รายงานวิจัยฉบับสมบูรณ์

โครงการวิจัยร่วมภาครัฐและเอกชน ปีงบประมาณ 2545
โครงการย่อยที่ 7 อัลกอริทึมการทำเหมืองข้อมูล

ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

บุญเสริม กิจศิริกุล

คำนำ

เอกสารนี้เป็นรายงานวิจัยฉบับสมบูรณ์โครงการวิจัยร่วมภาครัฐและเอกชน ปีงบประมาณ 2545 โครงการย่อยที่ 7 “อัลกอริทึมการทำเหมืองข้อมูล” ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย งานวิจัยนี้ทำการศึกษาวิจัยอัลกอริทึมการทำเหมืองข้อมูลและได้เขียนซอฟต์แวร์ทำเหมืองข้อมูลขึ้น โดยผู้วิจัยหวังว่าจะเป็นประโยชน์ต่อการทำเหมืองข้อมูลสำหรับผู้ที่สนใจ และสามารถนำไปใช้ในการเรียนการสอนในวิชาที่เกี่ยวข้องได้ไม่มากก็น้อย ซอฟต์แวร์นี้เปิดให้บริการแก่ผู้ที่สนใจที่เว็บไซต์ http://mind.cp.eng.chula.ac.th

ขอขอบคุณผู้ช่วยวิจัยและผู้มีส่วนร่วมในโครงการนี้ทุกท่าน ซึ่งได้แก่ ดร.สุกรี สินธุภิญโญ, ประเสริฐศักดิ์ ผุงประเสริฐยิ่ง, เจตน์ เอื้อเวชนิชกุล, พิริยะ ประเทืองวงศ์, ฐิมาพร เพชรแก้ว, ศิริพจน์ สุรบถโสภณ, ปัญจมา ศรีสิทธิชัยกุล, พรลดา ปิติพัฒนะโฆษิต และ วริญญา สันตโยดม

บุญเสริม กิจศิริกุล
สิงหาคม 2546

สารบัญ

สารบัญภาพ

สารบัญตาราง

อัลกอริทึมสำหรับการทำเหมืองข้อมูล

Algorithms for Data Mining

บทที่ 1: บทนำ

Chapter 1: Introduction

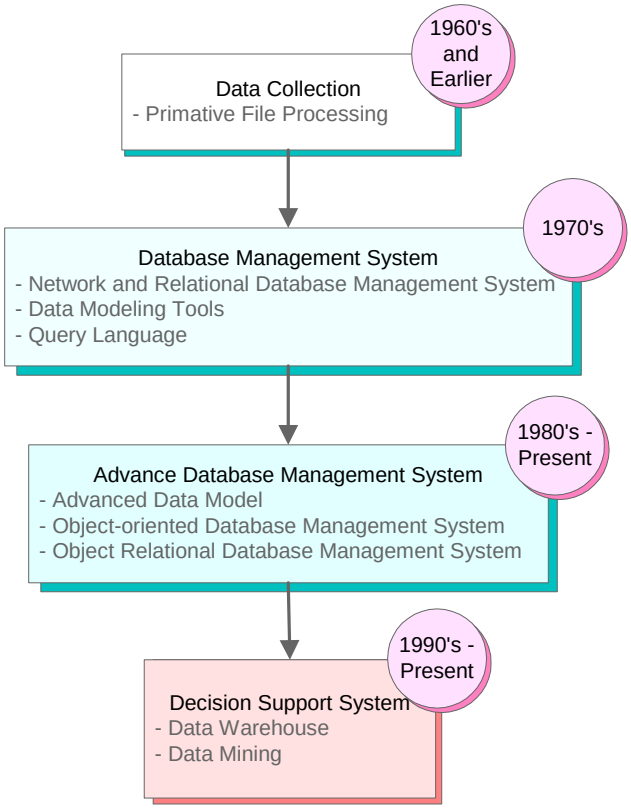
การทำเหมืองข้อมูล

We are drowning in data, but starving for knowledge.

Rutherford D. Rogers

การทำเหมืองข้อมูลคือกระบวนการที่กระทำกับข้อมูลจำนวนมากเพื่อค้นหารูปแบบและความสัมพันธ์ที่ซ่อนอยู่ในชุดข้อมูลนั้น ในปัจจุบัน การทำเหมืองข้อมูลได้ถูกนำไปประยุกต์ใช้ในงานหลายประเภท ทั้งในด้านธุรกิจที่ช่วยในการตัดสินใจของผู้บริหาร ในด้านวิทยาศาสตร์และการแพทย์ รวมทั้งในด้านเศรษฐกิจและสังคม

ดังแสดงในรูปที่ 1.1 การทำเหมืองข้อมูลเปรียบเสมือนอีกวิวัฒนาการหนึ่งในการจัดเก็บและตีความหมายข้อมูล จากเดิมที่มีการจัดเก็บข้อมูลอย่างง่ายๆ มาสู่การจัดเก็บในรูปฐานข้อมูลที่สามารถดึงค่าสารสนเทศของข้อมูลมาใช้ จนถึงการทำเหมืองข้อมูลที่สามารถค้นพบความรู้ที่ซ่อนอยู่ในข้อมูล



รูปที่ 1.1 วิวัฒนาการในการจัดเก็บและตีความหมายข้อมูล

ในการทำเหมืองข้อมูลนั้น ต้องใช้ความรู้จากศาสตร์หลายแขนง ได้แก่

- ฐานข้อมูล (database systems, data warehouses, On-Line Analytical Processing — OLAP) ซึ่งเป็นหลักในการจัดเก็บ รวบรวม และเตรียมข้อมูลที่ใช้ในการทำเหมือง
- การเรียนรู้ของเครื่อง (machine learning) ใช้เป็นอัลกอริทึมหลักที่ใช้ในการค้นหารูปแบบและความสัมพันธ์ที่ซ่อนอยู่ในข้อมูล
- หลักสถิติ (statistical and data analysis methods) สำหรับวิเคราะห์ข้อมูลเบื้องต้น ซึ่งอาจจะชี้ให้เห็นถึงรูปแบบและความสัมพันธ์ของข้อมูลที่ซ่อนอยู่ได้
- การทำจินตทัศน์ (visualization) เพื่อแสดงผลลัพธ์ รูปแบบ และความสัมพันธ์ของข้อมูลออกมาให้ผู้ใช้เข้าใจง่ายที่สุด ซึ่งเป็นประโยชน์ต่อการตีความและการนำผลลัพธ์นั้นไปใช้
- การโปรแกรมทางคณิตศาสตร์ (mathematical programming)
- การคำนวณประสิทธิภาพสูง (high performance computing) เนื่องจากข้อมูลที่มากจะทำให้การทำเหมืองข้อมูลใช้เวลานาน จึงจำเป็นต้องมีการคำนวณที่รวดเร็วรองรับ

นอกจากความรู้เหล่านี้แล้ว การทำเหมืองข้อมูลยังต้องอาศัยความรู้ในตัวข้อมูลที่ใช้ในการทำเหมืองด้วย เช่น ในการทำเหมืองของข้อมูลทางชีววิทยา จำเป็นต้องมีความรู้ด้านชีววิทยา หรือในการทำเหมืองข้อมูลเพื่อใช้ในด้านธุรกิจ จำเป็นต้องมีความรู้ในธุรกิจนั้นๆ

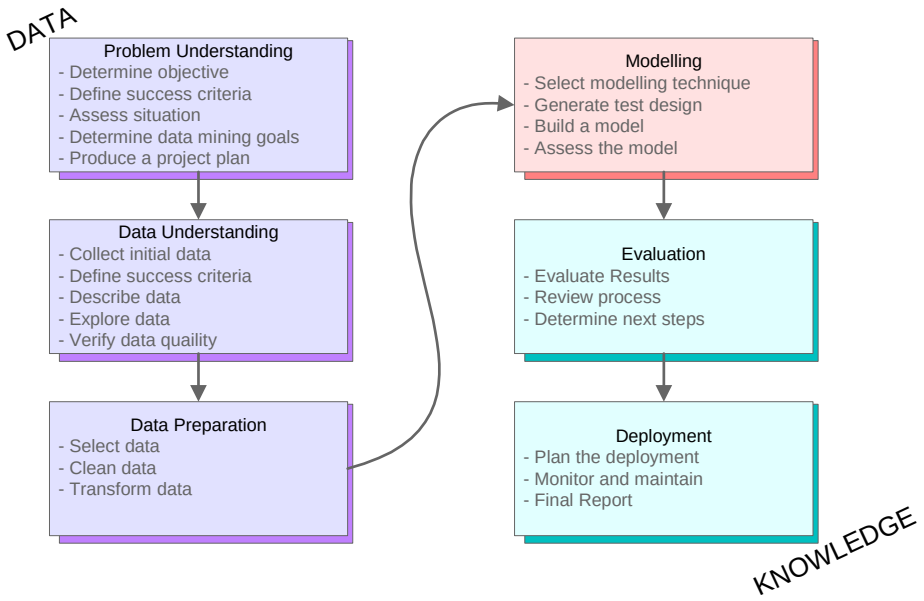
อัลกอริทึมในการทำเหมืองข้อมูล

เราสามารถแบ่งประเภทอัลกอริทึมในการทำเหมืองข้อมูลออกเป็นสองประเภท ได้แก่

1. การสร้างแบบจำลองในการทำนาย (predictive modeling, supervised modeling) ในที่นี้ ทุกข้อมูลจะมีคุณสมบัติหนึ่งเรียกว่าฉลาก (label) ซึ่งค่าของคุณสมบัตินี้จะเป็นค่าที่ใช้ในการทำนายผลของข้อมูล อัลกอริทึมประเภทนี้จะมุ่งเน้นในการแบ่งแยกข้อมูลออกเป็นกลุ่มตามค่าคุณสมบัติของฉลาก ซึ่งถ้าค่าคุณสมบัติของฉลากมีค่าไม่ต่อเนื่อง จะเรียกกระบวนการที่ใช้แบ่งแยกว่า การแยกแยะ (classification) ถ้าค่าคุณสมบัติของฉลากมีค่าต่อเนื่อง จะเรียกกระบวนการที่ใช้แบ่งแยกว่า การถดถอย (regression)
2. การสร้างแบบจำลองในการบรรยาย (descriptive modeling, unsupervised modeling) ในที่นี้ อาจเป็นการหาความสัมพันธ์ต่างๆ (association) หรือหาการจัดกลุ่มข้อมูล (clustering) ซึ่งไม่ได้มีจุดมุ่งหมายเพื่อการทำนาย

ขั้นตอนการทำเหมืองข้อมูล

การทำเหมืองข้อมูลประกอบด้วยขั้นตอนหลัก ดังแสดงในรูปที่ 1.2 ซึ่งได้แก่ขั้นตอนดังต่อไปนี้



รูปที่ 1.2 ขั้นตอนการทำเหมืองข้อมูล

1. การทำความเข้าใจปัญหา ประกอบด้วยกระบวนการย่อยดังนี้
   - ตั้งเป้าหมายว่าการทำเหมืองข้อมูลครั้งนี้ต้องการที่จะแก้ปัญหาใด เช่น ทำเหมืองข้อมูลเพื่อต้องการเพิ่มยอดขายสินค้า เป็นต้น
   - ตั้งเกณฑ์วัดความสำเร็จในการทำเหมืองข้อมูล ซึ่งอาจเป็นได้ทั้งความสำเร็จในด้านรูปธรรม เช่น สามารถเพิ่มยอดขายสินค้าได้ 5% และความสำเร็จในด้านนามธรรม เช่น สามารถค้นพบความรู้ใหม่จากข้อมูล
   - ประเมินสถานการณ์ในด้านต่างๆ เช่น ความรู้พื้นฐานในเรื่องที่จะทำเหมืองข้อมูลมีเพียงพอหรือไม่ และผลประโยชน์จากการทำเหมืองข้อมูลจะคุ้มค่ากับต้นทุนที่เสียไปหรือไม่ เป็นต้น
   - ตั้งเป้าหมายในเชิงการทำเหมืองข้อมูล ซึ่งต่างไปจากเป้าหมายหลักในการแก้ปัญหา เช่น เป้าหมายหลักคือต้องการเพิ่มยอดขายสินค้า เป้าหมายในเชิงการทำเหมืองข้อมูลคือ การหาลักษณะของลูกค้าที่มีแนวโน้มจะซื้อสินค้า
   - วางแผนการทำเหมืองข้อมูล ว่าจะเก็บข้อมูลอย่างไร และใช้อัลกอริทึมใดในการทำเหมืองข้อมูล

2. การทำความเข้าใจข้อมูล ประกอบด้วยกระบวนการย่อยดังนี้
    - เก็บรวบรวมข้อมูล
    - กำหนดคุณสมบัติของข้อมูลที่เก็บมาได้
    - สำรวจดูข้อมูลอย่างคร่าวๆ ถึงค่าสถิติต่างๆ ของข้อมูล
    - ตรวจสอบข้อมูลขั้นต้น โดยตรวจสอบทั้งความสมบูรณ์และความถูกต้องของข้อมูล
3. การเตรียมข้อมูล ประกอบด้วยกระบวนการย่อยดังนี้
    - คัดเลือกข้อมูลที่จะนำมาใช้
    - ปรับเปลี่ยนรูปแบบข้อมูล เช่น นำสองตารางในฐานข้อมูลมาเชื่อมต่อกัน
    - ทำความสะอาดข้อมูล เป็นกระบวนการเตรียมข้อมูลให้เหมาะสมที่สุดเพื่อนำไปใช้ในขั้นตอนต่อไป ซึ่งมีวิธีการต่างๆ หลายวิธี ได้แก่
        - การแก้ไขข้อมูลให้ถูกต้องสมบูรณ์ เช่น การแก้ไขค่าว่างของข้อมูลโดยใส่ค่า 0
        - ปรับเปลี่ยนข้อมูลให้มีค่าเหมาะสมในการตัดสินใจ เช่น ข้อมูลที่มีค่า "Coke" และ "Pepsi" อาจเปลี่ยนค่าให้เป็น "น้ำอัดลม"
        - เลือกข้อมูลเฉพาะที่สนใจ เช่น ต้องการหาลักษณะลูกค้าที่ซื้อรถสปอร์ต ไม่ควรนำรายชื่อพนักงานขายเข้ามาเกี่ยวข้อง
        - คอลัมน์ที่มีค่าสำหรับทุกแถวเป็นค่าเดียวกัน เช่น "สัญชาติไทย" หรือ คอลัมน์ที่มีค่าที่ไม่ซ้ำกันเลย เช่น “หมายเลขสมาชิก” ไม่ควรนำมาใช้ เนื่องจากไม่สามารถบอกรูปแบบของข้อมูลได้
4. การสร้างแบบจำลอง ประกอบด้วยกระบวนการย่อยดังนี้
    - เลือกอัลกอริทึมที่เหมาะสมในการทำเหมืองข้อมูล
    - กำหนดรูปแบบการทดสอบผลลัพธ์
    - สร้างแบบจำลองตามอัลกอริทึมที่เลือก
    - ทดสอบแบบจำลองที่ได้มานั้นว่ามีความถูกต้องและน่าเชื่อถือเพียงใด
5. การประเมิน อาจจะประเมินแบบจำลองที่สร้างขึ้นด้วยการลองนำไปใช้กับสถานการณ์จริง หรือนำไปใช้ในสถานการณ์ที่จำลองขึ้น เพื่อดูว่าแบบจำลองนี้ได้ผลหรือไม่เพียงใด และมีความผิดพลาดตรงไหน ถ้าผิดพลาด อาจจะต้องดำเนินการแก้ไขในกระบวนการก่อนหน้า ก่อนที่จะนำแบบจำลองนี้มาใช้งานจริง
6. การนำไปใช้ นำไปใช้และตรวจสอบผลว่าบรรลุเป้าหมายที่ตั้งไว้เพียงใด

ซอฟต์แวร์สำหรับทำเหมืองข้อมูล

ปัจจุบันมีซอฟต์แวร์สำหรับทำเหมืองข้อมูลมากมายดังรูปที่ 1.3 ซึ่งอาจแยกประเภทได้ดังนี้

- ตามประเภทฮาร์ดแวร์ แบ่งได้เป็น
  - ซอฟต์แวร์สำหรับเครื่องคอมพิวเตอร์ส่วนบุคคล ใช้ในการทำเหมืองของข้อมูลที่มีจำนวนไม่มาก และข้อมูลอาจเก็บอยู่ในรูปไฟล์
  - ซอฟต์แวร์สำหรับเครื่องคอมพิวเตอร์สมรรถนะสูง ใช้ในการทำเหมืองของข้อมูลที่มีจำนวนมาก อาจเป็นโปรแกรมแบบไคลเอนต์เซิร์ฟเวอร์ และข้อมูลที่ใช้อาจเก็บอยู่ในฐานข้อมูลขนาดใหญ่
- ตามประเภทการใช้งาน แบ่งได้เป็น
  - ใช้ในงานทั่วไป การพัฒนาซอฟต์แวร์ประเภทนี้ไม่ต้องแสดงถึงการทำงานของอัลกอริทึมมากนัก แต่เน้นให้ใช้งานได้สะดวก มีกระบวนการทำงานที่ผู้ใช้คุ้นเคย และมีการแสดงผลที่เข้าใจง่าย
  - ใช้ในงานด้านเทคนิค ในที่นี้ ผู้ใช้จะรู้กระบวนการของอัลกอริทึมที่ใช้ การพัฒนาซอฟต์แวร์จึงอาจให้มีการแสดงผลทางด้านเทคนิค และให้ผู้ใช้ปรับแต่งพารามิเตอร์ของอัลกอริทึมได้ตามสะดวก

เนื่องจากเป็นการค้นหารูปแบบและความสัมพันธ์ที่ซ่อนอยู่ การทำเหมืองข้อมูลจึงต้องอาศัยอัลกอริทึมที่สามารถค้นหารูปแบบในข้อมูล และการทำจินตทัศน์เพื่อแสดงรูปแบบนั้นออกมาให้ผู้ใช้ได้เกิดความเข้าใจมากที่สุด ซึ่งแต่ละซอฟต์แวร์ก็มีอัลกอริทึมและระบบจินตทัศน์แตกต่างกันไป

รูปที่ 1.3 ซอฟต์แวร์สำหรับทำเหมืองข้อมูล

แนวคิดหลักในการพัฒนาโปรแกรม

จุดประสงค์ในการพัฒนาโปรแกรม

ในการพัฒนาโปรแกรมเพื่อทำเหมืองข้อมูลมีจุดประสงค์และเป้าหมายดังนี้

(1) สามารถรองรับการเรียนรู้จากหลายเครื่องพร้อมกัน

(2) แยกกระบวนการเรียนรู้ออกจากกระบวนการอื่น เนื่องจากส่วนใหญ่กระบวนการเรียนรู้จะใช้เวลานาน

(3) สามารถรองรับข้อมูลในรูปแบบมาตรฐาน

(4) มีส่วนที่รองรับกระบวนการการทำเหมืองข้อมูล

(5) มีระบบจินตทัศน์ที่เหมาะสมกับผลลัพธ์ในการเรียนรู้แบบต่างๆ

(6) สามารถบันทึกข้อมูลต่างๆ ที่ได้จากกระบวนการเรียนรู้ครั้งหนึ่งๆ

(7) สามารถแยกแยะข้อมูลเพื่อใช้ในการตัดสินใจและวัดประสิทธิภาพของผลลัพธ์ที่ได้จากการเรียนรู้

(8) มีบันทึกถึงกระบวนการที่ได้กระทำไป

เพื่อให้บรรลุจุดประสงค์ในข้อ (1) และ (2) จึงทำการพัฒนาโปรแกรมให้อยู่ในรูปแบบไคลเอนต์เซิร์ฟเวอร์ และเพื่อจุดประสงค์ข้อ (3) จึงออกแบบโปรแกรมให้สามารถอ่านข้อมูลในรูปแบบจุลภาคคั่น (comma-separated data) ซึ่งเป็นรูปแบบข้อมูลที่ใช้กันทั่วไปในการทำเหมืองข้อมูล

มีส่วนของโปรแกรมในการปรับแต่งข้อมูลและจินตทัศน์ของชุดข้อมูล ซึ่งเป็นส่วนที่รองรับกระบวนการในการทำเหมืองข้อมูล (4) นอกจากนี้ยังมีระบบจินตทัศน์สำหรับผลลัพธ์ในการเรียนรู้แบบต่างๆ (5) และเพื่อจุดประสงค์ในข้อ (7) โปรแกรมสามารถแยกแยะข้อมูลในชุดข้อมูลที่ใช้ทดสอบและข้อมูลที่ป้อนเข้าไปได้

นอกจากนี้ โปรแกรมถูกพัฒนาให้สามารถบันทึกและเปิดข้อมูลต่างๆ ที่เกี่ยวข้องกับการทำเหมืองข้อมูลครั้งหนึ่งๆ (6) และมีบันทึกถึงกระบวนการต่างๆ ที่ได้กระทำไป (8)

เครื่องมือในการพัฒนา

ฮาร์ดแวร์

- Compaq Deskpro
  - Intel Pentium III – 1 GHz
  - RAM 256 MB
  - Hard disk 20 GB
  - Network Interface 10/100 Mbps
  - 17-inch Monitor

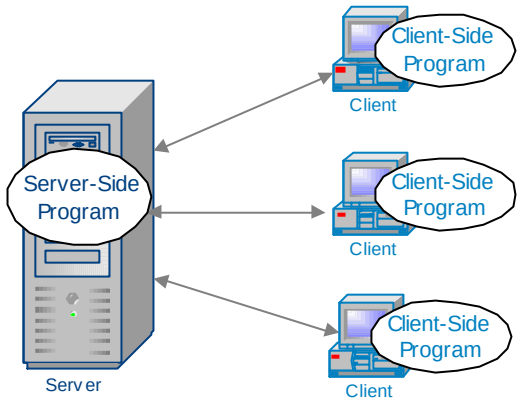
ซอฟต์แวร์

- Operating System: Windows XP Professional
- Development Tools
  - Microsoft Visual C++
  - Microsoft Visual Basic.NET

สถาปัตยกรรมของระบบ

ภาพรวมของระบบ

ระบบของโปรแกรมการทำเหมืองข้อมูลประกอบด้วยโปรแกรมทางฝั่งไคลเอนต์และโปรแกรมทางฝั่งเซิร์ฟเวอร์ที่ทำการเชื่อมต่อและแลกเปลี่ยนข้อมูลกัน โปรแกรมทางฝั่งไคลเอนต์เป็นโปรแกรมส่วนที่ติดต่อกับผู้ใช้ ขณะที่โปรแกรมทางฝั่งเซิร์ฟเวอร์เป็นโปรแกรมส่วนที่ทำการเรียนรู้ข้อมูลที่ใช้ในการเรียนรู้จะถูกส่งจากโปรแกรมทางฝั่งไคลเอนต์ไปที่โปรแกรมทางฝั่งเซิร์ฟเวอร์ เมื่อการเรียนรู้ของโปรแกรมทางฝั่งเซิร์ฟเวอร์เสร็จสิ้นลง ผลลัพธ์ที่ได้จะถูกส่งกลับมาที่โปรแกรมทางฝั่งไคลเอนต์เพื่อแสดงผลต่อผู้ใช้ ดังแสดงในรูปที่ 1.4

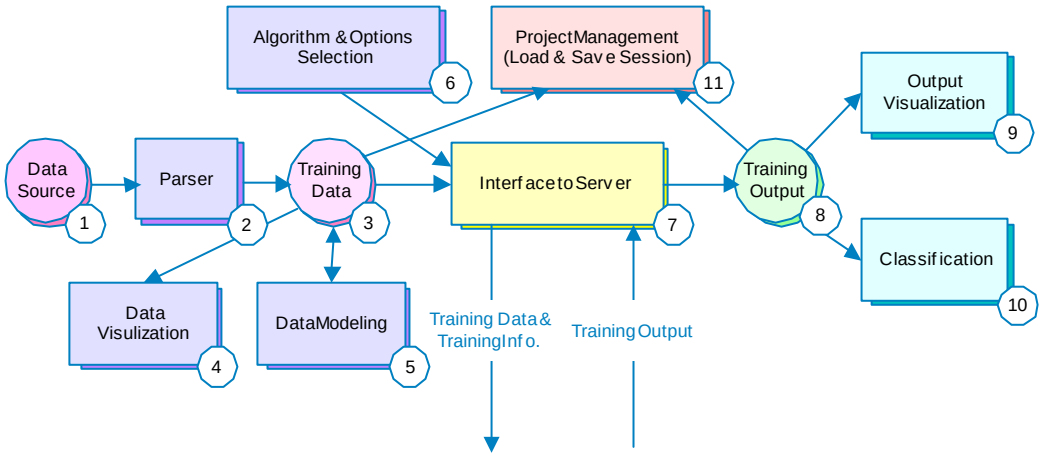


รูปที่ 1.4 ภาพรวมของระบบ

โปรแกรมทางฝั่งไคลเอนต์และโปรแกรมทางฝั่งเซิร์ฟเวอร์อาจจะอยู่ในเครื่องเดียวกันหรืออยู่กันคนละเครื่องก็ได้ ทั้งนี้ขึ้นกับระบบคอมพิวเตอร์ของผู้ใช้ ถ้าคอมพิวเตอร์ของผู้ใช้เป็นคอมพิวเตอร์ส่วนบุคคลที่ไม่ได้เชื่อมต่อกับเครื่องคอมพิวเตอร์อื่น ผู้ใช้สามารถใช้งานโปรแกรมได้โดยรันโปรแกรมทางฝั่งไคลเอนต์และโปรแกรมทางฝั่งเซิร์ฟเวอร์ไว้ที่เครื่องเดียวกัน และใช้งานโปรแกรมตามปกติ แต่ถ้าหากผู้ใช้มีเครื่องคอมพิวเตอร์สมรรถนะสูงเชื่อมต่อกับเครื่องคอมพิวเตอร์อื่นทางระบบเน็ตเวิร์ค เครื่องคอมพิวเตอร์สมรรถนะสูงนั้นสามารถเป็นเซิร์ฟเวอร์และรันโปรแกรมทางฝั่งเซิร์ฟเวอร์ได้ โดยมีเครื่องคอมพิวเตอร์เครื่องอื่นๆ รันโปรแกรมทางฝั่งไคลเอนต์ คอยส่งข้อมูลไปให้โปรแกรมทางฝั่งเซิร์ฟเวอร์ทำการเรียนรู้ และรอรับผลลัพธ์กลับมาแสดงผล

