำการตรวจสอบทางสถิติว่า 2 แอทริบิวใดๆ (เลือกพิจารณาทีละคู่ของแอทริบิวจากแอทริบิวทั้งหมด) มีความเหมือนกันหรือแตกต่างกันมากน้อยเพียงใด ตัวอย่างเช่น แอทริบิว A_1 และ A_2 ที่มีความเหมือนกันค่อนข้างมากจะทำให้เราทราบว่าเราควรที่จะต้องลบแอทริบิวใดแอทริบิวหนึ่งออกจากเซตของแอทริบิวที่จะทำการพิจารณา จากนั้นเราจะต้องทำการเลือกแอทริบิวที่น่าสนใจ (Attribute subset selection) เพื่อลดปริมาณแอทริบิวที่ต้องพิจารณา อันนำมาซึ่งการลดทรัพยากรในการคำนวณและอาจเป็นการเพิ่มความถูกต้องของการค้นหาผลลัพธ์ได้อีกด้วย
- **การเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลและการลดจำนวนข้อมูล (Data transformation and reduction)**—ข้อมูลที่เป็นอินพุตอาจมีช่วงของข้อมูลหรือค่าของข้อมูลที่มีระยะห่างค่อนข้างมาก ดังนั้น เราอาจทำการเปลี่ยนแปลง/เปลี่ยนรูปด้วยวิธีการ normalization ที่จะทำการปรับเปลี่ยนค่าในแอทริบิวหนึ่งๆให้อยู่ในช่วงที่กำหนด อาทิเช่น ช่วง -1.0 ถึง 1.0 หรือ ช่วง 0.0 ถึง 1.0 เป็นต้น การปรับเปลี่ยนช่วงของข้อมูลจะช่วยในเรื่องของการหาความแตกต่างระหว่างข้อมูลเรคคอร์ดต่างๆ (distance measure) นอกจากนั้นการเปลี่ยนแปลง/เปลี่ยนรูปข้อมูลยังสามารถทำได้ด้วยวิธีการ generalization ที่ซึ่งจะทำให้ข้อมูลมีความละเอียดน้อยลง ตัวอย่างเช่น แอทริบิวที่บ่งบอกถึงรายได้ที่เป็นค่าต่อเนื่อง (ค่าเงินจริงที่แต่ละบุคคลได้รับในแต่ละเดือน) จะสามารถถูกทำ generalize ให้อยู่ในช่วงที่ไม่ต่อเนื่องได้ อาทิเช่น low, medium และ high เป็นต้น ยิ่งไปกว่านั้น เรายังสามารถลดจำนวนข้อมูลที่ต้องทำการพิจารณาได้ด้วยการประยุกต์ใช้เทคนิคต่างๆ อาทิเช่น wavelet transformation และ principle component analysis รวมถึงเทคนิคการทำ binning, histogram analysis และ clustering เป็นต้น

6.2.2 การเปรียบเทียบประสิทธิภาพวิธีในการจำแนกและทำนายข้อมูล

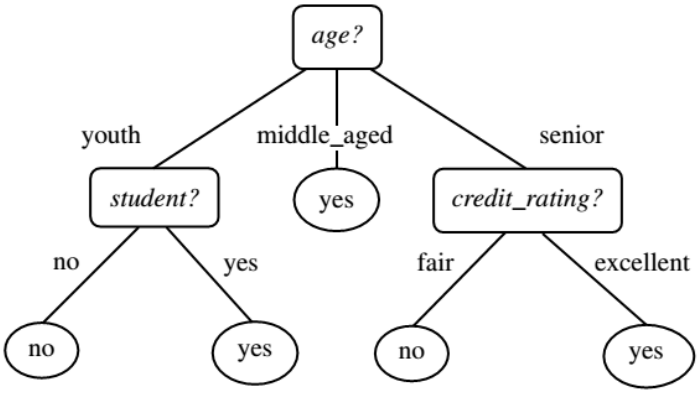
ในการเปรียบเทียบประสิทธิภาพของวิธีในการจำแนกและทำนายข้อมูลเราจะทำการประยุกต์ใช้เกณฑ์ดังต่อไปนี้

- **ความถูกต้อง (Accuracy)**—จะเกี่ยวข้องกับความสามารถของตัวจำแนกข้อมูลที่ถูกสร้างขึ้นที่จะสามารถจำแนกข้อมูลที่ไม่เคยพบเจอมาก่อนได้อย่างถูกต้อง โดยในการวัดความถูกต้องอาจประเมินได้จากการใช้ชุดข้อมูลหนึ่งๆ (หรือมากกว่าหนึ่งชุดก็ได้)ที่ แยกจากชุดข้อมูลเรียนรู้ (training dataset)
- **ความเร็ว (Speed)**—จะเกี่ยวข้องกับเวลาที่ใช้ในการคำนวณทั้งในส่วนของการสร้างตัวจำแนกข้อมูลและการจำแนก/ทำนายข้อมูล
- **ความทนทาน (Robustness)**—-จะเกี่ยวข้องกับความสามารถของตัวจำแนกหรือตัวทำนายข้อมูลที่จะทำการทำนายได้อย่างถูกต้องจากข้อมูลตั้งต้นที่มีสิ่งรบกวนหรือมีการขาดหายไปของข้อมูล
- **ความยืดหยุ่นต่อปริมาณข้อมูล (Scalability)**—จะเกี่ยวข้องกับความสามารถในการสร้างตัวจำแนกข้อมูลหรือตัวทำนายข้อมูลได้อย่างมีประสิทธิภาพเมื่อมีข้อมูลที่ต้องพิจารณาเป็นปริมาณมาก
- **ความสามารถในการเข้าใจ (Interpretability)**—เกี่ยวเนื่องกับระดับความสามารถที่จะถูกเข้าใจในตัวแจกแนกหรือทำนายข้อมูลจากผู้ใช้งาน

6.3 การจำแนกข้อมูลด้วยต้นไม้ตัดสินใจ

การจำแนกข้อมูลด้วยต้นไม้ตัดสินใจจะเป็นกระบวนการสร้างต้นไม้ขึ้นเพื่อใช้ในการตัดสินใจจากข้อมูลที่มีหมวดหมู่ข้อมูลแนบอยู่ด้วย ต้นไม้ตัดสินใจจะประกอบไปด้วย***โหนดต่างๆ (ที่ไม่ใช่โหนดใบ—nonleaf node)*** ที่ซึ่งถูกใช้ในการแสดงถึงเงื่อนไขหรือแอทริบิวหนึ่งๆของข้อมูล โดยที่แต่ละกิ่งก้านของโหนดหนึ่งๆจะหมายถึงค่าที่เป็นไปได้จากการทดสอบกับแอทริบิวนั้นๆ และจะประกอบไปด้วย***โหนดใบ (leaf node)*** ที่ซึ่งจะมีหมวดหมู่ข้อมูลจัดเก็บอยู่ โดยตัวอย่างต้นไม้ตัดสินใจจะถูกแสดงในรูปที่ 6.2 ที่จะแสดงการทำนายคุณลักษณะของลูกค้าที่จะทำการซื้อคอมพิวเตอร์จากร้านขายอุปกรณ์ไฟฟ้า โดยโหนดต่างๆที่ไม่ใช่โหนดใบจะถูกแทนด้วยสี่เหลี่ยม และโหนดใบจะถูกแทนด้วยวงรี ตามลำดับ จากรูปเราจะเห็นว่าโหนดใบจะเป็นโหนดที่บ่งบอกถึงข้อมูลหมวดหมู่ของคำตอบที่เราต้องการ อาทิ เช่น “yes” หมายถึง ลูกค้าจะซื้อคอมพิวเตอร์ และ “no” หมายถึงลูกค้าจะไม่ซื้อคอมพิวเตอร์ โดยต้นไม้ที่ถูกสร้างขึ้นอาจเป็นต้นไม้ที่มีลักษณะเป็นไบนารี่หรืออาจจะไม่เป็นไบนารี่ก็ได้

หลังจากทำการสร้างต้นไม้ตัดสินใจแล้ว เราจะสามารถใช้ต้นไม้ตัดสินใจในการจำแนกข้อมูลได้ โดยจะทำการจำแนกหมวดหมู่ของข้อมูลเรคคอร์ดหนึ่งๆ(ที่ประกอบไปด้วยแอทริบิวต่างๆ แต่เราจะไม่ทราบหมวดหมู่ข้อมูลในเรคคอร์ดนั้นๆ) ด้วยการเปรียบเทียบแอทริบิวที่อยู่ในโหนดรากกับค่าของแอทริบิวในเรคคอร์ดที่พิจารณา โดยจะทำการเปรียบเทียบจากโหนดรากไปจนถึงโหนดใบ เมื่อเราทราบถึงโหนดใบจะทำให้เราทราบถึงหมวดหมู่ข้อมูลของเรคคอร์ดที่ทำการพิจารณา



รูปที่ 6.2 ตัวอย่างต้นไม้ตัดสินใจสำหรับการจำแนกคุณลักษณะของลูกค้าที่ทำการซื้อคอมพิวเตอร์

การจำแนกหมวดหมู่ข้อมูลด้วยต้นไม้ตัดสินใจได้รับความนิยมเป็นอย่างมาก และถูกประยุกต์ใช้ในหลายๆงาน อาทิ เช่น การผลิตและการใช้ยา (medicine) การผลิตสินค้าต่างๆ (manufacturing and production) การวิเคราะห์ทางการเงิน (Financial analysis) ดาราศาสตร์ (astronomy) อณูชีววิทยา (molecular biology) เป็นต้น สาเหตุที่ต้นไม้ตัดสินใจได้รับความนิยมอันเนื่องมากจากเหตุผลหลายประการด้วยกัน เช่น 1) ไม่ต้องการองค์ความรู้ใดๆหรือการกำหนดค่าพารามิเตอร์ใดๆเพื่อที่จะทำการสร้างต้นไม้ตัดสินใจ 2) สามารถจัดการกับข้อมูลที่มีหลายๆมิติหรือข้อมูลที่มีหลายๆแอทริบิวได้ 3) ผลลัพธ์ที่ได้จะเป็นต้นไม้ตัดสินใจที่อยู่ในรูปแบบที่เข้าใจง่าย 4) ขั้นตอนการสร้างต้นไม้ตัดสินใจค่อนข้างง่ายและสามารถทำงานได้อย่างรวดเร็ว และ 5) มักจะให้ผลการจำแนกข้อมูลที่มีความถูกต้องค่อนข้างสูง ที่ซึ่งอาจขึ้นอยู่กับคุณลักษณะของข้อมูลที่เราทำการพิจารณาด้วยเช่นกัน

6.3.1 การสร้างต้นไม้ตัดสินใจ

ในช่วงปลายของยุค 1970 ได้มีนักวิจัยทางด้านการเรียนรู้ของเครื่อง (machine learning) คือ J. Ross Quinlan ได้คิดค้นอัลกอริทึมสำหรับสร้างต้นไม้ตัดสินใจที่มีชื่อว่า ID3 (Iterative Dichotomiser) ต่อมาเขาได้พัฒนาต่อยอด ID3 ไปเป็น C4.5 ซึ่งได้กลายมาเป็นอัลกอริทึมพื้นฐานที่ใช้สำหรับเปรียบเทียบประสิทธิภาพการทำงานของอัลกอริทึมต่างๆทางด้านการเรียนรู้แบบมีผู้สอน (Supervised Learning)

ID3 และ C4.5 ได้ทำการประยุกต์ใช้วิธีการเชิงละโมภ (greedy approach) ในการสร้างต้นไม้ภายใต้วิธีการแบบ “top-down recursive divide-and-conquer” โดยทำการพิจารณาชุดข้อมูลสำหรับเรียนรู้ (training data, เซตของเรคคอร์ดของข้อมูลที่แต่ละเรคคอร์ดจะประกอบไปด้วยเซตของแอทริบิวต่างๆและแอทริบิวที่บ่งบอกถึงหมวดหมู่ของข้อมูลเรคคอร์ดนั้นๆ) ด้วยการแบ่งข้อมูลออกเป็นส่วนย่อยๆในระหว่างกระบวนการสร้างต้นไม้ (ดังแดงรายละเอียดในรูปที่ 6.3)

จากรูปที่ 6.3 จะเป็นการอธิบายรายละเอียดต่างๆของขั้นตอนการสร้างต้นไม้ตัดสินใจ ที่ซึ่งจะสามารถอธิบายได้ดังนี้

- อัลกอริทึมการสร้างต้นไม้ตัดสินใจ จะต้องการอินพุต 3 อินพุตด้วยกันคือ 1) D หมายถึง ชุดข้อมูลที่จะทำการพิจารณา โดยในตอนเริ่มต้นของการทำงาน ชุดข้อมูล D จะประกอบไปด้วยข้อมูลทุกเรคคอร์ด 2) ลิสต์ของแอทริบิว ที่ซึ่งหมายถึง เซตของแอทริบิวที่ใช้ในการอธิบายคุณลักษณะของข้อมูลแต่ละเรคคอร์ดในชุดข้อมูล และ 3) วิธีการในการเลือกแอทริบิวที่ซึ่งเป็นวิธีการในการเลือกแอทริบิวที่ดีที่สุดที่สามารถแยกความแตกต่างระหว่างเรคคอร์ดต่างๆในชุดข้อมูลตามหมวดหมู่ของข้อมูล วิธีการเลือกแอทริบิวนี้จะใช้ตัวชี้วัดการเลือกแอทริบิว เช่น ค่าเกนความรู้ (information gain) หรือค่าดัชนีจีนี่ (gini index) เป็นต้น โดยในการเลือกตัวชี้วัดการเลือกแอทริบิว เราอาจต้องพิจารณาข้อบังคับและคุณลักษณะของต้นไม้ด้วย เช่น ค่าดัชนีจีนี่จะเป็นตัวชี้วัดที่จะสามารถทำงานได้กับต้นไม้แบบไบนารี่ แต่สำหรับค่าเกนความรู้จะสามารถทำงานได้กับต้นไม้ที่มีหลายกิ่งและยอมให้ทำการแตกกิ่งออกเป็นหลายทิศทาง (multi-way split)
- การสร้างต้นไม้จะเริ่มจากโหนด N เพียงแต่หนึ่งโหนดเท่านั้นที่ซึ่งแสดงถึงข้อมูลทั้งหมดในชุดข้อมูล D (ขั้นตอนที่ (1))
- ในกรณีที่ข้อมูลทุกเรคคอร์ดในชุดข้อมูล D มีหมวดหมู่ของข้อมูลเดียวกัน คือ C—โหนด N จะกลายเป็นโหนดใบ และจะมีหมวดหมู่ C แนบอยู่ (ขั้นตอนที่ (2) และ (3))—หมายเหตุ ขั้นตอนที่ (4) และ (5) คือ เงื่อนไขการจบการทำงาน

  ในขั้นตอนที่ (6) จะทำการเรียกใช้วิธีการเลือกแอทริบิวที่จะตัดสินใจเลือกเกณฑ์ในการแบ่งข้อมูล โดยผลลัพธ์ที่ได้จะเป็นแอทริบิวหนึ่งๆที่ดีที่สุดที่สามารถแบ่งข้อมูลเรคคอร์ดต่างๆพร้อมกับหมวดหมู่ของข้อมูลออกจากชุดข้อมูล การเลือกแอทริบิวยังสามารถบอกได้ว่ากิ่งใดจะเติบโตจากโหนด N บ้าง ในการแบ่งข้อมูลออกเป็นส่วนๆ ถ้าส่วนใดก็ตามที่ถูกแบ่งมีเรคคอร์ดทั้งหมดที่มีหมวดหมู่เดียวกับส่วนของข้อมูลนั้นๆจะถูกเรียกว่า “pure partition” แต่อย่างไรก็ตามในการแบ่งข้อมูลอาจมีการแบ่งข้อมูลที่ไม่ได้มีหมวดหมู่เดียวกันอยู่ในส่วนเดียวกันก็เป็นได้—โดยในการแบ่งข้อมูลเราจะต้องพยายามให้ส่วนที่ถูกแบ่งนั้นมีเรคคอร์ดของข้อมูลที่มีหมวดหมู่เหมือนกันมากที่สุดเท่าที่จะมากได้
- โหนด N จะถูกตั้งชื่อด้วยชื่อแอทริบิวที่ได้จากขั้นตอนที่ (6) และแต่ละกิ่งที่เติบโตจากโหนด N จะหมายถึงการเติบโตจากค่าที่เป็นไปได้ในแอทริบิวนั้นๆ โดยในการแบ่งข้อมูลและการเพิ่มกิ่งก้านให้แก่โหนด N (ในขั้นตอนที่ (10) และ (11)) จะเกิดขึ้นภายใต้เงื่อนไข 3 เงื่อนไขดังแสดงในรูปที่ 6.4 โดยกำหนดให้ A เป็นแอทริบิวที่ได้จากการเลือกหรือกล่าวอีกนัยหนึ่งว่า A เป็น splitting_attribute โดยที่ A มีค่าที่เป็นไปได้ทั้งหมดที่เกิดขึ้นในชุดข้อมูล เป็น $\{a_1, a_2, \dots, a_v\}$
  - กรณีที่ A มีค่าแบบไม่ต่อเนื่อง จะทำการแตกกิ่งจากโหนด N ไปตามค่าที่ปรากฏในแอทริบิว A โดยกิ่งหนึ่งจะถูกแทนด้วยค่า a_j หนึ่งๆในแอทริบิว ดังแสดงในรูปที่ 6.4(a) ที่ซึ่งแอทริบิว A คือแอทริบิวที่บ่งบอกถึงข้อมูลสีของเรคคอร์ด โดยมีสีที่เป็นไปได้ 5 สีด้วยกันคือ สีแดง เขียน น้ำเงิน ม่วง และ ส้ม ในกรณีนี้—เราจะทำการแตกกิ่งจากโหนด N ออกตามค่าที่เป็นไปได้ โดยในการแตกกิ่งจากโหนด N เราจะต้องทำการแบ่งข้อมูลจากชุดข้อมูล D ออกเป็นชุดข้อมูลย่อย $\{D_1, D_2, \dots, D_v\}$ โดยชุดข้อมูลย่อย D_j ใดๆจะสอดคล้องกับค่า a_j ที่

เกิดขึ้นในแอทริบิว A ที่ซึ่งทุกๆเรคคอร์ดในชุดข้อมูลย่อย D_j จะมีค่า a_j เกิดขึ้นในแอทริบิว A ดังนั้นเมื่อทำการแตกกิ่งสำหรับโหนด N (แอทริบิว A) แล้ว แอทริบิว A จะไม่ถูกพิจารณาอีกต่อไป สามารถลบแอทริบิว A ออกจากลิสต์ของแอทริบิวที่ทำการพิจารณาได้ (ขั้นตอนที่ (8) และ (9))

- กรณีที่ A มีค่าแบบไม่ต่อเนื่อง จะทำการตรวจสอบโหนด N ภายใต้ 2 เงื่อนไข โดยเราจะต้องทำการหาจุดแบ่งเงื่อนไข (split_point) ถ้าค่าในเรคคอร์ดใดๆก็ตามที่มีค่าน้อยกว่าหรือเท่ากับจุดแบ่งเงื่อนไข จะกำหนดให้ค่านั้นๆอยู่กลุ่มของชุดข้อมูลย่อยทางฝั่งซ้าย แต่ถ้าค่าในเรคคอร์ดมีค่ามากกว่าจุดแบ่งเงื่อนไขจะกำหนดให้อยู่ในกลุ่มของชุดข้อมูลย่อยทางฝั่งขวา จุดแบ่งเงื่อนไขจะได้มาจากการเรียกใช้ attribute_selection_method (โดยทั่วๆไปจุดแบ่งเงื่อนไขมักเป็นค่ากลางของค่าทั้งหมดที่เกิดขึ้นในแอทริบิว เช่น ในแอทริบิวมีค่าเกิดขึ้นอยู่ในช่วงระหว่าง $1-9$ ดังนั้นเราจะสามารถนำค่า 5 ซึ่งเป็นค่ากลางของข้อมูลมาใช้เป็นจุดแบ่งเงื่อนไขได้) ดังแสดงในรูป 6.4(b) ข้อมูลรายได้ของคนจะถูกแบ่งตามจุดแบ่งเงื่อนไขซึ่งมีค่าเท่ากับ $42{,}000$ ถ้าข้อมูลเรคคอร์ดใดๆก็ตามมีค่าเงินรายได้น้อยกว่าหรือเท่ากับ $42{,}000$ เรคคอร์ดนั้นๆจะถูกเก็บไว้ในชุดข้อมูลย่อย D_1 ในส่วนกรณีอื่นๆจะถูกเก็บอยู่ในชุดข้อมูลย่อย D_2 ตามลำดับ
- กรณีที่ A มีค่าแบบไม่ต่อเนื่อง และเราต้องการสร้างต้นไม้ให้อยู่ในรูปแบบของต้นไม้ไบนารี่ (อาจจะเกิดจากข้อจำกัดของตัวชี้วัดการเลือกแอทริบิวที่สามารถสร้างต้นไม้ได้ในลักษณะที่เป็นต้นไม้ไบนารี่เท่านั้น) เราจะต้องมีเซตของค่าในแอทริบิว A ที่ใช้ในการแบ่งชุดข้อมูลออกเป็นสองส่วน (เซต S_A ซึ่งถูกเรียกว่า splitting subset) โดยเซต S_A จะได้มาจากการเรียกใช้ attribute_selection_method เซต S_A จะประกอบไปด้วยค่าต่างๆที่เกิดขึ้นในแอทริบิว A จากตัวอย่างที่แสดงในรูป 6.4(c) เราจะเห็นว่าในเซต S_A ที่เป็น splitting subset จะประกอบไปด้วยสองแอทริบิวด้วยกันคือ สีแดงและสีเขียว ดังนั้น เมื่อทำการพิจารณาเรคคอร์ดหนึ่งๆแล้ว ถ้าค่าในแอทริบิว A ของเรคคอร์ดนั้นๆมีค่าอยู่ในเซต S_A เราจะทำการจัดเก็บเรคคอร์ดนั้นไว้ในชุดข้อมูลย่อย D_1 ส่วนในกรณีอื่นจะเก็บไว้ในชุดข้อมูลย่อย D_2 ตามลำดับ

- อัลกอริทึมการสร้างต้นไม้ตัดสินใจจะใช้กระบวนการแบบเวียนเกิด (recursive) ในการสร้างหรือแตกกิ่งต้นไม้/แบ่งข้อมูลออกเป็นชุดข้อมูลย่อยต่อไปเรื่อยๆ (ขั้นตอนที่ (14))
- การเวียนเกิดจะเสร็จสิ้นก็ต่อเมื่อการทำงานผ่านเงื่อนไขเงื่อนไขหนึ่งดังต่อไปนี้
  - ทุกๆเรคคอร์ดในชุดข้อมูล D ที่กำลังพิจารณาอยู่ขณะที่โหนด N มีหมวดหมู่ของข้อมูลเดียวกัน (ขั้นตอนที่ (2) และ (3))
  - ไม่มีแอทริบิวใดเหลืออยู่ในลิสต์ของแอทริบิวที่ยังไม่ทำการพิจารณา (ขั้นตอนที่ (4)) ในกรณีนี้เราจะทำแตกกิ่งที่เป็นโหนดใบให้แก่โหนด N แล้วทำการแนบหมวดหมู่ของข้อมูล C ให้แก่โหนดใบ โดยที่ C ได้มาจากการใช้ majority vote (ขั้นตอนที่ (5))

อัลกอริทึม การสร้างต้นไม้ตัดสินใจ (Generate_decision_tree)

อินพุต - ชุดข้อมูลสำหรับเรียนรู้ (D) ที่ประกอบไปด้วยเรคคอร์ดต่างๆพร้อมกับหมวดหมู่ข้อมูล
- ลิสต์ของแอทริบิว (attribute_list) ที่อธิบายถึงคุณลักษณะของข้อมูล
- วิธีการเลือกแอทริบิว (attribute_selection_method) เป็นวิธีการในการตัดสินว่าจะเลือกแอทริบิวใด (ที่ดีที่สุด) ในการแบ่งข้อมูลออกตามหมวดหมู่ของข้อมูล

เอาท์พุต ต้นไม้ตัดสินใจ

(1) สร้างโหนด N

(2) **if** ทุกๆเรคคอร์ดในชุดข้อมูล *D* มีหมวดหมู่ *C* เหมือนกันทั้งหมด **then**

(3) **return** N ในฐานะที่เป็นโหนดใบที่มีหมวดหมู่ C แนบอยู่

(4) **if** attribute_list เป็นเซตว่าง **then**

(5) **return** N ในฐานะที่เป็นโหนดใบที่มีหมวดหมู่ C (โดย C มาจากการใช้ majority vote กับหมวดหมู่ของข้อมูลที่ทำการพิจารณาทั้งหมด)

(6) ประยุกต์ใช้ **Attribute_selection_method** (D, attribute_list) เพื่อที่จะหาแอทริบิวที่ดีที่สุดที่จะใช้เป็นจุดแบ่งข้อมูล (find the “best” splitting_criterion)

(7) ทำการกำหนดชื่อของโหนด N ตามชื่อแอทริบิวที่ได้มาจากขั้นตอนที่ 6 (ตาม splitting_criterion)

(8) **if** แอทริบิวที่ได้จากขั้นตอนที่ 6 (splitting_attribute) มีค่าที่เป็นไปได้ในแอทริบิวแบบไม่ต่อเนื่อง (discrete-valued) และ attribute_selection_method ยอมให้ทำการแยกข้อมูลออกเป็นหลายๆส่วน (multi-way splits)

(9) ลบรายชื่อแอทริบิวที่ได้จากขั้นตอนที่ 6 ออกจากลิสต์ของแอทริบิว (attribute_list = attribute_list – splitting_attribute)

(10) **for** แต่ละค่าที่เป็นไปได้ j ของแอทริบิวจากขั้นตอนที่ 6

(11) กำหนดให้ D_j คือ เซตของเรคคอร์ดของข้อมูลใน D ที่มีค่าในแอทริบิวตรงกับค่า j (คือการแบ่งข้อมูลออกเป็นส่วนๆตามค่าที่เป็นไปได้ในแอทริบิวที่ทำการพิจารณา)

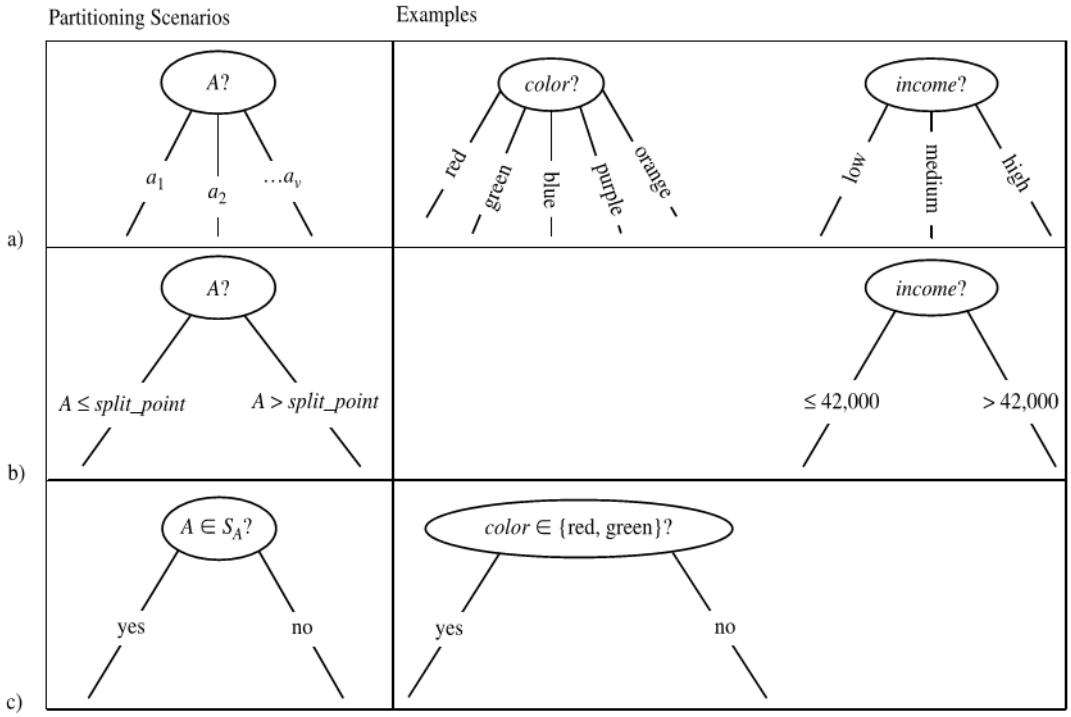
(12) **if** D_j เป็นเซตว่าง (ไม่มีเรคคอร์ดใดเลยใน D ที่มีค่าในแอทริบิวตรงกับค่า j)

(13) ทำการสร้างโหนดลูกให้กับโหนด N โดยโหนดที่สร้างขึ้นจะเป็นโหนดใบที่มี หมวดหมู่ C (โดย C มาจากการใช้ majority vote)

(14) **else** ทำการสร้างโหนดลูกให้กับโหนด N ด้วย Generate_decision_tree(Dj, attribute_list)

(15) **return** N

รูปที่ 6-3 อัลกอริทึมสำหรับการสร้างต้นไม้ตัดสินใจ



รูปที่ 6-4 กรณีที่เป็นไปได้ทั้ง 3 กรณีในการแบ่งข้อมูลจากชุดข้อมูล โดยกำหนดให้ A เป็นแอทริบิวที่ถูกเลือก (splitting_attribute) (a) กรณีที่ A มีค่าแบบไม่ต่อเนื่อง จะทำการแตกกิ่งตามค่าที่เกิดขึ้นใน A (b) กรณีที่ A เป็นค่าแบบต่อเนื่องจะทำการแตกกิ่งออกเป็น 2 กิ่งที่สอดคล้องกับ A ≤ split_point และ A > split_point และ (c) กรณีที่ A มีค่าแบบไม่ต่อเนื่อง แต่บังคับให้ต้องทำการสร้างเฉพาะต้นไม้แบบไบนารี่เท่านั้น ทำการเลือกค่าที่สนใจในเซต S_A แล้วทำการตรวจสอบว่าค่าในแอทริบิว A ของเรคคอร์ดหนึ่งๆอยู่ในเซต S_A หรือไม่

  - ไม่มีเรคคอร์ดใดๆเลยสำหรับชุดข้อมูลย่อย D_j (ขั้นตอนที่ 12) ที่ได้หลังจากการแตกกิ่งในกระบวนการสร้างต้นไม้ ในกรณีนี้—โหนดใบจะทำถูกสร้างด้วย majority vote ของหมวดหมู่ของข้อมูล
- ผลลัพธ์ที่ได้ก็คือต้นไม้ตัดสินใจที่ซึ่งจะถูกคืนค่าในขั้นตอนที่ (15)