โปรแกรมทางฝั่งไคลเอนต์

โปรแกรมทางฝั่งไคลเอนต์เป็นส่วนที่ติดต่อกับผู้ใช้ เกี่ยวข้องกับกระบวนการการทำเหมืองข้อมูลในขั้นตอนก่อนทำการเรียนรู้และในขั้นตอนหลังจากที่การเรียนรู้เสร็จสิ้น โปรแกรมทางฝั่งไคลเอนต์มีกระบวนการและข้อมูลที่เกี่ยวข้องดังรูปที่ 1.5 ด้านล่างนี้



รูปที่ 1.5 กระบวนการและข้อมูลต่างๆ ของโปรแกรมทางฝั่งไคลเอนต์

กระบวนการและข้อมูลของโปรแกรมทางฝั่งไคลเอนต์

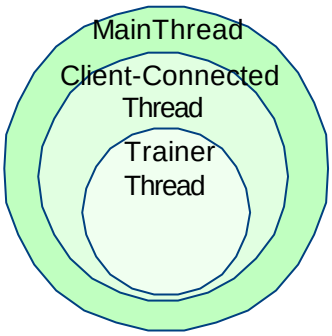
1. ชุดข้อมูล (Data Source) เป็นข้อมูลดิบที่เก็บมาเพื่อทำการเรียนรู้ ซึ่งข้อมูลนี้เอง ที่ผู้ใช้ต้องการนำมาค้นหารูปแบบ ความสัมพันธ์ และความรู้ต่างๆ ที่ซ่อนอยู่ข้างในรูปแบบของชุดข้อมูลสำหรับการเรียนรู้ของโปรแกรมจะกล่าวในหัวข้อการเตรียมข้อมูล
2. ส่วนอ่านไฟล์ (Parser) สำหรับอ่านชุดข้อมูลมาเก็บไว้ในโครงสร้างข้อมูลเพื่อใช้ในการเรียนรู้ต่อไป
3. ข้อมูลที่ใช้ในการเรียนรู้ (Training Data) เป็นโครงสร้างข้อมูลที่ใช้เก็บข้อมูลต่างๆ สำหรับการเรียนรู้
4. จินตทัศน์ของชุดข้อมูลที่ใช้ในการเรียนรู้ (Data Visualization) สำหรับรวบรวม คำนวณ และแสดงค่าทางสถิติต่างๆ ของชุดข้อมูล เพื่อให้ผู้ใช้ได้รู้ลักษณะเบื้องต้นของชุดข้อมูลก่อนที่จะทำการเรียนรู้
5. การปรับแต่งข้อมูล (Data Modeling) สำหรับปรับแต่งข้อมูลเพื่อให้เหมาะสมและสอดคล้องกับลักษณะงานที่ต้องการจะทำ รวมทั้งเป็นการกำจัดข้อมูลส่วนที่ไม่จำเป็นทิ้งไปก่อนที่จะทำการเรียนรู้
6. การเลือกอัลกอริทึมและกำหนดค่าพารามิเตอร์สำหรับการเรียนรู้ของอัลกอริทึมนั้น (Algorithm & Option Selections) เป็นการเลือกอัลกอริทึมให้สอดคล้องกับงานที่ต้องการจะทำ และผลลัพธ์ที่ต้องการจะได้ นอกจากนั้นยังต้องปรับแต่งค่าพารามิเตอร์ของแต่ละอัลกอริทึมตามที่ต้องการ ก่อนที่จะทำการเรียนรู้ชุดข้อมูลต่อไป
7. ส่วนเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ (Interface to Server) เป็นส่วนที่ทำการติดต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ สำหรับส่งข้อมูลที่ใช้ในการเรียนรู้ไปให้โปรแกรม

ทางฝั่งเซิร์ฟเวอร์ทำการเรียนรู้ และรับผลลัพธ์ของการเรียนรู้จากโปรแกรมทางฝั่งเซิร์ฟเวอร์

8. ผลลัพธ์จากการเรียนรู้ (Training Output) เป็นผลลัพธ์ที่ได้จากการเรียนรู้ ซึ่งจะแตกต่างกันไปตามแต่ละอัลกอริทึมที่ใช้ในการเรียนรู้
9. จินตทัศน์ของผลลัพธ์จากการเรียนรู้ (Output Visualization) เป็นการนำผลลัพธ์ที่ได้จากการเรียนรู้มาคำนวณ สรุปรวม และแสดงผล ซึ่งการแสดงผลนี้อาจแสดงผลทั้งในรูป ข้อมูล แผนภูมิ และรูปภาพ เพื่อให้ผู้ใช้เกิดความเข้าใจในผลลัพธ์มากที่สุด และสามารถที่จะมองหารูปแบบที่ซ่อนอยู่ในผลลัพธ์นั้น เพื่อประโยชน์ในการตัดสินใจและสามารถนำข้อมูลจากการเรียนรู้ไปใช้ให้เกิดประโยชน์สูงสุด
10. การแยกแยะข้อมูลที่ใช้ทดสอบ (Classification) สำหรับการเรียนรู้แบบแยกแยะนั้น การแยกแยะข้อมูลที่ใช้ทดสอบเป็นวิธีที่ใช้วัดประสิทธิภาพของผลลัพธ์ที่ได้จากการเรียนรู้ ถ้าการแยกข้อมูลที่ใช้ทดสอบให้ค่าความถูกต้องมาก อาจสรุปได้ว่าผลลัพธ์ที่ได้จากการเรียนรู้นี้มีประสิทธิภาพดี สามารถนำไปใช้แยกแยะข้อมูลที่ไม่เคยพบได้ดี
11. การจัดการข้อมูล (Project Management) เป็นการเก็บรวบรวมโครงสร้างข้อมูลต่างๆ ที่เกี่ยวเนื่องกับกระบวนการเรียนรู้เข้าด้วยกัน ทั้งโครงสร้างข้อมูลที่เก็บข้อมูลที่ใช้ในการเรียนรู้ และโครงสร้างข้อมูลที่เป็นผลลัพธ์จากการเรียนรู้ ทำให้ผู้ใช้มีความสะดวกในการจัดการ จัดเก็บ และค้นหาข้อมูลและผลลัพธ์ที่ได้จากกระบวนการทำเหมืองข้อมูลในแต่ละครั้ง

โปรแกรมทางฝั่งเซิร์ฟเวอร์

โปรแกรมทางฝั่งเซิร์ฟเวอร์มีกระบวนการหลักเพียงกระบวนการเดียว คือทำการเรียนรู้ข้อมูลที่ถูกส่งมาจากโปรแกรมทางฝั่งไคลเอนต์ตามอัลกอริทึมที่โปรแกรมทางฝั่งไคลเอนต์ระบุมา โปรแกรมทางฝั่งเซิร์ฟเวอร์จะต้องมีการออกแบบให้สามารถเชื่อมต่อและทำการเรียนรู้จากโปรแกรมหลายโปรแกรมทางฝั่งไคลเอนต์ได้ ทำให้ต้องมีการพัฒนาโปรแกรมทางฝั่งเซิร์ฟเวอร์โดยใช้เทรด (thread) ดังแสดงในรูปที่ 1.6 ซึ่งมีลำดับขั้นดังนี้



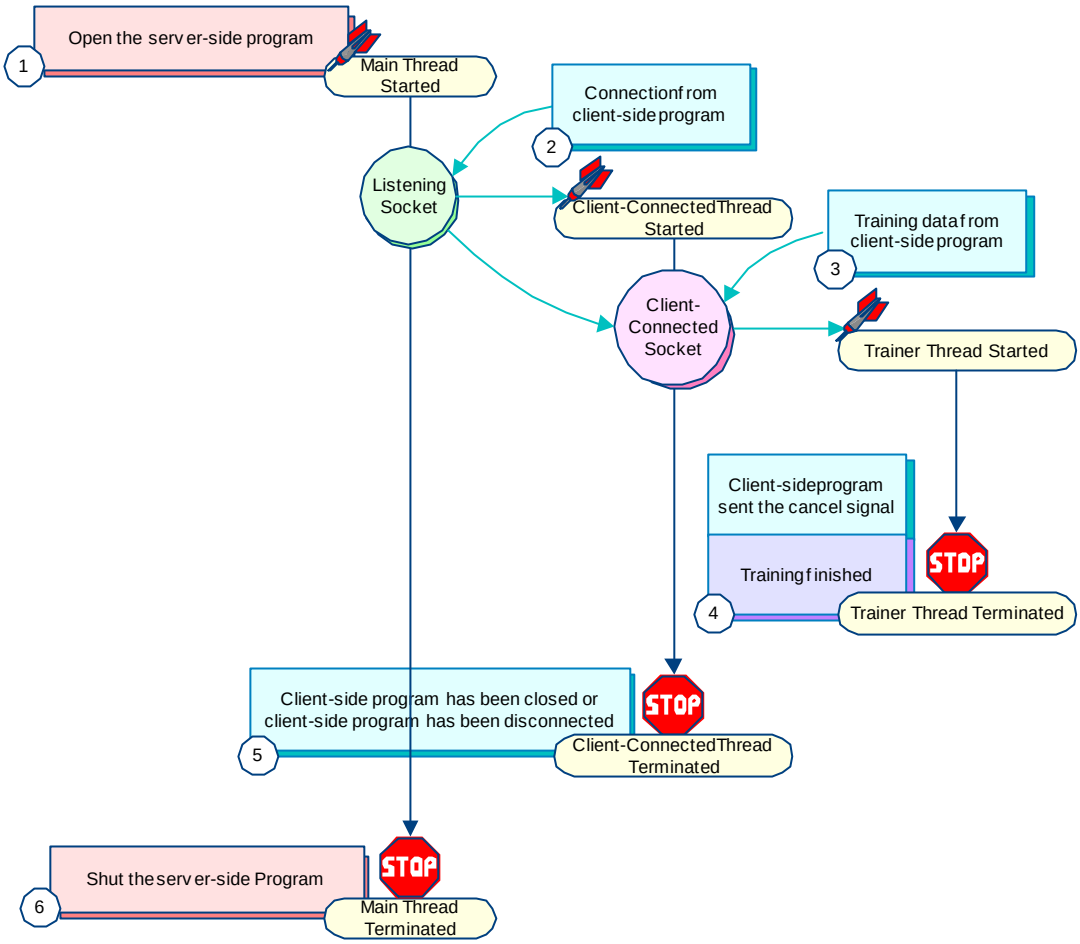
รูปที่ 1.6 ลำดับขั้นการทำงานของเทรดในโปรแกรมทางฝั่งเซิร์ฟเวอร์

เทรดที่ใช้ในโปรแกรมทางฝั่งเซิร์ฟเวอร์

1. เทรดหลัก (Main Thread) เป็นเทรดของโปรแกรมทางฝั่งเซิร์ฟเวอร์ ซึ่งจะทำงานเมื่อโปรแกรมทางฝั่งเซิร์ฟเวอร์ทำงาน และถูกทำลายเมื่อปิดโปรแกรมทางฝั่งเซิร์ฟเวอร์ เพราะฉะนั้น เทรดหลักจะมีเพียงเทรดเดียวในโปรแกรมแต่ละโปรแกรมทางฝั่งเซิร์ฟเวอร์
2. เทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ (Client-Connected Thread) สำหรับเชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ โดยทุกครั้งที่โปรแกรมทางฝั่งไคลเอนต์ทำการติดต่อเข้ามา โปรแกรมทางฝั่งเซิร์ฟเวอร์จะสร้างเทรดชนิดนี้ และเทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์จะถูกทำลายเมื่อโปรแกรมทางฝั่งไคลเอนต์ถูกปิด หรือเมื่อโปรแกรมทางฝั่งไคลเอนต์สั่งหยุดการเชื่อมต่อ เทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์นี้ทำการพัฒนาโดยใช้หลักการของเทรดติดต่อผู้ใช้ (user-interface thread) ในไลบรารีของ Microsoft Foundation Class (MFC)
3. เทรดเพื่อการเรียนรู้ (Trainer Thread) จะอยู่ในเทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ โดยทุกครั้งที่โปรแกรมทางฝั่งไคลเอนต์ส่งข้อมูลและสัญญาณการเรียนรู้เข้ามา เทรดเพื่อการเรียนรู้จะทำการเรียนรู้ด้วยอัลกอริทึมนั้นๆ เทรดเพื่อการเรียนรู้จะถูกทำลายเมื่อการเรียนรู้นั้นเสร็จสิ้น หรือมีคำสั่งจากโปรแกรมทางฝั่งไคลเอนต์ให้ยกเลิกการเรียนรู้นั้น เทรดเพื่อการเรียนรู้นี้พัฒนาโดยใช้หลักการของเทรดทำงาน (worker thread) ในไลบรารีของ MFC

ขั้นตอนการทำงานของโปรแกรมทางฝั่งเซิร์ฟเวอร์

โปรแกรมทางฝั่งเซิร์ฟเวอร์มีขั้นตอนการทำงานดังรูปที่ 1.7 ด้านล่างนี้



รูปที่ 1.7 ขั้นตอนการทำงานของโปรแกรมทางฝั่งเซิร์ฟเวอร์

1. เมื่อผู้ใช้เปิดโปรแกรมทางฝั่งเซิร์ฟเวอร์ เทรดหลักซึ่งก็คือเทรดของตัวโปรแกรมจะทำงาน พร้อมทั้งสร้างซอคเก็ตรับฟังเพื่อรอการติดต่อมาจากโปรแกรมทางฝั่งไคลเอนต์
2. เมื่อโปรแกรมทางฝั่งไคลเอนต์ส่งสัญญาณการเชื่อมต่อมา เทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์จะถูกสร้างขึ้นพร้อมด้วยซอคเก็ตที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ ซึ่งซอคเก็ตที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์นี้จะอยู่ในเทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ รอรับสัญญาณต่าง ๆ ที่โปรแกรมทางฝั่งไคลเอนต์ที่เชื่อมต่อด้วยส่งมา
3. เมื่อโปรแกรมทางฝั่งไคลเอนต์ส่งข้อมูลและสัญญาณการเรียนรู้เข้ามา เทรดเพื่อการเรียนรู้จะถูกสร้าง และทำการเรียนรู้ข้อมูลที่ถูกส่งมาตามอัลกอริทึมที่โปรแกรมทางฝั่งไคลเอนต์กำหนด

4. เมื่อทำการเรียนรู้เสร็จสิ้น หรือมีคำสั่งยกเลิกการเรียนรู้จากโปรแกรมทางฝั่งไคลเอนต์ เทรดเพื่อการเรียนรู้จะถูกทำลาย
5. เมื่อโปรแกรมทางฝั่งไคลเอนต์ถูกปิด หรือเมื่อโปรแกรมทางฝั่งไคลเอนต์สั่งหยุดการเชื่อมต่อ เทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์จะถูกทำลาย
6. เมื่อปิดโปรแกรมทางฝั่งเซิร์ฟเวอร์ เทรดหลักจะถูกทำลาย

นอกจากนี้ยังต้องพิจารณาถึงอีกสองกรณีที่เป็นไปได้ในการทำงานของเทรด คือ

- กรณีที่เทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ถูกทำลายโดยที่เทรดเพื่อการเรียนรู้ยังไม่ถูกทำลาย ซึ่งเกิดขึ้นเมื่อโปรแกรมทางฝั่งไคลเอนต์ถูกปิดหรือโปรแกรมทางฝั่งไคลเอนต์สั่งหยุดการเชื่อมต่อขณะที่เทรดเพื่อการเรียนรู้ยังคงทำงานอยู่ ในกรณีนี้ต้องทำลายเทรดเพื่อการเรียนรู้ก่อนที่จะทำลายเทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์
- กรณีที่เทรดหลักถูกทำลายโดยเทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ยังไม่ถูกทำลาย ซึ่งเกิดขึ้นเมื่อผู้ใช้ปิดโปรแกรมทางฝั่งเซิร์ฟเวอร์โดยที่โปรแกรมทางฝั่งไคลเอนต์ยังคงทำการเชื่อมต่ออยู่ ในกรณีนี้จะต้องทำลายเทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ทุกตัวเสียก่อน และถ้าหากเทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์นั้นยังคงมีเทรดเพื่อการเรียนรู้ทำงานอยู่ จะต้องทำลายเทรดเพื่อการเรียนรู้นั้นก่อนที่ทำลายเทรดที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์

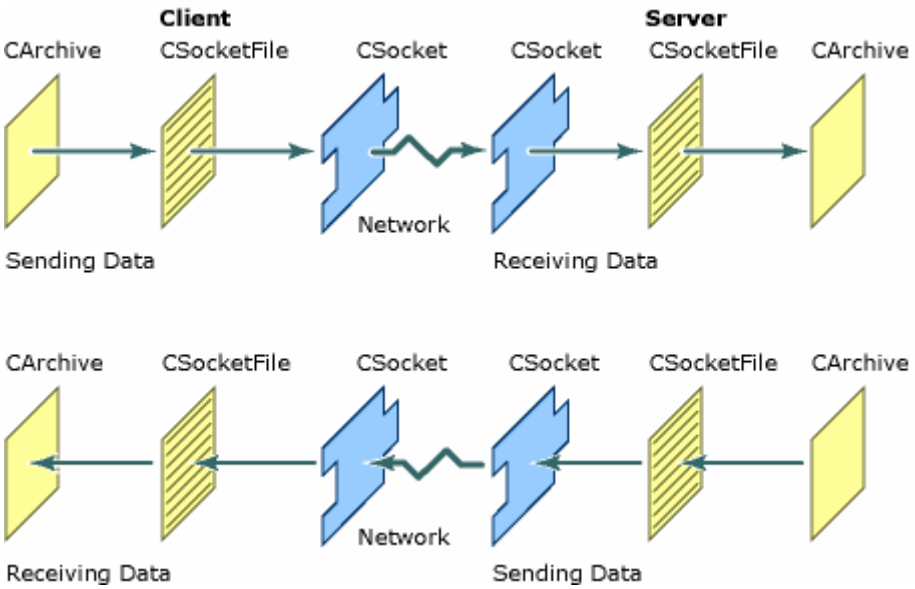
การเชื่อมต่อระหว่างโปรแกรมทางฝั่งไคลเอนต์และโปรแกรมทางฝั่งเซิร์ฟเวอร์

อุปกรณ์การเชื่อมต่อ

การเชื่อมต่อระหว่างโปรแกรมทางฝั่งไคลเอนต์และโปรแกรมทางฝั่งเซิร์ฟเวอร์พัฒนาโดยใช้หลักการของวินโดวส์ซอคเก็ต โดยเลือกใช้ CSocket ในไลบรารีของ MFC ซึ่งแสดงในรูปที่ 1.8 และมีลักษณะดังนี้

- เป็นซอคเก็ตแบบซิงโครนัส หรือบล็อกกิงซอคเก็ต ซึ่งจะไม่รับสัญญาณใดๆ จนกว่าจะทำงานฟังก์ชันที่ถูกเรียกใช้เสร็จ
- เป็นสตรีมซอคเก็ต ซึ่งรับรองว่าการส่งข้อมูลนั้นเป็นไปตามลำดับ ไม่มีข้อมูลใดหล่นหายหรือถูกส่งซ้ำ
- เชื่อมต่อกับ CArchive ในการรับส่งข้อมูลได้

ไลบรารีของ MFC ช่วยให้การรับส่งข้อมูลผ่านทางซอคเก็ตทำได้สะดวกขึ้น โดยการวาง CArchive และ CSocketFile ครอบ CSocket ทำให้การรับส่งข้อมูลเป็นการส่งตามลำดับ (serialize) ผ่านทาง CArchive แทนที่จะรับส่งผ่านทาง CSocket โดยตรง ทำให้สามารถรับส่งข้อมูลในระดับสูงได้ และช่วยลดขั้นตอนการเขียนโปรแกรมดังรูปที่ 1.8



รูปที่ 1.8 การรับส่งข้อมูลผ่านทาง CSocket และ CArchive

ขั้นตอนการเชื่อมต่อ

การสร้างซอคเก็ต การเชื่อมต่อ และการรับส่งข้อมูลของโปรแกรม เป็นไปดังตารางที่ 1.1

ตารางที่ 1.1 ขั้นตอนการรับส่งข้อมูล

Step	Server	Client
1	CSocket m_socListening ;	CSocket m_socServerConn;
2	m_socListening.Create(PORT);	m_socServerConn.Create();
3	m_socListening.Listen();	
4		m_socServerConn.Connect(m_strIP, PORT);
5	CSocket socClientConn; m_socListening.Accept(socClientConn);	
6	CSocketFile file(&socClientConn);	CSocketFile file(&socServerConn);
7	CArchive arIn(&file, CArchive::load); or CArchive arOut(&file, CArchive::store);	CArchive arIn(&file, CArchive::load); or CArchive arOut(&file, CArchive::store);
8	arIn >> dwValue; or arOut << dwValue;	arIn >> dwValue; or arOut << dwValue;

ขั้นตอนต่างๆ สามารถอธิบายได้ดังนี้

1. การประกาศตัวแปรของซอคเก็ต
2. การสร้างซอคเก็ต ในกรณีที่เป็นซอคเก็ตรับฟัง จะต้องระบุพอร์ตที่ใช้ในการเชื่อมต่อ ซึ่งในที่นี้ โปรแกรมทางฝั่งเซิร์ฟเวอร์จะมีหมายเลขพอร์ตเป็น 2545
3. ซอคเก็ตรับฟังทำการรับฟังการเชื่อมต่อ
4. ซอคเก็ตทางฝั่งไคลเอนต์ทำการเชื่อมต่อโดยระบุแอดเดรส และพอร์ตที่จะเชื่อมต่อ
5. สร้างซอคเก็ตที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ ซึ่งในที่นี้ต้องทำการสร้างเทรด และนำซอคเก็ตที่เชื่อมต่อกับโปรแกรมทางฝั่งไคลเอนต์ไว้ที่เทรด
6. สร้างซอคเก็ตไฟล์
7. สร้างอาร์ไคฟ์
8. ทำการรับส่งข้อมูลผ่านอาร์ไคฟ์

โปรโตคอลที่ใช้ในการเชื่อมต่อ

ข้อมูลที่รับส่งระหว่างโปรแกรมทางฝั่งไคลเอนต์และโปรแกรมทางฝั่งเซิร์ฟเวอร์ประกอบด้วยสองส่วน คือ

(1) ส่วนโปรโตคอลหรือเฮดเดอร์ เป็นส่วนที่ใช้สื่อสารว่าข้อมูลที่ส่งมาด้วยเป็นข้อมูลชนิดใด และจะเรียกใช้ฟังก์ชันใดในการดำเนินการข้อมูลนั้น

(2) ส่วนข้อมูล คือข้อมูลที่แลกเปลี่ยนกัน ได้แก่ชุดข้อมูลที่โปรแกรมทางฝั่งไคลเอนต์ส่งไปทำการเรียนรู้ หรือผลลัพธ์จากการเรียนรู้ที่โปรแกรมทางฝั่งเซิร์ฟเวอร์ส่งกลับมา

โปรโตคอลที่ใช้ในการเรียนรู้ของโปรแกรมประกอบด้วย

- SEARCH บอกโปรแกรมทางฝั่งเซิร์ฟเวอร์ว่าการเชื่อมต่อนี้เป็นการหาหมายเลขไอพีของเซิร์ฟเวอร์ที่รันโปรแกรมทางฝั่งเซิร์ฟเวอร์อยู่
- STOP บอกโปรแกรมทางฝั่งเซิร์ฟเวอร์ให้หยุดการเรียนรู้ที่ทำอยู่
- STOP_REPLY บอกโปรแกรมทางฝั่งไคลเอนต์ว่าการเรียนรู้ได้หยุดลง
- SERVER_SHUTDOWN บอกโปรแกรมทางฝั่งไคลเอนต์ว่าโปรแกรมทางฝั่งเซิร์ฟเวอร์ปิด
- CLIENT_DISCONNECT บอกโปรแกรมทางฝั่งเซิร์ฟเวอร์ว่าโปรแกรมทางฝั่งไคลเอนต์ตัดการเชื่อมต่อ
- VERIFY_LOGIN บอกโปรแกรมทางฝั่งเซิร์ฟเวอร์ว่าเป็นการตรวจสอบชื่อผู้ใช้และรหัสผ่าน
- VERIFY_LOGIN_REPLY บอกโปรแกรมทางฝั่งไคลเอนต์ถึงผลลัพธ์ในการตรวจสอบชื่อผู้ใช้และรหัสผ่าน
- LEARN_DT บอกโปรแกรมทางฝั่งเซิร์ฟเวอร์ให้ทำการเรียนรู้ต้นไม้ตัดสินใจ
- LEARN_DT_REPLY บอกโปรแกรมทางฝั่งไคลเอนต์ถึงผลลัพธ์ในการเรียนรู้ต้นไม้ตัดสินใจ
- LEARN_NN บอกโปรแกรมทางฝั่งเซิร์ฟเวอร์ให้ทำการเรียนรู้นิวรอลเน็ตเวิร์ก
- LEARN_NN_REPLY บอกโปรแกรมทางฝั่งไคลเอนต์ถึงผลลัพธ์ในการเรียนรู้นิวรอลเน็ตเวิร์ก
- LEARN_NB บอกโปรแกรมทางฝั่งเซิร์ฟเวอร์ให้ทำการเรียนรู้เบย์อย่างง่าย
- LEARN_NB_REPLY บอกโปรแกรมทางฝั่งไคลเอนต์ถึงผลลัพธ์ในการเรียนรู้เบย์อย่างง่าย
- LEARN_AR บอกโปรแกรมทางฝั่งเซิร์ฟเวอร์ให้ทำการค้นหากฎความสัมพันธ์
- LEARN_AR_REPLY บอกโปรแกรมทางฝั่งไคลเอนต์ถึงผลลัพธ์ในการค้นหากฎความสัมพันธ์

อัลกอริทึมสำหรับการทำเหมืองข้อมูล
Algorithms for Data Mining

บทที่ 2: ทฤษฎี
Chapter 2: Theories

ต้นไม้ตัดสินใจ

บทนำ (Introduction to Decision Trees)

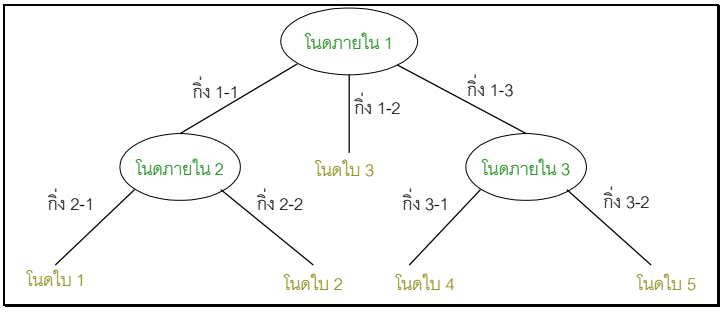
ต้นไม้ตัดสินใจนับว่าเป็นวิธีการเรียนรู้ที่ใช้มากที่สุดแบบหนึ่งในการเรียนรู้ของเครื่อง การเรียนรู้แบบนี้เป็นการเรียนรู้โดยการแยกแยะ (classification) ข้อมูลออกเป็นกลุ่ม (class) ต่างๆ โดยใช้คุณสมบัติ (attribute) ของข้อมูลในการแยกแยะ ต้นไม้ตัดสินใจที่ได้จากการเรียนรู้ทำให้ทราบว่าคุณสมบัติใดของข้อมูลที่เป็นตัวกำหนดการแยกแยะ และคุณสมบัติแต่ละตัวของข้อมูลมีความสำคัญมากน้อยต่างกันอย่างไร ซึ่งเป็นประโยชน์ช่วยให้ผู้ใช้สามารถวิเคราะห์ข้อมูลและตัดสินใจได้ถูกต้องยิ่งขึ้น

การแทนต้นไม้ตัดสินใจ (Decision Tree Representation)

ผลลัพธ์ของการเรียนรู้ต้นไม้ตัดสินใจจะแสดงในรูปต้นไม้ ซึ่งประกอบไปด้วย

1. **โนดภายใน** (internal node) คือ คุณสมบัติต่างๆ ของข้อมูล ซึ่งเมื่อข้อมูลใดๆ ตกลงมาที่โนด จะใช้คุณสมบัตินี้เป็นตัวตัดสินใจว่าข้อมูลจะไปในทิศทางใด โดยโนดภายในที่เป็นจุดเริ่มต้นของต้นไม้เรียกว่าโนดราก
2. **กิ่ง** (branch, link) เป็นค่าคุณสมบัติของคุณสมบัติในโนดภายในที่แตกกิ่งนี้ออกมา ซึ่งโนดภายในจะแตกกิ่งเป็นจำนวนเท่ากับจำนวนค่าคุณสมบัติของโนดภายในนั้น
3. **โนดใบ** (leaf node) คือกลุ่มต่างๆ ซึ่งเป็นผลลัพธ์ในการแยกแยะข้อมูล

ตัวอย่างของต้นไม้ตัดสินใจแสดงในรูปที่ 2.1



รูปที่ 2.1 การแทนต้นไม้ตัดสินใจ

ลักษณะการเรียนรู้ของต้นไม้ตัดสินใจ

- ผลการเรียนรู้แสดงอยู่ในรูปที่เข้าใจง่าย ทำให้ง่ายต่อการวิเคราะห์คุณสมบัติที่มีผลต่อการแยกแยะกลุ่มต่างๆ
- แต่ละเส้นทางจากโนดรากถึงโนดใบสามารถแสดงให้อยู่ในรูปกฎ IF-THEN ได้
- มีความทนทานต่อข้อมูลมีสัญญาณรบกวน (noisy data) เช่น คุณสมบัติที่ไม่เกี่ยวข้อง และค่าคุณสมบัติที่ผิดพลาดหรือขาดหาย
- การเรียนรู้มีความรวดเร็วเมื่อเทียบกับอัลกอริทึมสำหรับแยกแยะชนิดอื่น
- นำไปใช้ในการวิเคราะห์ความเสี่ยงของลูกหนี้ การวินิจฉัยทางการแพทย์ การวิเคราะห์กลุ่มดาว และงานทางด้านธุรกิจและวิทยาศาสตร์อื่นๆ

วิธีการเรียนรู้ของต้นไม้ตัดสินใจ (Decision Tree Learning)

การสร้างต้นไม้ตัดสินใจจะเป็นแบบการค้นหาจากบนลงล่างแบบตะกราม (top-down greedy search) โดยเริ่มจากการเลือกคุณสมบัติที่ดีที่สุดมาสร้างเป็นโนดราก เมื่อข้อมูลผ่านการแบ่งแยกที่โนดรากตามค่าคุณสมบัติของโนดรากแล้ว ก็จะหาคุณสมบัติที่ดีที่สุดของข้อมูลที่ผ่านการแบ่งแยกนั้นมาสร้างเป็นโนดลูกของโนดรากนั้นต่อไป และจะวนสร้างโนดลูกและต้นไม้ย่อยของแต่ละกิ่งไปเรื่อยๆ จนกว่าข้อมูลที่ผ่านการแบ่งแยกนั้นจะจัดอยู่ในกลุ่มเดียวกัน หรือจำนวนข้อมูลที่ผ่านการแบ่งแยกในกิ่งหนึ่งๆ มีค่าน้อยกว่าค่าที่กำหนดไว้

จากวิธีการข้างต้นก่อให้เกิดคำถามตามมาสองประการ คือ

1. จะนิยามคุณสมบัติที่ดีที่สุดว่าอย่างไร
2. การสร้างต้นไม้ตัดสินใจจากบนลงล่างแบบตะกรามจะให้ต้นไม้ที่ดีที่สุด (optimal tree) หรือไม่

สำหรับคำถามแรกนั้น แต่ละอัลกอริทึมก็ได้นิยามค่าความดีของคุณสมบัติแตกต่างกันไป อัลกอริทึม CART นิยามความดีของคุณสมบัติโดยใช้ค่าสัมประสิทธิ์จีนี (Gini) [Breiman, et] แต่ที่แพร่หลายที่สุดคือการใช้ค่ามาตรฐานเกน (Gain criterion) ของอัลกอริทึม ID3 [Quinlan, 1986] และ C4.5 [Quinlan, 1993] ซึ่งได้จากการคำนวณโดยอาศัยทฤษฎีสารสนเทศ (information theory) และค่าเอนโทรปี (entropy) ซึ่งจะกล่าวถึงในหัวข้อถัดไป

สำหรับคำถามที่สองนั้น มีการพิสูจน์มาแล้วว่าการสร้างต้นไม้ที่ดีที่สุดจากข้อมูลที่กำหนดให้เป็นปัญหาเอ็นพีสมบูรณ์ ซึ่งใช้งานไม่ได้อย่างมีประสิทธิภาพในทางปฏิบัติ อย่างไรก็ตาม มีการทดลองและรายงานผลว่าการสร้างต้นไม้ตัดสินใจแบบตะกรามก็ให้ผลลัพธ์เป็นต้นไม้ที่มีลักษณะใกล้เคียงกับต้นไม้ที่ดีที่สุดเช่นกัน

ค่ามาตรฐานเกน (Gain Criterion)

วิธีการสร้างต้นไม้ตัดสินใจแบบ ID3 จะใช้ค่ามาตรฐานเกนในการตัดสินใจเลือกคุณสมบัติที่จะใช้เป็นรากหรือโนดในต้นไม้ โดยการคำนวณค่าเกนของคุณสมบัติแต่ละตัวเมื่อทดลองใช้คุณสมบัตินั้นแบ่งตัวอย่าง แล้วเลือกคุณสมบัติที่มีค่าเกนสูงที่สุดมาเป็นรากหรือโนด ค่าเกนนี้คำนวณได้โดยใช้ความรู้จากทฤษฎีสารสนเทศ ซึ่งมีสาระสำคัญคือ ค่าสารสนเทศของข้อมูลขึ้นอยู่กับค่าความน่าจะเป็นของข้อมูล ซึ่งสามารถวัดอยู่ในรูปของบิต (bits) จากสูตร

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

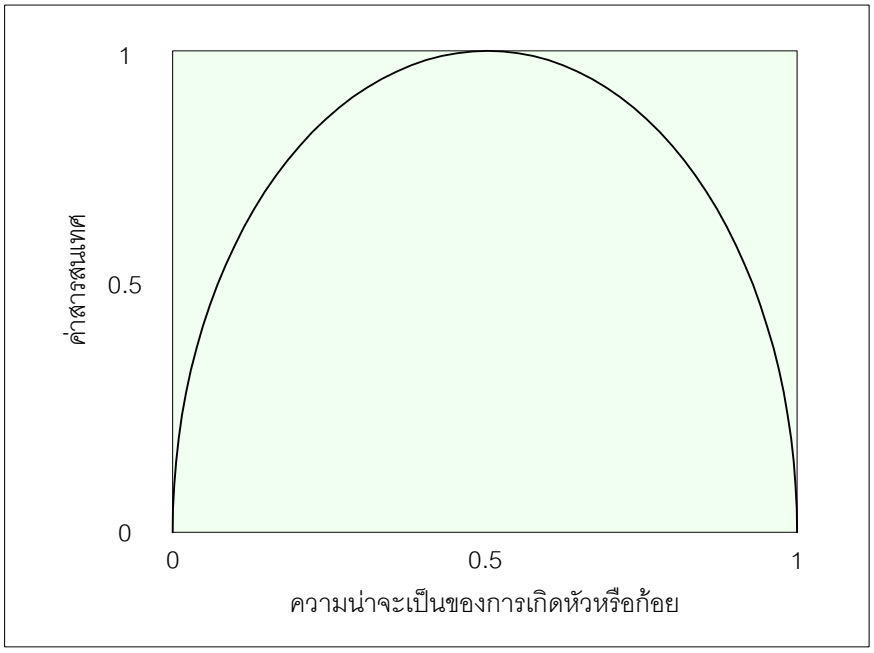
ถ้าให้ชุดของข้อมูล M ประกอบด้วยค่าที่เป็นไปได้ คือ $\{m_1, m_2, \ldots, m_n\}$ และให้ความน่าจะเป็นที่จะเกิดค่า m_i มีค่าเท่ากับ $P(m_i)$ จะได้ว่าค่าสารสนเทศของ M หรือค่าเอนโทรพีของ M เขียนแทนด้วย $I(M)$ คำนวณได้จากสูตร

$$I(M) = \sum_{i}^{n} -P(m_i)\log_2 P(m_i)$$

ตัวอย่างเช่น ในการโยนหัวโยนก้อย ชุดข้อมูล M จะประกอบด้วยค่าที่เป็นไปได้ {หัว, ก้อย} และถ้าให้ความน่าจะเป็นที่ออกหัวเท่ากับ P(หัว) และความน่าจะเป็นที่ออกก้อยเท่ากับ P(ก้อย) ดังนั้นค่าสารสนเทศของการโยนหัวโยนก้อย จะคำนวณได้จากสูตร

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

เมื่อความน่าจะเป็นของการเกิดหัวหรือก้อยมีค่าต่าง ๆ กันจะสามารถคำนวณค่าสารสนเทศของการโยนหัวโยนก้อยได้ต่าง ๆ กันดังรูปที่ 2.2 ซึ่งจะเห็นได้ว่าเมื่อออกหัวหมดหรือก้อยหมด ค่าสารสนเทศจะเป็น 0 และค่าสารสนเทศจะค่อย ๆ เพิ่มขึ้นจนสูงที่สุดเมื่อความน่าจะเป็นของการเกิดหัวเท่ากับความน่าจะเป็นของการเกิดก้อย แสดงให้เห็นว่าค่าสารสนเทศที่น้อยจะบ่งบอกว่าข้อมูลชุดนั้นมีความแตกต่างกันน้อยหรือเกือบจะเป็นพวกเดียวกัน แต่ถ้าค่าสารสนเทศสูงจะบ่งบอกว่าข้อมูลชุดนั้นมีความแตกต่างกันมาก หรือประกอบด้วยตัวอย่างหลายพวกที่มีจำนวนใกล้เคียงกัน



รูปที่ 2.2 ค่าสารสนเทศของการโยนหัวโยนก้อย

ในการเลือกคุณสมบัติที่จะมาเป็นโนดรากจะอาศัยค่ามาตรฐานเกน ซึ่งคำนวณจากค่าสารสนเทศทั้งหมดของชุดข้อมูลนั้นลบด้วยค่าสารสนเทศหลังจากเลือกคุณสมบัติใดคุณสมบัติหนึ่งเป็นราก ค่าสารสนเทศหลังจากแบ่งตามคุณสมบัติที่เลือกแล้วจะคำนวณได้จาก ค่าผลรวมของผลคูณระหว่างค่าสารสนเทศของแต่ละโนดกับอัตราส่วนของตัวอย่างในแต่ละกิ่งต่อตัวอย่างทั้งหมดที่โนดนั้น ๆ หรือความน่าจะเป็นของค่าที่เป็นไปได้ของแต่ละคุณสมบัติ

ถ้าให้ข้อมูลสอนคือ T และคุณสมบัติที่เป็นโนด คือ X และมีค่าทั้งหมดที่เป็นไปได้ n ค่า โนดปัจจุบันจะแบ่งตัวอย่าง T ออกตามกิ่งเป็น $\{t_1,t_2,\ldots,t_n\}$ ตามค่าที่เป็นไปได้ของ X ดังนั้นจึงสามารถคำนวณค่าสารสนเทศหลังจากแบ่งตามคุณสมบัติ X ดังนี้

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

ค่ามาตรฐานเกนของคุณสมบัติ X สามารถคำนวณได้จากการลบค่าสารสนเทศทั้งหมดที่โนดนี้กับค่าสารสนเทศที่ได้หลังจากแบ่งด้วยคุณสมบัติ X ดังนี้

$$Gain(X) = I(T) - I_x(T)$$

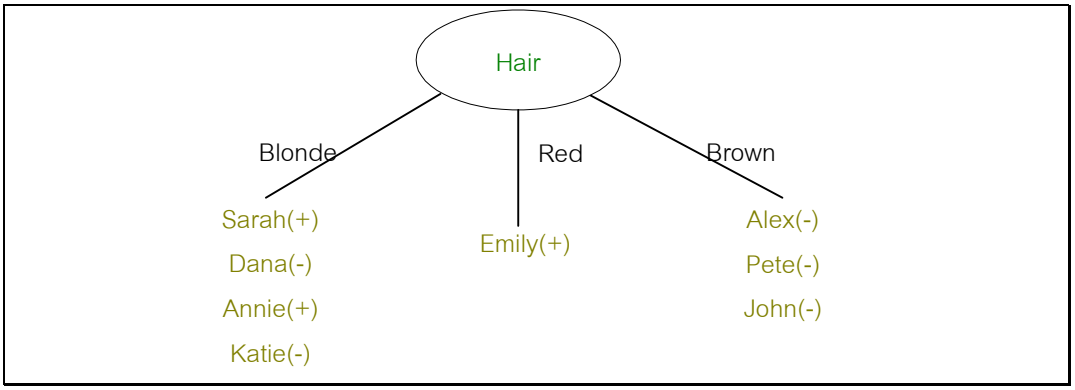
ตัวอย่างการเรียนรู้ต้นไม้ตัดสินใจโดยใช้ค่ามาตรฐานเกน

ต้องการศึกษาว่าปัจจัยใดมีผลทำให้เกิดผิวไหม้ในคนที่อาบแดดบ้าง จึงเก็บตัวอย่างทั้งหมด 8 ตัวอย่างดังตารางที่ 2.1

ตารางที่ 2.1 ตัวอย่างของข้อมูลสำหรับต้นไม้ตัดสินใจ

Name	Hair	Height	Weight	Lotion	Result
Sarah	Blonde	Average	Light	No	Sunburned (+)
Dana	Blonde	Tall	Average	Yes	None (−)
Alex	Brown	Short	Average	Yes	None (−)
Annie	Blonde	Short	Average	No	Sunburned (+)
Emily	Red	Average	Heavy	No	Sunburned (+)
Pete	Brown	Tall	Heavy	No	None (−)
John	Brown	Average	Heavy	No	None (−)
Katie	Blonde	Short	Light	Yes	None (−)

การสร้างต้นไม้ตัดสินใจจากกลุ่มตัวอย่างนี้ เริ่มด้วยการเลือกคุณสมบัติมาเป็นโนดเพื่อแบ่งแยกข้อมูล ซึ่งในที่นี้ มีคุณสมบัติที่สนใจอยู่ 4 อย่าง คือ Hair, Height, Weight และ Lotion คุณสมบัติที่ถูกเลือกมาสร้างโนดนั้นจะเลือกคุณสมบัติที่แบ่งแยกข้อมูลได้ดีที่สุด หรือมีค่ามาตรฐานเกนมากที่สุดนั่นเอง เช่น พิจารณาคุณสมบัติ Hair ซึ่งแบ่งแยกข้อมูลได้ดังรูปที่ 2.3



รูปที่ 2.3 การแบ่งแยกข้อมูลของคุณสมบัติ Hair

คำนวณหาค่ามาตรฐานเกนได้

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

= 0.45

ในทำนองเดียวกัน คุณสมบัติอื่นจะมีค่ามาตรฐานเกนเป็น

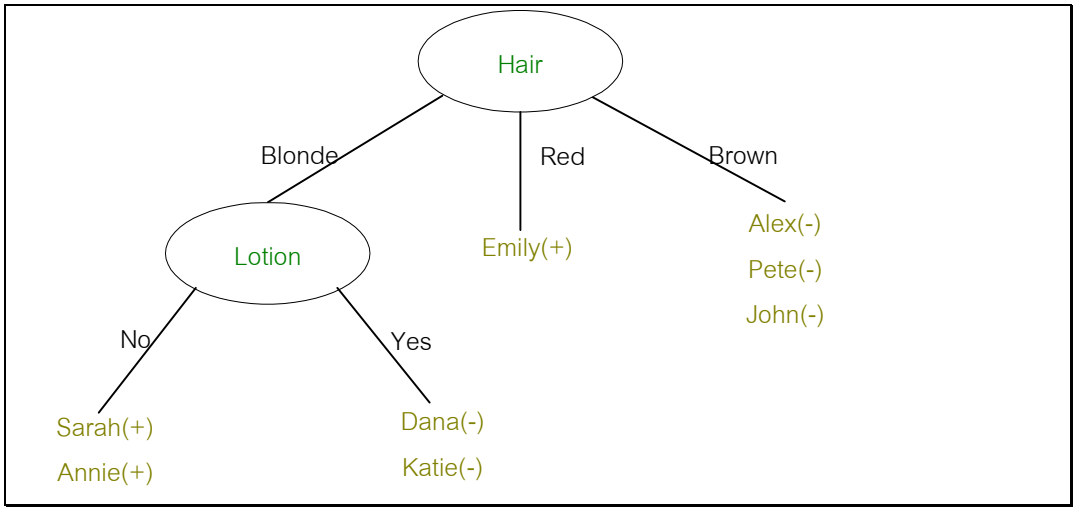
Gain (Height) = 0.26 Gain (Weight) = 0.01 Gain (Lotion) = 0.34

จึงเลือกคุณสมบัติ Hair มาเป็นโนดแรกของต้นไม้ตัดสินใจ แต่คุณสมบัติ Hair เพียงอย่างเดียวไม่สามารถแยกตัวอย่างบวกและลบออกจากกันได้ในกิ่งของค่าคุณสมบัติ Blonde จึงต้อง

พิจารณาคุณสมบัติอื่นเพื่อแบ่งแยกข้อมูลที่ตกลงมายังกิ่งนี้ โดยค่ามาตรฐานเกนของแต่ละคุณสมบัติมีค่าดังนี้

Gain (Height) = 0.00 Gain (Weight) = -0.5 Gain (Lotion) = 0.50

จึงเลือกคุณสมบัติ Lotion ซึ่งมีค่ามาตรฐานเกนมากที่สุดมาแบ่งแยกข้อมูลต่อไป ซึ่งพบว่าเมื่อแบ่งแยกแล้ว ข้อมูลที่ผ่านการแบ่งแยกมีกลุ่มเดียวกัน จึงได้ต้นไม้ตัดสินใจดังรูปที่ 2.4



รูปที่ 2.4 ต้นไม้ตัดสินใจที่เป็นผลลัพธ์จากการเรียนรู้

ค่ามาตรฐานอัตราส่วนเกน (Gain Ratio Criterion)

ใน ID3 จะใช้ค่ามาตรฐานเกนเป็นหลักในการเลือกคุณสมบัติที่จะใช้เป็นรากหรือโนด แต่ใน C4.5 ได้เพิ่มการใช้ค่ามาตรฐานอัตราส่วนเกน (Gain Ratio criterion) ในการตัดสินใจเลือกคุณสมบัติที่จะใช้เป็นรากหรือโนดอีกอย่างหนึ่ง เนื่องจากค่ามาตรฐานเกนจะมีอคติ (Bias) อย่างมากกับข้อมูลที่ประกอบด้วยคุณสมบัติที่มีค่าที่เป็นไปได้จำนวนมาก ๆ เช่นข้อมูลที่ประกอบด้วยคุณสมบัติหมายเลขประจำตัว ซึ่งปกติจะไม่ซ้ำกันในแต่ละตัวอย่าง ถ้าแบ่งข้อมูลตามคุณสมบัตินี้จะทำให้ได้จำนวนตัวอย่างเพียง 1 ตัวอย่างต่อ 1 กิ่งของต้นไม้ และชุดตัวอย่างย่อยที่ได้จะประกอบด้วยข้อมูลกลุ่มเดียว เมื่อคำนวณค่าสารสนเทศจากการแบ่งตัวอย่างบนคุณสมบัตินี้ จะได้เท่ากับ 0 ทำให้ค่าเกนที่ได้ในคุณสมบัตินี้จะสูงที่สุดเสมอ

การแก้ไขความอคติของค่ามาตรฐานเกนสามารถทำได้โดยการปรับค่ามาตรฐานเกนให้ถูกต้อง โดยใช้ค่าสารสนเทศของการแบ่งแยก (split information) ของคุณสมบัติแต่ละตัว ถ้าให้ T คือชุดของตัวอย่าง เมื่อแบ่งตัวอย่างนี้ตามคุณสมบัติ X จะได้ชุดของตัวอย่างย่อยในแต่ละกิ่ง คือ $\{t_1,t_2,\ldots,t_n\}$ จำนวน n ชุด ตามค่าที่เป็นไปได้ในคุณสมบัติ X เมื่อคำนวณค่าสารสนเทศของการแบ่งแยกได้ ดังนี้

$$\text{ค่าสารสนเทศของการแบ่งแยก} = -\sum_{i=1}^{n} \frac{|t_i|}{|T|} \log_2 \frac{|t_i|}{|T|}$$

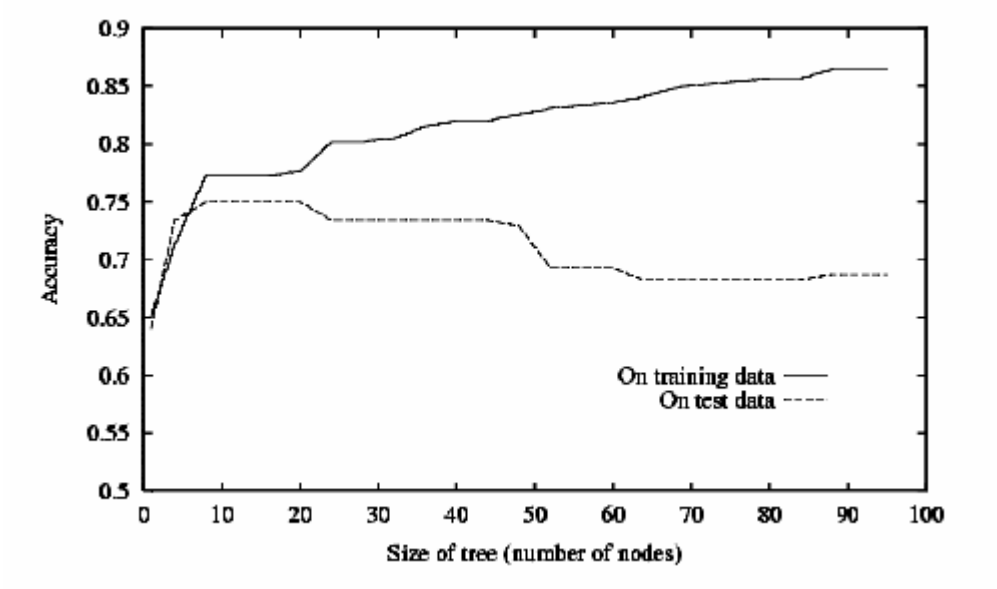
ค่าสารสนเทศของการแบ่งแยกนี้จะแสดงถึงระดับการกระจายของข้อมูล เมื่อแบ่งข้อมูลตัวอย่าง T เป็น n ชุดย่อยตามคุณสมบัติ X โดยค่านี้จะสูงสุดเมื่อ $|t_i|$ เป็น 1 เท่ากันในทุกกิ่ง และ

ลดลงเมื่อค่า $|t_i|$ เพิ่มขึ้น เมื่อนำค่านี้ไปหารค่ามาตรฐานเกนจะได้ค่ามาตรฐานอัตราส่วนเกน ซึ่งช่วยแก้ไขความอคติของค่ามาตรฐานเกนได้ โดยทำให้ค่ามาตรฐานอัตราส่วนเกนในการแบ่งด้วยคุณสมบัติที่มีการกระจายสูงถูกปรับลดลง ดังนั้นค่ามาตรฐานอัตราส่วนเกนในคุณสมบัติของตัวอย่างที่มีการกระจายตัวของข้อมูลสูงดังที่กล่าวมาแล้วจึงไม่มีค่าสูงที่สุดเสมอ โดย