จากอัลกอริทึมการสร้างต้นไม้ตัดสินใจในรูปที่ 6.3 เราสามารถคำนวณความซับซ้อนของอัลกอริทึมการสร้างต้นไม้จากชุดข้อมูล D ได้เป็น $O(n \times |D| \times \log(|D|))$ เมื่อ n คือ จำนวนแอทริบิวทั้งหมดที่อธิบายคุณลักษณะของข้อมูลแต่ละเรคคอร์ดในชุดข้อมูล D, $|D|$ คือ จำนวนเรคคอร์ดทั้งหมดในชุดข้อมูล D ที่เป็นข้อมูลสำหรับเรียนรู้

6.3.2 ตัวชี้วัดการเลือกแอทริบิว (Attribute selection measure)

ตัวชี้วัดการเลือกแอทริบิว (attribute selection measure) หรือที่รู้จักกันในนาม splitting rules คือ วิธีการ/เกณฑ์ในการเลือกแอทริบิว (splitting criterion) ที่ดีที่สุดที่จะถูกใช้เพื่อแบ่งส่วนชุดข้อมูล D ออกเป็นชุดข้อมูลย่อยๆ การแบ่งชุดข้อมูลจะทำการแบ่งตามค่าที่เกิดขึ้นในแอทริบิวหนึ่งๆที่ถูกเลือกมา ชุดข้อมูลย่อยหนึ่งๆจะประกอบไปด้วยเซตของเรคคอร์ดที่มีหมวดหมู่เหมือนกันมากที่สุด นอกเหนือจากการแบ่ง

ชุดข้อมูลแล้ว ตัวชี้วัดการเลือกแอทริบิวยังสามารถทำการจัดลำดับของแอทริบิวที่ใช้ในการอธิบายชุดข้อมูล โดยหลังจากทำการจัดลำดับ—แอทริบิวที่มีค่าคะแนนสูงสุดจะถูกเลือกไปเป็นแอทริบิวที่ใช้แบ่งข้อมูล (splitting attribute)—ในกรณีที่ splitting attribute มีค่าที่เกิดขึ้นในแอทริบิวแบบต่อเนื่องหรือเราถูกจำกัดให้สร้างต้นไม้ตัดสินใจแบบไบนารี่ เราจะต้องทราบถึง split point หรือ splitting subset เพื่อที่จะสามารถประมวลผลได้

ในการที่จะแบ่งชุดข้อมูล เราจะใช้นิยามดังต่อไปนี้—กำหนดให้ D คือ ชุดข้อมูลสอนที่ประกอบไปด้วยเซตของเรคคอร์ด ที่ซึ่งแต่ละเรคคอร์ดจะประกอบไปด้วยค่าในแอทริบิวต่างๆที่สามารถอธิบายถึงคุณลักษณะของข้อมูลเรคคอร์ดนั้นๆ รวมถึงค่าในแอทริบิวที่บ่งบอกถึงหมวดหมู่ของข้อมูลในเรคคอร์ดนั้นๆ สมมติให้แอทริบิวที่เป็นหมวดหมู่ของข้อมูลมีค่าที่เป็นไปได้ทั้งสิ้น m หมวดหมู่ ดังนั้น ในเรคคอร์ดหนึ่งจะมีหมวดข้อมูลเป็น C_i $(i = 1, 2, \dots, m)$ กำหนดให้ $C_{i,D}$ คือ เซตของเรคคอร์ดที่อยู่ในหมวดหมู่ C_i กำหนดให้ $|D|$ และ $|C_{i,D}|$ คือ จำนวนเรคคอร์ดที่อยู่ในชุดข้อมูล D และ $C_{i,D}$ ตามลำดับ

ค่าเกนความรู้ (Information gain)

ค่าเกนความรู้จะเป็นตัวชี้วัดการแบ่งข้อมูลออกเป็นชุดข้อมูลย่อยที่ได้รับความนิยมอย่างแพร่หลาย โดยค่านี้จะถูกประยุกต์ใช้ในอัลกอริทึม ID3 ที่ซึ่งจะทำการเลือกแอทริบิวสำหรับแบ่งข้อมูลจากแอทริบิวที่มีค่าเกนความรู้สูงที่สุด ที่ซึ่งจะเป็นการเลือกแอทริบิวที่ต้องการข้อมูลที่น้อยที่สุดในการระบุ/แบ่งข้อมูลออกเป็นชุดข้อมูลย่อย ในการดำเนินงานเริ่มต้น—เราจะต้องเริ่มจากการคำนวณหาค่า $Info(D)$ หรือที่เรียกอีกอย่างหนึ่งว่า ***ค่าเอนโทรปี (entropy of D)*** ที่จะหมายถึง ค่าเฉลี่ยของปริมาณข้อมูลที่ต้องการในการระบุถึงหมวดหมู่ของข้อมูลเรคคอร์ดหนึ่งๆในชุดข้อมูลที่จะขึ้นกับอัตราส่วนของจำนวนเรคคอร์ดที่สอดคล้องกับแต่ละหมวดหมู่ ที่ซึ่งสามารถคำนวณได้ดังนี้

$$Info(D) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

เมื่อ p_i คือ ค่าความน่าจะเป็นที่เรคคอร์ดหนึ่งๆจะมีหมวดหมู่ของข้อมูลเป็นหมวดหมู่ C_i ที่ซึ่งสามารถคำนวณได้จาก $\frac{|C_{i,D}|}{|D|}$

สมมติว่าเราต้องการที่จะแบ่งชุดข้อมูล D ออกเป็นชุดข้อมูลย่อยโดยใช้แอทริบิว A ที่มีค่าที่เกิดขึ้นในชุดข้อมูล v ค่าที่แตกต่างกัน $(\{a_1, a_2, \dots, a_v\})$ ถ้าแอทริบิว A มีค่าที่เกิดขึ้นเป็นแบบไม่ต่อเนื่อง เราจะพิจารณาการแบ่งชุดข้อมูลออกเป็น v ชุดข้อมูลย่อย โดยสามารถแบ่งได้เป็น $(\{D_1, D_2, \dots, D_v\})$ โดยที่ D_j ใดๆจะบรรจุไปด้วยเซตของเรคคอร์ดที่แอทริบิว A มีค่าเป็น a_j เป็นต้น โดยแต่ละชุดข้อมูลย่อยจะสอดคล้องกับกิ่งย่อยที่แตกออกจากโหนด N ที่กำลังทำการพิจารณาอยู่ ดังนั้นในการแบ่งชุดข้อมูลย่อย เราจะต้องพยายามทำให้แต่ละชุดข้อมูลย่อยประกอบไปด้วยเรคคอร์ดที่มีหมวดหมู่เหมือนกันทั้งหมด แต่อย่างไรก็ตาม

มักจะเป็นการยากที่จะมีเรคคอร์ดที่มีหมวดหมู่เหมือนกันมีค่าในแอทริบิวเหมือนกัน เนื่องจากในชุดข้อมูลย่อยที่ถูกแบ่งมักจะมีเรคคอร์ดที่มีหมวดหมู่ที่แตกต่างกัน ดังนั้นเราจะต้องทำการพิจารณาว่าเรคคอร์ดที่มีค่าในแอทริบิวเหมือนกันนั้นมีความเหมือนกันของหมวดหมู่เป็นเช่นไร โดยเราจะต้องทำการคำนวณค่า $Info_A(D)$ ที่ซึ่งหมายถึงจำนวนข้อมูลที่คาดว่าจะใช้สำหรับทำการแบ่งชุดข้อมูล D ออกเป็นชุดข้อมูลย่อย โดยทำการพิจารณาแอทริบิว A ที่ซึ่งสามารถคำนวณได้ดังนี้

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{D} \times Info(D_j)$$

เมื่อ $\frac{|D_j|}{|D|}$ หมายถึง จำนวนเรคคอร์ดในชุดข้อมูล D ที่มีค่าในแอทริบิว A เป็น a_j หารด้วยจำนวนเรคคอร์ดทั้งหมดในชุดข้อมูล D

หลังจากคำนวณค่า $Info(D)$ และค่า $Info_A(D)$ ค่าเกนความรู้สำหรับการพิจารณาแอทริบิว A จะสามารถคำนวณได้จากค่าความแตกต่างระหว่างปริมาณข้อมูลที่ต้องการในการระบุถึงหมวดหมู่ของข้อมูลสำหรับเรคคอร์ดหนึ่งๆกับจำนวนข้อมูลที่คาดว่าจะใช้สำหรับทำการแบ่งชุดข้อมูล D ออกเป็นชุดข้อมูลย่อย โดยทำการพิจารณาแอทริบิว A ที่ซึ่งสามารถแสดงการคำนวณได้ดังนี้

$$Gain(A) = Info(D) - Info_A(D)$$

ค่า $Gain(A)$ จะบ่งบอกปริมาณเกนความรู้ที่เราจะได้รับด้วยการพิจารณาแอทริบิว A และการแบ่งชุดข้อมูลออกเป็นชุดข้อมูลย่อยตามการพิจารณาแอทริบิว A ถ้าค่า $Gain(A)$ เป็นค่าเกนความรู้ที่มีค่ามากที่สุดในบรรดาค่าเกนความรู้ของแอทริบิวทั้งหมด แอทริบิว A จะเป็นแอทริบิวที่ดีที่สุดสำหรับการจำแนก/แบ่งข้อมูลเป็นชุดข้อมูลย่อย เนื่องจากแอทริบิว A ต้องการปริมาณข้อมูลที่น้อยที่สุดที่ใช้ในการจำแนกข้อมูล (minimal $Info_A(D)$)

ตัวอย่าง การสร้างต้นไม้ตัดสินใจด้วยค่าเกนความรู้ กำหนดให้ชุดข้อมูลสอน D จากร้านขายอุปกรณ์ไฟฟ้าประกอบไปด้วยเซตของเรคคอร์ด 14 เรคคอร์ดดังแสดงในตารางที่ 6.1 ที่ซึ่งเป็นข้อมูลคุณลักษณะของลูกค้าที่จะทำการซื้อ/ไม่ซื้อคอมพิวเตอร์ที่ประกอบไปด้วยข้อมูลหลัก 4 แอทริบิวที่มีค่าแบบไม่ต่อเนื่อง คือ อายุ รายได้ อาชีพนักศึกษาหรือไม่ และข้อมูลประวัติเครดิตทางการเงิน และ 1 แอทริบิวหมวดหมู่ของลูกค้า ที่ประกอบไปด้วย 2 หมวดหมู่ คือ หมวดหมู่ของลูกค้าที่ซื้อคอมพิวเตอร์จากร้านค้า และหมวดหมู่ของลูกค้าที่ไม่ได้ซื้อคอมพิวเตอร์จากร้านค้า เมื่อเราทำการสังเกตชุดข้อมูล เราจะเห็นว่ามี 9 เรคคอร์ดที่เป็นข้อมูลลูกค้าในหมวดหมู่ของการซื้อคอมพิวเตอร์ และมี 5 เรคคอร์ดที่เป็นลูกค้าที่ไม่ซื้อคอมพิวเตอร์ ตามลำดับ

ตารางที่ 6.1 ตัวอย่างข้อมูลสอนจากร้านขายอุปกรณ์ไฟฟ้า

RID	Age	Income	student	Credit_rating	Class:buys_computer
1	youth	High	no	fair	no
2	Youth	High	no	excellent	no
3	middle_aged	High	no	fair	yes
4	senior	medium	No	fair	yes
5	senior	Low	yes	fair	yes
6	senior	Low	yes	excellent	no
7	middle_aged	Low	yes	excellent	yes
8	youth	medium	no	fair	no
9	youth	Low	yes	fair	yes
10	senior	medium	yes	fair	yes
11	youth	medium	yes	excellent	yes
12	middle_aged	medium	no	excellent	yes
13	middle_aged	High	yes	fair	yes
14	senior	medium	no	excellent	no

ขั้นตอนแรกของการสร้างต้นไม้ตัดสินใจ คือ การสร้างโหนด N สำหรับข้อมูลทั้งหมดในชุดข้อมูล D จากนั้นทำการค้นหาเกณฑ์/แอทริบิวที่ใช้ในการแยกข้อมูลออกเป็นชุดย่อยๆด้วยการคำนวณหาค่าเกนความรู้ของแต่ละแอทริบิวในตารางที่ 6.1 (ทำการหาค่าเกนความรู้ของแอทริบิวอายุ รายได้ อาชีพที่เป็นนักศึกษาหรือไม่ และข้อมูลประวัติเครดิตทางการเงิน) แต่ก่อนที่จะทำการหาค่าเกนความรู้ของแต่ละแอทริบิว เราจะต้องทำการคำนวณค่าเอนโทรปีของชุดข้อมูล D เสียก่อน ที่ซึ่งสามารถคำนวณได้ดังนี้

$$Info(D) = -\frac{9}{14}log_2\left(\frac{9}{14}\right) - \frac{5}{14}log_2\left(\frac{5}{14}\right) = 0.940$$

ขั้นตอนต่อไป เราจะทำการคำนวณค่าเกนความรู้ของแต่ละแอทริบิว โดยเริ่มทำการพิจารณาแอทริบิวอายุเป็นลำดับแรก จากนั้นเราจะพิจารณาการกระจายของข้อมูลหมวดหมู่ต่างๆตามแต่ละค่าที่เป็นไปได้ของแอทริบิวอายุที่มีค่าที่เกิดขึ้น 3 ค่าด้วยกันคือ ‘youth’, ‘middle_aged’ และ ‘senior’ ตามลำดับ เมื่อพิจารณาช่วงอายุที่เป็น ‘youth’ เราจะเห็นว่ามี 2 เรคคอร์ดที่มีค่าอายุเป็น ‘youth’ และเป็นลูกค้าที่อยู่ในหมวดหมู่ที่ซื้อคอมพิวเตอร์ และมี 3 เรคคอร์ดที่มีค่าอายุเป็น ‘youth’ และอยู่ในหมวดหมู่ที่ไม่ซื้อคอมพิวเตอร์ ในส่วนของช่วงอายุที่เป็น ‘middle_aged’ จะมี 4 เรคคอร์ดที่มีค่าอายุเป็น ‘middle_aged’ และเป็นลูกค้าที่อยู่ในหมวดหมู่ที่ซื้อคอมพิวเตอร์ แต่จะไม่มีลูกค้ามีค่าอายุเป็น ‘middle_aged’ และอยู่ในหมวดหมู่ที่ไม่ซื้อคอมพิวเตอร์เลย และในส่วนของช่วงอายุที่เป็น ‘senior’ จะมี 3 เรคคอร์ดที่มีค่าอายุเป็น ‘senior’ และเป็นลูกค้าที่อยู่ในหมวดหมู่ที่ซื้อคอมพิวเตอร์ และ 2 เรคคอร์ดมีค่าอายุเป็น ‘senior’ และอยู่ในหมวดหมู่ที่ไม่ซื้อ

คอมพิวเตอร์ ตามลำดับ ดังนั้น เราสามารถคำนวณการกระจายของข้อมูลหมวดหมู่ต่างๆตามแต่ละค่าที่เป็นไปได้ของแอทริบิวอายุได้เป็น

$$Info_{age}(D) = \frac{5}{14} \times (-\frac{2}{5}\, log_2 \frac{2}{5} - \frac{3}{5}\, log_2 \frac{3}{5}) + \frac{4}{14} \times (-\frac{4}{4} log_2 \frac{4}{4} - \frac{0}{4} log_2 \frac{0}{4}) + \frac{5}{14} \times (-\frac{3}{5} log_2 \frac{3}{5} - \frac{2}{5} log_2 \frac{2}{5}) = 0.694$$

จากค่าการกระจายของข้อมูลหมวดหมู่ต่างๆตามแต่ละค่าที่เป็นไปได้ของแอทริบิวอายุ เราสามารถคำนวณค่าเกนความรู้ของการแบ่งชุดข้อมูลด้วยแอทริบิวอายุได้เป็น

$$Gain(age) = Info(D) - Info_{age}(D) = 0.940 - 0.694 = 0.246$$

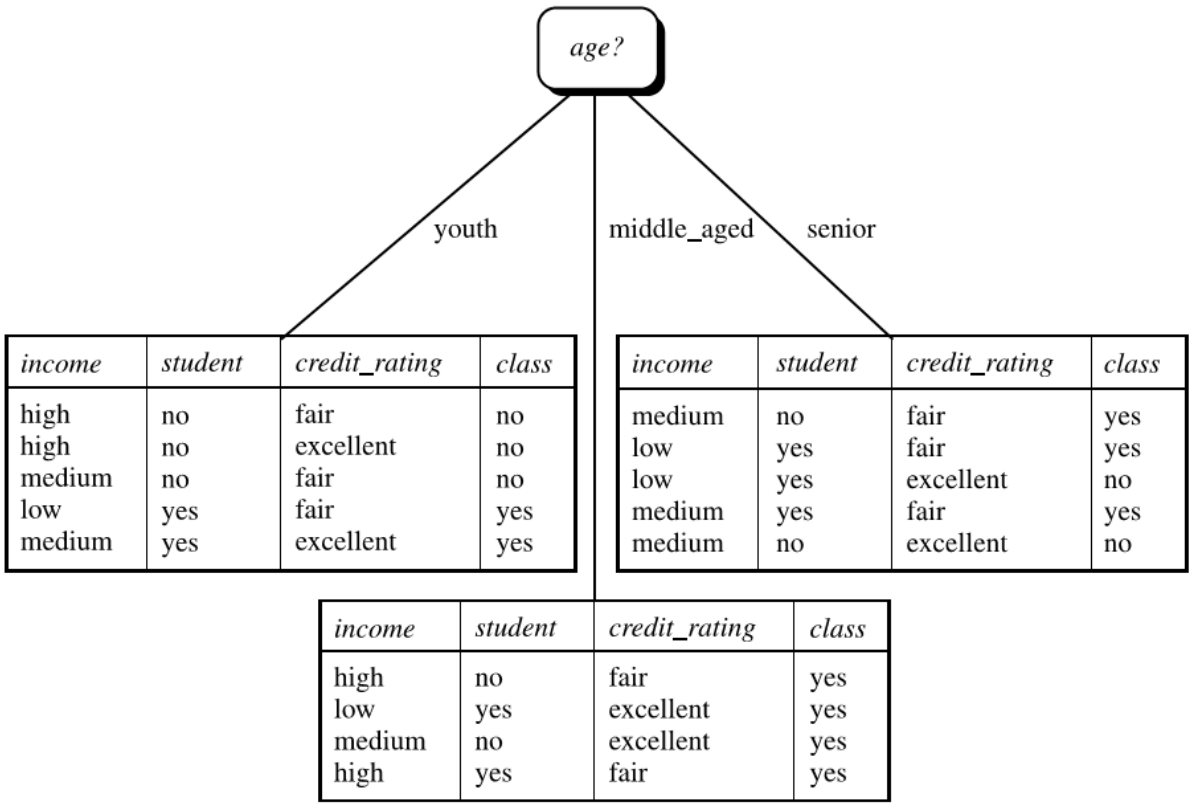
หลังจากคำนวณค่าเกนความรู้ของแอทริบิวอายุแล้ว เราจะต้องทำการคำนวณค่าเกนความรู้ของแอทริบิวอื่นๆด้วยทั้งหมดด้วยวิธีการเดียวกับข้างต้น ที่ซึ่งสามารถคำนวณได้เป็น $Gain(income) = 0.029$, $Gain(student) = 0.151$ และ $Gain(credit\_rating) = 0.048$ ตามลำดับ และ เมื่อทำการเปรียบเทียบค่าเกนความรู้ของทุกแอทริบิว เราจะสังเกตได้ว่าค่าเกนความรู้ของแอทริบิวอายุจะมีค่ามากที่สุด ดังนั้น เราจะเลือกแอทริบิวอายุให้เป็นแอทริบิวสำหรับแบ่งชุดข้อมูลออกเป็นออกเป็นชุดข้อมูลย่อย ด้วยการแนบชื่อของโหนด N ด้วยชื่อของแอทริบิวอายุ และทำการแตกกิ่งจากโหนด N ตามค่าทั้ง 3 ที่เกิดขึ้นในแอทริบิว N (‘youth’, ‘middle_aged’, และ ‘senior’) จากนั้นทำการแบ่งข้อมูลในชุดข้อมูล D ดังแสดงในรูปที่ 6.5 จากรูปเราจะสังเกตุได้ว่าช่วงอายุที่เป็น ‘middle_aged’ จะมีเฉพาะข้อมูลลูกค้าที่อยู่ในหมวดหมู่ที่ซื้อคอมพิวเตอร์ ดังนั้น เราจะทำการสร้างโหนดใบในกิ่งของช่วงอายุที่เป็น ‘middle_aged’ และกำหนดชื่อโหนดให้เป็น ‘หมวดหมู่ที่ซื้อคอมพิวเตอร์’ ตามลำดับ จากนั้นเราจะดำเนินการแบบเวียนเกิดกับแต่ละกิ่งของต้นไม้ที่ไม่ใช่โหนดใบ เมื่อดำเนินการสร้างต้นไม้จนเสร็จสิ้น เราจะได้ต้นไม้ตัดสินใจดังรูปที่ 6.2

จากตัวอย่างข้างต้นเราจะสังเกตุได้ว่าค่าที่เกิดขึ้นในทุกๆแอทริบิวจะมีค่าเป็นแบบไม่ต่อเนื่อง (หมายถึงเป็นค่าที่บ่งบอกถึงหมวดหมู่ที่แน่นอน ที่ซึ่งไม่ใช่ค่าที่แสดงถึงข้อมูลเชิงปริมาณ) แต่สำหรับแอทริบิวที่มีค่าแบบต่อเนื่อง เราจะต้องคำนวณค่าเกนความรู้ในลักษณะที่แตกต่างออกไป เพื่อให้มีความเข้าใจมากขึ้น ลองสมมิตว่าเรามีแอทริบิว A หนึ่งๆที่มีค่าเป็นแบบต่อเนื่อง อาทิเช่น แอทริบิวอายุที่ซึ่งบ่งบอกถึงตัวเลขอายุของลูกค้าแต่ละราย จากค่าที่มีลักษณะแบบต่อเนื่อง เราจะต้องทำการค้นหาจุดแบ่งที่ดีที่สุด (“best” split-point) สำหรับแอทริบิว A ที่ซึ่งจุดแบ่งจะทำหน้าที่เปรียบเสมือนค่าขีดแบ่งในการแบ่งย่อยข้อมูลโดยใช้แอทริบิว A—ดังนั้น เพื่อที่จะทำการหาจุดแบ่งที่ดีที่สุด เราจะเริ่มจากการเรียงลำดับค่าที่เกิดขึ้นในแอทริบิว A จากทุกๆเรคคอร์ดในชุดข้อมูลที่ทำการพิจารณา โดยเรียงลำดับจากน้อยไปมาก จากนั้นทำการค่ากลาง

(midpoint) ระหว่าง 2 ค่าใดที่มีค่าติดกัน โดยค่า a_i และ ค่า a_{i+1} ใดๆจะถูกพิจารณาเพื่อคำนวณค่ากลางได้ดังนี้

$$\frac{a_i + a_{i+1}}{2}$$

ค่ากลางที่คำนวณได้จะถูกนำมาพิจารณาเพื่อเป็นจุดแบ่งชุดข้อมูล ดังนั้น ถ้าแอทริบิว A มีค่าที่เกิดขึ้นในเรคคอร์ดทั้งหมด v ค่า เราจะสามารถหาค่ากลางได้ทั้งหมด $v-1$ ค่าด้วยกัน เมื่อเราคำนวณค่ากลางทั้งหมดแล้ว เราจะต้องทำการพิจารณาว่าค่ากลางใดเป็นค่าที่เหมาะจะเป็นจุดแบ่งด้วยการหาค่า $Info_A(D)$ เมื่อจำนวนกลุ่มของข้อมูลที่ต้องการแบ่งออกเป็นชุดข้อมูลย่อยๆจะมีค่าเท่ากับ 2 ซึ่งก็คือ 1) กลุ่มของเรคคอร์ดที่มีค่าในแอทริบิว A น้อยกว่าหรือเท่ากับค่ากลางที่ทำการพิจารณา และ 2) กลุ่มของเรคคอร์ดที่มีค่าในแอทริบิว A มากกว่าค่ากลางที่ทำการพิจารณา เมื่อทำการหาค่า $Info_A(D)$ ของทุกๆค่ากลางแล้ว เราจะเลือกค่ากลางที่มีค่า $Info_A(D)$ น้อยที่สุดเป็นจุดแบ่งสำหรับแอทริบิว A ตามลำดับ



income	student	credit_rating	class
high	no	fair	no
high	no	excellent	no
medium	no	fair	no
low	yes	fair	yes
medium	yes	excellent	yes

income	student	credit_rating	class
medium	no	fair	yes
low	yes	fair	yes
low	yes	excellent	no
medium	yes	fair	yes
medium	no	excellent	no

income	student	credit_rating	class
high	no	fair	yes
low	yes	excellent	yes
medium	no	excellent	yes
high	yes	fair	yes

รูปที่ 6.5 การแบ่งชุดข้อมูลออกเป็นชุดข้อมูลย่อยด้วยแอทริบิวอายุ

ค่าอัตราส่วนเกน (Gain ratio)

ค่าอัตราส่วนเกนจะเป็นตัวชี้วัดการแบ่งชุดข้อมูลออกเป็นชุดข้อมูลย่อยที่พัฒนามาจากค่าเกนความรู้ โดยเมื่อเราใช้ค่าเกนความรู้ในการแบ่งชุดข้อมูลจะทำให้เกิดความเอนเอียงเกิดขึ้นเมื่อแอทริบิวที่ทำการพิจารณามีค่าที่เกิดขึ้นเป็นจำนวนมาก โดยในการใช้ค่าเกนความรู้มักจะทำการเลือกแอทริบิวที่มีค่าที่เกิดขึ้น

เป็นจำนวนมาก ตัวอย่างเช่น ลองพิจารณาแอทริบิวรายการสินค้าที่มีค่าเป็นรหัสสินค้าต่างๆ (product_ID) ถ้าเราทำการแบ่งชุดข้อมูลตามแอทริบิวรายการสินค้าจะทำให้มีชุดข้อมูลย่อยเป็นจำนวนมาก (ชุดข้อมูลย่อยจะมีจำนวนเท่ากับรายการสินค้าทั้งหมดที่มีในชุดข้อมูล) โดยแต่ละชุดข้อมูลย่อยจะมีข้อมูลเพียงหนึ่งเรคคอร์ดเท่านั้นและจะทำให้ชุดข้อมูลย่อยนั้นๆมีข้อมูลที่มีหมวดหมู่ของข้อมูลเหมือนกันทั้งหมด (เพราะมีเพียงหนึ่งเรคคอร์ดเลยเหมือนกันทั้งหมดนั่นเอง) ที่ซึ่งจะทำให้ค่า $Info_{product\_ID}(D) = 0$ ที่ซึ่งจะทำให้ค่าเกนความรู้มีค่าสูง ($Gain(A) = Info(D) - Info_A(D) = Info(D) - 0 = Info(D)$) ดังนั้นแอทริบิวที่มีลักษณะคล้ายกับแอทริบิวรายการสินค้ามักจะถูกเลือกเพื่อใช้ในการแบ่งชุดข้อมูลเสมอ

จากความโน้มเอียงที่อาจเกิดขึ้นจากการใช้ค่าเกนความรู้ ได้มีนักวิจัยพยายามที่จะลดทอนความเอนเอียงลง โดยการพัฒนาตัวชี้วัดการแบ่งข้อมูลใหม่ ที่มีชื่อว่า อัตราส่วนเกน (gain ratio) ที่จะประยุกต์ใช้การทำนอร์มัลไลซ์ค่าเกนความรู้ด้วยการใช้ค่า “split information” ที่ซึ่งสามารถคำนวณได้ดังนี้

$$SplitInfo_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times log_2(\frac{|D_j|}{|D|})$$

โดยค่า $SplitInfo_A(D)$ จะแสดงถึงปริมาณข้อมูลที่ถูกพิจารณาโดยการแบ่งข้อมูลในชุดข้อมูล D ออกเป็น v ชุดข้อมูลย่อยตามค่าในแอทริบิว A โดยหลังจากทำการคำนวณหาค่า $SplitInfo_A(D)$ แล้ว เราจะสามารถคำนวณหาค่าอัตราส่วนเกนได้ดังนี้

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)}$$

เมื่อทำการคำนวณหาค่าอัตราส่วนเกนของทุกแอทริบิวที่ทำการพิจารณาแล้ว เราจะทำการเลือกแอทริบิวที่มีค่าอัตราส่วนเกนสูงที่สุดเพื่อเป็นแอทริบิวสำหรับแบ่งชุดข้อมูลออกเป็นชุดข้อมูลย่อยต่อไป (ข้อสังเกต ถ้าในการคำนวณค่า $SplitInfo_A(D)$ มีค่าเท่ากับ 0 จะทำให้การคำนวณค่าอัตราส่วนเกนของแอทริบิว A จะมีค่าไม่แน่นอน)

ตัวอย่าง การคำนวณค่าอัตราส่วนเกนสำหรับแอทริบิวรายได้ จากข้อมูลข้อมูลสอนดังตารางที่ 6.1 ที่ซึ่งเราจะทำการแบ่งข้อมูลออกเป็น 3 ชุดข้อมูลย่อยตามค่าที่เกิดขึ้นในแอทริบิวรายได้ ซึ่งก็คือ ‘low’, ‘medium’ และ ‘high’ ที่ซึ่งแต่ละค่าจะมีเรคคอร์ดที่มีค่านั้นๆทั้งสิ้น 4, 6 และ 4 เรคคอร์ดตามลำดับ ดังนั้นเราสามารถคำนวณค่าอัตราส่วนเกนสำหรับแอทริบิวรายได้ ได้เป็น

$$SplitInfo_{income}(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times log_2(\frac{|D_j|}{|D|})$$

$$SplitInfo_{income}(D) = -\frac{4}{14} \times log_2\left(\frac{4}{14}\right) - \frac{6}{14} \times log_2\left(\frac{6}{14}\right) - \frac{4}{14} \times log_2\left(\frac{4}{14}\right)$$

$$= 0.926$$

จากตัวอย่างการคำนวณค่าเกนความรู้ เราจะทราบว่าค่าเกนความรู้ของแอทริบิวรายได้จะมีค่าเท่ากับ $Gain(income) = 0.029$ ดังนั้นเราสามารถคำนวณหาค่าอัตราส่วนเกนได้เป็น

$$GainRatio(income) = \frac{Gain(income)}{SplitInfo_{income}(D)}$$

$$GainRatio(income) = \frac{0.029}{0.926} = 0.031$$

ดัชนีจีนี่ (Gini index)

ดัชนีจีนี่จะเป็นตัวชี้วัดที่จะทำการพิจารณาความไม่บริสุทธิ์ของชุดข้อมูล D ที่ซึ่งจะมีเซตของเรคคอร์ดที่มีหมวดหมู่ของข้อมูลไม่เหมือนกันอยู่ โดยเริ่มจากการคำนวนหาค่าความไม่บริสุทธิ์ของชุดข้อมูล D ดังนี้

$$Gini\ (D) = 1 - \sum_{i=1}^{m} p_i^2$$

เมื่อ p_i คือ ค่าความน่าจะเป็นที่เรคคอร์ดหนึ่งๆจะมีหมวดหมู่ของข้อมูลเป็นหมวดหมู่ C_i ที่ซึ่งสามารถคำนวณได้จาก $\frac{|C_{i,D}|}{|D|}$

ในการแบ่งชุดข้อมูลโดยใช้ดัชนีจีนี่จะสามารถแบ่งชุดข้อมูลออกเป็น 2 ชุดข้อมูลย่อยเท่านั้น เมื่อทำการพิจารณาแอทริบิว A ใดๆที่มีค่าที่เกิดขึ้นเป็นแบบไม่ต่อเนื่องและมีค่าที่แตกต่างกันทั้งสิ้น v ค่า ($\{a_1, a_2, \dots, a_v\}$) เราจะต้องทำการค้นหาการแบ่งชุดข้อมูลที่ดีที่สุดภายใต้การพิจารณาแอทริบิว A ด้วยการพิจารณาทุกสับเซตที่เป็นไปได้ของค่าที่เกิดขึ้นในแอทริบิว A ที่จะถูกจัดอยู่ในเซตย่อย 2 เซตด้วยกัน ตัวอย่างเช่น แอทริบิวรายได้ที่มีค่าที่เกิดขึ้นในแอทริบิว 3 ค่าด้วยกันคือ $\{low, medium, high\}$ ซึ่งจากทั้ง 3 ค่าที่เกิดขึ้น เราจะสามารถแบ่งค่าทั้ง 3 ออกเป็นเซตย่อย 2 เซตได้ดังต่อไปนี้

$$\{(\{low, medium\}, \{high\}), \langle\{low, high\}, \{medium\}\rangle, \langle\{medium, high\}, \{low\}\rangle,$$

$$\langle\{low\}, \{medium, high\}\rangle, \langle\{medium\}, \{low, high\}\rangle, \langle\{high\}, \{low, medium\}\rangle\}$$

เมื่อทำการแบ่งค่าที่เกิดขึ้นออกเป็น 2 เซตย่อย เราจะนำ 2 เซตใดๆที่แบ่งไว้แล้ว มาทำการคำนวณค่าน้ำหนักรวมของค่าความไม่บริสุทธิ์ โดยในการพิจารณาแอทริบิว A ใดๆ เราจะทำการแบ่งชุดข้อมูล D ออกเป็น 2 ชุดข้อมูลย่อย คือ D_1 และ D_2 ดังนั้น เราจะสามารถหาค่าดัชนีจีนี่เมื่อทำการพิจารณาแอทริบิว A ภายใต้การพิจารณา D_1 และ D_2 ได้เป็น

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2)$$

เมื่อทำการคำนวณค่าดัชนีจีนี่สำหรับ 2 เซตย่อยใดๆภายใต้แอทริบิว A 2 เซต D_1 และ D_2 ใดที่ให้ค่า $Gini_A(D)$ น้อยที่สุดจะถูกเลือกเป็นตัวแทนของแอทริบิว A

ในกรณีที่แอทริบิวที่เราทำการพิจารณามีค่าแบบต่อเนื่อง เราจะดำเนินการด้วยวิธีการเดียวกันกับการคำนวณหาค่าเกนความรู้ ด้วยการหาจุดแบ่งที่ดีที่สุด โดยทำการเรียงลำดับค่าทั้งหมดจากน้อยไปมาก แล้วทำการหาค่ากลางระหว่าง 2 ค่าใดๆ จากนั้นนำแต่ละค่ากลางมาพิจารณาเป็นจุดแบ่ง โดยค่าใดๆก็ตามที่มีค่าน้อยกว่าหรือเท่ากับค่ากลางจะถูกแบ่งไว้ในชุดข้อมูลย่อย D_1 และค่าใดก็ตามที่มีค่ามากกว่าค่ากลางจะถูกแบ่งไว้ในชุดข้อมูลย่อย D_2 ตามลำดับ จากนั้นค่ากลางใดที่ให้ค่า $Gini_A(D)$ น้อยที่สุดจะถูกเลือกเป็นตัวแทนของแอทริบิว A

ขั้นตอนสุดท้ายของการแบ่งชุดข้อมูลด้วยการใช้ค่าดัชนีจีนี่ เราจะต้องทำการคำนวณส่วนหักลบของความไม่บริสุทธ์ ซึ่งก็คือ $\Delta Gini(A)$ โดยเมื่อทำการคำนวณค่า $\Delta Gini(A)$ สำหรับทุกๆแอทริบิวแล้ว แอทริบิวใดก็ตามที่มีค่า $\Delta Gini(A)$ มากที่สุด (หรือมีค่า $Gini_A(D)$ น้อยที่สุด) แอทริบิวนั้นจะถูกเลือกเพื่อเป็นแอทริบิวสำหรับแบ่งชุดข้อมูลออกเป็นชุดข้อมูลย่อย

$$\Delta Gini(A) = Gini(D) - Gini_A(A)$$

ตัวอย่าง การสร้างต้นไม้ตัดสินใจด้วยค่าดัชนีจีนี่จากข้อมูลสอนดังตารางที่ 6.1 ที่จะประกอบไปด้วยข้อมูล 9 เรคคอร์ดที่มีหมวดหมู่ข้อมูลของลูกค้าที่ซื้อคอมพิวเตอร์ และมีข้อมูลอีก 5 เรคคอร์ดที่มีหมวดหมู่ข้อมูลของลูกค้าที่ไม่ซื้อคอมพิวเตอร์ ดังนั้นเมื่อเราพิจารณาที่โหนด N ที่ซึ่งเป็นโหนดเริ่มต้น เราจะเริ่มจากการคำนวนหาค่าความไม่บริสุทธิ์ของชุดข้อมูล D ดังนี้

$$Gini(D) = 1 - \left(\frac{9}{14}\right)^2 - \left(\frac{5}{14}\right)^2 = 0.459$$

จากนั้นจึงทำการพิจารณาแต่ละแอทริบิวเพื่อหาแอทริบิวเพื่อใช้ในการแบ่งชุดข้อมูล โดยถ้าเราเริ่มจากการพิจารณาแอทริบิว “รายได้” ที่ซึ่งมีค่าเป็นแบบไม่ต่อเนื่องและมีค่าที่เกิดขึ้นเท่ากับ 3 ค่า ดังนั้นเราจะต้องทำการแบ่งข้อมูลออกเป็น 2 เซตใดๆ เช่นถ้าเราแบ่งค่าออกเป็น 2 เซต คือ $\{low, medium\}$ และ $\{high\}$ ที่จะทำการแบ่งเป็นชุดข้อมูลย่อย D_1 และ D_2 โดยในเซตแรก คือ $\{low, medium\}$ จะมีข้อมูลจำนวนทั้งสิ้น 10 เรคคอร์ดที่มีค่าเป็น “low” หรือ “medium” แต่ในส่วนของเซตที่สอง คือ $\{high\}$ จะมีข้อมูลจำนวนทั้งสิ้น 4 เรคคอร์ดที่มีค่าเป็น “high” ดังนั้นเราจะสามารถคำนวณค่าดัชนีจีนี่ภายใต้ 2 เซตนี้ได้ดังนี้

$$\begin{aligned}
&Gini_{income \in \{low, medium\}}(D) \\
&= \frac{10}{14} Gini(D_1) + \frac{4}{14} Gini(D_2) \\
&= \frac{10}{14}\left(1 - \left(\frac{7}{10}\right)^2 - \left(\frac{3}{10}\right)^2\right) + \frac{4}{14}\left(1 - \left(\frac{2}{4}\right)^2 - \left(\frac{2}{4}\right)^2\right) \\
&= 0.44 \\
&= Gini_{income \in \{high\}}(D)
\end{aligned}$$

ต่อมาเมื่อเราทำการคำนวณค่าดัชนีจีนี่ของ 2 เซตอื่นๆ เช่น $\langle\{low, high\}, \{medium\}\rangle$ จะมีค่าดัชนีจีนี่เท่ากับ 0.315 และ $\langle\{medium, high\}, \{low\}\rangle$ จะมีค่าดัชนีจีนี่เท่ากับ 0.300 โดยหลังจากทำการคำนวณค่าดัชนีจีนี่ทั้งหมดจากทุกเซตที่แบ่งได้จากการพิจารณาแอทริบิว “รายได้” แล้ว เราจะทำการเลือกเซต $\langle\{medium, high\}, \{low\}\rangle$ ที่มีค่าดัชนีจีนี่น้อยที่สุดเป็นเซตของค่าที่ใช้แบ่งข้อมูลออกเป็นชุดข้อมูลย่อยภายใต้การพิจารณาแอทริบิว “รายได้” และจะสามารถคำนวณ $\Delta Gini(income)$ ได้เท่ากับ $0.459 - 0.300 = 0.159$

ต่อมาเมื่อพิจารณาแอทริบิว “อายุ” เราจะทราบว่าเซต $\langle\{youth, senior\}, \{middle\_aged\}\rangle$ จะให้ค่าดัชนีจีนี่ภายใต้การพิจารณาแอทริบิว “อายุ” น้อยที่สุดเท่ากับ 0.375 และเมื่อพิจารณาแอทริบิว “การเป็นนักเรียน” และ “ประวัติทางการเงิน” ที่ซึ่งจะเป็นแอทริบิวที่มีค่าในแอทริบิว 2 ค่าด้วยกัน ดังนั้นเราจะสามารถคำนวณค่าดัชนีจีนี่ของแต่ละแอทริบิวได้เป็น 0.367 และ 0.429 ตามลำดับ

เมื่อทำการคำนวณส่วนหักลบของความไม่บริสุทธิ์ ($\Delta Gini(A)$) ของทุกแอทริบิวแล้วเราจะทราบว่าแอทริบิว “รายได้” ภายใต้เซต $\langle\{medium, high\}, \{low\}\rangle$ จะมีค่าดัชนีจีนี่มากที่สุด ดังนั้นเราจะเลือกแอทริบิว “รายได้” เป็นแอทริบิวสำหรับแบ่งข้อมูลเป็นชุดข้อมูลย่อย ดังนั้น โหนด N ที่ทำการพิจารณาจะถูกกำหนดให้เป็นโหนดสำหรับแอทริบิว “รายได้” ที่ซึ่งจะสามารถแตกกิ่งเป็นกิ่งย่อยได้ 2 กิ่งคือ กิ่งสำหรับ $\{medium, high\}$ และ กิ่งสำหรับ $\{low\}$ ตามลำดับ

นอกเหนือจากตัวชี้วัดการเลือกแอทริบิวทั้ง 3 ที่ได้ศึกษาข้างต้น เราจะสังเกตได้ว่าค่าเกนความรู้จะมีความเอนเอียงกับแอทริบิวที่มีค่าที่เกิดขึ้นหลายๆค่า (multivalued attribute) ค่าอัตราส่วนเกนจะสามารถทำงานได้ดีกับการแบ่งข้อมูลที่ไม่สมดุล กล่าวคือ จะมีชุดข้อมูลย่อยหนึ่งๆที่มีเรคคอร์ดในชุดข้อมูลย่อยนั้นๆค่อนข้างน้อย ดัชนีจีนี่จะมีความเอนเอียงกับแอทริบิวที่มีค่าที่เกิดขึ้นหลายๆค่า และจะประสบความยุ่งยากในการดำเนินการกับข้อมูลที่มีหมวดหมู่ข้อมูลค่อนข้างมากอีกด้วย จากปัญหาดังกล่าว ยังมีตัวชี้วัดอื่นๆที่ได้รับความนิยมอย่างแพร่หลาย อาทิเช่น χ^2 ที่ซึ่งจะทำการวัดจากความไม่ขึ้นแก่กันของข้อมูล โดย χ^2 จะได้รับความนิยมเป็นอย่างมากในการสร้างต้นไม้ตัดสินใจสำหรับข้อมูลทางการตลาด นอกจากนั้นยังมีตัวชี้วัดอื่นๆอีก อาทิเช่น $\mathrm{C-SEP}$ (ทำงานได้ดีกว่าค่าเกนความรู้และค่าดัชนีจีนี่ในบางกรณี) $G-statistic$ (มีประสิทธิภาพในการทำงานใกล้เคียงกับ χ^2)และอีกหนึ่งตัวชี้วัดที่สำคัญ $Minimum\ Description\ Length$ (MDL) ที่ซึ่งจะมีความเอนเอียงน้อยกับแอทริบิวที่มีค่าที่เกิดขึ้นหลายๆค่า (multivalued attribute) MDL จะใช้เทคนิคการเข้ารหัสข้อมูลที่ซึ่งจะสามารถระบุถึงต้นไม้ตัดสินใจที่ดีที่สุดที่ซึ่งจะต้องการปริมาณข้อมูลน้อยในการเข้ารหัสข้อมูลได้ ยิ่งไปกว่านั้นยังมีตัวชี้วัดอื่นๆที่จะทำการพิจารณา/เลือกหลายๆแอทริบิว (Multivariate splits) ในครั้งหนึ่งๆ ที่ซึ่งจะทำการแบ่งข้อมูลออกเป็นชุดข้อมูลย่อยตามกลุ่มของแอทริบิวที่ทำการพิจารณา (นำค่าต่างๆมาประสมกันแล้วจึงแบ่งกลุ่มจากค่าเหล่านั้น)

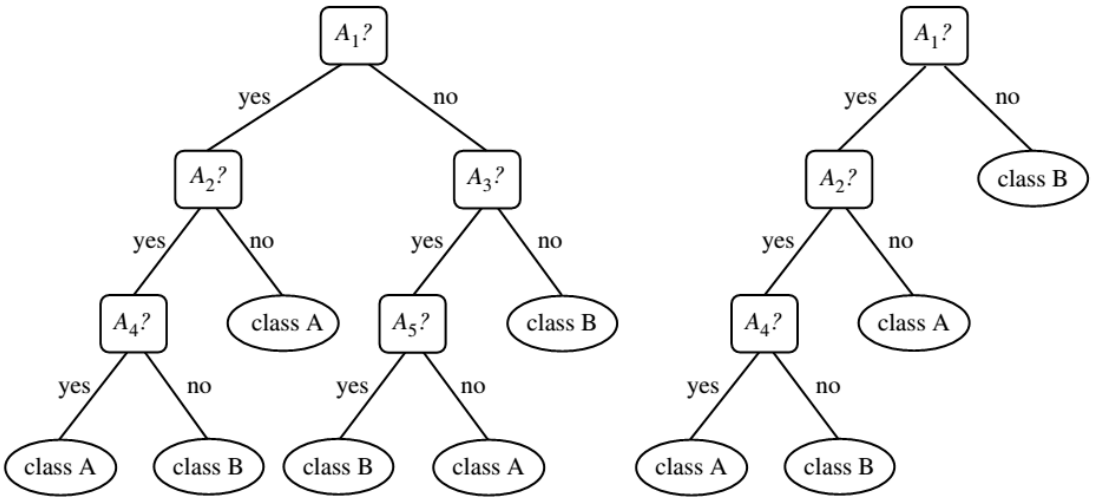
จากที่กล่าวมาทั้งหมดข้างต้น เรายังไม่สามารถสรุปได้ว่าตัวชี้วัดการเลือกแอทริบิวใดที่สามารถทำงานได้ดีที่สุด โดยการที่จะวัดว่าตัวชี้วัดใดทำงานดีหรือไม่จะวัดจาก 1) เปอร์เซ็นต์ความถูกต้อง 2) เวลาประมวลผล

ที่เพิ่มขึ้นเมื่อความสูงของต้นไม้เพิ่มขึ้นซึ่งการการวัดโดยเงื่อนไขต่างๆ จะยังไม่มีตัวชี้วัดใดที่สามารถทำงานได้ดีในทุกๆกรณี ที่ซึ่งแต่ละตัวชี้วัดจะมีบางกรณีที่ตัวชี้วัดนั้นๆสามารถทำงานได้ดีและก็จะมีบางกรณีที่ไม่สามารถทำงานได้ดีด้วยเช่นกัน

6.3.3 การตัดเล็มต้นไม้ตัดสินใจ (Tree pruning)

ในระหว่างการสร้างต้นไม้ตัดสินใจ เราจะสามารถสังเกตได้ว่ามีหลายกิ่งในต้นไม้ที่เป็นสิ่งผิดปกติในชุดข้อมูลสอนอันเนื่องมากจากข้อมูลที่ผิดปกติ ด้วยเหตุนี้ เราจึงจำเป็นที่จะต้องพยายามที่จะตัดเล็มต้นไม้เพื่อไม่ให้เกิดปัญหาเกี่ยวกับ ‘overfitting’ (ซึ่งเป็นเหตุการณ์ที่ตัวจำแนกข้อมูลสามารถเรียนรู้ได้ดีมากกับชุดข้อมูลสอนแต่มีความสามารถต่ำในการจำแนกข้อมูลที่ไม่เคยมาก่อน) โดยในการตัดเล็มต้นไม้จะใช้มาตรวัดเชิงสถิติมาตรวจวัดแต่ละกิ่งของต้นไม้ และทำการตัดเล็มกิ่งที่มีความน่าเชื่อถือต่ำที่สุดออกจากต้นไม้

ดังแสดงในรูปที่ 6.6 ที่ซึ่งต้นไม้ในฝั่งซ้ายจะเป็นต้นไม้ที่ยังไม่ได้ทำการจัดเล็ม ในขณะที่ต้นไม้ทางฝั่งขวาจะเป็นต้นไม้ที่ถูกตัดเล็มแล้ว ที่ซึ่งมักจะมีขนาดเล็กและความซับซ้อนน้อยกว่า รวมถึงมีความง่ายกว่าในการทำความเข้าใจในการจำแนกข้อมูลจากต้นไม้นั้นๆ นอกจากนั้น ต้นไม้ที่ถูกตัดเล็มแล้วจะสามารถจำแนกข้อมูลได้อย่างรวดเร็วยิ่งขึ้นและยังให้ผลของการจำแนกข้อมูลที่ดียิ่งขึ้น โดยจะไม่ขึ้นกับชุดข้อมูลการสอนมากจนเกินไป



รูปที่ 6.6 ตัวอย่างต้นไม้ตัดสินใจก่อนและหลังการตัดเล็ม

ในการตัดเล็มต้นไม้จะประกอบไปด้วยวิธีการตัดเล็ม 2 วิธีการด้วยกันคือ

- **Prepruning**—จะเป็นขั้นตอนการตัดเล็มกิ่งของต้นไม้ในขณะที่ทำการสร้างต้นไม้ โดยจะทำการตัดสินใจว่าจะทำการสร้างกิ่งของต้นไม้ต่อไปหรือไม่ (ทำการตัดสินใจว่าจะทำการแบ่งข้อมูลเป็นชุดข้อมูลย่อย/แตกกิ่งออกตามค่าที่เกิดขึ้นในชุดข้อมูลสอนหรือไม่) โดยในกรณีเกิดการตัดสินใจที่จะหยุดสร้างต้นไว้ โหนดที่กำลังพิจารณาอยู่จะเปลี่ยนสภาพเป็นโหนดใบ และเราจะทำการระบุถึงหมวดหมู่ด้วยการค้นหาหมวดหมู่ที่มีเรคคอร์ดที่สังกัดหมวดหมู่นั้นๆมากที่สุด โดยในระหว่างการสร้างต้นไม้

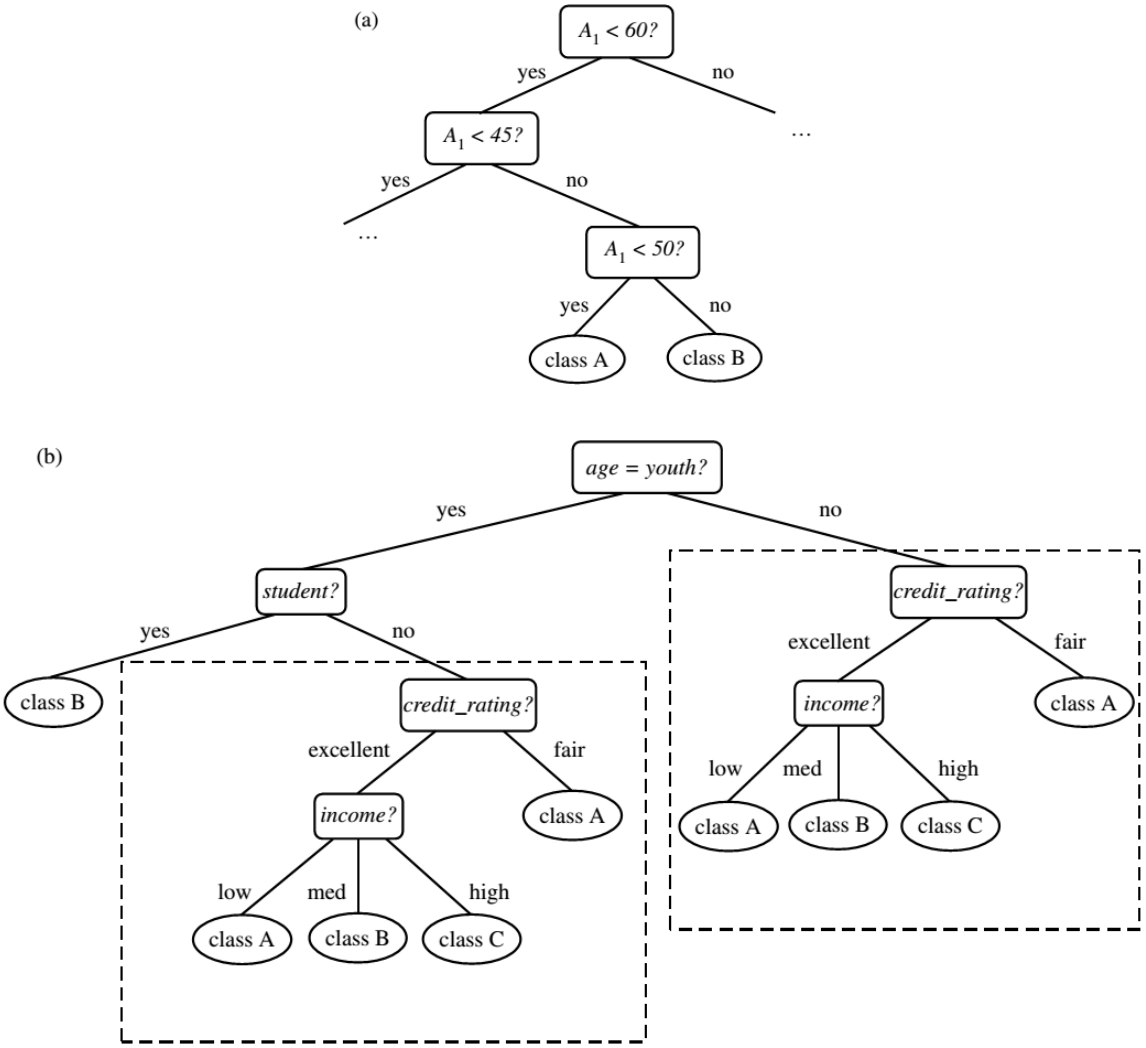
ตัวชี้วัดต่างๆ อาทิเช่น ค่าเกนความรู้ ค่าดัชนีจีนี่ และอื่นๆ จะถูกใช้ในการประเมินค่าความน่าจะเป็นในการแบ่งข้อมูลออกเป็นชุดย่อยๆ ที่ซึ่ง ณ ที่โหนดใดๆก็ตาม ถ้าทำการแบ่งข้อมูลออกเป็นชุดย่อยๆ แล้ว ทำให้ชุดข้อมูลย่อยมีค่าตามตัวชี้วัดน้อยกว่าค่าขีดแบ่งที่กำหนดไว้ก่อนหน้า (prespecified thresholod) เราจะทำการหยุดการแบ่งชุดข้อมูล แต่อย่างไรก็ตาม การที่จะสามารถกำหนดค่าขีดแบ่งให้มีค่าที่เหมาะสมสามารถทำได้ค่อนข้างยาก ถ้าค่าขีดแบ่งถูกกำหนดให้มีค่าสูงจะทำให้ต้นไม้ทำการแจกแนกข้อมูลอย่างง่ายๆโดยไม่สนใจความเป็นจริงที่เกิดขึ้น (oversimplified tree) แต่สำหรับกรณีที่ค่าขีดแบ่งกำหนดให้มีค่าน้อยจะทำให้ต้นไม้มีความซับซ้อนในการจำแนกข้อมูลสูง

- **Postprunning**—จะเป็นขั้นตอนการตัดเล็มกิ่งของต้นไม้หลังจากที่ทำการสร้างต้นไม้ตัดสินใจเสร็จสิ้น (หลังจากต้นไม้โตเต็มที่แล้ว) เมื่อทำการพิจารณาโหนดใดๆก็ตามพร้อมกับกิ่งย่อยที่แตกออกจากโหนดนั้นๆ กระบวนการตัดเล็มต้นไม้จะทำการตัดเล็มกิ่งย่อยจากโหนดที่พิจารณาด้วยการลบกิ่งย่อยออกทั้งหมดแล้วปรับให้กิ่งที่ทำการพิจารณาเปลี่ยนเป็นโหนดใบ และจะทำการระบุถึงหมวดหมู่ด้วยการค้นหาหมวดหมู่ที่มีเรคคอร์ดที่สังกัดหมวดหมู่นั้นๆมากที่สุด เพื่อให้มีความเข้าใจมากขึ้นลองพิจารณาโหนด “A_3” ในรูปต้นไม้ทางด้านซ้ายในรูปที่ 6.6 ถ้าโหนด “A_3” เป็นโหนดที่ต้องถูกตัดเล็มและมีข้อมูลที่อยู่ในหมวดหมู่ ‘B’ เป็นจำนวนมาก ดังนั้น กิ่งย่อยของโหนด “A_3” จะถูกตัดเล็มออกทั้งหมดและโหนด “A_3” จะถูกเปลี่ยนเป็นโหนดใบที่มีหมวดหมู่เป็นหมวดหมู่ ‘B’ ตามลำดับ

จากการตัดเล็มต้นไม้ทั้งสองวิธีข้างต้น ก่อนที่จะทำการประยุกต์ใช้วิธีการใดวิธีหนึ่ง เราควรที่จะต้องทำการพิจารณาถึงค่าความซับซ้อนของวิธีการตัดเล็มต้นไม้เสียก่อน โดยในการพิจารณาค่าความซับซ้อนจะพิจารณาอัตราการผิดพลาดของการจำแนกข้อมูลของต้นไม้ (error rate) ที่ซึ่งถูกคำนวนอยู่ในรูปของเปอร์เซ็นต์ โดยในการพิจารณาที่โหนด N ใดๆ เราจะทำการคำนวณค่าความซับซ้อนของกิ่งย่อยๆต่างๆของโหนด N และค่าความซับซ้อนของโหนด N แล้วนำมาเปรียบเทียบกัน ถ้าโหนด N มีค่าความซับซ้อนน้อยกว่าเราจะทำการตัดเล็มกิ่งย่อยของข้อมูล แต่ถ้าโหนด N มีค่าความซับซ้อนมากกว่าเราจะไม่ทำการตัดเล็มกิ่งของต้นไม้แต่อย่างใด โดยในการตัดเล็มต้นไม้เราอาจใช้การผสมผสานกันระหว่าง prepruning และ postpruning โดยเมื่อทำการพิจารณาถึงแต่ละขั้นตอนวิธี เราจะสังเกตได้ว่าการตัดเล็มต้นไม้แบบ postpruning จะเสียเวลาในการคำนวณมากกว่า แต่ก็นำมาซึ่งความน่าเชื่อถือที่มากกว่าด้วยเช่นกัน ซึ่ง ณ ปัจจุบัน เรายังไม่สามารถบอกกล่าวได้ว่าวิธีการใดสามารถทำการตัดเล็มต้นไม้ได้ดีกว่ากัน

แม้ว่าต้นไม้ที่ถูกตัดเล็มจะมีแนวโน้มที่จะมีขนาดเล็กกว่าต้นไม้ที่ยังไม่ได้รับการตัดเล็ม แต่ทว่า ต้นไม้ที่ตัดเล็มแล้วอาจยังมีขนาดใหญ่และมีความความซับซ้อนในการจำแนกข้อมูลอยู่ ต้นไม้ตัดสินใจอาจประสบความยากลำบากจาก ‘repetition’ และ ‘replication’ ก็เป็นได้ (ดังแสดงในรูปที่ 6.7(a) และ 6.7(b) ตามลำดับ) โดย repetition จะเกิดขึ้นเมื่อแอทริบิวหนึ่งๆถูกทำการทดสอบซ้ำแล้วซ้ำเล่าเมื่อเราทำการพิจารณากิ่งหนึ่งๆ ตัวอย่างเช่น เมื่อเราทำการพิจารณาเงื่อนไข “age < 60?” แล้วต่อมาเรายังคงทำการพิจารณาเงื่อนไข “age < 45?” และอื่นๆ แต่ในส่วนของ replication จะเป็นเหตุการณ์ที่ต้นไม้มีต้นไม้ย่อยซ้ำกัน ดังแสดงในรูปที่ 6.7(b) เราจะเห็นว่าต้นไม้จะมีต้นไม้ย่อยที่มีลักษณะเหมือนกันทั้งหมด ที่ซึ่งเมื่อเกิด

replication ขึ้นจะทำให้ความถูกต้องแม่นยำของการจำแนกข้อมูลถูกลดทอนลง ดังนั้น เพื่อที่จะทำการแก้ไขปัญหาการเกิด replication เราอาจอาศัยการแตกกิ่งโดยอาศัยการพิจารณาหลายๆแอทริบิว (multivariate split) ที่จะช่วยให้ไม่เกิดการซ้ำซ้อนกันของต้นไม้ย่อยได้



รูปที่ 6.7 ตัวอย่างต้นไม้ (a) เกิด repetition และ (b) เกิด replication

6.4 การจำแนกข้อมูลด้วยวิธีการเบย์เซียน (Baysian Classification)

การจำแนกข้อมูลด้วยวิธีการเบย์เซียนจะเป็นการสร้างตัวจำแนกข้อมูลด้วยการประยุกต์ใช้ค่าทางสถิติที่ซึ่งจะสามารถบ่งบอกถึงความน่าจะเป็นของข้อมูลเรคคอร์ดหนึ่งที่จะอยู่ในหมวดหมู่ของข้อมูลหนึ่งๆ วิธีการจำแนกข้อมูลนี้จะทำการประยุกต์ใช้ทฤษฎีของเบย์ (Bayes' theorem) ที่ซึ่งจะช่วยให้สามารถจำแนกข้อมูลได้อย่างรวดเร็วและมีความแม่นยำสูงอีกด้วย