ค่ามาตรฐานอัตราส่วนเกน = ค่ามาตรฐานเกน / ค่าสารสนเทศของการแบ่งแยก

การตัดเล็มต้นไม้ตัดสินใจ

ในการสร้างต้นไม้ตัดสินใจด้วยวิธีที่กล่าวมาแล้ว จะแบ่งข้อมูลจนกระทั่งได้ข้อมูลที่เป็นกลุ่มเดียวกันหมดหรือมีจำนวนข้อมูลเหลือน้อยเกินกว่าค่าที่กำหนดไว้ กล่าวได้ว่าต้นไม้ที่สร้างได้นี้เฉพาะเจาะจงกับข้อมูลที่ใช้สอนเป็นอย่างมาก หรือที่เรียกว่า โอเวอร์ฟิตติ้ง(overfitting) ถ้าข้อมูลที่ใช้สอนมีความผิดพลาดหรือมีน้อยเกินไปจะทำให้การแยกแยะข้อมูลใหม่ๆ ที่ไม่ใช่ข้อมูลที่ใช้สอนมีความผิดพลาดสูง โดยเฉพาะอย่างยิ่งเมื่อต้นไม้มีขนาดใหญ่ ตัวอย่างของการเกิดโอเวอร์ฟิตติ้งแสดงรูปที่ 2.5 ซึ่งแสดงให้เห็นว่าต้นไม้ตัดสินใจที่มีขนาดใหญ่ขึ้นจะให้ความถูกต้องบนข้อมูลสอนมากขึ้น (เส้นทึบในรูป) แต่เมื่อนำไปใช้งานจริงความถูกต้องจะลดลงตามลำดับของขนาดต้นไม้ ซึ่งวัดได้จากความถูกต้องบนตัวอย่างทดสอบ (เส้นประในรูป)



รูปที่ 2.5 ความถูกต้องจากการแยกแยะข้อมูลของต้นไม้ตัดสินใจเทียบกับขนาดของต้นไม้ตัดสินใจ

มีหลักการของ*ใบมีดโกนของอ็อคแคม* (*Occam's Razor*) ที่กล่าวว่า สมมุติฐานที่สั้นกว่าที่สามารถอธิบายข้อมูลได้เหมือนกันจะเป็นสมมติฐานที่ดีกว่า เพราะฉะนั้น ต้นไม้ที่สร้างได้จึงควรทำการตัดเล็มเพื่อให้ได้ต้นไม้ขนาดเล็กกว่าเดิมและลดความเฉพาะเจาะจงกับข้อมูลที่ใช้สอน การตัดเล็มต้นไม้อาจแบ่งได้เป็นสองประเภทคือ การตัดเล็มขณะที่เรียนรู้ (pre-pruning) และการตัดเล็มหลังการเรียนรู้ (post-pruning) จากต้นไม้ถูกสร้างเสร็จแล้ว

การตัดเล็มขณะเรียนรู้เกิดขึ้นขณะที่กำลังเรียนรู้และสร้างต้นไม้ตัดสินใจโดยดูว่าถ้าโนดลูกที่สร้างนั้นมีความผิดพลาดในการแยกแยะกลุ่มมากกว่าความผิดพลาดของกลุ่มที่มีอยู่เดิมเมื่อยังไม่ได้แยกแยะก็ไม่จำเป็นที่จะต้องสร้างโนดนั้นและตัดต้นไม้ย่อยที่มีโนดนั้นเป็นรากออกไป

การตัดเล็มหลังการเรียนรู้มีหลายวิธี อัลกอริทึม C4.5 จะตัดเล็มโดยใช้ค่าความผิดพลาด (error-based pruning) [Quinlan, 1993] คือจะมีการรวมต้นไม้ย่อยเข้าด้วยกันเป็นโนดเดียวก็ต่อเมื่อหลังจากรวมแล้วไม่ทำให้ค่าความผิดพลาดเพิ่มขึ้น โดยค่าความผิดพลาดที่ใช้ทดสอบกับข้อมูลที่ไม่เคยเห็นจะใช้ค่าจำกัดบนของการกระจายแบบไบโนเมียล (binomial distribution) ที่ระดับความเป็นอิสระเท่ากับ CF (confidence level) กล่าวคือถ้ามีข้อมูล N ตัวที่โนด และมีข้อมูล E ตัวเป็นข้อมูลที่มีกลุ่มไม่ถูกต้องหรือไม่ตรงกับกลุ่มส่วนใหญ่ ค่าความผิดพลาดที่โนดนี้จะเขียนสามารถเขียนได้ในรูป $U_{cf}(E,N)$

การประมาณค่าความผิดพลาดสำหรับโนดเมื่อใช้กับข้อมูลที่ไม่เคยเห็น จะอยู่บนข้อกำหนดที่ว่าขนาดของตัวอย่างสอนเท่ากับขนาดตัวอย่างของข้อมูลที่ไม่เคยเห็น ดังนั้นถ้าใบประกอบด้วยข้อมูลจำนวน N ตัว ค่าความผิดพลาดที่คาดไว้ของข้อมูลแต่ละตัวเท่ากับ $U_{cf}(E,N)$ ซึ่งสามารถคาดได้ว่าจะมีจำนวนข้อมูลที่แยกแยะผิดพลาดเท่ากับ N x $U_{cf}(E,N)$ ตัว เมื่อทดสอบบนข้อมูลที่ไม่เคยเห็น ซึ่งถ้าคำนวณจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดของแต่ละกิ่งรวมกันแล้วมากกว่าจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดของโนดที่แตกกิ่งนั้น ก็จะตัดโนดที่เป็นลูกในทุกกิ่งของโนดที่แตกกิ่งนั้นออกให้หมดจนเหลือเฉพาะโนดที่แตกกิ่งนั้นไว้โนดเดียว

การทำจินตทัศน์

นิยาม

- **การทำจินตทัศน์** คือ การนำเสนอสารสนเทศออกมาเป็นรูปภาพ โดยมีเป้าหมายหลักคือทำให้ผู้ชมเข้าใจเนื้อหาของสารสนเทศนั้นได้อย่างมีประสิทธิภาพ
- **สารสนเทศ** อาจเป็นได้ทั้ง ข้อมูล (data) กระบวนการ (process) ความสัมพันธ์ (relation) หรือแนวความคิด (concept)
- **การนำเสนอด้วยรูปภาพ** (graphical presentation) คือการจัดการเกี่ยวกับองค์ประกอบศิลป์ (graphical entity and graphical attribute) อันได้แก่ จุด เส้น ตัวอักษร สี ขนาด ตำแหน่ง รูปร่าง รูปทรง พื้นผิว ที่ว่าง แสงและเงา
- **การเข้าใจเนื้อหาของผู้ชม** หมายถึง การที่ผู้ชมสามารถตรวจจับรายละเอียดต่างๆ ของข้อมูล สามารถตรวจวัดข้อมูลในด้านต่างๆ พร้อมทั้งสามารถเปรียบเทียบข้อมูลซึ่งกันและกัน การทำให้ผู้ชมเข้าใจเนื้อหามากขึ้นนั้น อาจทำโดยการใช้ระบบโต้ตอบ (interactive technique) หรือทำให้ผู้ชมสามารถมองเห็นข้อมูลได้จากหลายมุมมอง เป็นต้น

จินตทัศน์ที่ดีควรเป็นอย่างไร

- **มีประสิทธิผล** (effective) ผู้ใช้สามารถเข้าใจถึงสิ่งที่ต้องการนำเสนอได้โดยง่าย
- **แม่นยำ** (accurate) ข้อมูลที่นำเสนอไม่ควรมีความคลาดเคลื่อน ความแม่นยำเป็นสิ่งจำเป็นอย่างมากต่อผู้ใช้ในการประเมินค่าเชิงปริมาณของข้อมูล
- **มีประสิทธิภาพ** (efficient) ไม่นำเสนอให้ดูยุ่งจนเกินไป ลดจุดและเส้นที่ไม่จำเป็น
- **มีความสวยงาม** (aesthetics)
- **ยืดหยุ่น** (adaptable) ตอบสนองความต้องการของผู้ใช้ได้หลายรูปแบบ

การทำจินตทัศน์ของต้นไม้ตัดสินใจ

ต้นไม้ถือเป็นโครงสร้างของกราฟรูปแบบหนึ่ง โดยเป็นกราฟที่ทุกคู่โนดมีเส้นทางต่อถึงกันได้เพียงเส้นทางเดียวเท่านั้น นอกจากนี้ต้นไม้ยังจัดเรียงโนดเป็นระดับชั้น โดยมีโนดรากอยู่ที่ระดับชั้นบนสุด การทำจินตทัศน์ของต้นไม้ตัดสินใจต้องคำนึงถึงสองเรื่องใหญ่ๆ คือ เรื่องการวาดต้นไม้ หรือการแสดงผลลัพธ์ของต้นไม้ตัดสินใจที่ได้จากการเรียนรู้ให้อยู่ในรูปต้นไม้ที่ดูเข้าใจได้ง่าย และเรื่องการนำเสนอข้อมูลที่อยู่ภายในต้นไม้ตัดสินใจ

การวาดต้นไม้

แนวคิดการวาดต้นไม้ในระยะเริ่มต้นเป็นแนวคิดเพื่อวาดรูปต้นไม้สองมิติแบบทั่วไป อาจเรียกได้ว่าเป็นวิธีวาดต้นไม้แบบคลาสสิก โดยเริ่มต้นจากการนิยามว่าภาพต้นไม้ที่ได้จากการวาดควรจะมีลักษณะใดจึงจะสวยงามและดูเข้าใจง่าย ซึ่งในแนวคิดของ Reingold และ Tilford [Reingold & Tilford, 1981] รูปต้นไม้ที่ดีควรมีลักษณะดังนี้

- โนดแต่ละโนดจะอยู่ในพื้นที่ต่างกัน
- ไม่มีโนดใดอยู่ใกล้รากมากกว่าโนดในระดับก่อนหน้าที่จะถึงโนดนี้
- โนดที่อยู่ในระดับเดียวกันจะอยู่ในแถวแนวนอนเดียวกัน และทุกแถวจะขนานกัน
- โนดในระดับเดียวกันจะเรียงตามลำดับการท่องต้นไม้ตามลำดับชั้น
- สำหรับต้นไม้ทวิภาค ลูกทางซ้ายจะอยู่ด้านซ้ายโนดแม่ ลูกทางขวาจะอยู่ด้านขวาของโนดแม่ ถ้ามีลูกเดียว ลูกนั้นจะอยู่ด้านล่างของโนดแม่
- โนดแม่จะอยู่ตรงกลางของโนดลูก
- ต้นไม้ย่อยควรมีลักษณะอย่างเดียวกันไม่ว่าจะอยู่ส่วนใด

ลักษณะในสี่ข้อแรกนี้จะเป็นหลักประกันว่าต้นไม้ที่วาดเป็นกราฟแบน (planar graph) คือไม่มีเส้นเชื่อมใดที่ตัดกัน ขณะอีกสามลักษณะต่อไปเป็นไปเพื่อความสมมาตรและความสวยงามของต้นไม้

ต้นไม้ลักษณะดังกล่าวสามารถสร้างขึ้นโดยใช้อัลกอริทึม Reingold-Tilford ซึ่งอาศัยการท่องต้นไม้จากล่างขึ้นบน เริ่มจากโนดใบที่ไม่มีลูก จนกระทั่งเมื่อถึงโนดที่มีลูกจะนำต้นไม้ย่อยที่เป็นลูกทุกต้นมาคำนวณหาตำแหน่งใหม่ที่สัมพัทธ์กับโนดนี้ซึ่งเป็นรากของต้นไม้ย่อยชั่วคราว โดยการวางตำแหน่งต้นไม้ย่อยจะต้องวางให้ไม่มีโนดใดทับกัน และมีระยะห่างระหว่างต้นไม้ย่อยเท่าๆ กัน

การนำเสนอข้อมูลของต้นไม้ตัดสินใจ

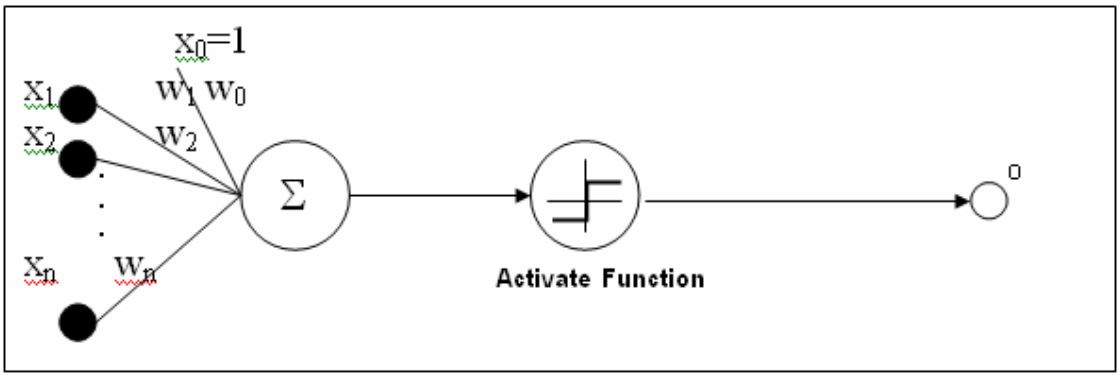
ข้อมูลของต้นไม้ตัดสินใจต้นหนึ่งๆ ที่เป็นผลลัพธ์จากกระบวนการเรียนรู้ต้นไม้ตัดสินใจประกอบไปด้วยข้อมูลโดยรวมของต้นไม้ และข้อมูลของแต่ละโนดและแต่ละกิ่งของต้นไม้ ซึ่งมีดังนี้

- **ข้อมูลของต้นไม้** ประกอบไปด้วย จำนวนโนดของต้นไม้ จำนวนกิ่งของต้นไม้ ความผิดพลาดในการแยกแยะของข้อมูลที่ใช้สอน ความผิดพลาดในการแยกแยะของข้อมูลที่ใช้ทดสอบ จำนวนข้อมูลทั้งหมด ตารางความผิดพลาดของการแยกแยะข้อมูล เส้นทางสำคัญของต้นไม้ที่ข้อมูลส่วนใหญ่ผ่าน ฯลฯ
- **ข้อมูลในแต่ละโนดของต้นไม้** ประกอบไปด้วย ชื่อโนด จำนวนข้อมูลที่ตกลงมาที่โนด จำนวนกลุ่มต่างๆ ของข้อมูลที่ตกลงมาที่โนด จำนวนข้อมูลที่มีกลุ่มไม่ตรงกับกลุ่มส่วนใหญ่ที่อยู่ในโนด ค่ามาตรฐานอัตราส่วนเกนของคุณสมบัติต่างๆ ที่ถูกนำมาคัดเลือกเพื่อสร้างโนด กฎของคุณสมบัติในเส้นทางที่จะมายังโนด ความสามารถในการแยกแยะกลุ่มของโนด ฯลฯ

การทำจินตทัศน์ของต้นไม้ตัดสินใจควรออกแบบให้ผู้ใช้สามารถเข้าถึงและดูข้อมูลเหล่านี้ได้โดยง่าย นอกจากนี้ยังอาจมีรายงานสรุปรวมเฉพาะทางด้านต่างๆ เช่น สรุปรวมการแยกแยะกลุ่มต่างๆ หรือสรุปรวมกฎทุกกฎของต้นไม้ เพื่อความสะดวกในการเข้าถึงและเปรียบเทียบข้อมูลของผู้ใช้

วิธีการเรียนรู้ของนิวรอลเน็ตเวิร์ก (Neural Network Learning)

แนวคิดของนิวรอลเน็ตเวิร์กได้มาจากการจำลองการทำงานของเซลล์สมองของมนุษย์ โดยหน่วยที่ย่อยที่สุดของนิวรอลเน็ตเวิร์กเรียกว่าเพอร์เซ็ปตรอน (Perceptron) ซึ่งเทียบได้กับเซลล์สมองของมนุษย์หนึ่งนิวรอน (neuron) เพอร์เซ็ปตรอนนี้จะทำหน้าที่รับอินพุตซึ่งเป็นเวคเตอร์ของจำนวนจริงเข้ามา พร้อมคำนวณค่าเหล่านี้โดยให้น้ำหนักของอินพุตแต่ละตัวแตกต่างกันดังแสดงในรูปที่ 2.6 เอาต์พุตที่ได้จะถูกนำไปคำนวณค่าผิดพลาด (error) เพื่อนำมาปรับน้ำหนักของอินพุตต่อไป



รูปที่ 2.6 เพอร์เซ็ปตรอน (Perceptron)

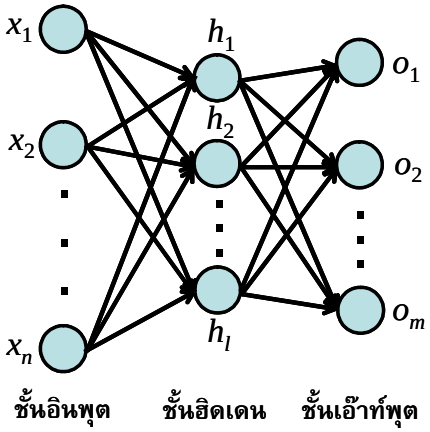
ในการเรียนรู้ของเพอร์เซ็ปตรอนมีกระบวนการดังนี้

- เริ่มจากการสุ่มค่าน้ำหนัก Wi
- เทียบเพอร์เซ็ปตรอนกับทุกตัวอย่างที่สอนทีละตัว และแก้ไขน้ำหนักเมื่อเพอร์เซ็ปตรอนแยกตัวอย่างผิดพลาด
- วนทำซ้ำกับตัวอย่างที่สอน จนกระทั่งเพอร์เซ็ปตรอนแยกตัวอย่างได้ถูกต้องทั้งหมด
- ในการแก้ไขน้ำหนัก น้ำหนักจะถูกปรับตาม $Wi \leftarrow Wi + \Delta Wi$

โดยที่ $\Delta Wi = \alpha \times (t-o) \times x_i = \alpha \times error \times input$ เมื่อ t เป็นผลลัพธ์ที่ถูกต้อง o เป็นผลลัพธ์ที่ได้จากเพอร์เซปตรอน และ α เป็นค่าที่แสดงอัตราการเรียนรู้

เพอร์เซ็ปตรอนเดี่ยวสามารถแสดงระนาบตัดสินใจแบบเชิงเส้น (linear decision surface) เท่านั้น ต้องใช้เน็ตเวิร์กแบบหลายชั้น (multilayer network) ถึงจะสามารถแสดงระนาบตัดสินใจแบบไม่เชิงเส้น (non-linear decision surface) ซึ่งมีความที่ซับซ้อนมากกว่าได้ ซึ่งในงานวิจัยนี้เราจะใช้ขั้นตอนวิธีแบ็กพรอพาเกชันขั้นตอนวิธีในการเรียนรู้

แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก (Backpropagation neural network) เป็นเน็ตเวิร์กที่มีได้หลายนิวรอนและมีได้หลายชั้น (multilayer) และทำงานกับฟังก์ชันซิกมอยด์ (Sigmoid function) ซึ่งเป็นฟังก์ชันที่สามารถแยกตัวอย่างได้แบบไม่เชิงเส้น ทำให้ทำงานได้ดีกว่าเพอร์เซ็ปตรอนเดี่ยว ๆ โครงสร้างของแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กแสดงในรูปที่ 2.7



รูปที่ 2.7 แบ็กพรอพาเกชันนิวรอลเน็ตเวิร์ก

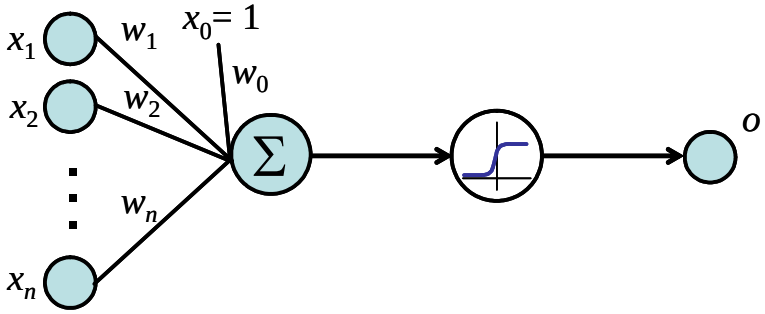
ตัวอย่างในรูปด้านบนแสดงเน็ตเวิร์กป้อนไปหน้าแบบหลายชั้น ซึ่งประกอบด้วยชั้นอินพุต ชั้นฮิดเดนหรือชั้นซ่อน และชั้นเอ๊าท์พุต ในรูปแสดงชั้นฮิดเดนเพียงชั้นเดียว แต่อาจมีมากกว่าหนึ่งชั้นก็ได้ และเส้นเชื่อมจะเชื่อมต่อเป็นชั้นๆ ไม่ข้ามชั้น จากชั้นอินพุตไปชั้นฮิดเดน ถ้ามีชั้นฮิดเดนมากกว่าหนึ่งชั้น ก็เชื่อมต่อกันไป และสุดท้ายจากชั้นฮิดเดนไปชั้นเอ๊าท์พุต เน็ตเวิร์กป้อนไปหน้าแบบหลายชั้นนี้จะไม่มีเส้นเชื่อมย้อนกลับจะมีแต่เส้นเชื่อมไปข้างหน้าอย่างเดียว กล่าวคือ ไม่มีเส้นเชื่อมจากโนด (node) ในชั้นเอ๊าท์พุตส่งกลับมายังโนดในชั้นฮิดเดนหรือชั้นอินพุต โนดแต่ละโนดแทนนิวรอนหนึ่งตัว

ในการปรับค่าเวกเตอร์น้ำหนักโดยอัลกอริทึมแบ็คพรอพาเกชัน [Rumelhart, et al., 1986] นั้น เราต้องนิยามค่าผิดพลาดการสอนสำหรับเน็ตเวิร์ก $E(\vec{w})$ จากนั้นจะหาค่าเวกเตอร์น้ำหนักที่ให้ค่าผิดพลาดต่ำสุด นิยามค่าผิดพลาดดังนี้

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

โดยที่ *outputs* คือเซตของโนดเอ๊าท์พุตในเน็ตเวิร์ก t_{kd} และ o_{kd} เป็นค่าเอ๊าท์พุตเป้าหมายและเอ๊าท์พุตที่ได้จากเน็ตเวิร์กตามลำดับของโนดเอ๊าท์พุตที่ k ของตัวอย่างตัวที่ d อัลกอริทึมแบ็คพรอพาเกชันจะค้นหาเวกเตอร์น้ำหนักที่ให้ค่าผิดพลาดต่ำสุด แต่ในกรณีของเน็ตเวิร์กป้อนไปหน้าแบบหลายชั้นนี้ ค่าต่ำสุดมีมากกว่าหนึ่งจุด ดังนั้นคำตอบของแบ็คพรอพาเกชันจึงเป็นค่าต่ำสุดเฉพาะที่

ฟังก์ชันกระตุ้นที่นิยมใช้ในแบ็กพรอพาเกชันนิวรอลเน็ตเวิร์กเป็นฟังก์ชันที่แบบไม่เชิงเส้น (nonlinear function) เรียกว่าฟังก์ชันซิกมอยด์ (Sigmoid function) ดังรูปที่ 2.8



รูปที่ 2.8 ฟังก์ชันซิกมอยด์

กระบวนการการเรียนรู้

ขั้นตอนของกระบวนการเรียนรู้มีดังต่อไปนี้

กำหนดให้ตัวอย่างที่ใช้ในการเรียนรู้แต่ละตัวอย่างอยู่ในรูป $< \vec{x}, \vec{t} >$

โดยที่ $\vec{x}$ เป็นอินพุตเวกเตอร์ของนิวรอลเน็ตเวิร์ก,

$\vec{t}$ เป็นเวกเตอร์เป้าหมายของนิวรอลเน็ตเวิร์ก

η เป็นค่าอัตราการเรียนรู้ (learning rate)

x_{ji} เป็นอินพุตของโนด j ซึ่งมาจากโนด i และ

w_{ji} เป็นค่าน้ำหนักของโนด j ซึ่งมาจากโนด i

1. สร้างนิวรอลเน็ตเวิร์กตามโครงสร้างที่ต้องการ กำหนดจำนวนนิวรอนของแต่ละชั้น
2. กำหนดค่าน้ำหนักเริ่มต้นแบบสุ่มให้มีค่าน้อยๆ (เช่น ระหว่าง -0.05 ถึง 0.05)
3. ทำการปรับค่าน้ำหนักด้วยขั้นตอนวิธีดังนี้

   สำหรับแต่ละอินพุตเวกเตอร์ $(\vec{x}, \vec{t})$ ในเซตตัวอย่างที่ใช้เรียนรู้

   - ใช้ตัวอย่าง x เป็นอินพุต ทำการคำนวณหาค่า o_u ของทุกๆ นิวรอนในนิวรอลเน็ตเวิร์ก
   - คำนวณค่าความคลาดเคลื่อน δ_k ของทุกนิวรอน k ในชั้นเอาต์พุต

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

   - คำนวณค่าความคลาดเคลื่อน δ_h ของทุกนิวรอน h ในชั้นฮิดเดน

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

   - ปรับค่าน้ำหนักของเส้นเชื่อม w_{ji}

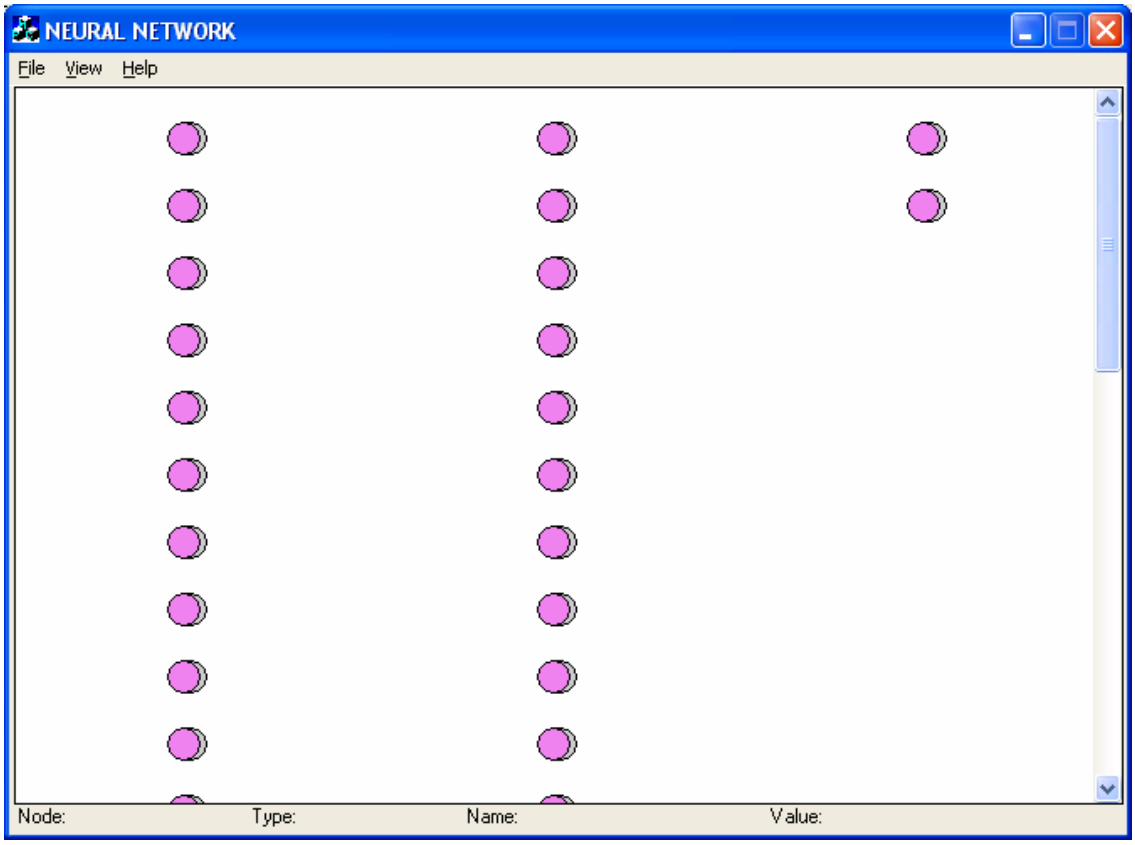
     $$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

   เมื่อ

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

การจินตทัศน์ของตัวแยกแยะแบบนิวรอลเน็ตเวิร์ก (Visualizing Neural Network Classifier)

การจินตทัศน์ข้อมูลที่ได้จากนิวรอลเน็ตเวิร์กที่จะพัฒนานี้จะใช้วิธีการแสดงข้อมูลในรูปลำดับชั้นของเน็ตเวิร์ก โดยจะแบ่งเป็น 3 ชั้น คือ ชั้นอินพุต ชั้นฮิดเดน และชั้นเอาท์พุต จะเห็นได้ว่ามีการแบ่งเน็ตเวิร์กเป็น 2 ระดับ โดยระดับแรกจะเป็นการนำเสนอแบบกว้าง ๆ เพื่อให้เห็นถึงโครงสร้างทั้งหมดของระบบโดยรวม ว่ามีคุณสมบัติใดบ้าง และมีกลุ่มใดบ้าง ดังรูปที่ 2.9



รูปที่ 2.9 โครงสร้างของจินตทัศน์นิวรอลเน็ตเวิร์ก

จากระดับแรกเราสามารถจะดูลึกถึงรายละเอียดได้โดยสามารถคลิกที่โนดของนิวรอลเน็ตเวิร์ก เพื่อดูในรายละเอียดที่เกี่ยวข้องกับโนดที่เราคลิกในระดับที่ 2 โดยมีรายละเอียดที่เกี่ยวข้องดังนี้

1. ชื่อของคุณสมบัติ
2. ชื่อของกลุ่ม
3. น้ำหนักของเส้นเชื่อม

รายละเอียดจะเป็นตัวเลือกที่ผู้ใช้สามารถเลือกดูได้ ซึ่งในตอนแรกจะไม่แสดง จนกว่าผู้ใช้จะต้องการทราบรายละเอียด ไม่ว่าจะเป็น ชื่อของคุณสมบัติ ชื่อของกลุ่ม น้ำหนักของเส้นเชื่อม โดยคลิกเมาส์เลือกตามที่ต้องการ มีตัวเลือก 2 ตัว คือ รายละเอียดของชื่อต่างๆ และรายละเอียดของน้ำหนัก

ถ้าต้องการให้กลับคืนสู่ระดับแรก สามารถคลิกเมาส์ที่ปุ่มขวา ก็จะทำให้กลับสู่หน้าแรก ก่อนคลิกดูรายละเอียดได้อีกครั้ง เพื่อให้สามารถคลิกเพื่อดูรายละเอียดของโนดอื่นๆ ได้อีกด้วย

การเรียนรู้เบย์อย่างง่าย

วิธีการเรียนรู้แบบเบย์ (Bayesian Learning)

วิธีการเรียนรู้แบบเบย์ใช้หลักการของความน่าจะเป็น โดยมีสมมติฐานว่าปริมาณของความสนใจขึ้นอยู่กับการกระจายความน่าจะเป็น (probability distribution) ดังนั้นการตัดสินใจที่ดีที่สุดจึงได้จากการวิเคราะห์ความน่าจะเป็นนี้กับข้อมูลที่นำมาแยกแยะ

วิธีการเรียนรู้เบย์อย่างง่าย (Naive Bayesian Learning)

การเรียนรู้เบย์อย่างง่ายเป็นวิธีการแยกแยะข้อมูลที่มีประสิทธิภาพวิธีหนึ่ง โดยผลลัพธ์ที่ได้นั้นเทียบได้กับผลลัพธ์จากอัลกอริทึมที่มีความซับซ้อนกว่า เช่น C4.5 [Dougherty, Kohavi & Sahami, 1995] การเรียนรู้เบย์อย่างง่ายมีพื้นฐานมาจากกฎของเบย์ แต่จะลดความซับซ้อนลงโดยจะเพิ่มสมมติฐานที่ว่าคุณสมบัติต่าง ๆ ของข้อมูลจะไม่ขึ้นต่อกัน หรือกล่าวได้ว่าความน่าจะเป็นของข้อมูลที่จะเป็นกลุ่ม C_i สำหรับข้อมูลที่มีคุณสมบัติ n ตัว $X=\{A_1,\ldots,A_n\}$ หรือใช้สัญลักษณ์ว่า $P(C_i \mid A_1,\ldots,A_n)$ คือ

$$P(C_i|A_1,\ldots,A_n) = \frac{P(A_1,\ldots,A_n|C_i)\times P(C_i)}{P(A_1,\ldots,A_n)} \quad \text{จากกฎของเบย์}$$

$$= \frac{\prod_{j=1}^{n} P(A_j|C_i)\times P(C_i)}{P(A_1,\ldots,A_n)} \quad \text{จากสมมติฐานที่ว่าคุณสมบัติแต่ละตัวไม่ขึ้นต่อกัน}$$

การนำวิธีการเรียนรู้เบย์อย่างง่ายไปใช้ มีวิธีการดังต่อไปนี้คือ

1. หา $P(C_i \mid A_1,\ldots,A_n)$ จากสมการด้านบนสำหรับทุกๆ กลุ่ม i
2. นำค่าที่ได้มาเปรียบเทียบกัน กลุ่มที่มีค่าความน่าจะเป็นสูงสุดคือคำตอบ

จากการที่ตัวแยกแยะเบย์อย่างง่ายขึ้นอยู่กับสมมติฐานที่ว่าคุณสมบัติแต่ละอย่างของข้อมูลไม่ขึ้นต่อกัน ทำให้ดูเหมือนจะนำไปใช้งานในทางปฏิบัติได้อย่างไม่มีประสิทธิภาพมากนัก แต่ผลจากการทดลองของ Domingos และ Pazzani [Domingos & Pazzani, 1996] ชี้ว่า ตัวแยกแยะเบย์อย่างง่ายสามารถแยกแยะข้อมูลได้อย่างมีประสิทธิภาพแม้สมมติฐานไม่เป็นจริง ด้วยเหตุนี้จึงทำให้วิธีการนี้เป็นที่นิยมนำไปประยุกต์ใช้กันมาก

การทำให้ข้อมูลเป็นแบบไม่ต่อเนื่อง (Discretization of Continuous Values)

ข้อมูลที่นำมาเรียนรู้จะมีอยู่สองชนิดคือ แบบต่อเนื่อง (continuous) และไม่ต่อเนื่อง (discrete) โดยปกติตัวแยกแยะเบย์อย่างง่ายสามารถแยกแยะข้อมูลที่ต่อเนื่องได้ แต่จากการทดลองในช่วงที่ผ่านมาแสดงให้เห็นว่าการทำให้ข้อมูลที่ต่อเนื่องทั้งหมดเป็นแบบไม่ต่อเนื่องก่อน (discretization) ที่จะนำมาแยกแยะ จะช่วยเพิ่มความถูกต้องของตัวแยกแยะได้ [Fayyad & Irani, 1993]

งานวิจัยเกี่ยวกับการทำให้ข้อมูลเป็นแบบไม่ต่อเนื่องยังคงเป็นหัวข้อวิจัยที่เป็นที่สนใจของนักวิจัยอยู่ มีหลายวิธีได้ถูกคิดขึ้นมา ยกตัวอย่างเช่น การแบ่งเป็นช่วง ช่วงละเท่า ๆ กัน (Binning), การคำนวณหาเอนโทรปีที่น้อยที่สุด (minimal entropy), การแบ่งโดยให้มีความถี่เท่ากัน (equal frequency interval), อัลกอริทึม1R [Holte, 1993] เป็นต้น

การจินตทัศน์ของตัวแยกแยะเบย์อย่างง่าย (Visualizing Naïve Bayesian Classifier)

การจินตทัศน์ข้อมูลที่ได้จากตัวแยกแยะเบย์อย่างง่ายที่จะพัฒนานี้จะใช้วิธีการแสดงข้อมูลในรูปแถวลำดับของแผนภูมิวงกลม [Becker, et al., 1997] ซึ่งเป็นวิธีที่ใช้ในโปรแกรม MineSet ของบริษัท Silicon Graphics Inc.

การค้นหากฎความสัมพันธ์

บทนำ (Introduction to Association Rule Discovery)

การค้นหากฎความสัมพันธ์ (association rule) ในฐานข้อมูลขนาดใหญ่ถือเป็นงานหลักงานหนึ่งในการทำเหมืองข้อมูล กฎความสัมพันธ์สามารถเขียนได้ในรูปเซตไอเท็มที่เป็นเหตุไปสู่เซตไอเท็มที่เป็นผลซึ่งมีรากฐานมาจากการวิเคราะห์ทางการตลาด เช่น ลูกค้าที่ซื้อผ้าอ้อมส่วนใหญ่จะซื้อจุกนมด้วย ก็สามารถเขียนกฎความสัมพันธ์ได้เป็น {ผ้าอ้อม} $\rightarrow$ {จุกนม} เป็นต้น พื้นฐานของการค้นหากฎความสัมพันธ์ประกอบด้วยนิยามต่างๆ เหล่านี้

- **เซตไอเท็ม** (I) คือเซตที่มีไอเท็มทั้งหมดเป็นสมาชิก ซึ่งไอเท็มในที่นี้อาจเป็นชื่อสินค้าหรือชื่อใดๆ ที่เป็นหน่วยพื้นฐานที่จะนำมาทำการเรียนรู้
- **ทรานแซคชัน** (T) เป็นเซตย่อยของเซตไอเท็ม โดยที่ $T \subseteq I$
- **เซตข้อมูล** (D) คือเซตที่มีทรานแซคชันทุกตัวเป็นสมาชิก
- เรากล่าวว่าทรานแซคชัน T บรรจุเซตย่อยของไอเท็ม X ก็ต่อเมื่อ $X \subseteq T$

เพราะฉะนั้นจึงนิยามกฎความสัมพันธ์ได้ว่า

- **กฎความสัมพันธ์** (association rule) คือการอุปนัยในรูปแบบ $X \rightarrow Y$ เมื่อ $X \subset Y$, $Y \subset I$ และ $X \cap Y = \phi$

นอกจากนี้ กฎความสัมพันธ์ทุกกฎจะประกอบไปด้วยค่าสนับสนุน (support) และค่าความมั่นใจ (confidence) ซึ่งมีนิยามดังนี้

- กฎความสัมพันธ์ $X \rightarrow Y$ มี**ค่าสนับสนุน**เท่ากับ s ในเซตข้อมูล D ก็ต่อเมื่อ s% ของทรานแซคชันใน D บรรจุ $X \cup Y$
- กฎความสัมพันธ์ $X \rightarrow Y$ มี**ค่าความเชื่อมั่น**เท่ากับ c ในเซตข้อมูล D ก็ต่อเมื่อ c% ของทรานแซคชันใน D ที่บรรจุ X บรรจุ Y ด้วย

ปัญหาการค้นหากฎความสัมพันธ์เป็นปัญหาทางคณิตศาสตร์ซึ่งสามารถนิยามได้ดังนี้

- **การค้นหากฎความสัมพันธ์** คือการหากฎความสัมพันธ์ทั้งหมดในทรานแซคชันทุกตัวของเซตข้อมูลที่กำหนดให้ โดยกฎความสัมพันธ์ที่หาได้ทั้งหมดจะต้องมีค่าสนับสนุนมากกว่าค่าสนับสนุนน้อยสุดที่กำหนดไว้ และมีค่าความมั่นใจมากกว่าค่าความมั่นใจน้อยสุดที่กำหนดไว้เช่นกัน

วิธีการค้นหากฎความสัมพันธ์ (Discovery of Association Rule)

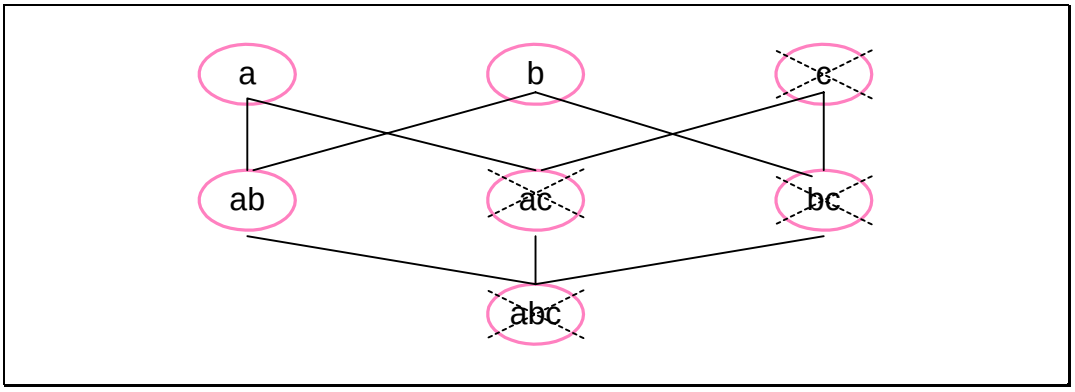
การค้นหากฎความสัมพันธ์สามารถแบ่งย่อยได้เป็นสองขั้นตอน คือการหาเซตไอเท็มที่มีค่าสนับสนุนมากกว่าค่าสนับสนุนน้อยสุดที่กำหนดให้ เรียกเซตนี้ว่า เซตไอเท็มปรากฏบ่อย – frequent itemset) และการนำเซตไอเท็มเหล่านี้มาสร้างเป็นกฎความสัมพันธ์ต่อไป

การหาเซตไอเท็มปรากฏบ่อย

การหาเซตไอเท็มที่ปรากฏบ่อยๆ เป็นปัญหาของการค้นหาในสเปซของการจัดหมู่ของเซตไอเท็มทั้งหมด ซึ่งสเปซในการค้นหานี้จะมีขนาดเพิ่มขึ้นเป็นเอ็กซ์โปเนนเชียลกับจำนวนไอเท็มทั้งหมดในเซตไอเท็ม ซึ่งถ้าเซตไอเท็มมีขนาดใหญ่ สเปซในการค้นหาจะมีขนาดใหญ่มากๆ ขึ้นหลายเท่าเป็นเงาตามตัว

แต่ในการค้นหานี้ไม่จำเป็นต้องไล่แจกแจงค้นหาในทุกการจัดหมู่ เพราะสามารถตัดเซตไอเท็มที่มีเซตย่อยเป็นเซตไอเท็มที่ไม่ใช่เซตไอเท็มปรากฏบ่อยออกได้ หรือกล่าวอีกนัยหนึ่งได้ว่า ถ้าแจกแจงแล้วพบเซตไอเท็มใดที่ไม่ใช่เซตไอเท็มปรากฏบ่อยก็ไม่จำเป็นต้องแจกแจงเซตไอเท็มอื่นๆ ที่มีเซตไอเท็มนี้เป็นเซตย่อยอีกต่อไป

ตัวอย่างเช่นไอเท็ม a, b และ c สามารถสร้างสเปซของเซตไอเท็มทั้งหมดได้ดังรูปที่ 2.10 ซึ่งถ้ารู้ว่าเซตไอเท็ม {c} ไม่ใช่เซตไอเท็มปรากฏบ่อยแล้ว ก็ไม่จำเป็นที่จะต้องสร้างหรือตรวจสอบ {ac}, {bc} และ {abc} ซึ่งมี {c} เซตย่อย



รูปที่ 2.10 สเปซการจัดหมู่ของสมาชิกในเซตไอเท็ม {a,b,c}

มีอัลกอริทึมหลายวิธีที่พยายามลดสเปซการค้นหาให้น้อยลงกว่านี้ โดยอัลกอริทึมบางวิธีได้ตัดเล็มสเปซให้เหลือเฉพาะเซตไอเท็มปรากฏบ่อยแบบปิดเท่านั้น

การค้นหาเซตไอเท็มแบบปิดนี้อาจทำได้ทั้งการค้นหาแบบแนวลึกก่อนและการค้นหาแบบแนวกว้างก่อน ซึ่งเมื่อทำการตรวจสอบเซตไอเท็มใดๆ จะต้องนับทรานแซคชันที่บรรจุเซตไอเท็มนั้น การนับทรานแซคชันนี้อาจทำได้ทั้งแบบการไล่นับจากเซตข้อมูล และการนับโดยการอินเตอร์เซกชันของเซตที่เก็บหมายเลขทรานแซคชัน

เราสามารถแบ่งอัลกอริทึมในการค้นหากฎความสัมพันธ์ได้เป็นสี่ประเภทใหญ่ๆ คือ

- การค้นหาแบบแนวกว้างก่อนที่ใช้การนับทรานแซคชัน
- การค้นหาแบบแนวกว้างก่อนที่ใช้การอินเตอร์เซกชันของเซตที่เก็บหมายเลขทรานแซคชัน
- การค้นหาแบบแนวลึกก่อนที่ใช้การนับทรานแซคชัน

- การค้นหาแบบแนวลึกก่อนที่ใช้การอินเตอร์เซกชันของเซตที่เก็บหมายเลขทรานแซคชัน

อัลกอริทึม Apriori [Agrawal & Srikant, 1994] เป็นอัลกอริทึมพื้นฐานที่แพร่หลายและใช้ในวงกว้าง โดยทำการค้นหาแบบแนวกว้างก่อนและใช้การนับทรานแซคชัน ซึ่งจะสร้างและตรวจสอบเซตไอเท็มปรากฏบ่อยทีละชั้น เริ่มจากเซตไอเท็มที่มีจำนวนสมาชิกเท่ากับหนึ่ง ถ้าเซตไอเท็มใดมีค่าสนับสนุนน้อยกว่าค่าสนับสนุนที่กำหนดให้ก็จะตัดเซตไอเท็มนั้นออก ไม่นำไปสร้างเซตไอเท็มในชั้นถัดไป การทำงานของอัลกอริทึมจะวนอย่างนี้ไปเรื่อยๆ จนกระทั่งไล่ไปทุกระดับชั้น หรือไม่เหลือเซตไอเท็มที่จะสร้างเซตไอเท็มในชั้นถัดไป

ในการนับจำนวนทรานแซคชัน อัลกอริทึม Apriori จะไล่ทรานแซคชันครั้งเดียวในแต่ละระดับชั้น ในการตรวจดูว่าทรานแซคชันนั้นบรรจุเซตไอเท็มใดบ้าง เพื่อความรวดเร็วจะเก็บเซตไอเท็มในแต่ละระดับชั้นทั้งหมดไว้ในโครงสร้างต้นไม้แฮช (hash tree)

อัลกอริทึมที่เราเลือกใช้ในงานวิจัยนี้คือ อัลกอริทึม CHARM [Mohammed & Ching-Jui, 2002] ซึ่งเป็นอัลกอริทึมที่ใช้การค้นหาแบบแนวลึกก่อนและการอินเตอร์เซกชันของเซตที่เก็บหมายเลขทรานแซคชัน ข้อดีของ CHARM มีดังนี้

- สามารถตัดเล็มสเปซการค้นหาได้มากกว่าโดยอาศัยคุณสมบัติปิดของเซตไอเท็ม โดยที่เซตไอเท็มที่ไม่ปิดจะถูกตัดออกไป
- ไม่ต้องไล่นับทรานแซคชันทุกครั้งเพราะเก็บหมายเลขของทรานแซคชันไว้ในเซตซึ่งผูกติดอยู่กับเซตไอเท็ม
- จากการเก็บหมายเลขของทรานแซคชันไว้ในเซตซึ่งผูกติดอยู่กับเซตไอเท็ม ทำให้สามารถรู้สเปซของทรานแซคชันในขณะที่ค้นหาเซตไอเท็ม
- เมื่อเข้าสู่กระบวนการหากฎความสัมพันธ์ จะช่วยลดจำนวนกฎความสัมพันธ์ที่ซ้ำซ้อน ทั้งนี้เนื่องจากได้ตัดเซตไอเท็มที่ไม่ปิดออกไปแล้วนั่นเอง

อัลกอริทึม CHARM

<u>ทฤษฎีพื้นฐาน</u>

สเปซการค้นหาของเซตไอเท็มแท้จริงแล้วก็คือ ความสัมพันธ์แบบเซตย่อยของสมาชิกในเซตที่เป็นเซตกำลังของเซตไอเท็มนั่นเอง หรือสามารถเขียนในรูปคณิตศาสตร์ได้เป็น $(P(I),\subseteq)$ กล่าวได้ว่าเซตกำลังของเซตไอเท็มที่มีความสัมพันธ์แบบเซตย่อยเป็นเซตแบบมีลำดับบางส่วน (partial order) เนื่องจากมีคุณสมบัติสะท้อน (reflexive) ปฏิสมมาตร (antisymmetric) และถ่ายทอด (transitive) ในขณะเดียวกันความสัมพันธ์แบบเซตย่อยก็ใช้ไม่ได้กับสมาชิกทุกคู่ในเซตกำลังของเซตไอเท็ม

แต่ละเซตไอเท็มจะถูกบรรจุอยู่ในทรานแซคชันจำนวนหนึ่ง ในอัลกอริทึม CHARM นี้จะนิยามเซตไอเท็มควบคู่ไปกับเซตของหมายเลขทรานแซคชันที่เซตไอเท็มนี้ถูกบรรจุอยู่ ทั้งนี้เพื่อความสะดวกในการหาเซตไอเท็มแบบปิด และเป็นการค้นหาเซตไอเท็มปรากฏบ่อยแบบใหม่ที่มีทั้งสเปซของทรานแซคชันควบคู่กันกับสเปซของเซตไอเท็ม ก่อนอื่น จะต้องมีการนิยามความสัมพันธ์ระหว่างเซตไอเท็มและเซตของหมายเลขทรานแซคชัน (δ) ดังนี้

$$\delta \subseteq I \times T, \quad x \,\delta\, y \leftrightarrow x \in T_y$$

จากนั้นจึงนิยามฟังก์ชันความสัมพันธ์ระหว่างเซตไอเท็มและเซตของหมายเลขทรานแซคชันสองฟังก์ชันดังนี้

$$t : P(I) \to P(T), \quad t(X) = \bigcap_{x \in X} \{ y \in T \mid x \, \delta \, y \}$$
$$i : P(T) \to P(I), \quad i(Y) = \bigcap_{y \in Y} \{ x \in I \mid x \, \delta \, y \}$$

ซึ่งทั้งสองฟังก์ชันก่อให้เกิดการเชื่อมต่อกันของเซตสองเซตแบบมีลำดับบางส่วน $(P(I), \subseteq)$ และ $(P(T), \subseteq)$ เรียกได้ว่าเป็นการเชื่อมต่อแบบกาลอยส์ (Galois connection) ซึ่งเป็นการเชื่อมต่อที่มีคุณสมบัติดังนี้

- ถ้า $X_1 \subseteq X_2$ แล้ว $t(X_1) \supseteq t(X_2)$
- ถ้า $Y_1 \subseteq Y_2$ แล้ว $i(Y_1) \supseteq i(Y_2)$
- $X \subseteq i(t(X)), \quad Y \subseteq t(i(Y))$

จากฟังก์ชันทั้งสองและคุณสมบัติสามประการของตัวดำเนินการปิด (closure operator) คือ ภาคขยาย (extension) ความเป็นทางเดียว (monotonicity) และ นิจพล (idempotency) เราสามารถพิสูจน์ได้ว่า

ถ้า $c_{it}(X) = i(t(X))$ แล้ว $c_{it} : P(I) \to P(T)$ เป็นตัวดำเนินการปิดของเซ็ตของไอเท็ม