6.4.1 ทฤษฎีของเบย์

ทฤษฎีของเบย์ถูกคิดค้นขึ้นโดย Thomas Bayes ที่ซึ่ง กำหนดให้ X คือ ข้อมูลเรคคอร์ดหนึ่งๆที่ประกอบไปด้วย n แอทริบิว กำหนดให้ H คือสมมติฐานที่ซึ่งข้อมูลเรคคอร์ด X จะมีหมวดหมู่ของข้อมูลเป็น C โดยในการจำแนกข้อมูล เราจะต้องทำการคำนวณค่าต่างๆ 4 ค่าด้วยกันคือ

- ค่าความน่าจะเป็นของสมมติฐาน H จะเป็นจริง เมื่อทำการพิจารณาข้อมูลเรคคอร์ด X ที่ซึ่งสามารถแทนได้ด้วย $P(H|X)$ ตัวอย่างเช่น ข้อมูลเรคคอร์ด X ที่ประกอบไปด้วย 2 แอทริบิว คือ อายุเท่ากับ 35 และ เงินเดือนเท่ากับ 40,000 บาท โดยสมมติฐาน H จะเป็นสมมติฐานที่นาย X จะทำการซื้อคอมพิวเตอร์ โดย $P(H|X)$ จะสะท้อนถึงความน่าจะเป็นที่นาย X จะทำการซื้อคอมพิวเตอร์เมื่อเราทราบถึงอายุและรายได้ของนาย X
- ค่าความน่าจะเป็นที่ข้อมูลทั้งหมดในชุดข้อมูลจะอยู่ในหมวดหมู่ C หนึ่งๆซึ่งก็คือค่า $P(H)$ ตัวอย่างคือ ค่าความน่าจะเป็นที่ลูกค้ารายหนึ่งจะทำการซื้อคอมพิวเตอร์ โดยที่เราไม่ต้องทำการพิจารณาถึงอายุและรายได้ของลูกค้าคนนั้นๆเลย
- ค่าความน่าจะเป็นที่จะมีเรคคอร์ดที่มีค่าเท่ากับ X เมื่อเราทราบถึงเรคคอร์ดที่อยู่ในหมวดหมู่ C ซึ่งก็คือค่า $P(X|H)$ ตัวอย่างคือ ความน่าจะเป็นที่จะมีลูกค้าที่มีอายุเท่ากับ 35 และ รายได้ 40,000 บาท จากลูกค้าที่ทำการซื้อคอมพิวเตอร์จากลูกค้าที่มีอายุ 35 และมีรายได้ 40,000 บาท ทั้งหมด
- ค่าความน่าจะเป็นที่จะเกิดข้อมูลในเรคคอร์ด X จากเรคคอร์ดทั้งหมด ซึ่งก็คือค่า $P(X)$ ตัวอย่างคือ ค่าความน่าจะเป็นที่จะมีลูกค้าอายุเท่ากับ 35 และ มีเงินเดือนเท่ากับ 40,000 บาท

จากค่าทั้ง 4 ข้างต้น เราจะสามารถประเมินหรือคำนวณหาค่า $P(H)$, $P(X|H)$ และ $P(X)$ ได้จากข้อมูลสอน โดยทฤษฎีของเบย์จะพยายามคำนวณหาค่า $P(H|X)$ จากค่า $P(H)$, $P(X|H)$ และ $P(X)$ ที่ซึ่งสามารถคำนวณได้ดังนี้

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

6.4.2 การจำแนกข้อมูลด้วยเบย์เซียนแบบง่าย (Naïve baysian classification)

การจำแนกข้อมูลด้วยเบย์เซียนแบบง่ายจะประกอบไปด้วยขั้นตอนต่างๆดังนี้

1. กำหนดให้ D คือ ชุดข้อมูลสอนที่ประกอบไปด้วยเซตของเรคคอร์ดของข้อมูลที่มีหมวดหมู่ของข้อมูลแนบอยู่ด้วย โดยข้อมูลแต่ละเรคคอร์ด X จะประกอบไปด้วยค่าในแอทริบิวต่างๆ n แอทริบิว $X = (x_1, x_2, \dots, x_n)$
2. สมมติว่าในชุดข้อมูล D จะมีข้อมูลทั้งสิ้น n หมวดหมู่ $(C_1, C_2, \dots, C_n)$ โดยที่ เมื่อทำการพิจารณาข้อมูลเรคคอร์ด X ตัวจำแนกข้อมูลจะทำการทำนายว่า X จะมีค่าความน่าจะเป็นที่จะมี***หมวดหมู่หนึ่งๆ***โดยจะสามารถคำนวณได้จาก

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}$$

โดยในการคำนวณ เราจะต้องพิจารณาข้อมูลเรคคอร์ด X แล้วทำการหาค่าความน่าจะเป็นกับทุกๆ หมวดหมู่ของข้อมูล จากนั้น เราจะทำการคำนวณหาค่า $P(C_i|X)$ ที่มากที่สุด ที่ซึ่งจะบ่งบอกถึงความจะเป็นที่มากที่สุดที่ข้อมูลเรคคอร์ด X จะมีหมวดหมู่ของข้อมูลเป็น C_i

3. เมื่อทำการพิจารณาค่า $P(X)$ เราจะสังเกตุได้ว่า ค่านี้จะมีค่าคงที่กับทุกๆหมวดหมู่ของข้อมูล ดังนั้น เราอาจจะสามารถละเลยค่านี้ได้ โดยทำการพิจารณาแค่เพียงค่า $P(X|C_i)P(C_i)$ ที่ซึ่งมีค่าสูงที่สุดในบรรดาหมวดหมู่ของข้อมูลทั้งหมด—ค่า $P(C_i)$ จะสามารถคำนวณได้จาก $|C_{i,D}|/D$ (ค่า $|C_{i,D}|$ คือ จำนวนเรคคอร์ดในชุดข้อมูล D ที่มีหมวดหมู่เป็น C_i)
4. ในกรณีที่ชุดข้อมูลมีจำนวนแอทริบิวที่ต้องทำการพิจารณาเป็นจำนวนมาก จะเป็นเหตุให้การคำนวณค่า $P(X|C_i)$ จะใช้เวลาค่อนข้างมาก โดยค่า $P(X|C_i)$ จะสามารถคำนวณได้ดังนี้

$$\begin{aligned} P(X|C_i) &= \prod_{k=1}^{n} P(x_k|C_i) \\ &= P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i) \end{aligned}$$

โดยที่แต่ละ x_k จะหมายถึงค่าที่เกิดขึ้นในแต่ละแอทริบิว A_k จากข้อมูลแต่ละเรคคอร์ด X โดยค่าที่เกิดขึ้นในแต่ละแอทริบิวอาจมีค่าเป็นแบบไม่ต่อเนื่องหรือเป็นแบบต่อเนื่อง โดยจะสามารถแบ่งเงื่อนไขการคำนวณของแต่ละ $P(x_k|C_i)$ ได้ดังนี้

   a. ถ้าแอทริบิว A_k มีค่าเป็นแบบไม่ต่อเนื่อง จะสามารถคำนวณ $P(x_k|C_i)$ ได้จากจำนวนเรคคอร์ดทั้งหมดใน D ที่มีหมวดหมู่เป็น C_i และ มีค่า x_k สำหรับแอทริบิว A_k หารด้วย $|C_{i,D}|$ (จำนวนเรคคอร์ดในชุดข้อมูล D ที่มีหมวดหมู่เป็น C_i)
   b. ถ้าแอทริบิว A_k มีค่าเป็นแบบต่อเนื่อง เราจะต้องทำการคำนวณเพิ่มขึ้น โดยเราจะใช้ “Gaussian distribution” พร้อมกับค่าเฉลี่ย μ และส่วนเบี่ยงเบนมาตราฐาน σ ที่ซึ่งสามารถคำนวนได้ดังนี้

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

ดังนั้น เราจะสามารถค่า $P(x_k|C_i)$ ได้เป็น

$$P(x_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

จากสมการข้างต้น เราจะต้องทำการคำนวณค่า μ_{C_i} ที่ซึ่งหมายถึงค่าเฉลี่ยของค่าที่เกิดขึ้นทั้งหมดในแอทริบิว A_k ที่อยู่ในหมวดหมู่ C_i และต้องทำการคำนวณค่า σ_{C_i} ที่ซึ่งหมายถึงส่วนเบี่ยงเบนมาตราฐานของค่าที่เกิดขึ้นในแอทริบิว σ_{C_i} ตามลำดับ เพื่อที่จะเข้าใจมากขึ้น ลองพิจารณาตัวอย่างดังต่อไปนี้ กำหนดให้ เรคคอร์ด X ที่ทำการพิจารณาจะมีข้อมูลอายุและเงินเดือนเป็น $X = (35, \$40{,}00)$ โดยเรคคอร์ด X จะเป็นข้อมูลที่อยู่ในหมวดหมู่การซื้อคอมพิวเตอร์ (กล่าวคือ “buy_computer = yes”) สมมติว่าค่าในแอริบิวอายุจะเป็นข้อมูลแบบต่อเนื่อง และสมมติว่าจากชุดข้อมูลสอน เราต้องการที่จะค้นหาลูกค้าในชุดข้อมูล D ที่ทำการซื้อคอมพิวเตอร์ โดยจะเป็นลูกค้าที่มีอายุเท่ากับ 38 ± 12 ที่ซึ่งจากการคำนวณเราจะได้ค่า $\mu = 38$ และ $\sigma = 12$ ตามลำดับ จากนั้นเราสามารถคำนวณหาค่าความน่าจะ

เป็น $P(age = 35|buys\_computer = yes)$ สำหรับลูกค้าที่มีอายุ 35 ปีและจะทำการซื้อคอมพิวเตอร์ได้

5. ในการที่จะทำนายหมวดหมู่ของข้อมูลเรคคอร์ด X ใดๆ เราจะต้องทำการคำนวณค่า $P(X|C_i)P(C_i)$ สำหรับทุกๆหมวดหมู่ C_i โดยผลลัพธ์ที่ได้จะเป็นหมวดหมู่ข้อมูลที่มีค่า $P(X|C_i)P(C_i)$ สูงที่สุด

ตัวอย่าง การทำนายหมวดหมู่ของข้อมูลโดยใช้การจำแนกข้อมูลด้วยเบย์เซียนแบบง่ายจากข้อมูลสอนดังตารางที่ 6.1 โดยชุดข้อมูลสอนที่จะประกอบไปด้วยข้อมูลหลัก 4 แอทริบิวที่มีค่าแบบไม่ต่อเนื่อง คือ อายุ รายได้ อาชีพนักศึกษาหรือไม่ และ ข้อมูลประวัติเครดิตทางการเงิน และ 1 แอทริบิวหมวดหมู่ของลูกค้า ที่ประกอบไปด้วย 2 หมวดหมู่ คือ หมวดหมู่ของลูกค้าที่ซื้อคอมพิวเตอร์จากร้านค้า และหมวดหมู่ของลูกค้าที่ไม่ได้ซื้อคอมพิวเตอร์จากร้านค้า โดยในการคำนวณ เราจะกำหนดให้ C_1 หมายถึง หมวดหมู่ของลูกค้าที่ซื้อคอมพิวเตอร์ และ C_1 หมายถึง หมวดหมู่ของลูกค้าที่ไม่ซื้อคอมพิวเตอร์ ตามลำดับ โดยเรคคอร์ดที่เราต้องการที่จะจำแนกข้อมูลจะมีลักษณะดังนี้

$$X = (age = youth, income = medium, student = yes, credit\_rating = fair)$$

ดังนั้นเราต้องการที่จะหาค่า $P(X|C_i)P(C_i)$ ที่สูงที่สุด โดยจากตัวอย่างเราจะสังเกตุได้ว่าจำนวนของหมวดหมู่ข้อมูลที่เป็นไปได้จะมี 2 หมวดหมู่ด้วยกันที่ซึ่งจะสามารถคำนวณหาค่าความน่าจะเป็นของเรคคอร์ดหนึ่งๆที่จะมีหมวดหมู่เป็นซื้อคอมพิวเตอร์และไม่ซื้อคอมพิวเตอร์ได้ดังนี้

$$P(buys\_computer = yes) = 9/14 = 0.643$$

$$P(buys\_computer = no) = 5/14 = 0.357$$

จากนั้นเราต้องทำการคำนวณค่า $P(x_k|C_i)$ สำหรับแต่ละแอทริบิวและแต่ละค่าที่เกิดขึ้นในเรคคอร์ด X ที่ซึ่งสามารถคำนวณได้ดังนี้

$$P(age = youth|buys\_computer = yes) = 2/9 = 0.222$$

$$P(age = youth|buys\_computer = no) = 3/5 = 0.600$$

$$P(income = medium|buys\_computer = yes) = 4/9 = 0.444$$

$$P(income = medium|buys\_computer = no) = 2/5 = 0.400$$

$$P(student = yes|buys\_computer = yes) = 6/9 = 0.667$$

$$P(student = yes|buys\_computer = no) = 1/5 = 0.200$$

$$P(credit\_rating = fair|buys\_computer = yes) = 6/9 = 0.667$$

$$P(credit\_rating = fair|buys\_computer = no) = 2/5 = 0.400$$

จากนั้นเราจะสามารถคำนวณค่า $P(X|C_i)$ ของหมวดหมู่การซื้อคอมพิวเตอร์และไม่ซื้อคอมพิวเตอร์ได้ดังนี้

$$P(X|buys\_computer = \boldsymbol{yes}) = P(age = youth|buys\_computer = yes) \times$$
$$P(income = medium|buys\_computer = yes) \times$$
$$P(student = yes|buys\_computer = yes) \times$$
$$P(credit\_rating = fair|buys\_computer = yes) \times$$
$$= 0.222 \times 0.444 \times 0.667 \times 0.667 = 0.044$$
$$P(X|buys\_computer = \boldsymbol{no}) = P(age = youth|buys\_computer = no) \times$$
$$P(income = medium|buys\_computer = no) \times$$
$$P(student = yes|buys\_computer = no) \times$$
$$P(credit\_rating = fair|buys\_computer = no) \times$$
$$= 0.600 \times 0.400 \times 0.200 \times 0.400 = 0.019$$

ดังนั้นเราจะสามารถหาหมวดหมู่ของข้อมูล C_i ที่ให้ค่า $P(X|C_i)P(C_i)$ ที่สูงที่สุด โดยสามารถคำนวณได้ดังนี้

$$P(X|buys\_computer = \boldsymbol{yes})P(buys\_computer = \boldsymbol{yes}) = 0.044 \times 0.643 = 0.028$$
$$P(X|buys\_computer = \boldsymbol{no})P(buys\_computer = \boldsymbol{no}) = 0.019 \times 0.357 = 0.007$$

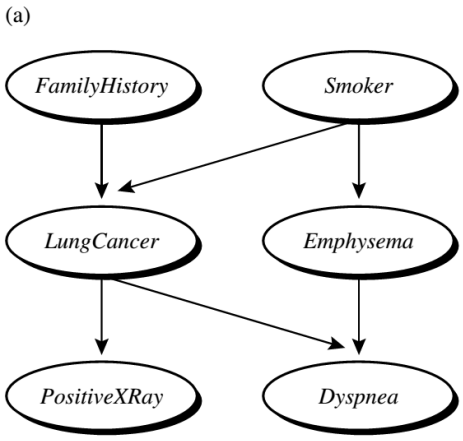
ดังนั้น จากการทำนายโดยใช้การจำแนกข้อมูลด้วยเบย์เซียนแบบง่ายจะให้ผลทำนายว่าข้อมูลเรคคอร์ด X จะอยู่ในหมวดหมู่การซื้อคอมพิวเตอร์ $(buys\_computer = \boldsymbol{yes})$

<u>*ข้อสังเกตุในการคำนวณ!!!*</u> ในการคำนวณหาค่า $P(X|C_i)$ จะเกิดจาก $P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$ ดังนั้น ถ้า $P(x_k|C_i)$ ใดๆมีค่าเป็น 0 ก็จะทำให้ $P(X|C_i)$ มีค่าเป็น 0 ไปด้วย ยิ่งไปกว่านั้นจะทำให้ $P(X|C_i)P(C_i)$ มีค่าเป็น 0 ด้วยเช่นกัน

ในการที่จะแก้ไขปัญหาดังกล่าว เราสามารถประยุกต์ใช้วิธีการประมาณค่าความน่าจะเป็น ที่เรียกว่า “Laplacian correction” หรือ “Laplacian estimation” ที่จะทำการเพิ่มจำนวนเรคคอร์ด 1 เรคคอร์ดให้แต่ละกรณีของแต่ละค่าที่จะเกิดขึ้นในแอทริบิวที่พิจารณา ตัวอย่างเช่น สมมติว่าชุดข้อมูลสอน D มีข้อมูลที่อยู่ในหมวดหมู่การซื้อคอมพิวเตอร์ทั้งสิ้น 1,000 เรคคอร์ด โดยเมื่อพิจารณาที่แอทริบิวรายได้ ปรากฏว่ามี 10 เรคคอร์ดที่มีรายได้สูง มี 990 เรคคอร์ดที่มีรายได้ปานกลางและไม่มีข้อมูลเรคคอร์ดใดเลยที่มีรายได้น้อย (มี 0 เรคคอร์ด ถ้าเราไม่ทำการประยุกต์ใช้ Laplacian correction จะทำให้เราได้ค่าความน่าจะเป็นเท่ากับ 0.010 (10/1000), 0.990 (990/1000) และ 0 ตามลำดับ แต่เมื่อเราประยุกต์ใช้ Laplacian correction ที่ซึ่งจะทำการเพิ่ม 1 เรคคอร์ดใรทุกๆค่าที่เกิดขึ้น ดังนั้น จำนวนเรคคอร์ดที่มีรายได้สูงจะมี 11 เรคคอร์ด รายได้ปานกลางมี 991 เรคคอร์ด และ รายได้น้อยมี 1 เรคคอร์ดตามลำดับ และ เมื่อเราทำการหาค่าความน่าจะเป็นจะมีค่าเท่ากับ 0.011(11/1003), 0.988(991/1003) และ 0.001(1/1003) ตามลำดับ

6.4.3 ข่ายงานความเชื่อเบย์ (Baysian belief networks)

การจำแนกข้อมูลด้วยเบย์เซียนอย่างง่ายจะยึดติดกับสมมติฐานที่ว่า “แอทริบิวต่างๆแต่ละตัวจะไม่ขึ้นต่อกัน” ที่ซึ่งจะทำให้เกิดข้อจำกัดในการทำงาน ด้วยเหตุนี้จึงมีแนวคิดที่จะทำการพัฒนาข่ายงานความเชื่อเบย์ (Baysian belief networks หรือที่เรียกในชื่ออื่นๆว่า “belief networks”, “Bayesian networks” และ “probabilistic networks”) เพื่ออธิบายความขึ้นต่อกันระหว่างเซตของแอทริบิวต่างๆ ด้วยการแสดงออกในรูปแบบกราฟฟิค โดยข่ายงานความเชื่อเบย์จะประกอบไปด้วย 2 ส่วนหลักๆ คือ directed acyclic graph (DAG) และ conditional probability tables (ดังแสดงในรูปที่ 6.8) แต่ละโหนดใน DAG จะแสดงถึงค่าต่างๆที่เกิดขึ้นในแอทริบิวหนึ่งที่ถูกสุ่มขึ้นมา และ แต่ละเส้นเชื่อมระหว่างโหนดจะมีข้อมูลความน่าจะเป็นของการขึ้นแก่กันแนบอยู่ ถ้าโหนด Y และ โหนด Z มีการเชื่อมโยงเข้าหากัน Y จะเป็นโหนดพ่อแม่หรือโหนดบรรพบุรุษของโหนด Z และ Z จะเป็นโหนดลูกโหนดหลานของ Y ตามลำดับ



(b)

	FH, S	FH, ~S	~FH, S	~FH, ~S
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

รูปที่ 6.8 ตัวอย่าง Baysian belief network (a) Directed acyclic graph และ (b) Conditional probability table ของแอทริบิว LungCancer ที่มีค่าความน่าจะเป็นแบบมีเงื่อนไขกับแอทริบิว FamilyHistory และ Smoking ตามลำดับ

จากรูปที่ 6.8(a) จะเป็นข่ายงานที่ประกอบไปด้วย 6 แอทริบิว/โหนด ที่มีค่าเป็นแบบ boolean คือ ‘yes’ หรือ ‘no’ โดยเส้นเชื่อมระหว่างโหนดจะแสดงถึงความขึ้นแก่กันของแอทริบิวต่างๆ ตัวอย่างเช่น คนที่เป็นโรคมะเร็งปอด (LungCancer, LC) อาจมีสาเหตุมาจากพันธุกรรมทางบรรพบุรุษที่เคยเป็นโรคนี้ (FamilyHistory, FH) หรือพฤติกรรมการสูบบุหรี่ของบุคคลนั้นๆ (Smoking, S) เป็นต้น ในขณะที่ผลการเอ็กซ์เรย์ที่เป็นบวก (PositiveXRay) จะไม่ขึ้นกับสาเหตุทางพันธุกรรมทางบรรพบุรุษที่เคยเป็นโรคนี้ หรือพฤติกรรมการสูบบุหรี่ของบุคคลนั้นๆ นอกจากนั้นเส้นเชื่อมระหว่างโหนดยังแสดงให้เห็นว่าแอทริบิวโรคมะเร็งปอดไม่ได้ขึ้นกับอวัยวะภายใน (Emphysema) เมื่อเราทราบถึงโหนดพ่อแม่ของโหนดทั้งสอง

ในการพิจารณาข่ายงานความเชื่อเบย์เราจะต้องทำการสร้างตารางความน่าจะเป็นแบบมีเงื่อนไข (conditional probability table, CPT) สำหรับแต่ละแอทริบิวที่เราทำการพิจารณา ที่ซึ่งจะเป็นการแสดงความขึ้นแก่กันระหว่างแอทริบิว Y กับแอทริบิวที่เกี่ยวเนื่องกับแอทริบิว Y ที่ซึ่งเราอาจหมายถึงแอทริบิวที่เป็น

โหนดพ่อแม่ของ Y โดยเราจะต้องทำการค้นหาค่าความน่าจะเป็นแบบมีเงื่อนไขของ Y และโหนดพ่อแม่ของ Y ที่ซึ่งสามารถคำนวณได้จาก $P(Y|Parents(Y))$ โดยจากรูป 6.8(b) จะแสดงถึงตารางความน่าจะเป็นแบบมีเงื่อนไขสำหรับแอทริบิว LungCancer ที่ซึ่งจะขึ้นกับโหนดพ่อแม่ของแอทริบิว LungCancer ซึ่งก็คือโหนดของ FamilyHistory และ Smoking โดยเราจะต้องพิจารณาถึงความน่าจะเป็นของกรณีที่เป็นไปได้ทั้งหมดของทั้ง 3 แอทริบิว ซึ่งก็คือ 1) LungCancer = 'yes', FamilyHistory = 'yes', Smoking = 'yes', 2) LungCancer = 'no', FamilyHistory = 'yes', Smoking = 'yes', 3) LungCancer = 'yes', FamilyHistory = 'yes', Smoking = 'no', 4) LungCancer = 'no', FamilyHistory = 'yes', Smoking = 'no', 5) LungCancer = 'yes', FamilyHistory = 'no', Smoking = 'yes', 6) LungCancer = 'no', FamilyHistory = 'no', Smoking = 'yes', 7) LungCancer = 'yes', FamilyHistory = 'no', Smoking = 'no' และ 8) LungCancer = 'no', FamilyHistory = 'no', Smoking = 'no' ตามลำดับ โดยเมื่อเราพิจารณาช่องทางซ้ายบนสุดและขวาบนสุดของตารางเราจะสังเกตุได้ว่า

$$P(LungCaner = 'yes'|FamilyHistory = 'yes', Smoker = 'yes') = 0.8$$

$$P(LungCaner = 'no'|FamilyHistory = 'no', Smoker = 'no') = 0.9$$

ดังนั้น เมื่อเราพิจารณาข้อมูลเรคคอร์ด $X = (x_1, x_2, \dots, x_n)$ หนึ่งๆที่ซึ่งถูกอธิบายได้ด้วยแอทริบิว $Y_1, Y_2, \dots, Y_n$ แล้ว เราจะต้องทำการแยกแยะระหว่างแอทริบิวที่ไม่ขึ้นแก่กัน โดยการสังเกตเส้นเชื่อมในกราฟว่ามีการเชื่อมโยงกันหรือไม่ จากนั้นเราจะต้องทำการคำนวณหาค่าความน่าจะเป็นแบบมีเงื่อนไขของค่า $x_1, x_2, \dots, x_n$ ที่ปรากฏในเรคคอร์ด X ที่ซึ่งสามารถคำนวณได้ดังนี้

$$P(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} P(x_i|Parents(Y_i))$$

เมื่อ $P(x_1, x_2, \dots, x_n)$ คือ ความน่าจะเป็นของการพิจารณาทุกๆค่า x_i ที่ปรากฏขึ้นในเรคคอร์ด X และ ค่า $P(x_i|Parents(Y_i))$ จะเป็นค่าความน่าจะเป็นแบบมีเงื่อนไขของค่า x_i กับแอทริบิวที่เป็นพ่อแม่ของแอทริบิวที่ทำการพิจารณา

โหนดหนึ่งๆในเครื่องข่ายที่ทำการพิจารณาอาจถูกเลือกเพื่อแสดงผล ที่ซึ่งจะแสดงถึงหมวดหมู่ของข้อมูล โดยในการเลือกโหนดที่แสดงผลเราอาจทำการเลือกได้มากกว่าหนึ่งโหนดก็เป็นได้

6.5 การจำแนกข้อมูลด้วยกฎ (Rule-Based Classification)

การจำแนกข้อมูลด้วยกฎจะเป็นโมเดลที่จะแสดงผลด้วยเซตของกฎที่มีลักษณะแบบ 'IF-THEN' โดยกฎหนึ่งๆจะถูกแสดงอยู่ในรูปฟอร์ม ดังนี้

IF condition **THEN** conclusion

ตัวอย่างเช่น

R1: **IF** age = youth AND student = yes **THEN** buys_computer = yes

จากรูปแบบฟอร์มของกฎและตัวอย่างกฎ R1 ข้างต้น จะประกอบไปด้วยข้อมูลสองส่วนด้วยกันคือ ส่วนแรก ส่วนเงื่อนไข 'IF' จะเป็นส่วนที่เรียกว่า 'rule antecedent' หรือ 'precondition' ประกอบไปด้วยเซตของแอทริบิวต่างๆประกอบกันเป็นเงื่อนไข ซึ่งจากกฎ R1 จะประกอบไปด้วยแอทริบิวที่บ่งบอกถึงคุณลักษณะของคน 2 แอทริบิวด้วยกัน คือ อายุอยู่ในช่วงหนุ่มสาว (age = youth) และเป็นนักเรียน (student = yes) โดยส่วนเงื่อนไขของกฎจะทำการเชื่อมโยงแอทริบิวต่างๆเข้าด้วยกันด้วยเครื่องหมาย AND ที่ซึ่งจะเป็นการบ่งบอกว่าข้อมูลจะต้องเป็นไปตามเงื่อนไขที่ถูกกำหนดไว้ทั้งหมด ในส่วนที่สอง จะเรียกว่า 'rule consequence' ที่จะมีหมวดหมู่ข้อมูลบรรจุอยู่ จากตัวอย่างกฎ R1 ข้างต้น เราจะทำการทำนายว่าลูกค้าที่มีคุณลักษณะเช่นไรที่จะทำการซื้อคอมพิวเตอร์ เป็นต้น

ในการเขียนกฎ เราสามารถเขียนกฎให้อยู่ในอีกรูปแบบหนึ่งที่ใช้เครื่องหมายคณิตศาสตร์ได้เป็น

R1: (age = youth) $\wedge$ (student = yes) $\Longrightarrow$ (buys_computer = yes)

โดยจากกฎข้างต้น ถ้ามีข้อมูลเรคคอร์ด X หนึ่งๆเป็นไปตามเงื่อนไขในส่วนของ rule antecedent เราจะสามารถบอกได้ว่ากฎ R1 คลอบคลุม (Covers) ข้อมูลเรคคอร์ด X หรือ เรคคอร์ด X ถูกคลอบคลุมด้วยกฎ R1 ดังนั้นเมื่อในชุดข้อมูลประกอบไปด้วยข้อมูลหลายเรคคอร์ด เราจะสามารถหาเปอร์เซ็นต์ของเรคคอร์ดทั้งหมดในชุดข้อมูลที่ถูกครอบคลุมโดยกฎ R1 ได้ ที่ซึ่งเราจะเรียกค่าเปอร์เซ็นต์ของการคลอบคลุมนั้นว่า 'Coverage' ที่ซึ่งสามารถคำนวณได้ ดังนี้

$$coverage\ (R) = \frac{n_{covers}}{|D|}$$

เมื่อ n_{covers} คือ จำนวนเรคคอร์ดในชุดข้อมูลที่ถูกคลอบคลุมโดยกฎ R และ $|D|$ คือ จำนวนเรคคอร์ดของข้อมูลทั้งหมดในชุดข้อมูล

ค่า coverage ของกฎจะถูกใช้ในการประเมินประสิทธิภาพการจำแนกข้อมูลของกฎว่ากฎหนึ่งๆที่สร้างขึ้นมีประสิทธิภาพอย่างไร และมีความถูกต้องของการจำแนกข้อมูล (การบ่งบอกถึงหมวดหมู่ข้อมูล) เป็นอย่างไร โดยในการวัดความถูกต้องในการจำแนกข้อมูลของกฎ (Accuracy) จะสามารถคำนวณได้ ดังนี้

$$accuracy\ (R) = \frac{n_{correct}}{n_{covers}}$$

เมื่อ $n_{correct}$ คือ จำนวนเรคคอร์ดที่ถูกจำแนกได้อย่างถูกต้องภายใต้กฎ R

ตัวอย่าง ลองพิจารณาข้อมูลในตาราง 6.1 ที่ประกอบไปด้วยข้อมูลคุณลักษณะของลูกค้าที่ทำการซื้อและไม่ซื้อคอมพิวเตอร์จากร้านจำหน่ายอุปกรณ์ไฟฟ้า งานของเราคือการทำนายว่าลูกค้ารายใดจะทำการซื้อคอมพิวเตอร์บ้าง ถ้าเราใช้กฎ R1 ข้างต้นในการทำนาย เราจะสามารถหาค่า coverage และ accuracy ของกฎ R1 เทียบกับข้อมูลเรคคอร์ดต่างๆในตาราง 6.1 ได้ดังนี้

$$coverage\ (R1) = \frac{2}{14} = 14.28\%$$

$$accuracy\ (R1) = \frac{2}{2} \quad = 100\%$$

เมื่อลองพิจารณาข้อมูลจากตาราง 6.1 เราจะเห็นว่ามีข้อมูลทั้งสิ้น 2 เรคคอร์ดจากทั้งหมด 14 เรคคอร์ดที่มี 'rule antecedent' เหมือนกับกฎ R1 ดังนั้นเราจะสามารถคำนวณค่า coverage ของกฎ R1 ได้เท่ากับ 14.28% และเมื่อลองพิจารณาข้อมูลอีกครั้งหนึ่ง เราจะเห็นว่ามีข้อมูลทั้งสิ้น 2 เรคคอร์ด ที่มี 'rule antecedent' และ 'rule consequent' เหมือนกับกฎ R1 จากเรคคอร์ดทั้งหมดที่ถูกคลอบคลุมด้วยกฎ R1 ดังนั้นเราจะสามารถคำนวณค่า accuracy ของกฎ R1 ได้เท่ากับ 100% ตามลำดับ

ในขั้นตอนนี้เราจะทำการศึกษาเกี่ยวกับการประยุกต์ใช้การจำแนกข้อมูลด้วยกฎเพื่อทำนายหมวดหมู่ข้อมูลของเรคคอร์ด X ใดๆ ถ้ามีกฎใดกฎหนึ่งที่สอดคล้องกับข้อมูลเรคคอร์ด X เราจะเรียกกฎนั้นว่า 'triggered' ตัวอย่างเช่น ถ้าเรากำหนดให้เรคคอร์ด X มีค่าดังนี้

$$X = (age = youth, income = medium, student = yes, credit_{rating} = fair)$$

เราต้องการที่จะระบุ/จำแนกหมวดหมู่ของเรคคอร์ด ตามหมวดหมู่การซื้อคอมพิวเตอร์ (buy_computer) เนื่องจาก X สอดคล้องกับกฎ $R1$ ถ้ากฎ $R1$ เป็นเพียงกฎเดียวที่สอดคล้องกับข้อมูลเรคคอร์ด X เราจะคืนค่าหมวดหมู่ของกฎ $R1$ ให้เป็นหมวดหมู่ของเรคคอร์ด X แต่ถ้ามีกฎมากกว่าหนึ่งที่สอดคล้องกับ X จะทำให้เกิดปัญหาเกิดขึ้นถ้ากฎเหล่านั้นมีหมวดหมู่ไม่เหมือนกัน หรืออีกกรณีหนึ่งคือไม่มีกฎที่สอดคล้องกับ X เลยจะทำให้เกิดปัญหาเช่นกัน

ในการที่จะแก้ไขปัญหาแรก เราจะต้องพิจารณาถึงกลยุทธ์ในการแก้ไขปัญหาความขัดแย้งกัน (ของกฎและหมวดหมู่ที่แตกต่างกัน) เพื่อที่จะบ่งชี้ได้ว่ากฎใดเหมาะสม/สอดคล้องกับ X มากที่สุดและจะนำหมวดหมู่ของกฎนั้นไปเป็นคำตอบหมวดหมู่สำหรับ X โดยกลยุทธ์แยกความขัดแย้งจะมีอยู่หลายกลยุทธ์ด้วยกันแต่จะมีอยู่ 2 กลยุทธ์ที่นิยมใช้กันในวงกว้างนั่นคือ 'size ordering' และ 'rule ordering' โดยวิธีการแบบ 'size ordering' จะทำการกำหนดความสำคัญที่สูงที่สุดให้กับกฎที่มีจำนวนแอทริบิวในฝั่งซ้ายมากที่สุดที่สอดกับ X โดยจะทำการเลือกหมวดหมู่ของกฎนั้นๆมาเป็นคำตอบสำหรับหมวดหมู่ของ X แต่สำหรับวิธีการแบบ 'rule ordering' จะทำการกำหนดความสำคัญให้กับกฎก่อนหน้าที่จะเปรียบเทียบกับ X โดยในการกำหนดความสำคัญ/เรียงลำดับกฎอาจจะเรียงลำดับตามความสำคัญของหมวดหมู่หรือตามกฎก็เป็นได้ โดยในการเรียงลำดับตามหมวดหมู่ กฎต่างๆจะถูกเรียงลำดับตามหมวดหมู่ โดยหมวดหมู่ที่มีความสำคัญมากจะอยู่ในลำดับต้นๆไล่เลียงมาตามลำดับความสำคัญ (โดยความสำคัญของแต่ละหมวดหมู่อาจหาได้จากการตอบหรือ

จำแนกข้อมูลผิดเมื่อทำการเปรียบเทียบกับชุดข้อมูลทดสอบก็เป็นได้) โดยในการเรียงลำดับกฎต่างๆ กฎที่มีหมวดหมู่เหมือนกันจะไม่ต้องทำการเรียงลำดับ เนื่องจากกฎเหล่านั้นอยู่ในหมวดหมู่เดียวกันและจะไม่ทำให้เกิดความขัดแย้งเกิดขึ้น แต่สำหรับการเรียงลำดับตามกฎจะเป็นการเรียงลำดับตามความสำคัญของแต่ละกฎที่ซึ่งจะสามารถวัดได้จากค่าความถูกต้อง (accuracy) ค่าเปอร์เซ็นต์ของการคลอบคลุม (coverage) จำนวนแอทริบิวที่อยู่ทางฝั่งซ้ายของกฎ หรือ อาจได้รับค่าความสำคัญจากผู้เชี่ยวชาญก็เป็นได้ เมื่อทำการเรียงกฎเสร็จสิ้นแล้ว กฎใดก็ตามที่มีค่าความสำคัญสูงที่สุดจะถูกเลือกไปเป็นคำตอบสำหรับหมวดหมู่ของเรคคอร์ด X โดยกฎอื่นๆจะถูกตัดทิ้งออกจากการพิจารณา

ในการที่จะแก้ไขปัญหาที่สอง ซึ่งก็คือไม่มีกฎใดเลยที่สอดคล้องกับข้อมูลเรคคอร์ด X จะสามารถแก้ไขด้วยการกำหนดกฎที่เป็นปริยาย (default rule) ที่ซึ่งจะมีหมวดหมู่โดยปริยาย (default class) แนบอยู่ด้วย โดยหมวดหมู่ปริยายนี้อาจจะมาจากผู้เชี่ยวชาญ หรืออาจจะมาจากหมวดหมู่ส่วนใหญ่ของข้อมูลสอนหรือเป็นหมวดหมู่โดยส่วนมากที่ไม่สอดคล้องกับกฎใดๆเลย

6.5.1 การสกัดกฎจากต้นไม้ตัดสินใจ

โดยส่วนใหญ่ของการสร้างต้นไม้ตัดสินใจ ต้นไม้ที่ถูกสร้างขึ้นจะมีขนาดค่อนข้างใหญ่ ซึ่งอาจทำให้เกิดความยุ่งยากในการตีความ ด้วยเหตุนี้ การสกัดกฎจากต้นไม้ตัดสินใจถูกออกแบบเพื่อทำการสร้างกฎจากต้นไม้ตัดสินใจที่อยู่ในรูปแบบ ‘IF-THEN’ ที่ทำให้มนุษย์สามารถข้าใจได้ง่าย โดยจะทำการสร้างกฎหนึ่งๆจากแต่ละกิ่งของต้นไม้เริ่มจากโหนดรากไปจนถึงโหนดใบ โดยแอทริบิวต่างๆที่อยู่ในกิ่งจะอยู่ในส่วนของ ‘IF’ และโหนดใบจะอยู่ในส่วนของ ‘THEN’ ตามลำดับ

ตัวอย่าง การสกัดกฎจากต้นไม้ตัดสินใจที่แสดงในรูปที่ 6.2 ที่ซึ่งสามารถสกัดออกเป็นกฎที่อยู่ในรูปแบบของ ‘IF-THEN’ ด้วยการท่องไปตามกิ่งต่างๆจากโหนดรากไปยังโหนดใบที่ซึ่งจะได้กฎต่างๆดังนี้

R1: **IF** age=youth	AND student=no	**THEN** buy_computer=no
R2: **IF** age=youth	AND student=yes	**THEN** buy_computer=yes
R3: **IF** age=middle_aged		**THEN** buy_computer=yes
R4: **IF** age=senior	AND credit_rating=excellent	**THEN** buy_computer=yes
R5: **IF** age=senior	AND credit_rating=fair	**THEN** buy_computer=no

จากกฎที่ได้จากต้นไม้ตัดสินใจ เราจะสังเกตได้ว่าแต่ละกิ่งจะหมายถึงหนึ่งกฎซึ่งในต้นไม้ตัดสินใจจะมีกิ่งสำหรับทุกกรณีที่เกิดขึ้น ด้วยเหตุนี้จะทำให้ไม่มีการขัดแย้งกันเมื่อต้องทำนายหมวดหมู่ของข้อมูลใดๆก็ตาม ดังนั้นในการสร้างกฎจากต้นไม้ตัดสินใจ เราจะไม่ต้องจัดเตรียมกฎโดยปริยายและหมวดหมู่โดยปริยายแต่อย่างใด และยังไม่ต้องทำการเรียงลำดับกฎแต่อย่างใดอีกด้วย แต่อย่างไรก็ตาม ด้วยการที่ต้นไม้ตัดสินใจมีขนาดใหญ่และอาจมีต้นไม้ย่อยที่ซ้ำกัน (ดังแสดงในรูปที่ 6.7) ที่ซึ่งจะทำให้กฎที่สกัดได้จะมีปริมาณมากและอาจมีกฎ

ที่ไม่เกี่ยวข้องรวมถึงมีกฎที่ซ้ำซ้อนกันอยู่เป็นจำนวนมาก ด้วยเหตุนี้เราอาจจำเป็นที่จะต้องทำการตัดเล็มกฎเพื่อให้กฎที่น้อยลงและเป็นกฎที่ไม่ซ้ำซ้อนกัน

6.5.2 การสร้างกฎเพื่อจำแนกข้อมูลด้วยอัลกอริทึม Sequential covering

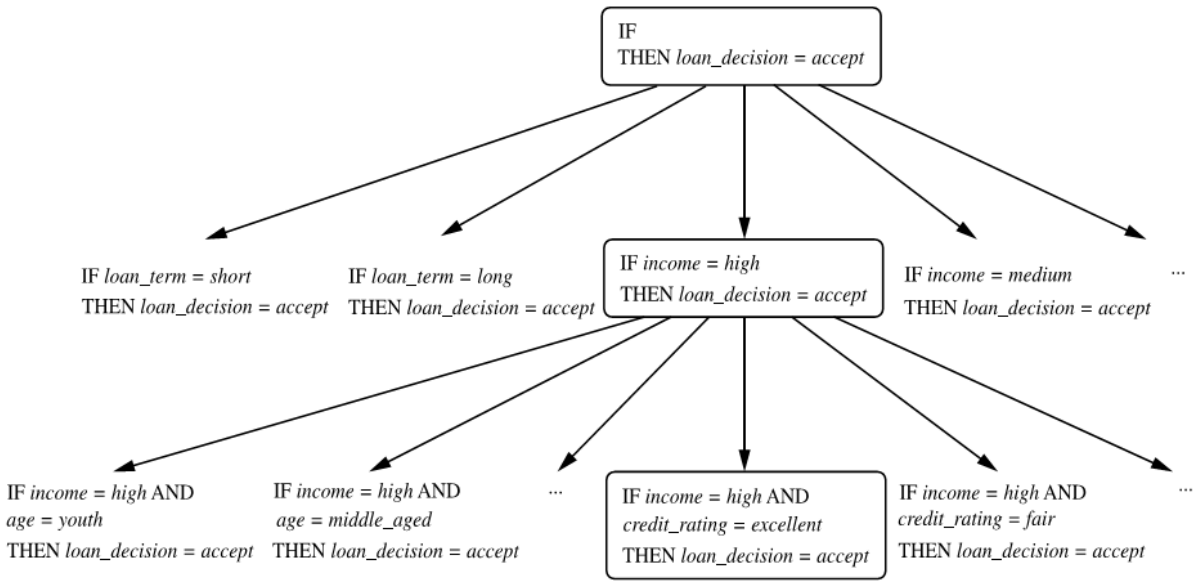
ในการที่จะสร้างกฎเพื่อจำแนกข้อมูลที่อยู่ในรูปแบบของ 'IF-THEN' โดยตรงจากชุดข้อมูลสอน (โดยไม่ทำการสร้างต้นไม้ตัดสินใจ) เราจะสามารถทำได้โดยการใช้อัลกอริทึม 'sequential covering' ที่ซึ่งจะเกิดจากแนวคิดที่จะทำการเรียนรู้หรือสร้างกฎแบบมีลำดับ โดยกฎที่สร้างได้จะเป็นกฎที่คลอบคลุมข้อมูลบางเรคคอร์ดในชุดข้อมูล

รูปที่ 6.9 จะแสดงรายละเอียดการทำงานของอัลกอริทึม 'sequential covering' ที่ซึ่งจะทำการพิจารณาทีละหมวดหมู่ข้อมูลแล้วทำการเรียนรู้/สร้างกฎต่างๆจากหมวดหมู่นั้นๆ โดยที่ เมื่อเราพิจารณาหมวดหมู่ C_i เราจะต้องการที่จะสร้างกฎที่สามารถคลอบคลุมข้อมูลเรคคอร์ดทั้งหมด (หรือส่วนใหญ่) ในชุดข้อมูลสอนที่มีหมวดหมู่ของข้อมูลเป็นหมวดหมู่ C_i โดยกฎที่ทำการสร้างขึ้นควรจะเป็นกฎที่มีค่าความถูกต้องสูง แต่ไม่จำเป็นต้องมีค่าเปอร์เซ็นต์ของการคลอบคลุมสูง เนื่องจากในการพิจารณาหมวดหมู่หนึ่งๆของข้อมูลเราสามารถสร้างกฎได้มากกว่าหนึ่งกฎ โดยกฎที่แตกต่างกันอาจคลอบคลุมข้อมูลเรคคอร์ดที่แตกต่างกันก็เป็นได้ โดยการสร้างกฎสำหรับหมวดหมู่ C_i ใดๆ จะเสร็จสิ้นก็ต่อเมื่อ 1) ไม่มีเรคคอร์ดของข้อมูลที่มีหมวดหมู่ C_i ให้พิจารณา หรือ 2) ไม่สามารถหากฎที่มีค่าความถูกต้องที่มากกว่าค่าขีดแบ่งที่ผู้ใช้กำหนด โดยในการหากฎแต่ละกฎเราจะประยุกต์ใช้ฟังก์ชัน Learn_One_Rule ที่ซึ่งจะทำการหา "กฎที่ดีที่สุด" สำหรับหมวดหมู่ที่พิจารณาจากชุดข้อมูลเรคคอร์ดในหมวดหมู่ C_i ณ ขณะนั้น

อัลกอริทึม การเรียนรู้กฎ IF-THEN จากชุดข้อมูลสอน (Sequential covering)
อินพุต - ชุดข้อมูลสำหรับเรียนรู้ (D) ที่ประกอบไปด้วยเรคคอร์ดต่างๆแนบด้วยหมวดหมู่ข้อมูล
- ลิสต์ของแอทริบิว (attr_vals) ที่อธิบายถึงคุณลักษณะต่างๆของข้อมูล
เอาท์พุต เซตของกฎที่เป็นแบบ IF-THEN
(1) กำหนดให้เซตของกฎ Rule_set = {}
(2) **for** แต่ละหมวดหมู่ c **do**
(3) **repeat**
(4) Rule = **Learn_One_rule**(D, attr_vals, c)
(5) ลบเรคคอร์ดที่ถูกครอบคลุมด้วยกฎ Rule ออกจากชุดข้อมูล D
(6) **Until** เงื่อนไขการหยุด
(7) เพิ่มกฎ Rule เข้าไปใน Rule_set
(8) **return** Rule_set

รูปที่ 6.9 อัลกอริทึม sequential covering พื้นฐาน

ในการค้นหากฎจะเริ่มจากการพิจารณาแอทริบิวทีละแอทริบิวแล้วค่อยเพิ่มการพิจารณาทีละหนึ่งไปเรื่อยๆ (ซึ่งวิธีการนี้จะเรียกว่า ‘general-to-specific’) ดังแสดงในตัวอย่างในรูปที่ 6.10 ที่จะแสดงถึงชุดข้อมูลสำหรับการกู้ยืมเงินที่ซึ่งจะประกอบไปด้วยข้อมูลเอกสารของผู้ที่ยื่นเรื่องทำการกู้-ยืมเงินแต่ละราย โดยข้อมูลแอทริบิวต่างๆที่จะทำการพิจารณาจะประกอบด้วย อายุ รายได้ การศึกษา ที่อยู่ เครดิต ระยะเวลาในการกู้ยืม และ ผลลัพธ์ของการกู้ยืมตามลำดับ (age, income, education level, residence, credit rating, loan term และ loan_decision ที่ซึ่งเป็นหมวดหมู่ของข้อมูล) โดยในการจำแนกข้อมูลเราจะทำการจำแนกว่าเอกสารใดที่จะได้รับการอนุมัติหรือไม่ได้รับการอนุมัติเงินกู้บ้าง ซึ่งจะมีหมวดหมู่ของข้อมูลเป็น ‘accept’ และ ‘non-accept’ โดยในการพิจารณาจะเริ่มจากการที่พิจารณาหมวดหมู่ของข้อมูล ที่ซึ่งจากรูปเราจะเห็นว่าเราจะทำการพิจารณาหมวดหมู่ที่ได้รับการอนุมัติ ซึ่งเราจะได้กฎเป็น **IF THEN** loan_decision=accept จากนั้นเราจะทำการเพิ่มแต่ละแอทริบิวที่เป็นไปได้ (attr, val) เข้ากับกฎข้างต้น โดยเราจะต้องทำการพิจารณาทุกๆแอทริบิวที่อยู่ในลิสต์ของแอทริบิวรวมถึงทุกๆค่าที่เกิดขึ้นในแต่ละแอทริบิวด้วย ตัวอย่างเช่น กิ่งซ้ายสุดของรูปที่ 6.10 จะเป็นการเพิ่มการพิจารณาแอทริบิว ‘loan_term=short’ เข้ากับกฎข้างต้น ที่ซึ่งจะทำให้กฎมีเงื่อนไขมากขึ้นคือ **IF** loan_term=short **THEN** loan_decision=accept ตามลำดับ โดยเมื่อทำการพิจารณาทุกๆแอทริบิวและทุกๆค่าที่เป็นไปได้ทั้งหมดแล้ว เราจะทำการเลือกกฎที่มีคุณภาพมากที่สุดเพื่อทำการพิจารณาต่อไป โดยการวัดคุณภาพของแต่ละกฎอาจจะวัดจากค่าเปอร์เซ็นต์ความถูกต้องของการจำแนกข้อมูลในชุดข้อมูลสอนได้อย่างถูกต้องหรือเราอาจจะประยุกต์ใช้การวัดคุณภาพโดยใช้มาตรวัดอื่นๆก็เป็นได้



รูปที่ 6.10 ตัวอย่างวิธีการสร้างกฎด้วยวิธีการแบบ ‘general-to-specific’

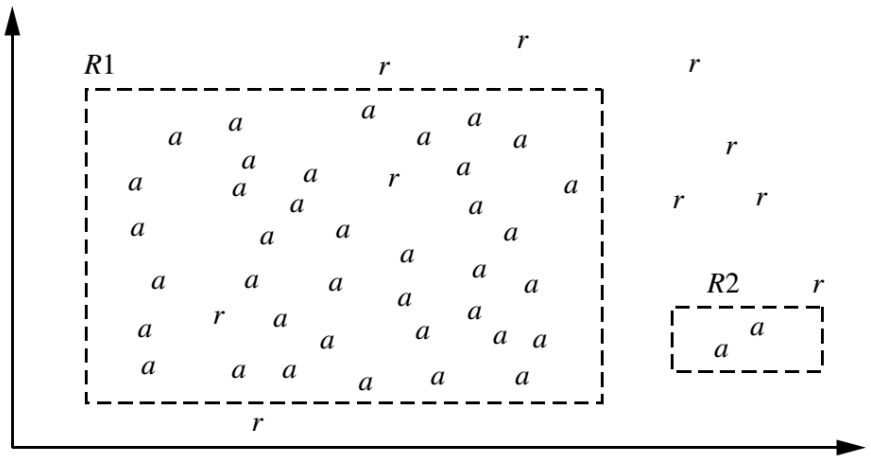
โดยจากรูปที่ 6.10 จะทำการเลือกกฎ **IF** income=high **THEN** loan_decision=accept ซึ่งมีคุณภาพสูงสุดจากการพิจารณาทุกๆแอทริบิวเพื่อเป็นกฎที่จะพิจารณาต่อไป ขั้นตอนต่อไปเราจะทำการเพิ่มแอทริบิวที่ยังไม่ถูกเลือกให้แก่กฎที่เลือกและทำกระบวนการซ้ำเดิมเพื่อให้ได้กฎที่เราต้องการ โดยในการเพิ่มแอทริบิวแต่ละครั้งเราจะทำการเลือกกฎที่จะสามารถคลอบคลุมข้อมูลเรคคอร์ดที่เป็นหมวดหมู่ที่ได้รับการ

อนุมัติได้มากขึ้น แต่เมื่อไรก็ตามที่เราไม่สามารถหากฎที่มีเปอร์เซ็นต์การคลอบคลุมที่มากขึ้น เราจะหยุดการเลือกแล้วทำการตรวจสอบว่ากฎที่ได้นั้นผ่านเกณฑ์คุณภาพที่ยอมรับได้หรือไม่ ถ้ากฎผ่านเกณฑ์คุณภาพเราจะทำการเก็บกฎนั้นไว้ใน Rule_Set ต่อไป

การวัดคุณภาพของกฎ (Rule quality measures)

ในการสร้างกฎจากฟังก์ชัน Learn_One_Rule เราจะต้องทำการวัดคุณภาพของกฎในทุกๆครั้งที่มีการเพิ่มแอทริบิวและค่าที่เกิดขึ้นเข้าไปในกฎ โดยในการเลือกกฎครั้งหนึ่งเราอาจประเมินจากค่าความถูกต้องของกฎที่จะสามารถจำแนกข้อมูลได้ แต่อย่างไรก็ตามการใช้ค่าความถูกต้องในการจำแนกข้อมูลจากการเลือกกฎเพียงอย่างเดียวอาจจะไม่เพียงพอ เพื่อที่จะเข้าใจเกี่ยวกับการใช้ค่าความถูกต้องในการจำแนกข้อมูลเพียงในการเลือกกฎเพียงมากขึ้น ลองพิจารณาตัวอย่างดังต่อไปนี้

ตัวอย่าง การเลือกกฎโดยประเมินจากค่าเปอร์เซ็นต์ความถูกต้องในการจำแนกข้อมูล โดยจะทำการพิจารณากฎ 2 กฎดังแสดงในรูปที่ 6.11 ที่ซึ่งคือ กฎ R1 และ กฎ R2 ที่ซึ่งจะเป็นกฎที่สามารถคลอบคลุมข้อมูลเรคคอร์ดที่อยู่ในหมวดหมู่ของการกู้ยืมเงินสำเร็จ โดยจากรูปค่า ‘a’ จะแสดงถึงเรคคอร์ดที่อยู่ในหมวดหมู่ของการกู้ยืมเงิน*สำเร็จ* และค่า ‘r’ จะแสดงถึงเรคคอร์ดที่อยู่ในหมวดหมู่ของการกู้ยืมเงิน*ไม่สำเร็จ* ตามลำดับ โดยกฎ R1 จะสามารถจำแนกข้อมูลได้อย่างถูกต้องทั้งสิ้น 38 เรคคอร์ดจากข้อมูล 40 เรคคอร์ดที่กฎ R1 คลอบคลุม และ กฎ R2 จะสามารถจำแนกข้อมูลได้อย่างถูกต้องเพียงแต่ 2 เรคคอร์ดจากจำนวนเรคคอร์ดที่คลอบคลุมทั้งหมด แต่อย่างไรก็ตาม เมื่อเราพิจารณาถึงค่าเปอร์เซ็นของความถูกต้องในการจำแนกข้อมูล (เพียงอย่างเดียว) เราจะเห็นว่ากฎ R1 จะมีค่าเปอร์เซ็นต์ความถูกต้องเท่ากับ 95% แต่ในส่วนของกฎ R2 จะมีค่าเปอร์เซ็นต์ความถูกต้องเท่ากับ 100% เมื่อเราใช้เกณฑ์เปอร์เซ็นต์ค่าความถูกต้องในการเลือกกฎ เราจะทำการเลือกกฎ R2 ที่จะมีค่าเปอร์เซ็นต์ความถูกต้องมากกว่า R1 แต่อย่างไรก็ตาม กฎ R2 นั้นไม่ได้ดีกว่ากฎ R1 เลย เนื่องจากค่าเปอร์เซ็นต์ความคลอบคลุมข้อมูลนั้นมีน้อยมาก



รูปที่ 6.11 ตัวอย่างกฎสำหรับการกู้ยืมที่ได้รับการอนุมัติ (loan_decision=accept)

จากตัวอย่างข้างต้นเราจะสังเกตได้ว่าการใช้เพียงแค่ค่าเปอร์เซ็นต์ความถูกต้องในการจำแนกข้อมูลจะไม่เพียงพอสำหรับการวัดคุณภาพของกฎ โดยเมื่อเราเปลี่ยนจากการใช้ค่าเปอร์เซ็นต์ความถูกต้องไปเป็นค่าความคลอบคลุมข้อมูลก็อาจจะไม่เพียงพอเช่นกัน ดังนั้นในการที่จะแก้ไขปัญหาดังกล่าว เราอาจทำการพิจารณาค่าทั้งสองร่วมกันเพื่อใช้ในการวัดคุณภาพของกฎ หรือเราอาจจะพิจารณาตัวชี้วัดคุณภาพอื่นๆอาทิเช่น ค่าเอนโทรปี ที่ซึ่งจะเกี่ยวเนื่องกับค่าเกนความรู้ หรืออาจใช้ตัวชี้วัดทางสถิติอื่นๆ โดยในการใช่ค่าเอนโทรปีในการวัดคุณภาพของกฎ เราจะทำการเลือกกฎที่มีค่าเอนโทรปีที่มีค่าน้อยที่สุด โดยการประยุกต์ใช้ค่าเอนโทรปีจะสามารถบอกได้ถึงปริมาณเรคคอร์ดที่ถูกคลอบคลุมโดยหมวดหมู่หนึ่งๆ และปริมาณเรคคอร์ดที่ถูกคลอบคลุมด้วยหมวดหมู่อื่นๆได้

6.6 การจำแนกข้อมูลด้วยการส่งค่าย้อนกลับ (Classification by Backpropagation)

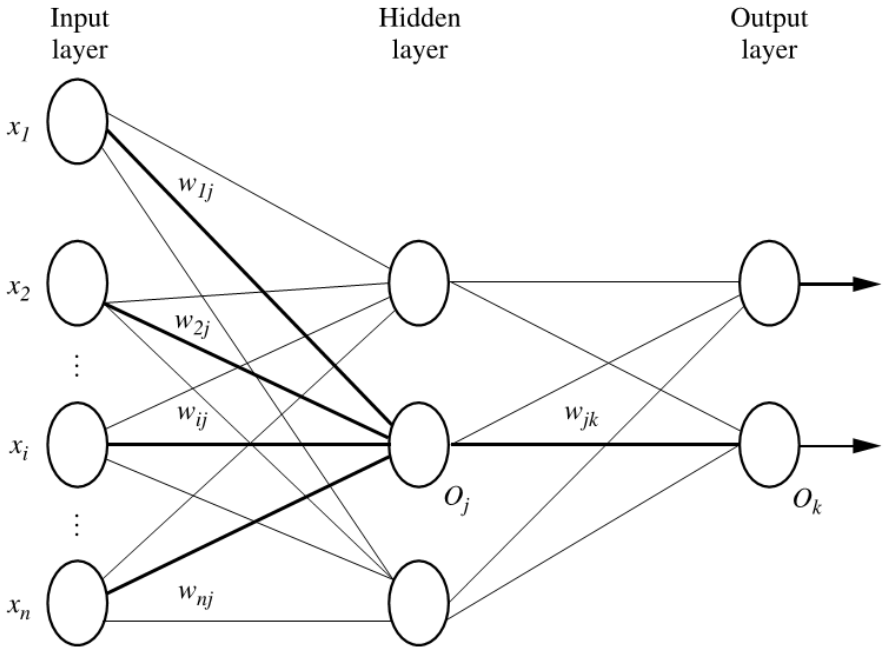
การส่งค่าย้อนกลับจะเป็นอัลกอริทึมเครือข่ายประสาทเทียม—ที่ซึ่งคือ เซตของการเชื่อมต่อกันระหว่างโหนดของอินพุตและเอาท์พุต โดยแต่ละการเชื่อมต่อกันจะมีค่าน้ำหนักเกี่ยวเนื่องด้วย เมื่อทำการพิจารณาชุดข้อมูลสอน เราจะทำการสร้างเครือข่ายด้วยการปรับค่าน้ำหนักที่เชื่อมต่อระหว่างโหนดต่างๆ เพื่อที่จะสามารถคาดเดาหมวดหมู่ของข้อมูลได้อย่างถูกต้อง โดยในการปรับค่าน้ำหนักจะใช้เวลาค่อนข้างมาก เนื่องจากจะต้องทำการปรับค่าน้ำหนักหลายครั้ง และในการสร้างเครือข่ายจะต้องการให้ผู้ใช้ทำการกำหนดค่าพารามิเตอร์ต่างๆเพื่อกำหนดโครงสร้างของเครือข่ายด้วย

ข้อดีของเครือข่ายประสาทเทียม คือ 1) มีความคงทนต่อสิ่งรบกวน (noise) สูง 2) มีความสามารถในการจำแนกข้อมูลเมื่อยังไม่ได้นำข้อมูลมาสอน 3) สามารถใช้ในการการจำแนกข้อมูลเมื่อทราบ/รับรู้/มีความรู้เกี่ยวเพียงเล็กน้อยเกี่ยวกับความสัมพันธ์ระหว่างแอทริบิวต่างๆกับหมวดหมู่ต่างๆ 4) สามารถทำงานได้ดีกับข้อมูลอินพุตและเอาท์พุตที่มีค่าแบบไม่ต่อเนื่อง 5) ถูกประยุกต์ใช้อย่างแพร่หลายในงานหลายๆด้าน อาทิเช่น การรู้จำรายมือ การวิเคราะห์ข้อมูลเกี่ยวกับพยาธิต่างๆ และการสอนให้คอมพิวเตอร์ออกเสียงภาษาอังกฤษ เป็นต้น และ 6) สามารถนำไปพัฒนาแบบขนาน (parallel, parallelization) ได้ เครือข่ายประสามเทียมได้ถูกพัฒนาอย่างต่อเนื่องที่ซึ่งทำให้มีอัลกอริทึมถูกคิดค้นขึ้นเป็นจำนวนมาก แต่อย่างไรก็ตาม มีอัลกอริทึมหนึ่งที่ได้รับความนิยมเป็นอย่างมากและถูกประยุกต์ใช้อย่างแพร่หลาย นั่นคือ อัลกอริทึมการส่งค่าย้อนกลับ (Backpropagation) ที่ซึ่งเราจะได้ศึกษารายละเอียดการทำงานในส่วนนี้