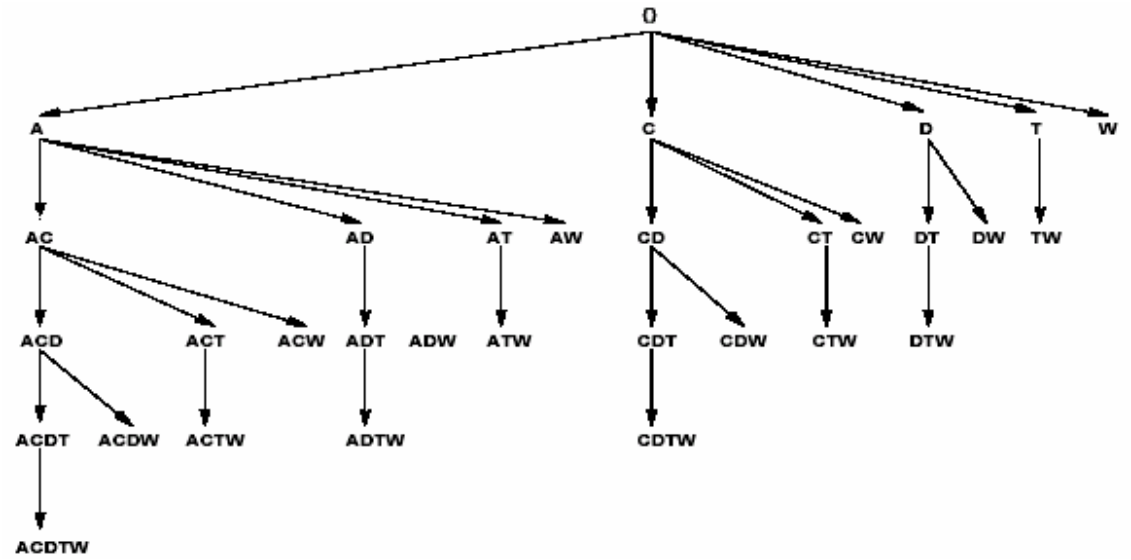
เพราะฉะนั้น สามารถนิยามเซตไอเท็มแบบปิดได้ดังนี้

X เป็นเซตของไอเท็มแบบปิด ก็ต่อเมื่อ $X = c_{it}(X)$

ซึ่งสามารถพิสูจน์ต่อไปได้อีกว่า ไม่จำเป็นที่จะต้องหาทุกเซตไอเท็มปรากฏบ่อย แค่หาเพียงเซตไอเท็มปรากฏบ่อยแบบปิดก็ให้ความถูกต้องและเพียงพอต่อความต้องการแล้ว

<u>การค้นหาเซตไอเท็มปรากฏบ่อย</u>

การค้นหาเซตไอเท็มปรากฏบ่อยตามอัลกอริทึม CHARM ใช้วิธีการค้นหาแบบแนวลึกก่อน โดยในระดับชั้นแรกจะแตกกิ่งเท่ากับจำนวนสมาชิกในเซตไอเท็ม การค้นหาเริ่มจากลูกตัวแรกจะสร้างลูกโดยอาศัยการยูเนียนของเซตตัวเองและเซตอื่นที่อยู่ในระดับเดียวกัน จากนั้นจึงวนซ้ำสร้างโนดใหม่ไปเรื่อยๆ ต้นไม้ที่ได้จะมีการจัดหมู่ทุกแบบของสมาชิกของเซตไอเท็ม ดังแสดงในรูปที่ 2.11



รูปที่ 2.11 โครงสร้างการค้นหาแบบแนวลึกก่อนของเซตไอเท็ม {A,C,D,T,W}

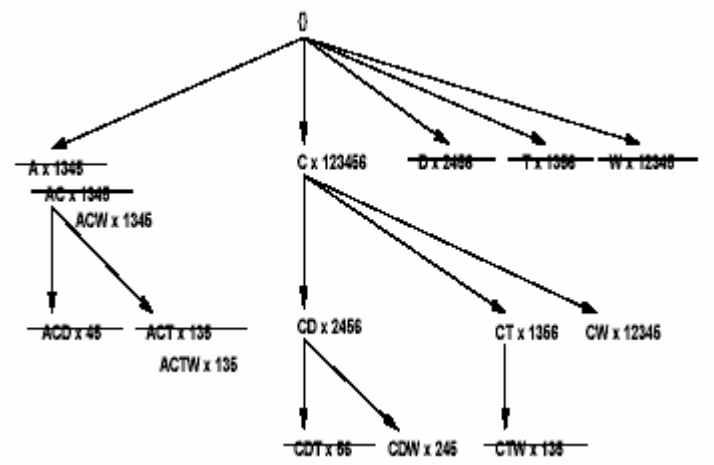
นอกจากจะพิจารณาถึงเซตไอเท็มแล้ว ในทุกโนดของต้นไม้จะต้องพิจารณาเซตของหมายเลขทรานแซคชันควบคู่กันไป โดยเริ่มต้นเมื่อสร้างโนดของเซตไอเท็มที่มีจำนวนสมาชิกเพียงตัวเดียว (โนดในระดับชั้นแรก) ก็จะเก็บเซตของหมายเลขทรานแซคชันที่บรรจุเซตไอเท็มนี้อยู่ควบคู่ไปด้วยกัน ขั้นตอนการสร้างโนดใหม่จากการยูเนียนก็จะสร้างเซตของหมายเลขทรานแซคชันใหม่โดยการอินเตอร์เซกชันควบคู่กันไปด้วย เนื่องจาก $t(X_1 \cup X_2) = t(X_1) \cap t(X_2)$ (สามารถพิสูจน์ได้จากนิยามฟังก์ชัน t)

เมื่อมีการเก็บเซตของหมายเลขทรานแซคชันที่บรรจุเซตไอเท็มแล้ว การตัดเซตไอเท็มที่ไม่ใช่เซตไอเท็มปรากฏบ่อยจะกระทำได้โดยง่าย เพียงนับจำนวนสมาชิกของเซตของหมายเลขทรานแซคชันเท่านั้น

นอกจากจะตัดเซตไอเท็มที่ไม่ใช่เซตไอเท็มปรากฏบ่อยแล้ว ยังตัดเซตไอเท็มที่ไม่ปิดออกด้วย โดยมีหลักการตัดเล็มเซตไอเท็มที่ไม่ปิดสี่ประการคือ

- ถ้า $t(X_1) = t(X_2)$ แล้ว กล่าวได้ว่า $X_1 \cup X_2$ เป็นเซตไอเท็มแบบปิดของทั้ง X_1 และ X_2 ทำให้สามารถแทนที่ X_1 ด้วย $X_1 \cup X_2$ และตัด X_2 ออกจากการพิจารณาได้
- ถ้า $t(X_1) \subset t(X_2)$ แล้ว กล่าวได้ว่า $X_1 \cup X_2$ เป็นเซตไอเท็มแบบปิดของ X_1 ทำให้สามารถแทนที่ X_1 ด้วย $X_1 \cup X_2$ แต่ไม่สามารถตัด X_2 ออกจากการพิจารณาได้
- ถ้า $t(X_1) \supset t(X_2)$ แล้ว กล่าวได้ว่า $X_1 \cup X_2$ เป็นเซตไอเท็มแบบปิดของ X_2 ทำให้สามารถแทนที่ X_2 ด้วย $X_1 \cup X_2$ แต่ไม่สามารถตัด X_1 ออกจากการพิจารณาได้
- ถ้า $t(X_1) \neq t(X_2)$ แล้ว ในกรณีนี้ $X_1 \cup X_2$ ไม่ได้เป็นเซตไอเท็มแบบปิดของทั้ง X_1 และ X_2 จึงไม่สามารถแทนที่ หรือตัด X_1 และ X_2 ออกจากการพิจารณาได้

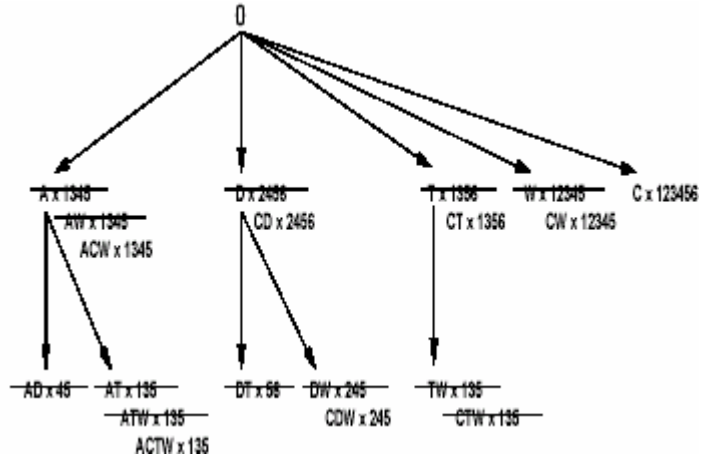
รูปที่ 2.12 ด้านล่างแสดงถึงตัวอย่างการค้นหาเซตไอเท็มปรากฏบ่อยแบบปิดโดยใช้อัลกอริทึม CHARM ซึ่งมีการตัดทั้งเซตไอเท็มที่ไม่ปราฏบ่อย และเซตไอเท็มที่ไม่ปิด โดยใช้ค่าสนับสนุนน้อยสุดที่กำหนดให้เท่ากับ 3 โดยที่โนดในต้นไม้แสดง ‘เซตไอเท็ม x จำนวนของไอเท็ม’



รูปที่ 2.12 การค้นหาเซตไอเท็มปรากฏบ่อยแบบปิดโดยใช้อัลกอริทึม CHARM

นอกจากนี้แล้ว การค้นหาตามแบบอัลกอริทึม CHARM ยังมีเทคนิคอีกหลายอย่างที่ช่วยให้การค้นหาถูกต้องและรวดเร็วขึ้น ได้แก่

- การเรียงลำดับโนดในระดับชั้นแรกใหม่ตามลำดับจากน้อยไปมากตามจำนวนสมาชิกในเซตของหมายเลขทรานแซคชัน ดังแสดงรูปที่ 2.13 เนื่องจากลำดับการสร้างโนดใหม่เรียงจากซ้ายไปขวา การกระทำเช่นนี้จะทำให้เซตของหมายเลขทรานแซคชันของโนดทางด้านซ้ายมีโอกาสเป็นเซตย่อยของเซตของหมายเลขทรานแซคชันของโนดทางด้านขวามากขึ้น ซึ่งจะทำให้โอกาสในการตัดโนดมีมากขึ้น และโอกาสในการสร้างโนดใหม่มีน้อยลง ซึ่งเห็นได้ชัดจากต้นไม้ข้างบนเมื่อทำการเรียงลำดับโนดในระดับชั้นแรกใหม่จะได้ต้นไม้ดังรูป ซึ่งมีจำนวนโนดน้อยลงเมื่อเทียบกับต้นไม้ในรูปที่ 2.12



รูปที่ 2.13 การค้นหาโดยใช้อัลกอริทึม CHARM เมื่อมีการเรียงลำดับโนดในระดับชั้นแรกใหม่

- ผลลัพธ์ที่ได้จากการค้นหาบางทีอาจจะยังไม่เป็นเซตไอเท็มแบบปิดได้ เช่น CTW ในรูปด้านบน การจะตรวจสอบว่าเซตไอเท็มที่หามาได้เป็นเซตไอเท็มแบบปิดหรือไม่ เพื่อความรวดเร็วจะเก็บเซตไอเท็มที่หามาได้ไว้ในตารางแฮช ซึ่งมีกุญแจแฮชเป็นเศษจากการหารของผลรวมของหมายเลขทรานแซคชันทั้งหมดที่บรรจุเซตไอเท็มไว้ด้วยจำนวนช่องทั้งหมดในตาราง เซตไอเท็มที่หามาได้จะถูกตรวจสอบกับเซตไอเท็มแบบปิดในตารางแฮชช่องของตัวเองว่าเซตไอเท็มที่หามาได้นี้เป็นเซตไอเท็มแบบปิดหรือไม่

แทนที่จะต้องตรวจสอบกับเซตไอเท็มทุกเซตซึ่งเสียเวลาเป็นอย่างมาก และเพื่อให้มีความรวดเร็วยิ่งขึ้น การตรวจสอบในช่องตารางแฮชจะเริ่มจากเซตไอเท็มแบบปิดที่มีสมาชิกมากกว่าไปสู่เซตไอเท็มแบบปิดที่มีสมาชิกน้อยกว่า เพราะเซตไอเท็มแบบปิดที่มีสมาชิกมากกว่ามีโอกาสที่จะมีเซตไอเท็มที่หามาได้เป็นเซตย่อยมากกว่า ซึ่งถ้าพบก่อนจะทำให้ไม่จำเป็นต้องนำไปเปรียบเทียบกับเซตไอเท็มแบบปิดตัวถัดไป

การหากฎความสัมพันธ์จากเซตไอเท็มปรากฏบ่อย

เมื่อได้เซตไอเท็มปรากฏบ่อยมาแล้ว จำเป็นต้องหากฎความสัมพันธ์จากเซตไอเท็มปรากฏบ่อยนั้น โดยกฎความสัมพันธ์ที่ได้จะต้องมีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นน้อยสุดที่กำหนดให้ การหากฎความสัมพันธ์จากเซตไอเท็มปรากฏบ่อยถือเป็นปัญหาการค้นหาในสเปซของการจัดหมู่เช่นกัน โดยกฎความสัมพันธ์ทั้งหมดที่เป็นไปได้คือการจัดหมู่ทุกแบบของสมาชิกในเซตไอเท็มปรากฏบ่อย การจัดหมู่นี้อาจจะเป็นการจัดหมู่ของกฎความสัมพันธ์ส่วนที่เป็นเหตุ หรือการจัดหมู่ของกฎความสัมพันธ์ส่วนที่เป็นผลก็ได้ ในที่นี้ให้กฎความสัมพันธ์อยู่ในรูปแบบ $(F - S) \rightarrow S$ เมื่อ S คือเซตไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นผล และ F คือเซตไอเท็มปรากฏบ่อย ในที่นี้ $S \subset F$ และ $S \neq \phi$ จากรูปแบบของกฎความสัมพันธ์บอกได้ว่าการค้นหาเฉพาะการแจกแจงของกฎความสัมพันธ์ส่วนที่เป็นผล ก็สามารถนำไปสู่กฎความสัมพันธ์ส่วนที่เป็นเหตุ และกฎความสัมพันธ์ได้

การลดขนาดของสเปซการค้นหาสามารถทำได้โดยใช้ทฤษฎีที่ว่า

ถ้า $F - S \rightarrow S$ ไม่ใช่กฎความสัมพันธ์ที่มีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นน้อยสุดที่กำหนดให้ แล้ว $F - \widetilde{S} \rightarrow \widetilde{S}$ ก็จะไม่ใช่เช่นกัน เมื่อ $S \subseteq \widetilde{S}$

ซึ่งทฤษฎีนี้สามารถพิสูจน์ได้ และผลจากทฤษฎีสามารถนำมาใช้สร้างเป็นอัลกอริทึมสำหรับหากฎความสัมพันธ์จากเซตไอเท็มปรากฏบ่อยได้

อัลกอริทึมสำหรับการทำเหมืองข้อมูล

Algorithms for Data Mining

บทที่ 3: การพัฒนา

Chapter 3: Implementation

ต้นไม้ตัดสินใจ

การเรียนรู้ต้นไม้ตัดสินใจ

การพัฒนาโปรแกรมการเรียนรู้ต้นไม้ตัดสินใจในงานวิจัยนี้ได้ยึดตามแบบโปรแกรม C4.5 โดยมีการปรับปรุงแก้ไขในบางส่วนเพื่อให้เข้ากับรูปแบบการแสดงผลของโปรแกรม โครงสร้างข้อมูล และขั้นตอนที่ใช้ในการเรียนรู้เป็นดังนี้

โครงสร้างข้อมูล

โครงสร้างข้อมูลหลักในการเรียนรู้ต้นไม้ตัดสินใจมีอยู่โครงสร้างเดียวคือ โครงสร้างข้อมูลของต้นไม้ ต้นไม้ตัดสินใจจะถูกเก็บลงในโครงสร้างข้อมูลที่ประกอบด้วยโนดหลายๆ โนดที่อ้างถึงกัน ซึ่งแต่ละโนดจะมีข้อมูลดังนี้ โดยในวงเล็บคือชื่อตัวแปรที่ใช้เก็บข้อมูล

- **ชนิดของโนด** (NodeType) เป็นเลขจำนวนเต็มบอกว่าโนดนี้เป็นโนดชนิดใด (เป็นโนดใบ โนดของคุณสมบัติแบบต่อเนื่อง หรือ โนดของคุณสมบัติแบบไม่ต่อเนื่อง)
- **ค่าคุณสมบัติของโนด** (Tested) เก็บค่าคุณสมบัติของโนดนี้
- **หมายเลขกลุ่ม** (Leaf) บอกถึงหมายเลขของกลุ่มที่มีจำนวนมากสุดในข้อมูลที่ตกลงมายังโนดนี้
- **จำนวนข้อมูล** (Items) บอกจำนวนข้อมูลที่ตกลงมายังโนดนี้
- **การกระจายตัวของกลุ่ม** (ClassDist) บอกว่าในโนดนี้มีแต่ละกลุ่มเป็นจำนวนเท่าใด
- **ความผิดพลาด** (Errors) บอกถึงจำนวนข้อมูลที่มีกลุ่มไม่ตรงกับกลุ่มส่วนใหญ่ (ไม่ตรงกับหมายเลขกลุ่ม) ในโนดนี้
- **จำนวนกิ่ง** (Forks) บอกจำนวนกิ่งที่โนดนี้แตกไป ซึ่งเท่ากับค่าคุณสมบัติของคุณสมบัติในโนดนี้
- **ค่าตัด** (Cut) เป็นค่าที่ตัดแบ่งสำหรับโนดที่มีคุณสมบัติแบบต่อเนื่องให้มีค่าคุณสมบัติสองค่า คือค่าที่น้อยกว่าหรือเท่ากับค่าที่ตัด และค่าที่มากกว่าค่าที่ตัด
- **โนดลูก** (Branch) คือตัวชี้ไปถึงโนดที่เป็นลูกของโนดนี้ตามค่าคุณสมบัติต่างๆ
- **ค่าคุณสมบัติที่เรียงลำดับตามค่ามาตรฐานอัตราส่วนเกน** (AttSort) เป็นการเรียงลำดับของทุกคุณสมบัติที่นำมาใช้คัดเลือกเป็นโนดนี้ตามค่ามาตรฐานอัตราส่วนเกนจากน้อยไปมาก
- **ค่ามาตรฐานอัตราส่วนเกนที่เรียงลำดับ** (Worth) เก็บค่ามาตรฐานอัตราส่วนเกนของแต่ละคุณสมบัติที่นำมาใช้คัดเลือกเป็นโนดนี้เรียงลำดับจากน้อยไปมาก

ซึ่งทุกตัวแปรในที่นี้จำเป็นต่อการนำไปใช้ในการแสดงผลลัพธ์ของต้นไม้ตัดสินใจที่ได้จากการเรียนรู้ต่อไป

ขั้นตอนวิธี

การเรียนรู้ของต้นไม้ตัดสินใจประกอบด้วยสองขั้นตอนหลัก คือ

1. การสร้างต้นไม้ตัดสินใจ
2. ตัดเล็มต้นไม้ตัดสินใจที่สร้างได้

การสร้างต้นไม้ตัดสินใจ

สามารถสร้างโดยตรงตามวิธีการเรียนรู้ของต้นไม้ตัดสินใจ คือเริ่มต้นจากการเลือกคุณสมบัติที่มีค่ามาตรฐานอัตราส่วนเกนสูงสุดมาสร้างเป็นโนดรากก่อน จากนั้นจึงแตกข้อมูลไปตามกิ่งต่างๆ ของโนดราก โดยที่แต่ละกิ่ง ถ้าข้อมูลยังไม่เป็นกลุ่มเดียวกัน หรือยังมีจำนวนมากกว่าจำนวนน้อยสุดที่กำหนด ก็จะใช้ข้อมูลเหล่านี้มาหาคุณสมบัติที่ดีที่สุดที่จะมาสร้างเป็นโนดที่เป็นโนดลูกของโนดรากต่อไป และที่โนดนี้ก็จะแตกข้อมูลและวนซ้ำสร้างโนดตามวิธีการเดิมไปเรื่อยๆ จนเมื่อเสร็จสมบูรณ์แล้วจะได้ผลลัพธ์เป็นต้นไม้ตัดสินใจที่ใช้แยกแยะข้อมูลได้

การตัดเล็มต้นไม้ตัดสินใจ

การตัดเล็มต้นไม้ตัดสินใจเป็นกระบวนการที่ทำจากล่างขึ้นบน โดยเริ่มจากการคำนวณจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดของโนดใบแต่ละโนด แล้วส่งผลลัพธ์ไปยังโนดที่อยู่เหนือขึ้นไป โนดที่อยู่เหนือขึ้นไปก็จะคำนวณจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดของตัวเองเปรียบเทียบกับจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดโดยรวมของโนดใบทุกโนดที่เป็นลูกของตัวเอง ผลการเปรียบเทียบแยกได้เป็นสองกรณีคือ

- ถ้าจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดของตัวเองมีค่าน้อยกว่าจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดรวมของทุกโนดใบ ก็จะยุบตัวเองกลายเป็นโนดใบและตัดโนดใบที่เป็นลูกทิ้ง พร้อมส่งจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดของตัวเองไปยังโนดที่อยู่เหนือขึ้นไป
- ถ้าจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดของตัวเองมีค่ามากกว่าจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดรวมของทุกโนดใบ ก็จะคงสภาพเดิม พร้อมส่งจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดรวมของทุกโนดใบไปยังโนดที่อยู่เหนือขึ้นไป

โนดที่อยู่เหนือขึ้นไปก็จะทำซ้ำกระบวนการเดิม คือเปรียบเทียบจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดของตัวเองกับจำนวนข้อมูลที่คาดว่าจะแยกแยะผิดพลาดรวมของทุกโนดลูก ไปจนกระทั่งถึงโนดรากของต้นไม้

การทำจินตทัศน์ต้นไม้ตัดสินใจ

โครงสร้างข้อมูล

โครงสร้างข้อมูลที่ใช้เพื่อทำจินตทัศน์ของต้นไม้ตัดสินใจมีดังนี้

- **โครงสร้างข้อมูลของโนดในต้นไม้** เก็บตำแหน่งของโนดในต้นไม้ และข้อมูลของแต่ละโนดในต้นไม้ตัดสินใจที่เป็นผลลัพธ์จากการเรียนรู้
- **โครงสร้างข้อมูลของกิ่งในต้นไม้** เก็บตำแหน่งของกิ่งในต้นไม้ และข้อมูลของแต่ละกิ่งในต้นไม้ตัดสินใจที่เป็นผลลัพธ์จากการเรียนรู้
- **โครงสร้างข้อมูลของต้นไม้** เก็บข้อมูลสำหรับการหาตำแหน่งโนดเพื่อใช้ในการวาดต้นไม้ และข้อมูลของต้นไม้ตัดสินใจที่เป็นผลลัพธ์จากการเรียนรู้ รวมทั้งเก็บข้อมูลของทุกโนดและทุกกิ่งในต้นไม้ สำหรับข้อมูลสำหรับการหาตำแหน่งของแต่ละโนดมีดังนี้
  - *height* เป็นค่าความสูงของต้นไม้
  - *ltop* เป็นอาร์เรย์เก็บระยะห่างระหว่างโนดซ้ายสุดของระดับชั้นนี้กับรากของต้นไม้
  - *rtop* เป็นอาร์เรย์เก็บระยะห่างระหว่างโนดขวาสุดของระดับชั้นนี้กับรากของต้นไม้
  - *lmax* เก็บระยะห่างระหว่างรากกับโนดที่อยู่ซ้ายสุดของต้นไม้
  - *rmax* เก็บระยะห่างระหว่างรากกับโนดที่อยู่ขวาสุดของต้นไม้
- **กองซ้อน** สำหรับเก็บต้นไม้ย่อยไว้เพื่อนำมาสร้างเป็นต้นไม้ใหญ่ต่อไป

ขั้นตอนวิธี

การทำจินตทัศน์ของต้นไม้ตัดสินใจเริ่มต้นจากการหาตำแหน่งของแต่ละโนดในต้นไม้ ซึ่งมีขั้นตอนดังนี้

1. ท่องต้นไม้ตัดสินใจจากล่างขึ้นบน เมื่อเจอโนดที่เป็นใบจะนำโนดนี้ใส่ลงในกองซ้อนรอการดึงออกเพื่อนำไปสร้างเป็นต้นไม้ใหญ่กว่าต่อไป
2. เมื่อเจอโนดที่ไม่เป็นใบ หรือคุณสมบัติในต้นไม้ตัดสินใจ จะทำการดึงต้นไม้ย่อยออกจากกองซ้อน โดยดึงออกมาเท่ากับจำนวนลูกของโนด และนำต้นไม้ย่อยที่ดึงออกมามาสร้างเป็นต้นไม้ใหม่ที่มีโนดนี้เป็นราก
3. ในการสร้างต้นไม้ใหม่จากต้นไม้เดิมนั้น ทำได้โดยการรวมต้นไม้เดิมเข้าด้วยกัน โดยรักษาระยะห่างระหว่างต้นไม้สองต้นใดๆ ให้เป็นระยะห่างคงที่ระยะหนึ่ง โดยจะดูค่า ltop และ rtop ในทุกระดับชั้นของต้นไม้ย่อยที่อยู่ติดกัน แล้วนำมาคำนวณว่ารากของต้นไม้ย่อยที่จะนำมารวมกันควรอยู่ห่างกันเท่าใดต้นไม้ย่อยจึงจะไม่ชนกัน และรักษาระยะห่างได้พอดี
4. นอกจากนี้แล้ว ยังมีกรณีที่แม้ต้นไม้ย่อยที่ติดกันจะไม่ชนกัน แต่ต้นไม้ย่อยที่ไม่ติดกันแต่ความสูงมากกว่าต้นไม้ย่อยที่อยู่ตรงกลางอาจจะชนกันได้ จึงต้องตรวจสอบกรณีนี้ด้วย
5. เมื่อสร้างเป็นต้นไม้ใหม่แล้ว นำต้นไม้นั้นใส่ลงกองซ้อน และวนทำเช่นนี้ไปเรื่อยๆ จนกระทั่งท่องต้นไม้ครบทุกโนด