6.6.1 โครงข่ายประสาทเทียมหลายลำดับชั้น

อัลกอริทึมการส่งค่าย้อนกลับจะทำการเรียนรู้โดยใช้เครือข่ายประสาทเทียมหลายลำดับชั้น ที่ซึ่งจะประกอบไปด้วย 1 ลำดับชั้นสำหรับข้อมูลอินพุ (input layer) 1 ลำดับชั้นหรือมากกว่านั้นสำหรับ "hidden layer" และ 1 ลำดับชั้นสำหรับข้อมูลเอาท์พุต (output layer) ดังแสดงในรูปที่ 6.12 จากรูปเราจะสังเกตุได้ว่าแต่ละลำดับชั้นจะประกอบไปด้วยโหนดต่างๆ โดยที่อินพุตสำหรับเครือข่ายจะสอดคล้องกับแทริบิวต่างๆของเรคคอร์ดหนึ่งๆจากชุดข้อมูลสอน อินทุตจะถูกป้อนเข้าไปยังโหนดต่างๆใน input layer จากนั้นอินพุต

เหล่านั้นจะผ่าน input layer และด้วยค่าคน้ำหนักต่างๆและเดินทางไปสู่ลำดับชั้นต่อไป ซึ่งก็คือ hidden layer โดยที่เอาท์พุตของ hidden layer ลำดับชั้นหนึ่งๆอาจเป็นอินพุตของ hidden layer ลำดับชั้นหนึ่งๆก็เป็นได้ โดยจำนวนลำดับชั้นของ hidden layer จะมีจำนวนเท่าไรก็ได้ (แต่โดยทั่วๆไปมักใช้เพียงแค่ลำดับชั้นเดียว) เมื่อผ่านลำดับชั้นที่เป็น hidden layer แล้วเราจะได้ค่าน้ำหนักที่ซึ่งจะใช้เป็นอินพุตสำหรับ output layer ที่ซึ่งจะคืนค่าผลของการจำแนก/ทำนายหมวดหมู่ของข้อมูลสำหรับเรคคอร์ดหนึ่งๆที่เป็นอินพุต โดยเมื่อเราพิจารณาที่โหนดต่างๆใน output layer เราจะเห็นว่า แต่ละโหนดจะทำการรวมค่าน้ำหนักทั้งหมดจากลำดับชั้นก่อนหน้า แล้วทำการประยุกต์ใช้ nonlinear (activation) function เพื่อทำการกำหนดค่าน้ำหนัก (ดังแสดงในรูปที่ 6.13)



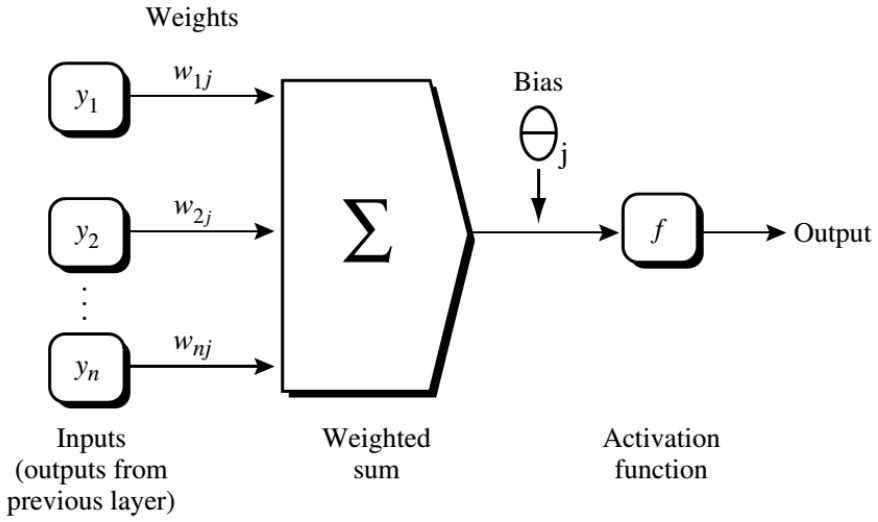
รูปที่ 6.12 ตัวอย่างโครงข่ายประสาทเทียม

6.6.2 การกำหนดโครงสร้างของเครือข่าย

ก่อนที่จะทำการเรียนรู้โดยการประยุกต์ใช้โครงข่ายประสาทเทียม เราจะต้องทำการกำหนดโครงสร้างของโครงข่ายประสาทเทียมเสียก่อน ที่ซึ่งจะเป็นการกำหนดจำนวนโหนดใน input layer จำนวนโหนดใน hidden layer และ จำนวนโหนดใน output layer ตามลำดับ โดยในการกำหนดโหนดใน input layer เราจะสามารถแบ่งการพิจารณาออกเป็น 2 กรณี ตามลักษณะของแอทริบิวที่ทำการพิจารณา โดยถ้าแอทริบิวที่พิจารณามีค่าเป็นแบบต่อเนื่อง (มีค่าในเชิงตัวเลขที่บ่งบอกถึงปริมาณ) เราควรที่จะต้องทำการนอร์มัลไลซ์ข้อมูลให้อยู่ในช่วง 0.0 – 1.0 เสียก่อนเพื่อที่จะช่วยให้กระบวนการเรียนรู้นั้นเป็นไปอย่างรวดเร็ว และ เราอาจทำการกำหนดโหนดเพียงหนึ่งโหนดสำหรับแอทริบิวที่มีค่าแบบต่อเนื่องเท่านั้น แต่สำหรับแอทริบิวที่มีค่าแบบไม่ต่อเนื่อง (มีค่าแบบหมวดหมู่หรือมีค่าเป็นช่วง) เราอาจทำการกำหนดให้แต่ละโหนดจะแทนค่าที่ปรากฏขึ้นในแอทริบิวนั้น ตัวอย่างเช่น แอทริบิว A มีค่าแบบไม่ต่อเนื่องที่ซึ่งจะมีค่าที่ปรากฏทั้งสิ้น 3 ค่าด้วยกันคือ $\{a_0, a_1, a_2\}$ เราอาจทำการกำหนดโหนด I_0, I_1 และ I_2 สำหรับค่าที่ปรากฏขึ้นทั้ง 3 ค่า โดยกำหนดให้โหนด

ที่ I_0 จะเป็นโหนดสำหรับค่า a_0 โหนดที่ I_1จะเป็นโหนดสำหรับค่า a_1 และ โหนดที่ I_2 จะเป็นโหนดสำหรับค่า a_2 ตามลำดับ โดยในขั้นตอนเริ่มต้นทุกๆโหนดจะถูกกำหนดให้มีค่าเป็น 0

ในส่วนของการกำหนดจำนวนโหนดใน hidden layer จะไม่มีกฎเกณฑ์ที่ตายตัว โดยเราจะสามารถกำหนดอย่างไรก็ได้ แต่จำนวนโหนดที่กำหนดอาจส่งผลกระทบต่อความถูกต้องในการเรียนรู้ของโครงข่ายประสาทเทียม นอกจากนั้นการกำหนดค่าน้ำหนักเริ่มต้นของแต่ละโหนดก็ส่งผลต่อความถูกต้องในการเรียนรู้ด้วยเช่นกัน แต่อย่างไรก็ตาม เมื่อเราทำการสร้างโครงข่ายและทำการเรียนรู้แล้ว เราจะสามารถตรวจวัดค่าความถูกต้องของโครงข่ายว่าเป็นที่ยอมรับได้หรือไม่ ถ้าไม่สามารถยอมรับได้ เราสามารถดำเนินการเรียนรู้ใหม่โดยการปรับเปลี่ยนโครงสร้างของโครงข่ายให้เหมาะสมมากยิ่งขึ้น



รูปที่ 6.13 ตัวอย่างการเชื่อมต่อระหว่าง input layer ไปยัง hidden layer และจาก hidden layer ไปยัง output layer

6.6.3 อัลกอริทึมการส่งค่าย้อนกลับ

การทำงานของอัลกอริทึมการส่งค่าย้อนกลับจะทำการเรียนรู้โดยใช้กระบวนการทำซ้ำกับชุดข้อมูลสอนแล้วทำการเปรียบเทียบการทำนาย/คาดเดาหมวดหมู่ของข้อมูลเรคคอร์ดหนึ่งๆ โดยในการพิจารณาข้อมูลเรคคอร์ดหนึ่งๆจะทำให้ค่าน้ำหนักที่โหนดหนึ่งๆมีค่าเปลี่ยนแปลงไป โดยจะเปลี่ยนแปลงไปเพื่อที่จะทำให้ค่าความผิดพลาดนั้นน้อยที่สุดเท่าที่จะเป็นไปได้ โดยในการเปลี่ยนแปลงค่าน้ำหนักนั้นจะทำในทิศทางย้อนกลับ ที่ซึ่งจะเริ่มจากการพิจารณาที่ output layer แล้วย้อนกลับไปยัง hidden layer ตามลำดับ โดยกระบวนการเรียนรู้จะสิ้นสุดเมื่อค่าน้ำหนักไม่เปลี่ยนแปลง โดยการทำงานของอัลกอริทึมการส่งค่าย้อนกลับจะสามารถอธิบายได้ดังนี้

การทำงานของอัลกอริทึมการส่งค่าย้อนกลับจะเริ่มจาก **1) การกำหนดค่าน้ำหนัก** จะเป็นการกำหนดค่าน้ำหนักด้วยวิธีการสุ่มให้มีค่าน้อยๆ (อาทิเช่น อยู่ในช่วง -1.0 ถึง 1.0 หรือ -0.5 ถึง 0.5)

นอกจากนั้นเราจะต้องทำการกำหนดค่าความเอียนเองของแต่ละโหนดด้วยวิธีการสุ่มให้มีค่าน้อยๆด้วยเช่นกัน จากนั้นเมื่อทำการพิจารณาแต่ละเรคคอร์ด X เราจะต้องทำการดังต่อไปนี้

1) การพิจารณาอินพุตด้วยการพิจารณาไปข้างหน้า (Propagate the input forward)—จะเริ่มจากการพิจารณาอินพุตโดยทำการเทียบกับโหนดใน input layer เพื่อทำการส่งต่อไปยัง hidden layer โดยการพิจารณาอินพุตของแอทริบิว A ในเรคคอร์ด X เราจะทำการหาโหนดใน input layer I_j ที่มีค่าตรงกับค่าที่ปรากฏในแอทริบิว A ที่ซึ่งเราจะได้เอาท์พุต O_j จากนั้นจะทำการคำนวณค่าที่เป็นอินพุตและเอาท์พุตของแต่ละโหนดใน hidden layer เพื่อที่จะเข้าใจการคำนวณค่าที่เป็นอินพุตและเอาท์พุตมากขึ้น ลองพิจารณารูปที่ 6.13 ที่ซึ่งแต่ละโหนดใน hidden layer จะได้รับอินพุตมาจากโหนดใน input layer จากนั้นคำนวณค่าอินพุตของ hidden layer ด้วยการคูณแต่ละค่าเอาท์พุต O_j ด้วยค่าน้ำหนักที่สอดคล้องกับโหนดนั้นๆแล้วรวมไว้จากนั้นบวกด้วยค่าความเอนเอียงอีกครั้งหนึ่ง ที่ซึ่งจะสามารถคำนวณได้ดังนี้

$$I_j = \sum_i w_{ij} O_j + \theta_j$$

เมื่อค่า w_{ij} คือ น้ำหนักของการเชื่อมต่อระหว่างโหนดที่ i ในลำดับชั้นที่กำลังพิจารณากับโหนดที่ j ในลำดับชั้นก่อนหน้า ค่า O_j คือ ค่าเอาท์พุตจากโหนดที่ i ในลำดับชั้นก่อนหน้า และค่า θ_j คือ ค่าความเอนเอียงที่ซึ่งทำหน้าที่คล้ายกับค่าขีดแบ่ง (threshold) ที่ซึ่งจะเปลี่ยนแปลงกิจกรรมของโหนดที่พิจารณา

แต่ละโหนดใน hidden layer และ output layer จะทำการคำนวณค่าอินพุตแล้วทำการประยุกต์ใช้ “activation function” กับค่าอินพุตที่คำนวณไว้แล้ว (ดังแสดงในรูปที่ 6.13 ที่ซึ่งแสดงด้วย “f”) โดย “activation function” จะเป็นสัญลักษณ์ของการกระตุ้นการทำงานของเซลล์ประสาทในส่วนของโหนดนั้นๆ เมื่อทำการประยุกต์ใช้ “activation function” แล้วเราจะได้เอาท์พุตของโหนด j ที่ทำการพิจารณา ที่ซึ่งสามารถคำนวณได้ดังนี้

$$O_j = \frac{1}{1 + e^{-I_j}}$$

เมื่อเราทำการคำนวณค่าเอาท์พุต O_j สำหรับแต่ละโหนดใน hidden layer และรวมถึง output layer แล้ว ถ้าเราทำการเก็บ/บันทึกค่าเอาท์พุตเหล่านั้นไว้ ย่อมจะเป็นการดีสำหรับการดำเนินการส่งค่าความผิดพลาดย้อนกลับที่ซึ่งจะต้องใช้ค่าเหล่านั้นอีกครั้งหนึ่ง

2) การส่งค่าความผิดพลาดย้อนกลับ (Backpropagate the error)—ค่าความผิดพลาดจะถูกส่งย้อนกลับด้วยการอัพเดทค่าน้ำหนักและค่าความเอนเอียงที่ส่งผลกระทบต่อผลการทำนายของโครงข่ายประสาทเทียม โดยสำหรับแต่ละโหนดที่ j ใน output layer เราจะทำการคำนวณค่าความผิดพลาด Err_j โดย

$$Err_j = O_j(1 - O_j)(T_j - O_j)$$

เมื่อ O_j คือ ค่าเอาท์พุตจริงที่ได้จากโหนดที่ j และค่า T_j จะเป็นค่าหมวดหมู่/ค่าเป้าหมายของเรคคอร์ดในชุดข้อมูลสอนที่ทำการพิจารณา

ในการที่จะคำนวณค่าความผิดพลาดของโหนดที่ j หนึ่งๆใน hidden layer เราจะทำการพิจารณาค่าน้ำหนักรวมของค่าความผิดพลาดของโหนดในลำดับชั้นถัดไปที่เชื่อมต่อกับโหนด j ที่ซึ่งค่าความผิดพลาดของโหนดที่ j จะสามารถคำนวณได้จาก

$$Err_j = O_j(1 - O_j)\sum_k Err_k w_{jk}$$

เมื่อ w_{jk} คือ ค่าน้ำหนักของการเชื่อมต่อระหว่างโหนดที่ j ใน hidden layer ที่ทำการพิจารณากับโหนดที่ k ในลำดับชั้นถัดไป และค่า Err_k คือ ค่าความผิดพลาดของโหนดที่ k

ค่าน้ำหนักจะถูกอัพเดทเพื่อแพร่กระจายความผิดพลาด โดยค่าน้ำหนักจะถูกอัพเดทดังนี้

$$\Delta w_{ij} = (l)Err_j O_i$$
$$w_{ij} = w_{ij} + \Delta w_{ij}$$

เมื่อค่า Δw_{ij} คือ ค่าความเปลี่ยนแปลงที่จะเกิดขึ้นกับค่าน้ำหนัก w_{ij} และค่า l คือ ค่าอัตราการเรียนรู้ที่ซึ่งมีค่าคงที่อยู่ระหว่าง 0.0 ถึง 1.0

ค่าความเอนเอียงจะถูกอัพเดทเพื่อแพร่กระจายความผิดพลาดด้วยเช่นกัน โดยค่าความเอนเอียงจะถูกอัพเดทดังนี้

$$\Delta\theta_j = (l)Err_j$$
$$\theta_j = \theta_j + \Delta\theta_j$$

เมื่อค่า $\Delta\theta_j$ คือ ค่าความเปลี่ยนแปลงที่จะเกิดขึ้นกับค่าความเอนเอียง θ_j

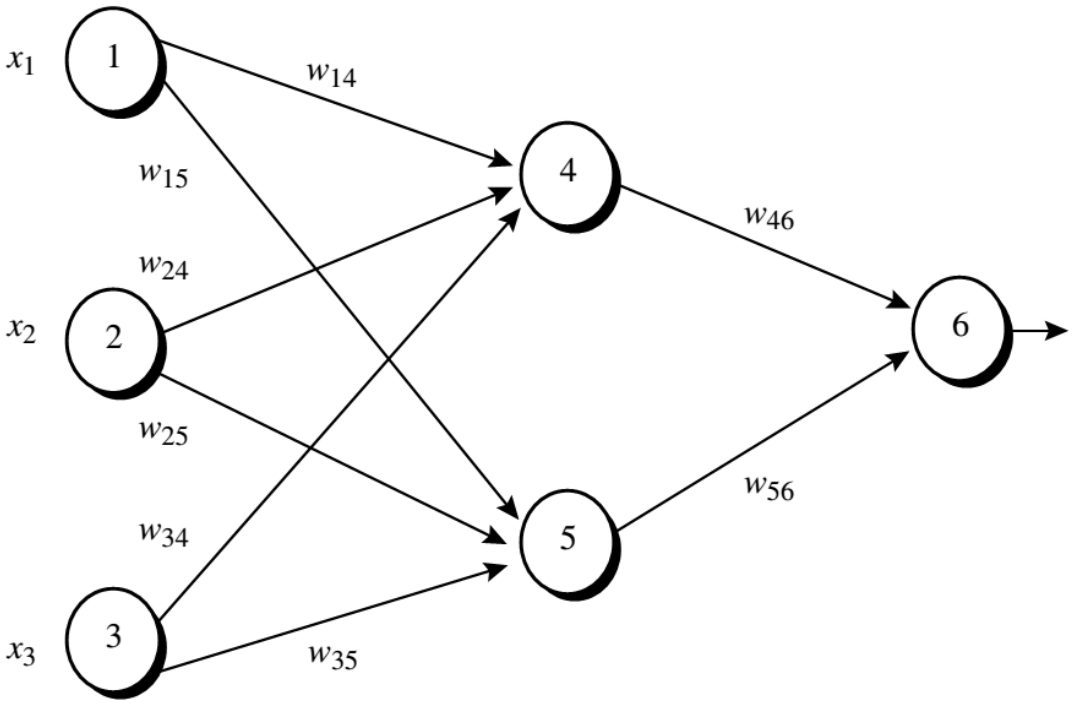
3) **การหยุดการทำงาน (Terminating condition)**—กระบวนการเรียนรู้จะหยุดการทำงานก็ต่อเมื่อ
   - ทุกๆค่า Δw_{ij} มีค่าน้อยมาก ที่ซึ่งน้อยกว่าค่าขีดแบ่งที่กำหนด หรือ
   - ค่าเปอร์เซ็นต์ในการจำแนกข้อมูลผิดพลาดมีค่าน้อยกว่าค่าขีดแบ่งที่กำหนด หรือ
   - กระบวนการเรียนรู้ได้ถูกดำเนินการซ้ำจนครบตามจำนวนรอบที่กำหนด

   โดยในทางปฏิบัติแล้วเราจะทำการกำหนดจำนวนรอบในการดำเนินงาน ที่ซึ่งจะทำให้กระบวนการเรียนรู้หยุดการทำงานก่อนที่ค่าน้ำหนักต่างๆจะมีค่าคงที่

จากกระบวนการทำงานทั้งหมดข้างต้น เราจะสังเกตุได้ว่าประสิทธิภาพการคำนวณของอัลกอริทึมการส่งค่าย้อนกลับจะขึ้นอยู่กับเวลาที่ใช้ในกระบวนการเรียนรู้ (ปรับค่าน้ำหนัก) เมื่อเรากำหนดให้ข้อมูลในชุดข้อมูลสอนประกอบด้วย $|D|$ เรคคอร์ด และ มีค่าน้ำหนักที่ต้องทำการพิจารณาทั้งสิ้น w ค่า ดังนั้นเราจะสามารถคำนวณเวลาที่ใช้ในการเรียนรู้ในแต่ละรอบได้เป็น $O(|D| \times w)$

ตัวอย่าง การเรียนรู้ด้วยอัลกอริทึมการส่งค่าย้อนกลับ รูปที่ 6.14 จะแสดงตัวอย่างโครงสร้างเครือข่ายประสาทเทียมที่ถูกกำหนดโครงสร้างไว้เพื่อสำหรับเรียนรู้ กำหนดให้ค่าอัตราการเรียนรู้มีค่าเท่ากับ 0.9 และกำหนดให้

ค่าน้ำหนักและค่าความเอนเอียงของแต่ละโหนดมีค่าดังแสดงในตารางที่ 6.2 นอกจากนั้นเราจะทำการพิจารณา ***ข้อมูลเรคคอร์ด $X = (1, 0, 1)$ ที่ซึ่งมีหมวดหมู่ข้อมูลเป็น 1***



รูปที่ 6.14 ตัวอย่างการกำหนดโครงข่ายประสาทเทียมเพื่อเรียนรู้

ตารางที่ 6.2 การกำหนดค่าน้ำหนักและค่าความเอนเอียงให้กับโหนดต่างๆในโครงข่ายประสาทเทียม

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
1	0	1	0.2	-0.3	0.4	0.1	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1

เมื่อทำการพิจารณาข้อมูลเรคคอร์ด $X = (1,0,1)$ ด้วยการพิจารณาอินพุตด้วยการพิจารณาไปข้างหน้า (Propagate the input forward) เราจะสามารถคำนวณค่าอินพุตและเอาท์พุตเมื่อทำการพิจารณาโหนดต่างๆใน hidden layer และ โหนดใน output layer ได้ดังแสดงในตารางที่ 6.3 จากนั้นเราจะทำในขั้นตอนต่อไป ซึ่งก็คือการคำนวณค่าความผิดพลาดของแต่ละโหนดด้วยการส่งค่าความผิดพลาดย้อนกลับ (Backpropagate the error) ที่ซึ่งจะเป็นการคำนวณค่าน้ำหนักและค่าความเอนเอียงใหม่ดังแสดงตัวอย่างการคำนวณในตารางที่ 6.4 และ 6.5 ตามลำดับ หลังจากคำนวณค่าน้ำหนักและค่าความเอนเอียงเสร็จสิ้น เราจะถือว่าเป็นการเสร็จสิ้นการพิจารณาข้อมูลเรคคอร์ด X และเราจะทำการพิจารณาข้อมูลเรคคอร์ดอื่นในลำดับถัดไป

ตารางที่ 6.3 การคำนวณค่าอินพุตและเอาท์พุตเมื่อทำการพิจารณาโหนดที่ I_j

โหนดที่ j	ค่าอินพุตที่ได้ในการพิจารณาโหนดที่ I_j	ค่าเอาท์พุตที่ได้จากการพิจารณาโหนดที่ I_j (O_j)
4	$0.2 + 0 - 0.5 - 0.4 = -0.7$	$\frac{1}{1+e^{0.7}} = 0.332$
5	$-0.3 + 0 + 0.2 + 0.2 = 0.1$	$\frac{1}{1+e^{-0.1}} = 0.525$
6	$(-0.3)(0.332) - (0.2)(0.525) + 0.1 = -0.105$	$\frac{1}{1+e^{0.105}} = 0.474$

ตารางที่ 6.4 การคำนวณค่าความผิดพลาดของแต่ละโหนด

โหนดที่ j	Err_j
6	$(0.474)(1-0.474)(1-0.474) = 0.1311$
5	$(0.525)(1-0.525)(0.1311)(-0.2) = -0.0065$
4	$(0.332)(1-0.332)(0.1311)(-0.3) = -0.0087$

ตารางที่ 6.5 การคำนวณค่าน้ำหนักและค่าความเอนเอียง

ค่าน้ำหนัก/ค่าความเอนเอียง	ค่าใหม่ที่ถูกอัพเดท
w_{46}	$-0.3 + (0.9)(0.1311)(0.332) = -0.261$
w_{56}	$-0.2 + (0.9)(0.1311)(0.525) = -0.138$
w_{14}	$0.2 + (0.9)(-0.0087)(1) = 0.192$
w_{15}	$-0.3 + (0.9)(-0.0065)(1) = -0.306$
w_{24}	$0.4 + (0.9)(-0.0087)(0) = 0.4$
w_{25}	$0.1 + (0.9)(-0.0065)(0) = 0.1$
w_{34}	$-0.5 + (0.9)(-0.0087)(1) = -0.508$
w_{35}	$0.2 + (0.9)(-0.0065)(1) = 0.194$
θ_6	$0.1 + (0.9)(0.1311) = 0.218$
θ_5	$0.2 + (0.9)(-0.0065) = 0.194$
θ_4	$-0.4 + (0.9)(-0.0087) = -0.408$

6.7 การจำแนกข้อมูลด้วยการวิเคราะห์กฎความสัมพันธ์ของข้อมูล (Classification by Association Rule Analysis, Associative Classification)

จากที่เราได้ศึกษาในบทที่แล้ว เราได้ทราบว่ารูปแบบปรากฏบ่อยและกฎความสัมพันธ์ของข้อมูลจะประกอบไปด้วยรายการ (item) ต่างๆที่ปรากฏร่วมกันบ่อยๆ เพื่อใช้อธิบายถึงรูปแบบที่แอบแฝงอยู่ชุดข้อมูล

แต่ในส่วนนี้เราจะทำการศึกษาเกี่ยวกับการจำแนกข้อมูลจากกฎความสัมพันธ์ที่ซึ่งจะเป็นการค้นหาความสัมพันธ์ของรูปแบบปรากฏบ่อย (ที่ซึ่งถูกอธิบายด้วยแอทริบิวและค่าที่เกิดขึ้นในแอทริบิว) และหมวดหมู่ของข้อมูล

การค้นหากฎความสัมพันธ์ของข้อมูลเพื่อการจำแนกข้อมูลจะประกอบไปด้วยขั้นตอนการทำงานหลัก 2 ขั้นตอนด้วยกัน คือ (i) การค้นหารูปแบบปรากฏบ่อย และ (ii) การสร้างกฎจากรูปแบบที่ได้จากขั้นตอนแรก ที่ซึ่งจะได้เป็นกฎดังตัวอย่างดังนี้

$$age = youth \wedge credit = OK \Rightarrow buys_{computer} = yes \ [support = 20\%, confidence = 93\%]$$

โดยจากกฎเราจะเห็นว่าเราสามารถประเมินค่าความถูกต้องของกฎได้จากค่าความเชื่อมั่น (confidence) ที่ซึ่งจะแสดงถึงอัตราส่วนระหว่างการเกิดขึ้นร่วมกันของทั้ง 3 รายการหารด้วยการเกิดขึ้นร่วมกันของ 2 รายการแรกที่ไม่รวมหมวดหมู่

$$\frac{support(age = youth \wedge credit = OK \wedge buys\_computer = yes)}{support(age = youth \wedge credit = OK)}$$

นอกจากนั้นเรายังหาค่าความครอบคลุม (coverage) ของกฎหนึ่งได้จากค่าสนับสนุน (support) ที่ซึ่งจะบ่งบอกถึงอัตราเปอร์เซ็นต์การปรากฏขึ้นของกฎเทียบกับเรคคอร์ดของข้อมูลทั้งหมด

กำหนดให้ D คือชุดข้อมูลที่ซึ่งแต่ละเรคคอร์ดจะถูกอธิบายด้วยแอทริบิวทั้งสิ้น n แอทริบิว $(A_1, A_2, \dots, A_n)$ และ 1 แอทริบิวที่เป็นหมวดหมู่ของข้อมูล (A_{class}) แอทริบิวทั้งหมดที่ทำการพิจารณาจะต้องถูกทำให้มีค่าแบบไม่ต่อเนื่องทั้งหมด ที่ซึ่งจะเป็นหมวดหมู่ของข้อมูล ตัวอย่างเช่น ข้อมูลในแอทริบิว age จะถูกแปลงจากอายุที่เป็นตัวเลขให้กลายเป็นหมวดหมู่หรือช่วงของอายุ ดังนี้คือ age=youth age=middle หรือ age=old เป็นต้น แต่ละรายการ (item) $p = (A_i, v)$ จะประกอบไปด้วยข้อมูลสองส่วน คือ แอทริบิวหนึ่งๆและค่าที่เกิดขึ้นในแอทริบิวนั้นๆ ดังนั้นเมื่อทำการพิจารณารายการ p และ ข้อมูลเรคคอร์ด $X = (x_1, x_2, \dots, x_n)$ เราจะสามารถกล่าวได้ว่า X สอดคล้องกับ $p = (A_i, v)$ ก็ต่อเมื่อ ค่า $x_i = v$ เมื่อ x_i คือ ค่าที่เกิดขึ้นในแอทริบิวที่ i^{th} ที่เกิดขึ้นในเรคคอร์ด X ดังนั้น กฎความสัมพันธ์ของข้อมูลจะมีรายการต่างๆเกิดขึ้นในส่วนทางฝั่งซ้ายและจะมีหมวดหมู่ของข้อมูลทางฝั่งขวา ที่ซึ่งจะอยู่ในรูปแบบ $p_1 \wedge p_2 \wedge \dots \wedge p_l \Rightarrow A_{class}$ พร้อมกับค่าสนับสนุนและค่าความเชื่อมั่นตามลำดับ

อัลกอริทึมแรกในการจำแนกข้อมูลด้วยกฎความสัมพันธ์ คือ “CBA (Classification-Based Assoication)” โดย CBA จะประยุกต์ใช้กระบวนการทำซ้ำในการค้นหารูปแบบปรากฏบ่อยแล้วค่อยๆเพิ่มความยาวของรูปแบบไปเรื่อยๆ (การทำงานจะคล้ายกับอัลกอริทึม “*Apriori*”) จากนั้นจะทำการสร้างกฎความสัมพันธ์ โดยเมื่อกฎใดก็ตามที่มีรูปแบบหรือสมาชิกทางฝั่งซ้ายเหมือนกัน เราจะทำการเลือกกฎที่มีค่า

ความเชื่อมั่นมากที่สุดเพื่อแสดงผล โดยในการจำแนกหมวดหมู่ของเรคคอร์ดใดๆก็ตาม จะใช้กฎที่มีค่าความเชื่อมั่นสูงที่สุดที่สอดคล้องกับเรคคอร์ดนั้นๆ

นอกเหนือจากอัลกอริทึม CBA ยังมีอีกอัลกอริทึมหนึ่งที่ซึ่งได้รับความนิยม นั่นคือ "CMAR (Classification based on Multiple Association Rules)" ที่ซึ่งจะทำการประยุต์ใช้ "*FP-growth*" ในการค้าหารูปแบบปรากฏบ่อย (รายละเอียดของ FP-growth จะถูกอธิบายในบทที่ 5) จากนั้น CMAR จะใช้โครงสร้างต้นไม้ในการจัดเก็บกฎต่างๆที่ถูกสร้างขึ้น รวมถึงการประยุกต์ใช้วิธีการตัดเล็มต้นไม้โดยใช้ค่าความเชื่อมั่น ค่าสหสัมพันธ์ และค่าความครอบคลุม ตามลำดับ โดยในการตัดเล็มต้นไม้จะถูกดำเนินการก็ต่อเมื่อมีกฎหนึ่งถูกเพิ่มเข้าสู่ต้นไม้ ตัวอย่างเช่น กฎ R_1 และ R_2 เป็นกฎที่ถูกพิจารณา ที่ซึ่งรูปแบบทางฝั่งซ้ายของกฎ R_1 จะมีจำนวนแอทริบิวน้อยกว่ากฎ R_2 และค่า $confidence(R_1) \geq confidence(R_2)$ ด้วยเหตุนี้กฎ R_2 จะถูกลบทิ้งจากการพิจารณา นอกจากนั้น CMAR ยังทำการตัดเล็มกฎที่ซึ่งมีค่าสหสัมพันธ์ระหว่างรูปแบบทางฝั่งซ้ายและหมวดหมู่ข้อมูลเป็น 0 หรือมีค่าลบ (ในการคำนวณค่าสหสัมพันธ์จะใช้ χ^2)

ในการจำแนกหมวดหมู่ข้อมูลของเรคคอร์ด X ใดๆ โดยใช้ CMAR โดยในการจำแนก ถ้ามีเพียงกฎเดียวที่สอดคล้องกับ X เราจะทำการจำแนกหมวดหมู่ของ X ตามหมู่ของกฎที่สอดคล้องกับ X แต่สำหรับกรณีที่มีกฎมากกว่าหนึ่งกฎสอดคล้องกับ X เราจะทำการจัดกลุ่มกฎที่สอดคล้องกับ X ตามหมวดหมู่ จากนั้น CMAR จะใช้ χ^2 ในการหาค่าสหสัมพันธ์ของกฎในกลุ่มนั้นๆ จากนั้นจะทำการตอบหมวดหมู่ให้แก่เรคคอร์ด X โดยจะเลือกจากกลุ่มของหมวดหมู่ที่มีค่าสหสัมพันธ์มากที่สุด ซึ่งการดำเนินการดังกล่าวจะช่วยให้ค่าความถูกต้องของ CMAR นั้นสูงกว่า CBA

6.8 การเรียนรู้แบบเกียจคร้าน (Lazy learners)

จากเนื้อหาก่อนหน้าในบทนี้ เราได้ทำการศึกษาการจำแนกข้อมูลด้วยต้นไม้ตัดสินใจ การจำแนกข้อมูลด้วยเบเซียน การจำแนกข้อมูลด้วยกฎต่างๆ การจำแนกข้อมูลด้วยการส่งค่าย้อนกลับ และ การจำแนกข้อมูลจากกฎความสัมพันธ์ของข้อมูล ที่ซึ่งจะเป็นวิธีการจำแนก/เรียนรู้แบบกระตือรือร้น (Eager learners) ที่จะประกอบไปด้วยขั้นตอนการทำงาน 2 ขั้นตอน คือ การสร้างตัวจำแนกข้อมูลจากชุดข้อมูลที่เป็นอินพุต และการจำแนกข้อมูลที่ไม่เคยพบเจอมาก่อนจากตัวจำแนกข้อมูล จากขั้นตอนการทำงานทั้งสอง เราจะสังเกตุได้ว่าทุกขั้นตอนข้างต้นจะเป็นแบบกระตือรือร้นที่จะต้องทำการสร้างโมเดล/ตัวจำแนกข้อมูลก่อนที่จะทำการจำแนกข้อมูล/บ่งบอกหมวดหมู่ของข้อมูลเรคคอร์ดหนึ่งๆที่ปรากฏในภายหลัง แต่เนื้อหาในส่วนนี้จะทำการศึกษาเกี่ยวกับการเรียนรู้แบบเกียจคร้าน (Lazy learners) ที่ซึ่งจะเป็นโมเดลที่จะไม่ต้องทำการสร้างตัวจำแนกข้อมูลจากชุดข้อมูลเมื่อได้รับอินพุต แต่จะดำเนินการเพียงแค่เก็บข้อมูลเหล่านั้นไว้ แล้วรอจนกระทั่งเราได้รับข้อมูลที่ต้องการจำแนก/ทำนายหมวดหมู่ของข้อมูล แล้วจึงเริ่มดำเนินการวิเคราะห์ข้อมูล โดยการดำเนินการวิเคราะห์ข้อมูลจะถูกดำเนินการทุกๆครั้งที่เราได้รับข้อมูลที่ต้องการจำแนกหมวดหมู่ข้อมูลเท่านั้น แต่เมื่อไรก็ตามที่ต้องทำการจำแนก/ทำนายหมวดหมู่ของข้อมูล การเรียนรู้แบบเกียจคร้านจะต้องทำการคำนวณเป็นจำนวนมาก โดยอัลกอริทึมที่มีการทำการแบบเกียจคร้านที่มีชื่อเสียงและเป็นที่ยอมรับจะมีอยู่

ด้วยกัน 2 โมเดลด้วยกัน ซึ่งก็คือ ขั้นตอนวิธีการค้นหาเพื่อนบ้านใกล้สุด k ตัว (k-Nearest-Neighbor Classifiers) และ การให้เหตุผลตามกรณี (Case-based reasoning classifiers) โดยเนื้อหาในวิชานี้จะทำการอธิบายเกี่ยวกับขั้นตอนวิธีการค้นหาเพื่อนบ้านใกล้สุด k ตัวที่ซึ่งมีขั้นตอนการทำงานดังนี้

ขั้นตอนวิธีการค้นหาเพื่อนบ้านใกล้สุด k ตัว (k-Nearest-Neighbor Classifiers)

การจำแนกข้อมูลด้วยวิธีการค้นหาเพื่อนบ้านใกล้สุดจะเป็นการเรียนรู้โดยการเปรียบเทียบกันระหว่างเรคคอร์ดของข้อมูลที่ต้องการจำแนก/ทำนายหมวดหมู่กับเรคคอร์ดทั้งหมดในชุดข้อมูลสอนที่มีลักษณะเหมือนกันหรือใกล้เคียงกันด้วยการพิจารณาข้อมูลแอทริบิวต่างๆ โดยข้อมูลเรคคอร์ดหนึ่งๆจะสามารถถูกมองว่าเป็นจุดหนึ่งในระนาบ n มิติ (เมื่อ n คือจำนวนแอทริบิวทั้งหมด) ถ้าเรานำข้อมูลทุกๆเรคคอร์ดในชุดข้อมูลสอนมาวางในระนาบ n มิติ จากนั้นนำข้อมูลเรคคอร์ดที่ต้องการจำแนก/ทำนายหมวดหมู่มาวางในระนาบด้วยเช่นกัน จากนั้นพิจารณาหาว่ามีข้อมูลจุดใดบ้าง (เรคคอร์ดใดบ้าง) ที่มีคุณลักษณะใกล้เคียงกับเรคคอร์ดที่ต้องการจำแนกหมวดหมู่มากที่สุดเป็นจำนวน k เรคคอร์ด

ในการที่จะหาความเหมือนกัน ต่างกัน หรือค่าความใกล้เคียงกันระหว่าเรคคอร์ดใดๆ เราจะสามารถประยุกต์ใช้มาตรวัดความแตกต่างต่างๆ อาทิเช่น ‘Euclidean distance’ ที่ซึ่งจะทำการพิจารณา 2 เรคคอร์ด $X_1 = (x_{11}, x_{12}, \dots, x_{1n})$ และ $X_2 = (x_{21}, x_{22}, \dots, x_{2n})$ ใดๆ จากนั้นทำการพิจารณาความแตกต่างระหว่างค่าที่ปรากฏขึ้นในแต่ละแอทริบิวของทั้งสองเรคคอร์ดแล้วนำมารวมเป็นค่าความแตกต่างรวมที่ซึ่งจะสามารถคำนวณได้ดังนี้

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

จากสมการข้างต้น เมื่อเราทำการพิจารณาค่าที่เกิดขึ้นในแอทริบิว A_i ใดๆ เราจะต้องทำการพิจารณาว่าค่าที่ปรากฏขึ้นมีลักษณะใด—สำหรับแอทริบิวที่มีค่าเป็นตัวเลข เราจะสามารถหาความแตกต่างระหว่างค่า x_{1i} และ x_{2i} ได้โดยตรง แต่อย่างไรก็ดี ถ้าเราทำการหาความแตกต่างจากข้อมูล x_{1i} และ x_{2i} โดยตรง โดยที่ไม่ทำการประมวลผลใดๆ จะทำให้อาจเกิดความเอนเอียงในค่าความแตกต่างได้ (แอทริบิวที่มีค่าความแตกต่างมากๆจะชี้นำแอทริบิวอื่นๆทั้งหมด) ดังนั้นเราควรจะต้องทำการนอร์มัลไลซ์ข้อมูลเสียก่อน โดยการปรับข้อมูลให้อยู่ในช่วงที่กำหนดเช่น อยู่ในช่วง 0 – 1 ด้วย Min-max normailization ดังนี้

$$v' = \frac{v - min_A}{max_A - min_A}$$

เมื่อค่า min_A และ max_A คือ ค่าต่ำสุดและค่าสูงสุดที่ปรากฏขึ้นในแอทริบิว A ที่ทำการพิจารณา

หลังจากทำการหาค่าความแตกต่างระหว่างเรคคอร์ดที่ต้องการจำแนกหมวดหมู่กับเรคคอร์ดทั้งหมดในชุดข้อมูลสอนแล้ว เราจะทำการจำแนก/ทำนายหมวดหมู่ข้อมูลด้วยการพิจารณาจากหมวดหมู่ของข้อมูล k

เรคคอร์ดที่มีค่าความแตกต่างน้อยที่สุด k อันดับแรก (เรียกว่า 'เพื่อนบ้านใกล้สุด k อันดับ,k-nearest neighbor') เมื่อเทียบกับเรคคอร์ดที่ต้องการจำแนกหมวดหมู่ด้วยการค้นหาว่าใน k เรคคอร์ด เราพบเจอหมวดหมู่ข้อมูลใดมากที่สุด เราจะสรุปว่าเรคคอร์ดที่ต้องการจำแนกหมวดหมู่จะมีหมวดหมู่ตามหมวดหมู่ที่มากที่สุดที่พิจารณาก่อนหน้า

แต่ในการคำนวณหาค่าความแตกต่างระหว่างเรคคอร์ด เมื่อทำการพิจารณาแอทริบิวที่มีค่าที่ไม่ได้เป็นตัวเลข (มีค่าเป็นแบบไม่ต่อเนื่อง/มีค่าเป็นแบบหมวดหมู่) อาทิเช่น แอทริบิวค่าสี เราอาจใช้การเปรียบเทียบค่าระหว่างในแอทริบิวระหว่าง 2 เรคคอร์ด X_1 และ X_2 ได้โดยตรง โดยที่ถ้าค่าในแอทริบิวที่ทำการพิจารณาของทั้งสองเรคคอร์ดมีค่าเท่ากัน (เช่น ทั้งสองเรคคอร์ด X_1 และ X_2 มีค่าสีเป็นสีเขียว) เราจะทำการกำหนดให้ค่าความแตกต่างของเรคคอร์ด X_1 และ X_2 เมื่อทำการแอทริบิวค่าสีมีค่าเป็น 0 แต่ถ้าค่าของทั้งสองเรคคอร์ดมีค่าแตกต่างกัน (เช่น เรคคอร์ด X_1 มีค่าสีเป็นสีเขียว และ X_2 มีค่าสีเป็นสีส้ม) เราจะทำการกำหนดให้ค่าความแตกต่างมีค่าเท่ากับ 1 ตามลำดับ แต่อย่างไรก็ตามเราสามารถกำหนดค่าความแตกต่างให้กับการเปรียบเทียบแต่ละค่าได้ ตัวอย่างเช่น เราอาจจะกำหนดให้ค่าความแตกต่างระหว่างสีเขียวและสีดำให้มีค่าเท่ากับ 0.7 และอาจกำหนดให้ความแตกต่างระหว่างสีเขียวและสีฟ้าให้มีค่าเท่ากับ 0 ก็ได้ ขึ้นอยู่กับลักษณะและการใช้งานข้อมูล

ในกรณีที่ข้อมูลในแอทริบิว A ใดๆในเรคคอร์ X_1 และ/หรือ X_2 ที่พิจารณาไม่มีค่าที่ปรากฏ (X_1 และ/หรือ X_2 มี missing value เกิดขึ้นใน แอทริบิว A) เราจะทำการสมมติให้ค่าความแตกต่างของเรคคอร์ด X_1 และ X_2 เมื่อทำการแอทริบิว A มีค่าสูงสุดเท่าที่เป็นไปได้ สมติว่าค่าที่เป็นไปได้ในแอทริบิวที่ทำการพิจารณาจะสามารถถูกแปลงให้อยู่ในช่วงตัวเลข $[0,1]$ ดังนั้น เมื่อทำการพิจารณาแอทริบิวที่มีลักษณะเป็นแบบหมวดหมู่ (มีค่าแบบไม่ต่อเนื่อง) เราจะทำการกำหนดให้ค่าความแตกต่างมีค่าเท่ากับ 1 สำหรับกรณีที่มีการขาดหายไปของค่าในเรคคอร์ดใดเรคคอร์หนึ่งหรือมีการขาดหายไปของค่าในทั้งสองเรคคอร์ดที่ทำการพิจารณา แต่สำหรับสำหรับการขาดหายไปของค่าในแอทริบิวที่มีค่าเชิงตัวเลข (มีค่าแบบต่อเนื่อง) เราจะทำการกำหนดให้ค่าความแตกต่างมีค่าเท่ากับ 1 สำหรับกรณีที่มีการขาดหายไปของค่าในทั้งสองเรคคอร์ดที่ทำการพิจารณา แต่สำหรับการขาดหายไปของค่าในเรคคอร์ดใดเรคคอร์หนึ่ง เราจะสามารถทำการนอร์มัลไลซ์ข้อมูลได้ โดยการหาค่าผลต่างจาก $|1-v'|$หรือ $|1-v'|$ เมื่อ v' คือ ข้อมูลของเรคคอร์ดที่มีข้อมูล โดยวิธีการนอร์มัลไลซ์นี้จะช่วยให้เราหาค่าความแตกต่างได้อย่างสมบูรณ์มากขึ้น

นอกเหนือจากการพิจารณากรณีต่างๆข้างต้น เราจะต้องพิจารณาถึงปัจจัยต่างๆอีกมากมาย อาทิเช่น 1) จำนวนเพื่อนบ้าน k ที่ต้องทำการพิจารณา ว่าควรมีค่าหรือมีจำนวนเท่าที่เหมาะสมเท่ากับเท่าไร โดยในการที่จะกำหนดค่า k ที่เหมาะสมนั้นจะสามารถทำได้ในการทดลอง โดยจะเริ่มจากการกำหนดให้ค่า k แล้วทำการทดสอบถึงเปอร์เซ็นต์ของความผิดพลาดในการจำแนกข้อมูล ต่อมาจะทำการเพิ่มค่า k แล้วทำการวนการจำแนกข้อมูลซ้ำเดิมอีก กระบวนการทำงานจะสามารถถูกทำซ้ำไปเรื่อยๆตามค่า k สูงสุดที่เรากำหนด จากนั้นเราจะทำการเลือกค่า k ที่ให้เปอร์เซ็นต์ของความผิดพลาดในการจำแนกข้อมูลน้อยที่สุด ซึ่งโดยทั่วๆไปของการทำงาน เราจะสังเกตได้ว่า เมื่อจำนวนเรคคอร์ในชุดข้อมูลสอนมีจำนวนมาก จะทำให้ค่า k ที่ให้เปอร์เซ็นต์

ของความผิดพลาดในการจำแนกข้อมูลน้อยที่สุดมีค่าสูงตามไปด้วย 2) การหาค่าความแตกต่างของเรคคอร์ดที่ทำการพิจารณา โดยทำการพิจารณาแอทริบิวต่างๆอาจจะให้ผลลัพธ์ที่ไม่ดีหรืออาจให้เปอร์เซ็นต์ของความผิดพลาดในการจำแนกข้อมูลที่ไม่ดีอันเนื่องมาจากข้อมูลรบกวน (noise) หรือจากการที่เราทำการพิจารณาแอทริบิวที่ไม่เกี่ยวข้องมากจนเกินไป ดังนั้นเราอาจจำเป็นต้องคำนึงถึงหรือทำการเลือกมาตรวัดความแตกต่างอื่นๆเป็นทางเลือกเสริม และ 3) ประสิทธิภาพในการจำแนกข้อมูล กล่าวคือ ในการจำแนกข้อมูลอาจใช้เวลานานเมื่อมีจำนวนเรคคอร์ดในชุดข้อมูลสอนเป็นจำนวนมาก และเราทำการกำหนดค่า k สูง ดังนั้น เราอาจจะต้องทำการลดเวลาในการคำนวณโดยทำการจัดเรียงลำดับของข้อมูลเรคคอร์ดต่างๆไว้ก่อนหน้า โดยอาจทำการเก็บไว้ในต้นไม้ที่ซึ่งจะทำให้เราสามารถลดการเปรียบเทียบหาความแตกต่างระหว่างเรคคอร์ดใดๆได้ โดยประสิทธิภาพที่ใช้ในการทำการจะเป็น $O(\log|D|)$ หรือ เราอาจประยุกต์ใช้วิธีการทำงานแบบขนาน (Parallel) ในการคำนวณเพื่อที่จะลดการเวลาในการทำลงก็ได้

6.9 การทำนายข้อมูล (Prediction)

ในการจำแนกข้อมูลเราจะทำการพิจารณาเรคคอร์ดของข้อมูลที่ไม่ทราบหมวดหมู่ที่แน่นอน แล้วทำการกำหนดหรือบ่งบอกหมวดหมู่ของข้อมูลให้แก่เรคคอร์ดนั้นๆที่ซึ่งหมวดหมู่ของข้อมูลจะมีค่าเป็นแบบไม่ต่อเนื่อง แต่ในบางสถานการณ์เราอาจต้องการที่จะทำนายหมวดหมู่ของข้อมูลที่มีค่าเป็นแบบต่อเนื่อง/ค่าที่บ่งบอกในเชิงปริมาณ ตัวอย่างเช่น เราอาจจะต้องการทำนายค่าเงินเดือนของพนักงานคนหนึ่งๆที่มีประสบการณ์ทำงานมาแล้ว 10 ปี หรือ ต้องการที่จะทำนายยอดขายเมื่อกำหนดราคาขายของสินค้ามาให้ จากความต้องการดังกล่าว เราจะต้องทำการทำนายค่าเชิงตัวเลขโดยการใช้หลักการทางสถิติที่เรียกว่า การถดถอย (regression)

การวิเคราะห์การถดถอยจะเป็นโมเดลสำหรับการหาความสัมพันธ์ระหว่าง “predictor variable” ที่ซึ่งเป็นค่าในแอทริบิวต่างๆที่ใช้ในการอธิบายคุณลักษณะของข้อมูลเรคคอร์ดหนึ่งๆ กับ “response variable” ที่ซึ่งจะเป็นค่าเชิงตัวเลขที่เราต้องการที่จะทราบหรือทำนาย การวิเคราะห์ความถดถอยจะสามารถทำงานได้ดีหรือเป็นทางเลือกที่ดีในการทำนายข้อมูล ที่ซึ่งค่าในแอทริบิวต่างๆมีค่าในเชิงตัวเลข การทำงานของการวิเคราะห์การถดถอยจะสามารถแบ่งได้เป็นหลายกรณีตามชนิดของข้อมูลที่ทำการพิจารณาและตามผลลัพธ์ที่ต้องการ โดยจะสามารถแบ่งได้เป็นกรณีต่างๆดังนี้

6.9.1 การวิเคราะห์การถดถอยเชิงเส้นตรง

การวิเคราะห์การถดถอยเชิงเส้นตรง (Linear regression analysis) จะเป็นการทำนายข้อมูลที่มีค่าเชิงตัวเลขที่เกี่ยวข้องกับ response variable “y” โดยการพิจารณาค่า predictor variable “x” เพียงแค่ค่าเดียวด้วยการประยุกต์ใช้ฟังก์ชันเชิงเส้นตรง (Linear function) ที่ซึ่งสามารถคำนวณได้จาก

$$y = b + wx$$

เมื่อ b คือ ค่าสัมประสิทธิ์ความถดถอยที่ซึ่งจะเป็นตัวกำหนดจุดตัดแกน y (y-intercept) และ w คือ ค่าสัมประสิทธิ์ความถดถอยที่ซึ่งจะเป็นตัวกำหนดความลาดเอียงของเส้นตรง โดยจากสมการข้างต้น เราสามารถพิจารณาค่าสัมประสิทธิ์ b และ w เป็นค่าน้ำหนักที่ซึ่งจะสามารถเปลี่ยนการเขียนสมการได้เป็น

$$y = w_0 + w_1 x$$

จากนั้นเราจะต้องทำการคำนวณค่าสัมประสิทธิ์ต่างๆด้วยวิธีการ “least squares” ที่ซึ่งจะทำการประมาณเส้นตรงที่ดีที่สุดที่จะลดค่าความผิดพลาดระหว่างข้อมูลจริงและเส้นตรงที่ถูกประมาณให้มีค่าน้อยที่สุดเท่าที่จะเป็นไปได้ โดยในการพิจารณาชุดข้อมูลสอนใดๆก็ตามที่ประกอบไปด้วยค่า predictor variable (x) ที่ซึ่งจะสอดคล้องกับค่า response variable (y) ค่าสัมประสิทธิ์ต่างๆจะสามารถคำนวณได้ดังนี้

$$w_1 = \frac{\sum_{i=1}^{|D|}(x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{|D|}(x_i - \overline{x})^2}$$

$$w_0 = \overline{y} - w_1 \overline{x}$$

เมื่อ $\overline{x}$ คือ ค่าเฉลี่ยของค่า $x_1, x_2, \dots, x_{|D|}$ ที่ทำการพิจารณา และ $\overline{y}$ คือ ค่าเฉลี่ยของค่า $y_1, y_2, \dots, y_{|D|}$ ที่ทำการพิจารณา

ตัวอย่าง การทำนายข้อมูลด้วยการวิเคราะห์การถดถอยเชิงเส้นตรง จากข้อมูลอินพุตในตารางที่ 6.6 ที่ซึ่งจะแสดงข้อมูลประสบการณ์ทำงาน (predictor variable) และค่าเงินเดือนที่ได้รับ (response variable)

ตารางที่ 6.6 ตัวอย่างข้อมูลประสบการณ์ทำงานและเงินเดือนที่ได้รับ

x ประสบการณ์ทำงาน (ปี)	y เงินเดือนที่ได้รับ ($\times \$1{,}000s$)
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
6	83

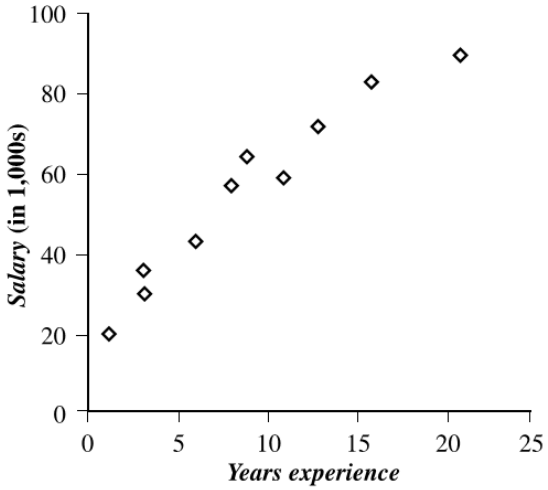
เมื่อเรานำข้อมูลจากตาราง 6.6 มาสร้างกราฟ เราจะได้กราฟดังแสดงในรูปที่ 6.15 ที่ซึ่งจะแสดงให้เราเห็นถึงความสัมพันธ์ระหว่างตัวแปร x และ y โดยความความสัมพันธ์ระหว่างเงินเดือนที่ได้รับกับจำนวนปีที่ทำงานจะสามารถคำนวณได้จาก $y = w_0 + w_1 x$ เมื่อเราการพิจารณาข้อมูลในตารางที่ 6.6 เราจะสามารถ

คำนวณค่าเฉลี่ยที่ปรากฏใน predictor variable x และ response variable y ได้เป็น $\overline{x} = 9.1$ และ $\overline{y} = 55.4$ ที่ซึ่งจะสามารถนำไปคำนวณค่าน้ำหนัก w_0 และ w_1 ได้ดังนี้

$$w_1 = \frac{(3-9.1)(30-55.4)+(8-9.1)(57-55.4)+\ldots+(16-9.1)(83-55.4)}{(3-9.1)^2+(8-9.1)^2+\ldots+(16-9.1)^2} = 3.5$$

$$w_0 = 55.4-(3.5)(9.1) = 23.6$$

จากนั้นเราจะสามารถทำนายเงินเดือนที่จะได้รับเมื่อทราบจำนวนปีที่ทำงานได้โดยใช้สมการ $y = 23.6+3.5x$ ดังนั้นเราสามารถทำนายค่าเงินเดือนของพนักงานที่ทำงานมาเป็นเวลา 10 ปีได้เท่ากับ $23.6+3.5(10) = \$58{,}600$



รูปที่ 6.15 ตัวอย่างแผนภาพกราฟจากข้อมูลประสบการณ์ทำงานและเงินเดือนที่ได้รับ

การวิเคราะห์การถดถอยพหุคูณ (Multiple linear regression) จะเป็นการพัฒนาต่อยอดจากการวิเคราะห์การถดถอยเชิงเส้นตรง โดยจะทำการพิจารณาค่า predictor variable หลายๆค่า ที่ซึ่งจะหมายถึงการพิจารณาแอทริบิว $A_1, A_2, \ldots, A_n$ ที่ใช้ในการอธิบายข้อมูลเรคคอร์ด X ใดๆ ดังนั้นเรคคอร์ดของข้อมูล X ใดๆจะประกอบไปด้วย $x_1, x_2, \ldots, x_n$ ที่เป็นค่าที่เกิดขึ้นในแอทริบิวที่ทำการพิจารณา เมื่อเราทำการพิจารณาชุดข้อมูล D ที่ประกอบไปด้วยเรคคอร์ด $X_1 = (x_{1,1}, x_{1,2}, \ldots, x_{1,n}, y_1), X_2 = (x_{2,1}, x_{2,2}, \ldots, x_{2,n}, y_2),$ $\ldots, X_{|D|} = (x_{|D|,1}, x_{|D|,2}, \ldots, x_{|D|,n}, y_{|D|})$ เราจะสามารถทำนายข้อมูลด้วยการวิเคราะห์การถดถอยพหุคูณด้วยการใช้ค่า predictor variable หลายๆค่าที่ซึ่งจะสามารถคำนวณได้จาก

$$y = w_0 + w_1x_1 + w_2x_2+\ldots+w_nx_n$$

เมื่อ ค่าน้ำหนัก $w_1, w_2, \ldots, w_n$ จะสามารถคำนวณได้ด้วยวิธีการเดียวกับการวิเคราะห์การถดถอยเชิงเส้นตรงด้วยการพิจารณาแอทริบิวต่างๆได้โดยตรง

6.9.2 การวิเคราะห์การถดถอยแบบไม่เป็นเส้นตรง (Nonlinear Regression analysis)

ในบางกรณี เราอาจต้องทำการจัดการกับข้อมูลที่ไม่สามารถทำนายได้โดยเส้นตรงหรือความสัมพันธ์เชิงเส้น แต่เราอาจจัดการได้โดยการเปลี่ยนฟังก์ชัน เช่น เปลี่ยนจากฟังก์ชันเชิงเส้นตรงไปเป็นฟังก์ชันโพลีโนเมียล เป็นต้น เราจะสามารถทำการแปลงฟังก์ชันโพลีโนเมียลให้กลายเป็นฟังก์ชันเชิงเส้นตรงที่ซึ่งจะทำให้เราสามารถจัดการกับข้อมูลได้ง่ายขึ้น ลองพิจารณาตัวอย่างดังต่อไปนี้

ตัวอย่าง การแปลงการวิเคราะห์การถดถอยแบบโพลีโนเมียลไปเป็นการวิเคราะห์การถดถอยเชิงเส้นตรง ด้วยการพิจารณาความสัมพันธ์ของลูกบาศ์กโพลีโนเมียล $y = w_0 + w_1x + w_2x^2 + w_3x^3$ นี้แล้วทำการแปลงให้ไปเป็นการวิเคราะห์การถดถอยเชิงเส้นตรงที่ซึ่งจะสามารถทำได้โดยการกำหนดตัวแปรขึ้นใหม่โดยการกำหนดให้ $x_1 = x, x_2 = x^2$ และ $x_3 = x^3$ จากนั้นเราจะสามารถทำการวิเคราะห์การถดถอยเชิงเส้นตรงได้โดย $y = w_0 + w_1x_1 + w_2x_2 + w_3x_3$ ที่ซึ่งสามารถคำนวณได้โดยง่าย

นอกเหรือจากฟังก์ชันโพลีโนเมียลแล้ว เราอาจเปลี่ยนการใช้ฟังก์ชันไปเป็นฟังก์ชันอื่นๆได้อีก แต่อย่างไรก็ตามเมื่อทำการเปลี่ยนฟังก์ชันแล้ว เราจะต้องทำการเปลี่ยน/แปลงการใช้ฟังก์ชันเหล่านั้นให้กลับมาเป็นฟังก์ชันเชิงเส้นตรงที่ซึ่งจะทำให้เราสามารถทำนายข้อมูลได้อย่างแม่นยำ แต่อย่างไรก็ดีในการทำด้วยการวิเคราะห์การถดถอยจะมีปัจจัยหนึ่งที่เราต้องพิจารณาคือ ประสิทธิภาพในการคำนวณ ที่ซึ่งเมื่อเราพิจารณาแอทริบิวหรือ predictor variable มากขึ้นก็ย่อมจะทำให้ประสิทธิภาพในการคำนวณลดลง ดังนั้น ก่อนที่จะทำการวิเคราะห์การถดถอย เราควรที่จะต้องทำการเลือกแอทริบิวที่เหมาะสมกับการวิเคราะห์ โดยทำการเลือกแอทริบิวเพียงบางส่วน และตัดทอนแอทริบิวบางส่วนที่จะไม่ส่งผลต่อการคำนวณ นอกจากนั้นความถูกต้องของการวิเคราะห์ก็เป็นอีกปัจจัยหนึ่งที่เราจะต้องพิจารณา ที่ซึ่งการวิเคราะห์การถดถอยจะสามารถทำนายข้อมูลได้อย่างแม่นยำในกรณีที่ข้อมูลที่เป็นอินพุตจะไม่มีข้อมูลแปลกปลอม (outlier)