นิวรอลเน็ตเวิร์ก

ส่วนการเรียนรู้

ขั้นตอนการเรียนรู้

มีสามขั้นตอนในการเรียนรู้ตัวแยกแยะแบบนิวรอลเน็ตเวิร์กคือ

1. สุ่มค่าน้ำหนักเส้นเชื่อมขึ้นมาค่าน้อยๆ ชุดหนึ่ง พร้อมทั้งสร้างตารางซิกมอยด์
2. ทำการเรียนรู้ (training)
3. ทดสอบตัวแยกแยะ (classifier testing)

การสุ่มค่าน้ำหนักและสร้างตารางซิกมอยด์ (Randomness and Creating Sigmoid Table)

สำหรับการสุ่มค่าน้ำหนักเส้นเชื่อมนั้น กระทำเพื่อหาค่าเริ่มต้นในการปรับน้ำหนักของเส้นเชื่อม ส่วนตารางซิกมอยด์ (sigmoid table) เป็นตารางที่ถูกสร้างเพื่อเก็บค่าจากฟังก์ชันซิกมอยด์ ถูกเรียกใช้เมื่อได้ผลลัพธ์จากการคำนวณของอินพุตที่เข้ามากับน้ำหนักของเส้นเชื่อมรวมกันทุกเส้น ตารางนี้สร้างขึ้นเพื่อเพิ่มความรวดเร็วในการคำนวณค่าฟังก์ชันซิกมอยด์ เนื่องจากฟังก์ชันซิกมอยด์จะถูกเรียกใช้บ่อยครั้งมาก และเป็นฟังก์ชันทางคณิตศาสตร์ที่มีความซับซ้อน เราจึงคำนวณค่าที่เป็นไปได้ไว้ก่อน แล้วเก็บไว้ในตารางซิกมอยด์ และเรียกใช้ได้อย่างรวดเร็วกว่าการคำนวณฟังก์ชันโดยตรง

การเรียนรู้ (Training)

ตัวแยกแยะจะทำการนับจำนวนที่มีของข้อมูลที่นำมาเรียนรู้เก็บไว้ในอาเรย์ของ กลุ่มคุณสมบัติ ค่าคุณสมบัติ รวมทั้งน้ำหนักเส้นเชื่อม

การทดสอบ (Testing)

เป็นการทดสอบเพื่อหาค่าความแม่นยำ และคอนฟิวชันเมตริกซ์ (confusion matrix) ของตัวแยกแยะ โดยจะทำการแยกแยะข้อมูลทดสอบทีละข้อมูล โดยนำกลุ่มที่ได้จากตัวแยกแยะไปตรวจสอบกับกลุ่มของข้อมูลทดสอบ แล้วคำนวณหาค่าความน่าจะเป็นที่จะแยกข้อมูลได้ถูกต้อง

ส่วนการแยกแยะ

วิธีการแยกแยะแบบนิวรอลเน็ตเวิร์กนั้นเป็นวิธีที่ค่อนข้างซับซ้อน ค่าเอาท์พุตของโนดฮิดเดน Ci จากโนดอินพุตหาได้จากสูตร

$$Ci = \oint(\sum_{i=1}^{m} Wi * Ai)$$

โดยที่ $\oint$, m,Wi และ Ai แทนฟังก์ชันซิกมอยด์, จำนวนคุณสมบัติ, น้ำหนักของเส้นเชื่อมระหว่างโนดอินพุตกับโนดฮิดเดน และ ค่าของคุณสมบัติที่แปลงเป็นตัวเลข (ทั้งคุณสมบัติต่อเนื่องและไม่ต่อเนื่อง) ตามลำดับ

หลังจากนั้นค่า Ci ที่ได้จะนำไปเป็นป้อนเข้าสู่โนดเอาท์พุตต่อไป ซึ่งจะเข้าสูตรเหมือนข้างต้น โดยแทน Ci ลงไปใน Ai และคำนวณค่าเอาท์พุต Co ได้ดังนี้

$$Co = \oint(\sum_{i=1}^{n} Wh * Ci)$$

โดยที่ $\oint$, n,Wh และ Ci แทนฟังก์ชันซิกมอยด์, จำนวนโนดฮิดเดน, น้ำหนักของเส้นเชื่อมระหว่างโนดฮิดเดนกับโนดเอาท์พุต และ ค่าเอาท์พุตจากโนด ตามลำดับ

ค่า Co ซึ่งจะมีอยู่ทั้งหมดตามจำนวนกลุ่ม จะถูกนำมาเปรียบเทียบกันเพื่อหาค่าที่มากที่สุด ซึ่งจะนำมาเป็นคำตอบของการแยกแยะ

การเรียนรู้เบย์อย่างง่าย

ส่วนการเรียนรู้

แนวคิดสำคัญ

การแยกแยะด้วยตัวแยกแยะเบย์อย่างง่ายเป็นวิธีการที่ไม่ซับซ้อน ซึ่งอาจไม่จำเป็นจะต้องมีการเรียนรู้ก่อนจะนำไปใช้ แต่เพื่อให้โปรแกรมทำงานได้เร็วที่สุด การเรียนรู้โดยการคำนวณค่าต่างๆ ของข้อมูลที่นำมาเรียนรู้เก็บไว้ก่อนจึงเป็นแนวทางที่ดี วิธีการคือเลือกคำนวณค่าที่จะเป็นต้องใช้บ่อยๆ ในการแยกแยะซึ่งลดการคำนวณซ้ำโดยไม่จำเป็นลง

ขั้นตอนการเรียนรู้โดยสังเขป

มีสามขั้นตอนในการเรียนรู้ตัวแยกแยะเบย์อย่างง่ายคือ

1. ทำให้ข้อมูลเป็นแบบไม่ต่อเนื่อง (discretization)
2. ทำการเรียนรู้ (training)
3. ทดสอบตัวแยกแยะ (classifier testing)

การทำให้ข้อมูลเป็นแบบไม่ต่อเนื่อง (Discretization)

เพื่อให้ผลการแยกแยะมีประสิทธิภาพมากขึ้น จึงต้องทำให้ข้อมูลที่นำมาเรียนรู้แบบต่อเนื่องเป็นแบบไม่ต่อเนื่องก่อนนำไปแยกแยะ มีอยู่ 2 วิธีที่นำมาใช้ในโปรแกรมนี้ คือ การแบ่งเป็นช่วงที่มีความกว้างเท่ากันโดยให้ผู้ใช้สามารถกำหนดจำนวนช่วงได้ (equal-width binning) และการแบ่งโดยให้มีเอนโทรปีน้อยที่สุด (minimal entropy)

การแบ่งเป็นช่วงที่มีความกว้างเท่ากัน ทำการหาค่าที่มากที่สุด และน้อยที่สุด แล้วทำการคำนวณค่าจุดแบ่งจากจำนวนช่วงที่ผู้ใช้กำหนด

การแบ่งโดยให้มีเอนโทรปีน้อยที่สุด คำนวณหาค่าเอนโทรปีของเซต S ที่น้อยที่สุดที่เกิดจากการแบ่งด้วยตัวแบ่ง T_i ของคุณสมบัติ A กำหนดโดย $E(A,T,S)$ คือ

$$E(A,T;S)=\frac{|S_1|}{|S|}Ent(S_1)+\frac{|S_2|}{|S|}Ent(S_2)$$

โดยกำหนดฟังก์ชันเอนโทรปี $Ent(S)=-\sum_{i=1}^{n}P(C_i)\log P(C_i)$

ตัวแบ่ง T_i ที่ทำให้ได้ค่า $E(A,T,S)$ น้อยที่สุดจะถูกเลือกมาเป็นตัวแบ่งของเซต S และได้เป็นเซตย่อย S_1 และ S_2 หลังจากนั้นก็ให้ทำการเรียกแบบเวียนบังเกิดบนเซตย่อย S_1 และ S_2 ไปเรื่อยๆ โดยจะหยุดทำการเรียกแบบเวียนบังเกิดเมื่อ

- ข้อมูลในเซต S ทุกตัวมีกลุ่มเหมือนกัน
- จำนวนข้อมูลในเซต S มีน้อยกว่า 1 เปอร์เซ็นต์ของจำนวนข้อมูลทั้งหมด

การเรียนรู้ (Training)

ตัวเรียนรู้จะทำการนับจำนวนที่มีของข้อมูลที่นำมาเรียนรู้เก็บไว้ในอาเรย์สามมิติของ กลุ่ม คุณสมบัติ และค่าคุณสมบัติ นอกจากนี้ทำจะการคำนวณขนาดของส่วนแบ่งต่าง ๆ ของแผนภูมิเพื่อใช้ในการจินตทัศน์ด้วย

การทดสอบ (Testing)

เป็นการทดสอบเพื่อหาค่าความแม่นยำ และคอนฟิวชันเมตริกซ์ (confusion matrix) ของตัวแยกแยะ โดยจะทำการแยกแยะข้อมูลทดสอบทีละข้อมูล โดยนำกลุ่มที่ได้จากตัวแยกแยะไปตรวจสอบกับกลุ่มของข้อมูลทดสอบ แล้วคำนวณหาค่าความน่าจะเป็นที่จะแยกข้อมูลได้ถูกต้อง

ส่วนการแยกแยะ

วิธีการแยกแยะเบย์อย่างง่ายนั้นเป็นวิธีที่ไม่ซับซ้อน โดยกลุ่มที่น่าจะเป็นไปได้มากที่สุด C หาได้จากสูตร

$$C = \underset{i=1}{\overset{n}{Max}} \frac{\prod_{j=1}^{m} P(A_j|C_i) \times P(C_i)}{P(A_1,...,A_n)}$$

$$= \underset{i=1}{\overset{n}{Max}} \prod_{j=1}^{m} P(A_i|C_i) \times P(C_i)$$

โดยที่ C_i, A_j, m และ n คือกลุ่มที่ i, คุณสมบัติตัวที่ j, จำนวนคุณสมบัติ และจำนวนกลุ่มตามลำดับ

การนำสูตรนี้ไปใช้จริง มีข้อควรคำนึงถึงและวิธีการแก้ไขดังต่อไปนี้

- จากสูตรนี้ผลการแยกแยะจะไม่ถูกต้องถ้าพจน์ใดพจน์หนึ่งเป็นศูนย์ สามารถหลีกเลี่ยงได้โดยทำการปรับเรียบ (smoothing) โดยกำหนดค่า x ค่าหนึ่ง นำไปบวกทุกๆ พจน์ จะได้เป็น

$$= \underset{i=1}{\overset{n}{Max}} \prod_{j=1}^{m} \left(P(A_i|C_i) + x\right) \times P(C_i)$$

- ในกรณีที่ไม่ทราบค่าคุณสมบัติที่นำมาแยกแยะ ก็ไม่ต้องนำคุณสมบัตินั้นมาคิดในสูตร

เนื่องจากผลที่ได้จากสูตรเกิดจากการคูณจำนวนเท่ากับจำนวนคุณสมบัติของข้อมูลที่นำมาแยกแยะ ถ้ามีจำนวนคุณสมบัติมาก ค่าที่ได้จะเล็กเกินกว่าที่ตัวแปรชนิด double จะเก็บได้ เพราะฉะนั้นจึงต้องทำการนอร์มอลไลซ์ (normalize) ค่าที่คำนวณได้ในทุกๆ รอบของการวนลูปหาค่า $\underset{i=1}{\overset{n}{Max}} \prod_{j=1}^{m} P(A_i|C_i)$ โดยให้ค่าที่น้อยที่สุดเป็น 0 และค่าที่มากที่สุดเป็น 1 เพราะฉะนั้นค่าความน่าจะเป็นที่คำนวณได้จึงไม่ใช่ค่าความน่าจะเป็นสัมบูรณ์แต่เป็นค่าความน่าจะเป็นสัมพัทธ์

การค้นหากฎความสัมพันธ์

การค้นหาเซตไอเท็มปรากฏบ่อย

การพัฒนาโปรแกรมการค้นหาเซตไอเท็มปรากฏบ่อยในงานวิจัยนี้ได้ยึดตามอัลกอริทึม CHARM โดยมีโครงสร้างข้อมูลและขั้นตอนวิธีเป็นดังนี้

โครงสร้างข้อมูล

- **โครงสร้างข้อมูลเซต** เป็นโครงสร้างพื้นฐานหน่วยย่อยที่สุดเพื่อใช้สำหรับเก็บเซตไอเท็มและเซตของหมายเลขทรานแซคชันที่บรรจุเซตไอเท็ม โครงสร้างข้อมูลเซตนี้จำเป็นต้องมีตัวดำเนินการยูเนียนและอินเตอร์เซคชัน นอกจากนี้เพื่อความสะดวกควรมีตัวดำเนินการอื่นๆ อีก เช่น ตัวดำเนินการลบ (เพื่อใช้หาว่าเซตที่พิจารณาเป็นเซตย่อยของเซตใดหรือไม่)
- **โครงสร้างต้นไม้ในการค้นหา** ประกอบด้วยโนดหลายๆ โนดที่อ้างถึงกัน ซึ่งแต่ละโนดจะมีข้อมูลดังนี้
  - เซตไอเท็ม
  - เซตของหมายเลขทรานแซคชันที่บรรจุเซตไอเท็มที่อยู่ในโนดนี้
  - ตัวชี้ไปถึงโนดที่เป็นลูก
- **ตารางแฮชและลิงค์ลิสต์** สำหรับเก็บเซตไอเท็มแบบปิดที่หามาได้เพื่อทำการเปรียบเทียบกับเซตไอเท็มที่เข้ามาใหม่ว่าเซตไอเท็มนั้นเป็นเซตไอเท็มแบบปิดหรือไม่ ตารางแฮชนี้กำหนดให้มีขนาดช่องตามความเหมาะสม ในแต่ละช่องจะประกอบด้วยลิสต์ของโนดเรียงต่อกันตามจำนวนสมาชิกของเซตไอเท็มจากมากไปน้อย แต่ละลิสต์ของโนดมีข้อมูลดังนี้
  - เซตไอเท็ม
  - เซตของหมายเลขทรานแซคชันที่บรรจุเซตไอเท็มที่อยู่ในโนดนี้
  - ตัวชี้ไปถึงลิสต์ของโนดตัวถัดไป

ขั้นตอนวิธี

การค้นหาเซตไอเท็มปรากฏบ่อยประกอบด้วยสามขั้นตอนหลัก คือ

1. การไล่และจัดเก็บเซตไอเท็มที่มีจำนวนสมาชิกเท่ากับหนึ่ง และเซตของหมายเลขทรานแซคชันที่สัมพันธ์กับเซตไอเท็มนั้น
2. จัดเรียงลำดับโนดในระดับชั้นแรกใหม่ตามลำดับจากน้อยไปมากตามจำนวนสมาชิกในเซตของหมายเลขทรานแซคชัน
3. ไล่ค้นหาข้อมูลตามแนวลึกก่อนตามแบบอัลกอริทึม CHARM พร้อมเปรียบเทียบและจัดเก็บเซตไอเท็มปรากฏบ่อยแบบปิดที่หามาได้ลงในตารางแฮช

การค้นหากฎความสัมพันธ์จากเซตไอเท็มปรากฏบ่อย

มีโครงสร้างข้อมูลและขั้นตอนวิธีเป็นดังนี้

โครงสร้างข้อมูล

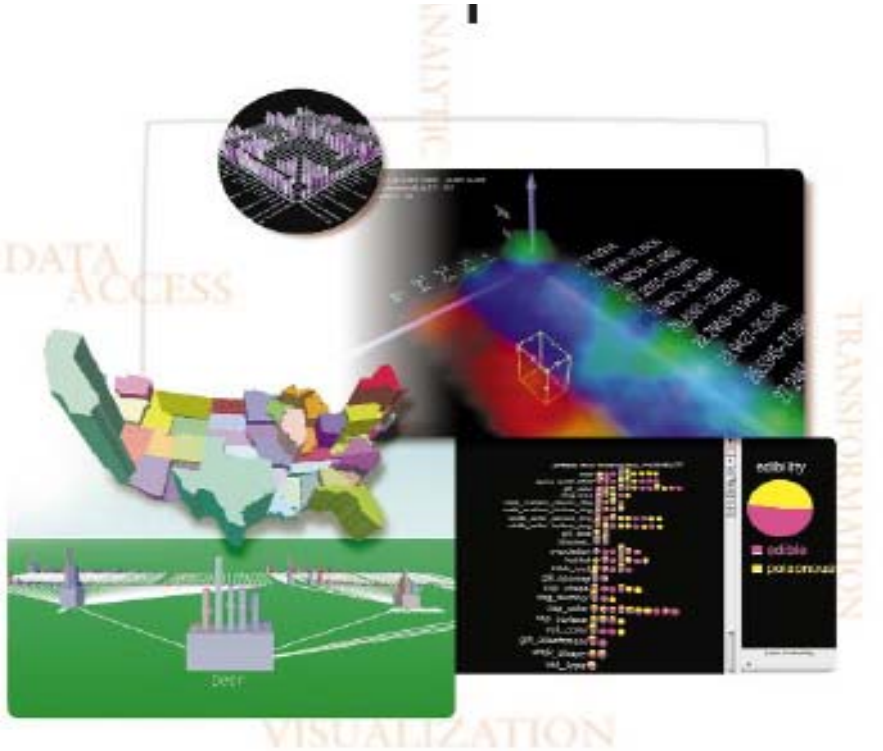
- **โครงสร้างข้อมูลเพื่อเก็บเซตไอเท็มที่ปรากฏบ่อย** เพื่อเก็บเซตไอเท็มปรากฏบ่อยแต่ละเซต ซึ่งประกอบด้วยข้อมูลภายในดังนี้
  - เซตไอเท็มปรากฏบ่อย
  - เซตของหมายเลขทรานแซคชันที่บรรจุเซตไอเท็มปรากฏบ่อย
  - ค่าสนับสนุนของเซตไอเท็มปรากฏบ่อย
  - จำนวนกฎความสัมพันธ์ทั้งหมดของเซตไอเท็มปรากฏบ่อย
  - เซตไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นผลในแต่ละกฎความสัมพันธ์
  - เซตไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นเหตุในแต่ละกฎความสัมพันธ์
  - ค่าความมั่นใจในแต่ละกฎความสัมพันธ์
- **โครงสร้างข้อมูลโดยรวม** ซึ่งจะเก็บข้อมูลโดยรวมทั้งหมดในการค้นหากฎความสัมพันธ์ ประกอบด้วยข้อมูลดังต่อไปนี้
  - อาร์เรย์ของโครงสร้างข้อมูลที่เก็บเซตไอเท็มปรากฏบ่อย
  - จำนวนกฎความสัมพันธ์ทั้งหมด

ขั้นตอนวิธี

ในขั้นตอนนี้จะใช้วิธีการค้นหาเซตไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นผลตามแนวกว้างทีละระดับชั้น พร้อมตัดเซตไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นผลที่ทำให้เกิดกฎความสัมพันธ์ที่มีค่าความมั่นใจน้อยกว่าค่าความมั่นใจที่กำหนดให้ออกไป ไม่นำไปสร้างเป็นเซตไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นผลในระดับชั้นใหม่ และทำการค้นหาทีละระดับชั้นอย่างนี้เรื่อยไปจนกระทั่งเซตไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นผลมีจำนวนสมาชิกเท่ากับสมาชิกของเซตไอเท็มปรากฏบ่อย หรือจนกระทั่งไม่เหลือเซตไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นผลที่จะสร้างในระดับชั้นถัดไป

อัลกอริทึมสำหรับการทำเหมืองข้อมูล
Algorithms for Data Mining

บทที่ 4: การใช้ซอฟต์แวร์
Chapter 4: Using the Software



การติดตั้งโปรแกรม

ความต้องการขั้นต่ำของระบบ

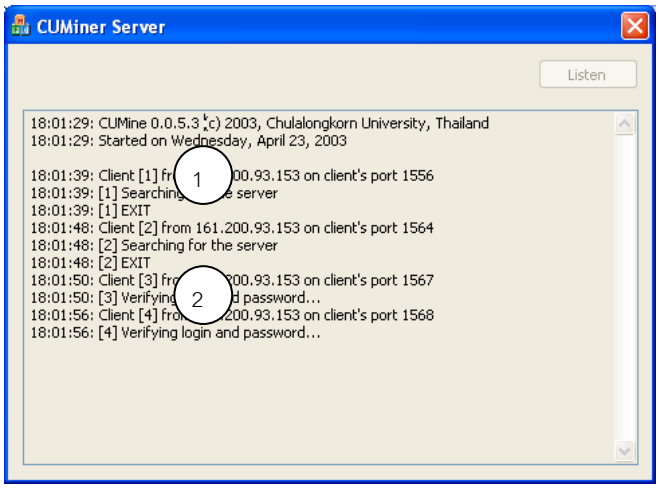
CPU	500 MHz
Ram	64 MB
Hard disk	20MB
Screen Resolution	800 x 600 pixels

การติดตั้งโปรแกรม

สำหรับการติดตั้งโปรแกรม ทำโดยโปรแกรมเซตอัพ ซึ่งจะให้เลือกว่าจะลงโปรแกรมฝั่งไหน เช่น ถ้าเป็นเครื่องทางฝั่งเซิร์ฟเวอร์ ก็จะคลิกเพื่อติดตั้งโปรแกรมทางฝั่งเซิร์ฟเวอร์ ซึ่งจะทำการลงตัวโปรแกรมทางฝั่งเซิร์ฟเวอร์ หรือถ้าเป็นเครื่องทางฝั่งไคลเอนต์ โปรแกรมจะทำการลงโปรแกรมทางฝั่งไคลเอนต์

โปรแกรมทางฝั่งเซิร์ฟเวอร์

หน้าจอแสดงผล



รูปที่ 4.1 หน้าจอแสดงผลของโปรแกรมทางฝั่งเซิร์ฟเวอร์

โปรแกรมทางฝั่งเซิร์ฟเวอร์จะแสดงงานที่ตัวโปรแกรมได้กระทำอยู่ หรือที่ทำผ่านมาในอดีต เปรียบได้กับล็อกไฟล์ (log file) ที่เก็บประวัติการทำงาน

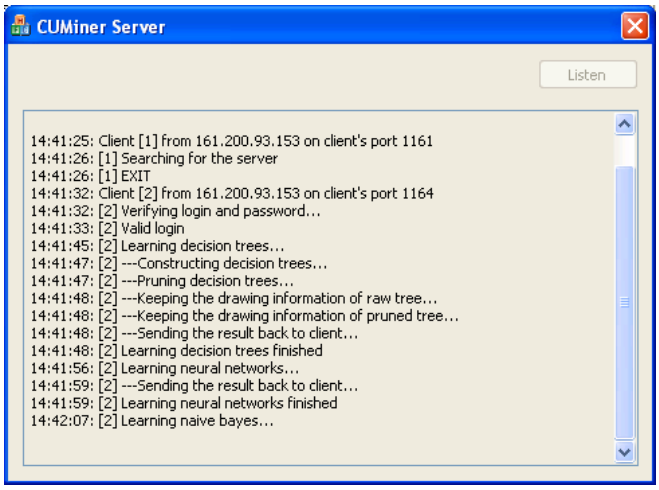
รูปที่ 4.1 แสดงให้เห็นถึงการทำงานของโปรแกรมทางฝั่งเซิร์ฟเวอร์ในการโต้ตอบกับโปรแกรมทางฝั่งไคลเอนต์ ดังนี้

เมื่อโปรแกรมทางฝั่งไคลเอนต์ต้องการหาว่ามีเซิร์ฟเวอร์ตัวไหนบ้างที่กำลังรันโปรแกรมทางฝั่งเซิร์ฟเวอร์ จึงทำการไล่หา เมื่อพบแล้ว โปรแกรมทางฝั่งเซิร์ฟเวอร์จะเก็บข้อมูลให้ทราบว่ามีโปรแกรมทางฝั่งไคลเอนต์จากเครื่องใดทำการติดต่อเข้ามาบ้าง ยกตัวอย่างเช่น ในรูปมีโปรแกรมทางฝั่งไคลเอนต์จากเครื่องที่มีหมายเลขไอพี 161.200.93.153 ทำการติดต่อเข้ามา โดยเครื่องทางฝั่งไคลเอนต์นี้ได้ทำการเปิดโปรแกรมทางฝั่งไคลเอนต์อยู่ทั้งหมดสองโปรแกรม

เมื่อโปรแกรมทางฝั่งไคลเอนต์เลือกเซิร์ฟเวอร์ที่ต้องการทำการเรียนรู้ได้แล้ว โปรแกรมทางฝั่งไคลเอนต์จะส่งชื่อผู้ใช้กับรหัสผ่านเข้ามาที่โปรแกรมทางฝั่งเซิร์ฟเวอร์ ยืนยันการเข้าใช้ของโปรแกรมทางฝั่งไคลเอนต์

เมื่อมีการส่งสัญญาณจากโปรแกรมทางฝั่งไคลเอนต์เพื่อส่งข้อมูลมาเรียนรู้ โปรแกรมทางฝั่งเซิร์ฟเวอร์จะมีการบันทึกและเริ่มกระบวนการในการเรียนรู้ดังรูปที่ 4.2

รูปที่ 4.2 แสดงตัวอย่างในขณะเรียนรู้ ซึ่งจะเห็นได้ว่า โปรแกรมทางฝั่งเซิร์ฟเวอร์แสดงให้เห็นถึงการเรียนรู้ต้นไม้ตัดสินใจซึ่งจะมีรายละเอียดในการทำงาน พร้อมทั้งการเรียนรู้นิวรอลเน็ตเวิร์ก และมีการส่งผลกลับไปให้โปรแกรมทางฝั่งไคลเอนต์เรียบร้อยแล้ว พร้อมทั้งแสดงสถานะว่า ขณะนี้เซิร์ฟเวอร์กำลังทำการเรียนรู้เบย์อย่างง่ายอยู่