6.10 การประเมินความถูกต้องและความผิดพลาด (Accuracy and error measures)

ในการสร้างและใช้งานตัวจำแนกข้อมูลเราจะต้องทำการประเมินความถูกต้องและความผิดพลาดของตัวจำแนกข้อมูล ที่ซึ่งจะสื่อให้เราทราบว่าตัวจำแนกข้อมูลที่สร้างขึ้นนั้นมีประสิทธิภาพในการจำแนกข้อมูลอย่างไร ดังนั้น เนื้อหาในส่วนนี้จะเป็นการศึกษาเกี่ยวกับการชี้วัดค่าความถูกต้องของตัวจำแนกข้อมูล เทคนิคต่างๆในการประเมินความถูกต้องของตัวจำแนกข้อมูล วิธีการในการเพิ่มความถูกต้องให้กับตัวจำแนกข้อมูล และ อภิปรายเกี่ยวกับการเลือกวิธีการในการจำแนกข้อมูลว่าเราควรที่จะเลือกโดยใช้วิธีการใดในกรณีใด

6.10.1 ตัวชี้วัดความผิดพลาดจากการจำแนกข้อมูล

หลังจากทำการสร้างตัวจำแนกข้อมูล เราจะต้องทำการตรวจสอบความถูกต้องในการจำแนกข้อมูลเพื่อทำการประเมินว่าค่าความถูกต้องที่จะได้รับจากตัวจำแนกข้อมูลที่สร้างขึ้นสามารถยอมรับได้หรือไม่ โดยใน

การตรวจสอบ***ความถูกต้องในการจำแนกข้อมูล*** เราจะใช้ข้อมูลทดสอบที่มีหมวดหมู่ของข้อมูลแนบอยู่ด้วย (เป็นข้อมูลอีกชุดหนึ่งนอกเหนือจากชุดข้อมูลสอน เมื่อทำการทดสอบแล้วเราจะทราบถึงจำนวนหรือเปอร์เซ็นต์ของเรคคอร์ดในชุดข้อมูลทดสอบที่สามารถจำแนกข้อมูลได้อย่างถูกต้อง นอกจากนั้นเรายังต้องทำการพิจารณาถึง***อัตราความผิดพลาดหรืออัตราการจำแนกข้อมูลผิด*** (misclassification rate) ของตัวจำแนกข้อมูล M ใดๆที่สร้างขึ้น ที่ซึ่งจะสามารถคำนวณได้จาก $1 - acc(M)$ เมื่อ $acc(M)$ คือ ค่าความถูกต้องในการจำแนกข้อมูลของตัวจำแนกข้อมูล M

ในการที่จะวิเคราะห์หาความถูกต้องและความผิดพลาดในการจำแนกข้อมูล เราอาจทำการประยุกใช้ “confusion matrix” ที่ซึ่งจะเป็นตารางที่มีขนาด $m \times m$ โดยข้อมูลแต่ละเซลล์ $cm_{i,j}$ หมายถึง จำนวนเรคคอร์ดที่มีหมวดหมู่ i แต่ถูกตัวจำแนกข้อมูลจำแนกหมวดหมู่เป็นหมวดหมู่ j โดยที่ถ้าตัวจำแนกข้อมูลที่เราทำการทดสอบเป็นตัวจำแนกข้อมูลที่มีความถูกต้องสูง จะทำให้เซลล์ $cm_{1,1}$, $cm_{2,2}$ ไปจนถึง เซลล์ $cm_{m,m}$ จะมีจำนวนเรคคอร์ดเป็นจำนวนมากและเซลล์ $cm_{i,j}$ อื่นๆจะมีค่าเท่ากับ 0 หรือมีค่าใกล้ๆ 0 ตัวอย่างเช่น ถ้าชุดข้อมูลที่ทำการพิจารณามีหมวดหมู่ข้อมูล 2 หมวดหมู่ เมื่อทำการพิจารณาเรคคอร์ดต่างๆในชุดข้อมูล เราอาจทำการแบ่งข้อมูลเรคคอร์ดต่างๆเป็น “***positive tuples***” (เป็นเรคคอร์ดที่มีหมวดหมู่ข้อมูลเป็นหมวดหมู่ที่เราสนใจ อาทิ เช่น หมวดหมู่ของลูกค้าที่ซื้อคอมพิวเตอร์) และ “***negative tuples***” (เช่น หมวดหมู่ของลูกค้าที่ไม่ซื้อคอมพิวเตอร์) จากนั้นเมื่อทำการจำแนกข้อมูลเราจะพบเจอกับ

1) ***True positives*** ที่ซึ่งเป็น *positive tuples* ที่ถูกจำแนกหมวดหมู่ได้อย่างถูกต้อง
2) ***True negatives*** ที่ซึ่งเป็น *negative tuples* ที่ถูกจำแนกหมวดหมู่ได้อย่างถูกต้อง
3) ***False positives*** ที่ซึ่งเป็น *negative tuples* ที่ถูกจำแนกข้อมูลผิดพลาด (เช่น เรคคอร์ดของข้อมูลที่มีหมวดหมู่ที่ไม่ซื้อคอมพิวเตอร์แต่ถูกจำแนกโดยตัวจำแนกข้อมูลว่าเป็นหมวดหมู่ที่ซื้อคอมพิวเตอร์)
4) ***False negatives*** ที่ซึ่งเป็น *positive tuples* ที่ถูกจำแนกข้อมูลผิดพลาด (เช่น เรคคอร์ดของข้อมูลที่มีหมวดหมู่ที่ซื้อคอมพิวเตอร์แต่ถูกจำแนกโดยตัวจำแนกข้อมูลว่าเป็นหมวดหมู่ที่ไม่ซื้อคอมพิวเตอร์)

โดยกรณีที่พบทั้ง 4 ข้อมูลสามารถนำมาสร้างเป็น confusion matrix (ดังแสดงในรูปที่ 6.16) เพื่อนำมาวิเคราะห์หาความถูกต้องและความผิดพลาดของการจำแนกข้อมูลต่อไปได้

		Predicted class	
		C_1	C_2
Actual class	C_1	true positives	false negatives
	C_2	false positives	true negatives

รูปที่ 6.16 ตัวอย่าง confusion matrix สำหรับ positive และ negative tuples

การประยุกต์ใช้เปอร์เซ็นต์ความถูกต้องในการจำแนกข้อมูลได้ถูกประยุกต์ใช้อย่างแพร่หลาย แต่อย่างไรก็ดี การใช้เพียงแค่เปอร์เซ็นต์ความถูกต้องเพียงอย่างเดียวอาจจะประเมินความถูกต้องของตัวจำแนกข้อมูลได้ไม่ดีนัก ตัวอย่างเช่น เมื่อเราทำการพิจารณาข้อมูลสอนที่เกี่ยวข้องกับหมวดหมู่ของผู้ป่วยที่เป็นมะเร็งและไม่เป็นมะเร็ง เมื่อทำการทดสอบปรากฏว่าเปอร์เซ็นต์ความถูกต้องมีค่าเท่ากับ 90% ซึ่งมีค่าสูง แต่อย่างไรก็ดี เมื่อพิจารณาลึกลงไปอีกเราพบว่ามีเพียงแค่ 3-4% ของเรคคอร์ดทั้งหมดที่พิจารณาเท่านั้น ที่อยู่ในหมวดหมู่ของผู้ป่วยที่เป็นมะเร็ง ดังนั้นเราจะเห็นว่าค่าเปอร์เซ็นต์ความถูกต้อง 90 เปอร์เซ็นต์ไม่อาจถูกใช้เพื่อวัดความถูกต้องของตัวจำแนกข้อมูลได้อย่างแม่นยำ—โดยค่าความถูกต้อง 90 เปอร์เซ็นต์อาจเกิดจากการที่ตัวจำแนกข้อมูลสามารถจำแนกหรือทำนายหมวดหมู่ของเรคคอร์ดขอผู้ป่วยที่ไม่เป็นโรคมะเร็งได้อย่างถูกต้อง แต่นั่นจะไม่ใช่สิ่งสำคัญเท่าไรนัก เนื่องจากวัตถุประสงค์หลักของเราคือต้องการระบุ/จำแนกผู้ป่วยที่เป็นโรคมะเร็ง ด้วยเหตุนี้เราอาจทำการพิจารณาค่า “***sensitivity***” และ “***specificity***” ที่ซึ่งค่า sensitivity จะอ้างถึงอัตราส่วนระหว่างจำนวน positive tuples ที่ถูกจำแนกได้อย่างถูกต้อง แต่สำหรับค่า specificity จะถูกคำนวณจากอัตราส่วนระหว่างจำนวน negative tuples ที่ถูกจำแนกได้อย่างถูกต้อง นอกจากนั้นอาจจะประยุกต์ใช้ค่า “***precision***” เพื่อใช้ในการคำนวณเปอร์เซ็นต์ของเรคคอร์ดของหมวดหมู่ที่เป็น positive ที่ซึ่งเป็นถูกจำแนกเป็นหมวดหมู่ positive โดยค่าทั้ง 3 จะสามารถคำนวณได้ดังนี้

$$sensitivity = \frac{|true\ positives|}{|postive\ tuples|}$$

$$specificity = \frac{|true\ negatives|}{|negative\ tuples|}$$

$$precision = \frac{|true\ positive|}{(|true\ positive|+|false\ positive|)}$$

จากการคำนวณค่าต่างๆข้างต้น เราสามารถคำนวณค่าความถูกต้องของตัวจำแนกข้อมูลได้ดังนี้

$$accuracy = sensitivity\frac{|positive\ tuples|}{(|postive\ tuples|+|negative\ tuples|)}$$

$$+specificity\frac{|negative\ tuples|}{(|postive\ tuples|+|negative\ tuples|)}$$

6.10.2 ตัวชี้วัดความผิดพลาดจากการทำนายข้อมูล

กำหนดให้ D^T คือ ชุดข้อมูลทดสอบที่อยู่ในรูปแบบ $(X_1, y_1), (X_2, y_2), ..., (X_d, y_d)$ ที่ซึ่ง X_i เป็นเซตของเรคคอร์ดที่มีหมวดหมู่ y_i และ d คือ จำนวนเรคคอร์ดทั้งหมดในชุดข้อมูลทดสอบ D^T โดยในการทำนายข้อมูลจะคืนค่าเป็นเปอร์เซ็นต์ของความเป็นหมวดหมู่ข้อมลหนึ่งๆ แทนที่จะบอกถึงหมวดหมู่ของข้อมูลหนึ่งๆที่แน่นอน การประเมินความถูกต้องในการทำนายหมวดหมู่ข้อมูล y_i' สำหรับ X_i จะสามารถทำได้ค่อนข้างยาก ดังนั้นในการประเมินความถูกต้อง เราจะไม่ทำการทำนายหมวดหมู่ y_i' สำหรับ X_i แล้วนำไปเทียบกับหมวดหมู่ y_i ที่แท้จริงของ X_i แต่เราจะทำการพิจารณาหาความแตกต่างระหว่างผลลัพธ์ที่ได้จากการทำนายกับค่าที่แท้จริง โดยเราสามารถใช้ “***Loss functions***” ในการตรวจวัดความผิดพลาดระหว่างหมวดหมู่ที่แท้จริง y_i เทียบกับหมวดหมู่ที่ได้จากการทำนาย y_i' ที่ซึ่งจะมีฟังก์ชันการทำงานดังนี้

$$Absolute\ Error: |y_i - y_i'|$$

$$Squared\ Error: (y_i - y_i')^2$$

จากฟังก์ชันข้างต้น เราสามารถคำนวณหาข้อผิดพลาดในการทดสอบ (error rate) ที่ซึ่งเป็นค่าเฉลี่ยของความผิดพลาดของการทำนายหมวดหมู่ของทุกเรคคอร์ดในชุดข้อมูลทดสอบได้ดังนี้

$$Mean\ Absolute\ Error: \frac{\sum_{i=1}^{d} |y_i - y_i'|}{d}$$

$$Mean\ Squared\ Error: \frac{\sum_{i=1}^{d} (y_i - y_i')^2}{d}$$

แต่อย่างไรก็ดี ในบางครั้ง เราอาจจะต้องการที่จะทำการคำนวณหาค่าความผิดพลาดที่มีความสัมพันธ์กับสิ่งที่จะเป็นเมื่อเราทำการทำนายหมวดหมู่ข้อมูล ที่ซึ่งจะเป็นการคำนวณอัตราส่วนระหว่างผลรวมของความแตกต่างของผลลัพธ์จากการทำนายกับค่าจริงเทียบกับการสูญเสียที่เกิดจากการทำนายข้อมูล โดยค่าความสัมพันธ์กับค่าความผิดพลาดสามารถคำนวณได้ดังนี้

$$Relative\ Absolute\ Error: \frac{\sum_{i=1}^{d} |y_i - y_i'|}{\sum_{i=1}^{d} |y_i - \bar{y}|}$$

$$Relative\ Squared\ Error: \frac{\sum_{i=1}^{d} (y_i - y_i')^2}{\sum_{i=1}^{d} (y_i - \bar{y})^2}$$

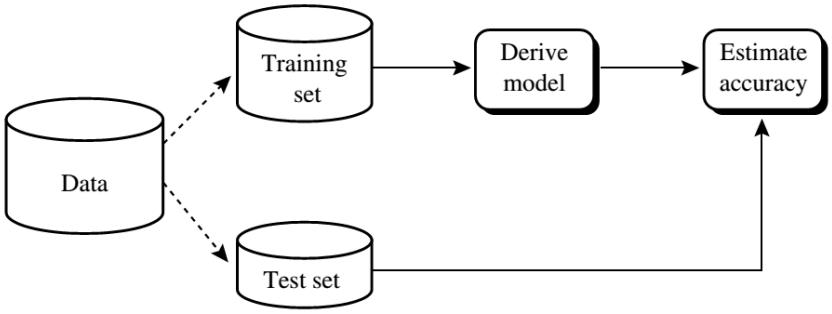
เมื่อ $\bar{y}$ คือ ค่าเฉลี่ยของค่าหมวดหมู่ y_i ในชุดข้อมูลสอน ที่ซึ่งสามารถคำนวณได้จาก $\bar{y} = \frac{\sum_{i=1}^{t} y_i}{d}$

6.10.3 การประเมินความถูกต้องของตัวจำแนกข้อมูลหรือตัวทำนายข้อมูล

จากตัวชี้วัดความถูกต้องของการจำแนก/ทำนายข้อมูลข้างต้น เราจะสามารถประยุกต์ใช้ตัวชี้วัดเหล่านั้นเพื่อที่จะประเมินความถูกต้องของตัวจำแนกข้อมูลหรือตัวทำนายข้อมูลได้อย่างน่าเชื่อถือได้อย่างไร??? ดังนั้นในการที่จะประเมินความถูกต้องของตัวจำแนกข้อมูลหรือตัวทำนายข้อมูลได้อย่างน่าเชื่อถือ เราสามารถประยุกต์ใช้เทคนิคต่างๆที่ได้รับความนิยมกันอย่างแพร่หลายดังต่อไปนี้

วิธีการ Holdout

Holdout จะทำการแบ่งชุดข้อมูลออกเป็น 2 ชุดข้อมูลย่อยด้วยวิธีการสุ่ม โดยชุดข้อมูลที่ได้จะเป็นชุดข้อมูลสอนและชุดข้อมูลทดสอบ ซึ่งโดยปกติ—ชุดข้อมูลสอนจะมีปริมาณข้อมูลเท่ากับ 2 ใน 3 ของชุดข้อมูลทั้งหมด และ ชุดข้อมูลทดสอบจะมีปริมาณข้อมูลเท่ากับ 1 ใน 3 ของชุดข้อมูล หลังจากแบ่งชุดข้อมูลแล้ว—ชุดข้อมูลสอนจะถูกใช้ในการสร้างตัวจำแนกข้อมูล และชุดข้อมูลทดสอบจะถูกใช้ในการทำสอบตัวจำแนกที่สร้างขึ้น (ดังแสดงในรูปที่ 6.17)



รูปที่ 6.17 การประเมินความถูกต้องด้วยวิธีการ Holdout

วิธีการ Cross-validation

วิธีการ Cross-valiation หรือที่นิยมเรียกว่า "***k-fold cross validation***" จะเริ่มจากการสุ่มแบ่งข้อมูลออกเป็น k ส่วน ที่ซึ่งจะมีข้อมูลไม่ซ้ำกัน โดยชุดข้อมูลย่อย $D_1, D_2, \dots, D_k$ จะมีจำนวนเรคคอร์ดเท่ากัน จากนั้นจะทำการเรียนรู้และทดสอบตัวจำแนกข้อมูลทั้งสิ้น k ครั้ง โดยในการทำงานในรอบที่ i เราจะกำหนดให้ชุดข้อมูลย่อย D_i ให้เป็นชุดข้อมูลทดสอบ และชุดข้อมูลอื่นๆเป็นชุดข้อมูลสอน ตัวอย่างเช่น ในการทำงานรอบแรก ชุดข้อมูลย่อย $D_2, \dots, D_k$ จะเป็นชุดข้อมูลสอน และชุดข้อมูลย่อย D_1 จะเป็นชุดข้อมูลทดสอบ แต่สำหรับการทำงานในรอบที่สอง ชุดข้อมูลย่อย $D_1, D_3, \dots, D_k$ จะเป็นชุดข้อมูลสอน และชุดข้อมูลย่อย D_2 จะเป็นชุดข้อมูลทดสอบ ตามลำดับ

การทำงานของ *k-fold cross validation* จะแตกต่างจากวิธีการ *Hold out* และ *Random subsampling* ตรงที่ข้อมูลแต่ละเรคคอร์ดจะถูกพิจารณาเป็นจำนวนเท่าๆกันที่ซึ่งจะทำหน้าที่เป็นข้อมูลสอนทั้งสิ้น $k-1$ และทำหน้าที่เป็นข้อมูลทดสอบ 1 ครั้ง โดยการประเมินค่าความถูกต้องจะสามารถทำได้ในแต่ละรอบที่ k^{th} ของการทำงาน โดยในการประเมินค่าความถูกต้องจากการจำแนกข้อมูลในการทำงานรอบที่ k^{th} จะสามารถคำนวณได้จากจำนวนเรคคอร์ดที่ถูกจำแนกได้อย่างถูกต้องหารด้วยจำนวนเรคคอร์ดทั้งหมดในชุดข้อมูล แต่สำหรับการทำนายข้อมูล เราจะสามารถประเมินค่าความผิดพลาดได้จากจำนวนค่าความสูญเสียในรอบที่ k^{th} หารด้วยจำนวนเรคคอร์ดทั้งหมดในชุดข้อมูล (ข้อสังเกตุ โดยส่วนใหญ่ของการทำลองมักจะกำหนดค่า $k = 10$ ที่ซึ่งจะเรียกว่า 10-fold cross-validation)

วิธีการ Bootstrap

Bootstrap จะทำการสุ่มเลือกเรคคอร์ดที่ใช้สำหรับสอนด้วยการทำ "replacement" โดยที่ เมื่อเราทำการเลือกเรคคอร์ดหนึ่งๆเพื่อเป็นส่วนประกอบของชุดข้อมูลสอน หลังจากนั้นเราอาจจะทำเลือกเรคคอร์ดเดิมอีกครั้งก็เป็นได้ (วิธีการนี้จะสามารถเลือก/พิจารณาเรคคอร์ดหนึ่งๆได้มากกว่าหนึ่งครั้ง)

6.10.4 วิธีการเพิ่มความถูกต้อง

วิธีการเพิ่มความถูกต้องให้กับตัวจำแนก/ทำนายข้อมูลที่ได้รับความนิยมจะมีอยู่ 2 วิธีหลัก คือ “Bagging” และ “Boosting” ที่ซึ่งจะเป็นการนำโมเดลต่างๆมาทำงานร่วมกัน ที่ซึ่งจะมีรายละเอียดดังนี้

วิธีการ Bagging

Bagging จะเริ่มจากการแบ่งชุดข้อมูลออกเป็นชุดข้อมูลย่อยๆด้วยวิธีการ Bootstrap สมมติว่าเรากำหนดให้ k คือจำนวนชุดข้อมูลย่อยที่ต้องการ เมื่อทำการแบ่งชุดข้อมูลเราจะได้ $D_1, D_2, \dots, D_k$ เป็นชุดข้อมูลย่อย จากนั้นจะทำการพิจารณาแต่ละชุดข้อมูลย่อย D_i แล้วทำการสร้างตัวจำแนกข้อมูล M_i ด้วยวิธีการต่างๆ (อาทิเช่น ต้นไม้ตัดสินใจ การจำแนกข้อมูลด้วยเบย์เซียน โครงข่ายประสาทเทียมและการส่งค่าย้อนกลับ และวิธีการอื่นๆ) โดยทำการเลือกวิธีการใดวิธีการหนึ่งๆต่อชุดข้อมูลย่อย D_i หนึ่งๆ เมื่อทำการสร้างตัวจำแนกข้อมูให้แก่ทุกชุดข้อมูลย่อยแล้ว จะเข้าสู่การจำแนก/การทำนายข้อมูล โดยเมื่อทำการพิจารณาข้อมูลเรคคอร์ด X ใดๆ ถ้างานที่ทำเป็นการจำแนกข้อมูล เราจะทำการจำแนกข้อมูลจากทุกตัวจำแนกข้อมูล $M, M_2, \dots, M_k$ ที่ซึ่งจะได้หมวดหมู่ข้อมูลหนึ่งๆจากแต่ละตัวจำแนกข้อมูล จากนั้นเราจะทำการคืนค่าหมวดหมู่ที่ปรากฏมากที่สุดในบรรดาตัวจำแนกข้อมูลทั้งหมด แต่ถ้างานที่ทำเป็นการทำนายข้อมูล เราจะต้องทำการทำนายข้อมูลจากทุกๆตัวทำนายข้อมูล $M, M_2, \dots, M_k$ ที่ซึ่งจะได้ค่าทศนิยมของเรคคอร์ด X (โดยจะได้ 1 ค่าต่อการดำเนินการกับ M_i หนึ่งๆ) หลังจากนั้นจะทำการหาค่าเฉลี่ยของผลการทำนายที่ได้จากทุกตัวทำนายข้อมูล $M, M_2, \dots, M_k$ แล้วคืนค่าให้แก่ผู้ใช้ต่อไป

วิธีการ Boosting

Boosting จะเป็นวิธีการที่คล้ายๆกับ Bagging ที่ซึ่งจะทำการแบ่งข้อมูลออกเป็น k ชุดข้อมูลย่อย ที่ซึ่งเราจะได้ชุดข้อมูลย่อยเป็น $D_1, D_2, \dots, D_k$ จากนั้นจะทำการกำหนดค่าน้ำหนักให้แต่ละชุดข้อมูลย่อย จากนั้นจะใช้แต่ละชุดข้อมูลย่อย D_i เพื่อสร้างตัวจำแนกข้อมูล M_i แล้วจะทำการปรับค่าน้ำหนักเพื่อให้ที่จะทำให้ตัวจำแนกข้อมูลตัวที่ M_{i+1} จะถูกพิจารณาอย่างระมัดระวังมากยิ่งขึ้นกับเรคคอร์ดที่ถูกทำนายอย่างผิดพลาดจากตัวจำแนกข้อมูล M_i เมื่อทำการพิจารณาทุกๆตัวจำแนกข้อมูล เราจะรวมผลโหวดของทุกๆการจำแนกข้อมูลที่ซึ่งจะถูกคูณด้วยค่าน้ำหนักของแต่ละชุดข้อมูล

Adaboost เป็นวิธีการเพิ่มความถูกต้องแบบ Boosting ที่ได้รับความนิยม โดยการทำงานของ Adaboost จะมีรายละเอียดดังแสดงในรูปที่ 6.18

อัลกอริทึม Adaboost

อินพุต - ชุดข้อมูล D ที่มีหมวดหมู่ข้อมูลทั้งสิ้น d หมวดหมู่, k จำนวนรอบในการทำงาน (ใน 1 รอบการทำงานจะทำการสร้าง 1 ตัวจำแนกข้อมูล), อัลกอริทึมในการสร้างตัวจำแนกข้อมูล

เอาท์พุต โมเดลสำหรับการจำแนกข้อมูล

(1) กำหนดค่าน้ำหนักให้แก่แต่ละเรคคอร์ดข้อมูลในชุดข้อมูล D ให้มีค่าเท่ากับ $\frac{1}{d}$

(2) for แต่ละรอบการทำงาน $i = 1$ ถึง k do

(3) ทำการสุ่มเลือกชุดข้อมูล D_i แบบ 'replacement'

(4) ทำการสร้างตัวจำแนกข้อมูล M_i จากชุดข้อมูล D_i

(5) ทำการคำนวณค่าความผิดพลาดของตัวจำแนกข้อมูล M_i ด้วย $error(M_i) = \sum_j^d w_j \times err(X_j)$ เมื่อ $err(X_j)$ คือ ค่าความผิดพลาดของการจำแนกข้อมูลของเรคคอร์ด X_j (ถ้าเรคคอร์ด X_j ที่ทำการพิจารณาถูกจำแนกหมวดหมู่ผิดพลาดจะทำให้ $err(X_j)$ มีค่าเท่ากับ 1 แต่ถ้าเรคคอร์ด X_j ถูกจำแนกหมวดหมู่ได้อย่างถูกต้องจะทำให้ $err(X_j)$ มีค่าเท่ากับ 1)

(6) if $error(M_i) > 0.5$

(7) ทำการล้างค่าน้ำหนักด้วยการกำหนดค่าน้ำหนักใหม่ให้มีค่าเท่ากับ $\frac{1}{d}$ แล้วย้อนกลับไปดำเนินการในข้อที่ (3)

(8) for แต่ละเรคคอร์ดข้อมูลในชุดข้อมูลย่อย D_i ที่ถูกจำแนกหมวดหมู่ได้อย่างถูกต้อง do

(9) ทำการอัพเดทค่าน้ำหนักของเรคคอร์ดด้วยการคูณด้วย $\frac{error(M_i)}{(1-error(M_i))}$

(10) ทำการนอร์มอลไลซ์ค่าน้ำหนักเรคคอร์ดทั้งหมดในชุดข้อมูลด้วยการคูณด้วยผลรวมของค่าน้ำหนักเก่าแล้วหารด้วยผลรวมของค่าน้ำหนักใหม่ (จะทำให้ค่าน้ำหนักของเรคคอร์ดที่ถูกจำแนกหมวดหมู่ผิดมีค่าสูงขึ้นแต่จะทำให้ค่าน้ำหนักของเรคคอร์ดที่ถูกจำแนกหมวดหมู่ถูกมีค่าลดลง)

การจำแนกข้อมูลเรคคอร์ด X

(1) กำหนดค่าน้ำหนักของแต่ละหมวดหมู่ข้อมูลให้มีค่าเท่ากับ 0

(2) for แต่ละรอบการทำงาน $i = 1$ ถึง k do

(3) ทำการโหวดค่าน้ำหนักของตัวจำแนกข้อมูล M_i ($w_i = \frac{1 - error(M_i)}{error(M_i)}$)

(4) ทำการคำนวณผลของการทำนายสำหรับเรคคอร์ด X จาก M_i ($c = M_i(X)$)

(5) return หมวดหมู่ข้อมูลที่มีค่าน้ำหนักสูงที่สุด

รูปที่ 6.18 รายละเอียด Adaboost สำหรับการเพิ่มความถูกต้องในการจำแนกข้อมูล

คำถามท้ายบท

1. จงอธิบายขั้นตอนการทำการหลักของการจำแนกข้อมูลด้วยต้นไม้ตัดสินใจ
2. เพราะเหตุใดการตัดเล็มต้นไม้จึงเป็นขั้นตอนที่เป็นประโยชน์ในการจำแนกข้อมูลด้วยต้นไม้ตัดสินใจ
3. จงอธิบายขั้นตอนการทำการหลักของการจำแนกข้อมูลด้วบเบย์เซียนอย่างง่าย
4. กำหนดให้ชุดข้อมูลสอนเกี่ยวกับข้อมูลพนักงานประกอบไปด้วยข้อมูล 3 แอทริบิว (แผนก ช่วงอายุ และ ช่วงเงินเดือน) และ หมวดหมู่ของข้อมูล (สถานะ “junior” และ “senior”) ดังแสดงในตารางข้างล่างนี้ แต่เมื่อเราพิจารณาตารางข้อมูลจะสังเกตุได้ว่า ยังมีอีก 1 แอทริบิว นั่นคือ “count” ที่ซึ่งจะบ่งบอกถึงจำนวนเรคคอร์ดข้อมูลที่มีค่าในแอทริบิว แผนก ช่วงอายุ ช่วงเงินเดือน และ สถานะ ตามค่าในเรคคอร์ดนั้นๆ

Department	age	salary	status	count
Sales	31...35	46K...50K	senior	30
Sales	26...30	26K...30K	junior	40
Sales	31...35	31K...35K	junior	40
Systems	21...25	46K...50K	junior	20
Systems	31...35	66K...70K	senior	5
Systems	26...30	46K...50K	junior	3
Systems	41...45	66K...70K	senior	3
Marketing	36...40	46K...50K	senior	10
Marketing	31...35	41K...45K	junior	4
Secretary	46...50	36K...40K	senior	4
Secretary	26...30	26K...30K	junior	6

   (a) จงทำการสร้างต้นไม้ตัดสินใจจากข้อมูลในตารางข้างต้น
   (b) กำหนดให้ข้อมูลเรคคอร์ดหนึ่งๆประกอบไปด้วย (“systems”, “26...30”, “46K...50K”) จงทำการจำแนกข้อมูลด้วยต้นไม้ตัดสินใจ
   (c) กำหนดให้ข้อมูลเรคคอร์ดหนึ่งๆประกอบไปด้วย (“systems”, “26...30”, “46K...50K”) จงทำการจำแนกข้อมูลด้วยเบย์เซียนอย่างง่าย
5. “associative classification” คืออะไร
6. เหตุใด “associative classification” สามารถให้ผลการจำแนกข้อมูลที่มีความถูกต้องสูง
7. จงเปรียบเทียบข้อดีและข้อเสียของการเรียนรู้แบบกระตือรือร้น (eager learning เช่น ต้นไม้ตัดสินใจ, เบย์เซียน) และ การเรียนรู้แบบเกียจคร้าน (lazy learning เช่น การค้นหาเพื่อนบ้านใกล้สุด k ตัว)
8. จงอธิบายการทำงานการค้นหาเพื่อนบ้านใกล้สุด k ตัว และทำการบอกถึงหมวดหมู่ของข้อมูลเรค-คอร์ด (“systems”, “26...30”, “46K...50K”) จากชุดข้อมูลสอนในตารางข้างต้น
9. จงอธิบายวิธีในการตรวจสอบความถูกต้องและความผิดพลาดในการจำแนก/ทำนายข้อมูล
10. จงอธิบายวิธีในการเพิ่มประสิทธิภาพในการจำแนก/ทำนายข้อมูล