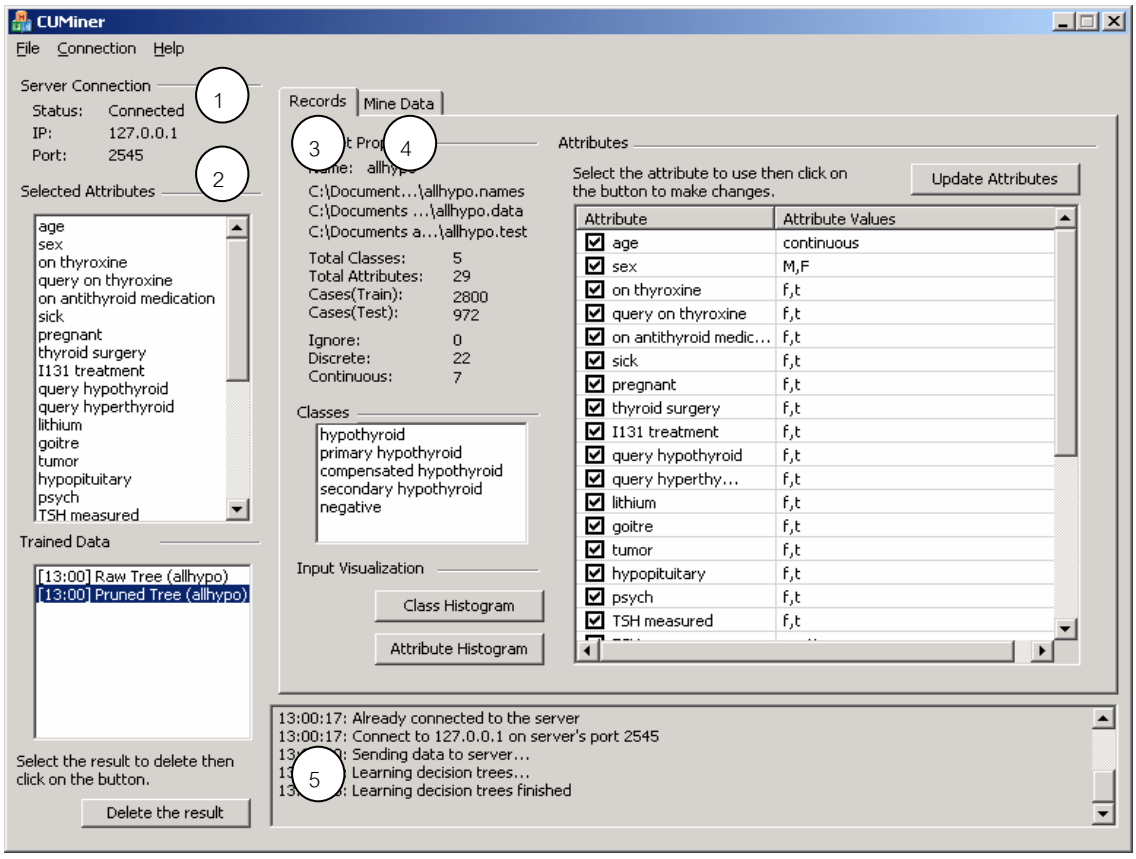


รูปที่ 4.2 แสดงการทำงานของตัวเซิร์ฟเวอร์ขณะทำการเรียนรู้

โปรแกรมทางฝั่งไคลเอนต์

หน้าจอแสดงผล

เมื่อทำการเปิดโปรแกรมทางฝั่งไคลเอนต์ จะปรากฏหน้าจอดังรูปที่ 4.3 ด้านล่างนี้



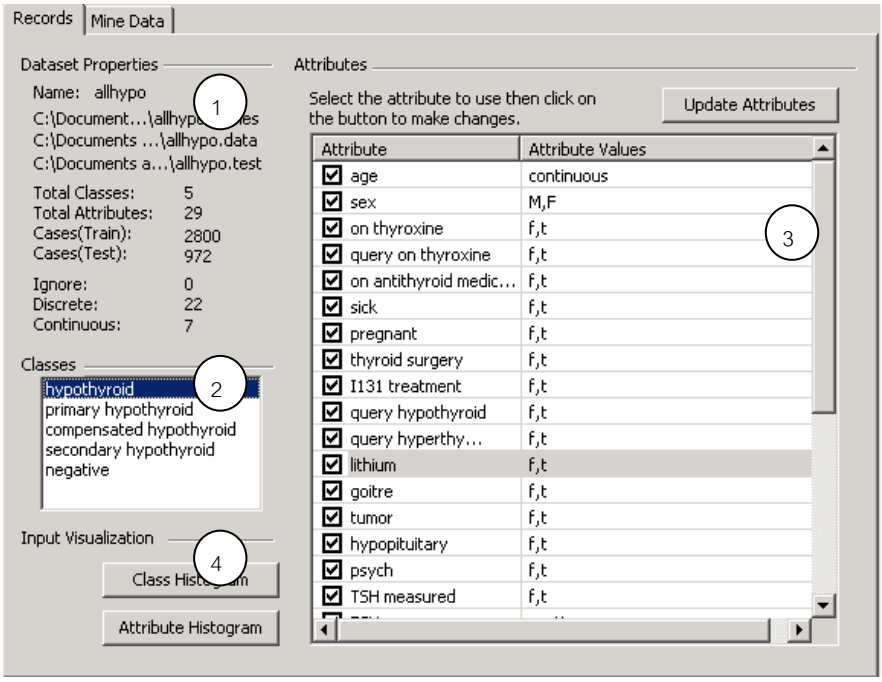
รูปที่ 4.3 ส่วนประกอบต่างๆ ของหน้าจอ

หน้าจอแสดงผลของโปรแกรมทางฝั่งไคลเอนต์ประกอบด้วยส่วนสำคัญดังนี้

1. **รายการ** ประกอบด้วยรายการหลักคือ รายการสำหรับการเปิดและบันทึกไฟล์ รายการสำหรับการเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ และรายการช่วยเหลือ
2. **แถบด้านซ้าย** ส่วนบนสุดของแถบด้านซ้ายแสดงรายละเอียดในการเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ ส่วนถัดลงมาเป็นกล่องรายการแสดงคุณสมบัติทั้งหมดที่ใช้ในการเรียนรู้ และส่วนล่างสุดเป็นกล่องรายการแสดงผลลัพธ์ที่ได้จากการเรียนรู้
3. **แท็บแสดงรายละเอียดในชุดข้อมูล** สำหรับใช้ปรับแต่งและทำจินตทัศน์ของชุดข้อมูล ซึ่งจะกล่าวถึงในหัวข้อถัดไป
4. **แท็บแสดงรายละเอียดในการเรียนรู้** สำหรับทำการเรียนรู้และประมวลผลลัพธ์ที่ได้จากการเรียนรู้ ซึ่งจะกล่าวถึงในหัวข้อถัดไป
5. **บันทึก** บอกรายละเอียดและขั้นตอนในกระบวนการต่างๆ ที่ทำเสร็จสิ้นลงแล้ว

แท็บแสดงรายละเอียดในชุดข้อมูล

แท็บแสดงรายละเอียดในชุดข้อมูลมีหัวแท็บชื่อ Records ดังรูปที่ 4.4 ซึ่งเป็นแท็บแรกในจำนวนสองแท็บของหน้าจอไคลเอนต์ มีรายละเอียดดังนี้



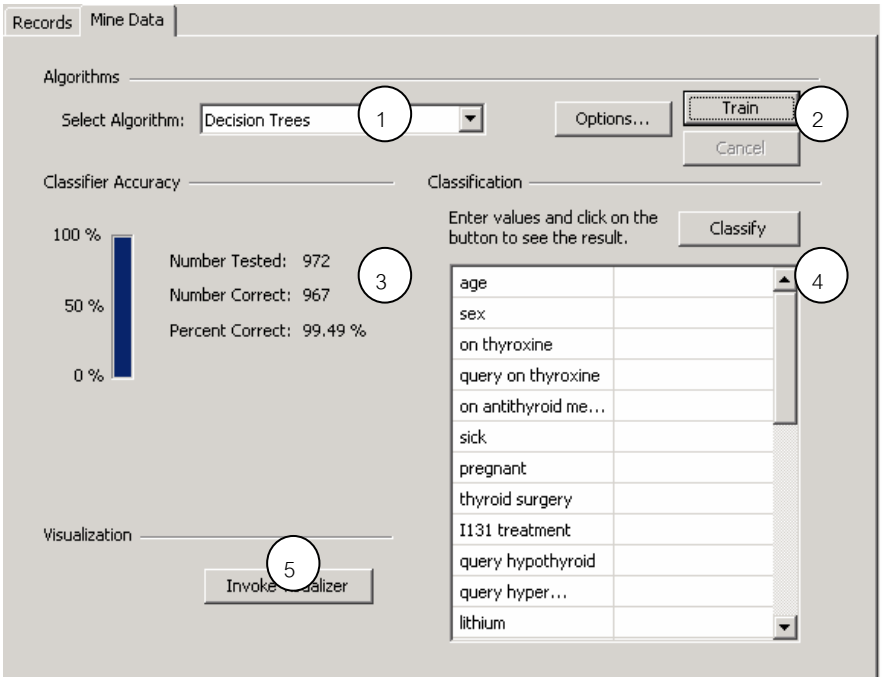
รูปที่ 4.4 แท็บแสดงรายละเอียดในชุดข้อมูล

แท็บแสดงรายละเอียดในชุดข้อมูลประกอบด้วยส่วนสำคัญดังนี้

1. **ส่วนแสดงรายละเอียดของชุดข้อมูลที่ใช้ในการเรียนรู้** แสดงชื่อชุดข้อมูลที่ใช้ในการเรียนรู้ ชื่อไฟล์ที่ใช้ในการเรียนรู้ รวมถึงจำนวนกลุ่มและจำนวนคุณสมบัติในชุดข้อมูล
2. **กล่องรายการแสดงกลุ่ม** แสดงกลุ่มที่มีอยู่ในชุดข้อมูลที่ใช้ในการเรียนรู้
3. **ส่วนการปรับแต่งชุดข้อมูลที่ใช้ในการเรียนรู้** สำหรับเลือกค่าคุณสมบัติที่ต้องการเพื่อจะนำมาทำการเรียนรู้
4. **ส่วนจินตทัศน์ชุดข้อมูลที่ใช้ในการเรียนรู้** สำหรับทำจินตทัศน์ของชุดข้อมูลที่ใช้ในการเรียนรู้

แท็บแสดงรายละเอียดในการเรียนรู้

แท็บแสดงรายละเอียดในการเรียนรู้มีหัวแท็บชื่อ Mine Data ดังรูปที่ 4.5 ซึ่งเป็นแท็บที่สองในจำนวนสองแท็บของหน้าจอไคลเอนต์ มีรายละเอียดดังนี้



รูปที่ 4.5 แท็บแสดงรายละเอียดในการเรียนรู้

แท็บแสดงรายละเอียดในการเรียนรู้ประกอบด้วยส่วนสำคัญดังนี้

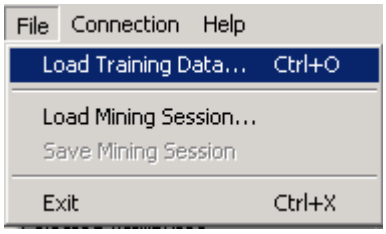
1. **กล่องคอมโบอัลกอริทึม** แสดงชื่ออัลกอริทึมสำหรับเลือกใช้ในการเรียนรู้
2. **ปุ่มที่ใช้ในการเรียนรู้** ประกอบด้วยปุ่มสำหรับกำหนดค่าพารามิเตอร์ที่ใช้ในการเรียนรู้ (Options…) ปุ่มสำหรับการเรียนรู้ (Train) และปุ่มสำหรับยกเลิกการเรียนรู้ (Cancel)
3. **ส่วนแสดงผลลัพธ์ในการแยกแยะข้อมูลที่ใช้ทดสอบ** จะแสดงผลลัพธ์ในการแยกแยะชุดข้อมูลที่ใช้ทดสอบ ว่าผลลัพธ์จากการเรียนรู้ที่ได้ให้ความถูกต้องเท่าใดในการแยกแยะชุดข้อมูลที่ใช้ในการทดสอบ
4. **ส่วนแยกแยะข้อมูล** ประกอบด้วยตารางสำหรับกรอกค่าคุณสมบัติของคุณสมบัติต่างๆ ในข้อมูล และส่วนแสดงผลการแยกแยะว่าข้อมูลตัวนี้น่าจะเป็นกลุ่มใด
5. **ส่วนจินตทัศน์ผลลัพธ์ที่ได้จากการเรียนรู้** สำหรับทำจินตทัศน์ของผลลัพธ์ที่ได้จากการเรียนรู้

กระบวนการเรียนรู้

การรับข้อมูลที่ใช้ในการเรียนรู้

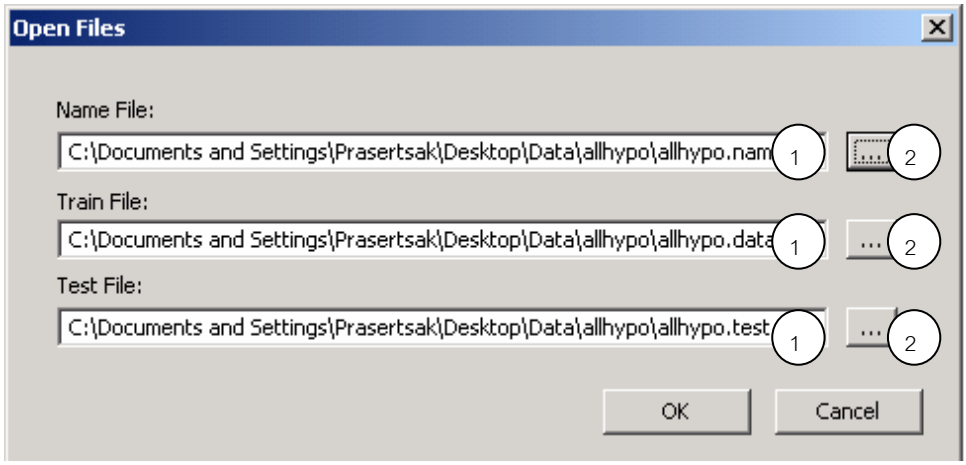
รูปแบบของไฟล์ที่ใช้ในการเรียนรู้จะกล่าวถึงในหัวข้อการเตรียมข้อมูล โดยการรับข้อมูลที่ใช้ในการเรียนรู้มีขั้นตอนดังนี้

- เลือกรายการ File->Load Training Data… (ดูรูปที่ 4.6 ประกอบ)



รูปที่ 4.6 รายการการรับข้อมูลที่ใช้ในการเรียนรู้

- จะปรากฏกล่องโต้ตอบเพื่อให้ใส่ชื่อของ ไฟล์ชื่อ ไฟล์ข้อมูลที่ใช้ในการเรียนรู้ และไฟล์ข้อมูลที่ใช้ในการทดสอบ อาจใส่ชื่อของไฟล์ที่กล่องข้อความ หรือกดปุ่ม … เพื่อเลือกไฟล์ที่ต้องการดังรูปที่ 4.7



รูปที่ 4.7 กล่องโต้ตอบเพื่อรับข้อมูลที่ใช้ในการเรียนรู้

- เมื่อกดปุ่ม OK ที่กล่องโต้ตอบ โปรแกรมจะอ่านข้อมูลเหล่านั้นเข้ามา ซึ่งอาจใช้เวลานานถ้าหากไฟล์ข้อมูลมีขนาดใหญ่ เมื่ออ่านข้อมูลการเรียนรู้เสร็จสิ้น รายละเอียดของข้อมูลที่ใช้ในการเรียนรู้จะถูกแสดงผลที่แถบด้านซ้ายและแท็บแสดงรายละเอียดในชุดข้อมูลในหน้าจอการแสดงผลของโปรแกรมทางฝั่งไคลเอนต์ ข้อมูลที่อ่านเข้ามาสามารถดูรายละเอียดและปรับแต่งได้

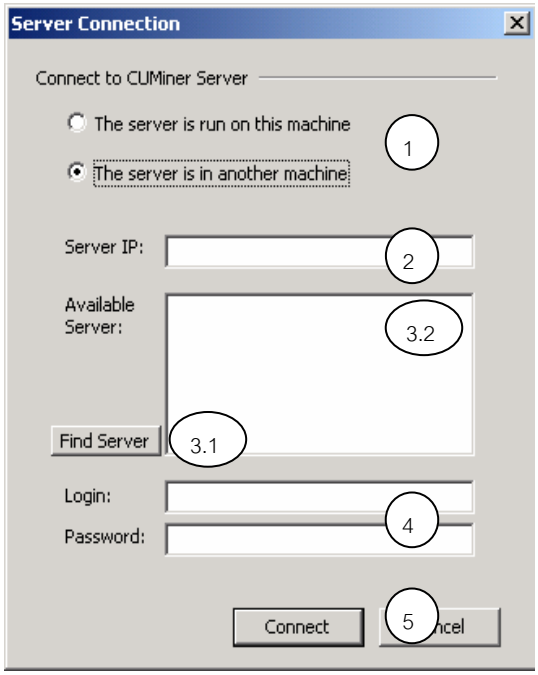
การเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์

เนื่องจากโปรแกรมทางฝั่งไคลเอนต์ไม่มีความสามารถในการเรียนรู้ ก่อนที่จะทำการเรียนรู้จึงต้องเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ก่อน ซึ่งมีขั้นตอนดังนี้

- เลือกรายการ Connection->Connect… (ดูรูปที่ 4.8 ประกอบ)

รูปที่ 4.8 รายการการเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์

- จะปรากฏกล่องโต้ตอบเพื่อเลือกการเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ ดังรูปที่ 4.9 โดยผู้ใช้สามารถเลือกได้ว่าจะเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ที่รันอยู่ในเครื่องเดียวกัน หรือรันอยู่ที่เครื่องอื่น (1) ในกรณีที่โปรแกรมทางฝั่งเซิร์ฟเวอร์รันอยู่ที่เครื่องอื่น ผู้ใช้สามารถเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์โดยใส่หมายเลขไอพีของเครื่องคอมพิวเตอร์ที่จะเชื่อมต่อลงในกล่องข้อความ (2) หรืออาจจะทำการหาเครื่องที่อยู่ในเครือข่ายเดียวกันที่รันโปรแกรมทางฝั่งเซิร์ฟเวอร์อยู่โดยกดปุ่ม Find Server (3.1) หมายเลขไอพีของเครื่องในเครือข่ายที่รันโปรแกรมทางฝั่งเซิร์ฟเวอร์อยู่จะปรากฏที่กล่องรายการ (3.2) ในการเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ที่รันอยู่ในเครื่องอื่น ผู้ใช้ต้องใส่หมายเลขผู้ใช้และรหัสผ่าน (4) เพื่อให้ได้สิทธิในการเข้าไปใช้โปรแกรมทางฝั่งเซิร์ฟเวอร์ที่รันอยู่ในเครื่องนั้น



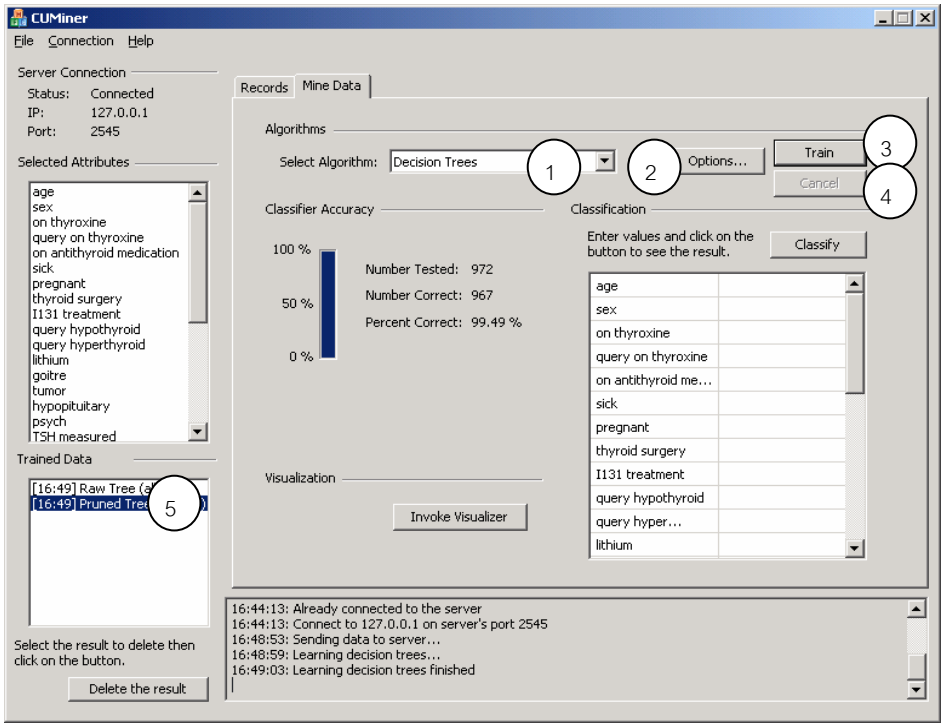
รูปที่ 4.9 กล่องโต้ตอบเพื่อเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์

- เมื่อกดปุ่ม Connect โปรแกรมจะทำการเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์ เมื่อการเชื่อมต่อเสร็จสิ้น รายละเอียดของการเชื่อมต่อจะถูกแสดงผลที่แถบด้านซ้ายส่วนบนในหน้าจอการแสดงผลของโปรแกรมทางฝั่งไคลเอนต์

การเรียนรู้

เมื่อผู้ใช้รับข้อมูลที่ใช้ในการเรียนรู้เข้ามา และทำการเชื่อมต่อกับโปรแกรมทางฝั่งเซิร์ฟเวอร์เรียบร้อยแล้ว ผู้ใช้จะสามารถทำการเรียนรู้ชุดข้อมูลที่รับเข้ามาได้ ขั้นตอนการเรียนรู้ของโปรแกรมแสดงในรูปที่ 4.10 โดยมีขั้นตอนดังนี้

- เลือกแท็บแสดงรายละเอียดการเรียนรู้
- ที่กล่องคอมโบ Select Algorithm: เลือกอัลกอริทึมที่จะใช้ในการเรียนรู้ (1) แต่ละอัลกอริทึมมีวิธีการเรียนรู้และผลลัพธ์ต่างกันไป
- กดปุ่ม Options... เพื่อกำหนดพารามิเตอร์ที่ใช้ในการเรียนรู้ ซึ่งต่างกันไปตามแต่ละอัลกอริทึม (2)
- กดปุ่ม Train เพื่อทำการเรียนรู้ (3)
- ขณะที่ทำการเรียนรู้ ถ้าต้องการยกเลิกการเรียนรู้นั้น กดปุ่ม Cancel (4)
- เมื่อการเรียนรู้เสร็จสิ้น จะได้ผลลัพธ์จากการเรียนรู้มาปรากฏที่แถบด้านซ้ายส่วนล่างสุด (5) ซึ่งสามารถใช้ผลลัพธ์นี้ในการแยกแยะข้อมูล หรือนำมาทำจินตทัศน์ต่อไป



รูปที่ 4.10 การเรียนรู้

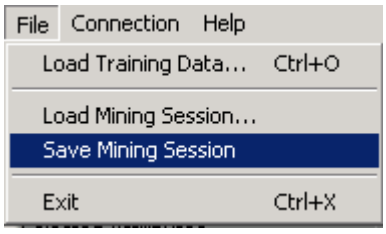
การบันทึกและการเปิดไฟล์ข้อมูลการทำเหมือง

ในการทำเหมืองข้อมูลครั้งหนึ่งๆ จะประกอบไปด้วยข้อมูลที่เกี่ยวข้องดังนี้

- ชุดข้อมูลที่ใช้ในการเรียนรู้
- ผลลัพธ์จากการเรียนรู้ของข้อมูล

ซึ่งข้อมูลที่เกี่ยวข้องทั้งหมดสามารถบันทึกลงในไฟล์นามสกุล .msf โดยมีขั้นตอนดังนี้

- เลือกรายการ File->Save Mining Session (ดูรูปที่ 4.11 ประกอบ)

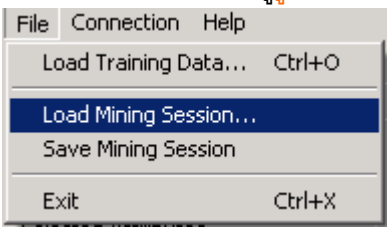


รูปที่ 4.11 รายการการบันทึกไฟล์

- พิมพ์ชื่อไฟล์ และกดปุ่ม Save

การเปิดไฟล์นามสกุล .msf เพื่อดึงข้อมูลสามารถทำได้ดังนี้

- เลือกรายการ File->Load Mining Session... (ดูรูปที่ 4.12 ประกอบ)



รูปที่ 4.12 รายการการเปิดไฟล์

- เลือกไฟล์ และกดปุ่ม Open

การเตรียมข้อมูล

ลักษณะข้อมูลที่ใช้ในการเรียนรู้

ข้อมูลที่ใช้ในการเรียนรู้แต่ละตัวจะประกอบไปด้วยกลุ่มที่ข้อมูลตัวนั้นสังกัดอยู่ และค่าของคุณสมบัติต่างๆ ในข้อมูล ข้อมูลทุกตัวที่ใช้ในการเรียนรู้จะมีเซตของคุณสมบัติเหมือนๆ กัน โดยคุณสมบัติเหล่านี้อาจมีค่าคุณสมบัติเป็นได้ทั้งแบบต่อเนื่องและไม่ต่อเนื่องอย่างใดอย่างหนึ่ง ข้อมูลตัวหนึ่งๆ สามารถสังกัดกลุ่มได้เพียงหนึ่งกลุ่ม และมีค่าของคุณสมบัติใดๆ ได้เพียงค่าเดียว

ตัวอย่างเช่น ข้อมูลการวินิจฉัยโรคคอพอกจากสถาบันวิจัยการ์วานในซิดนีย์ ได้เก็บข้อมูลของคนหลายๆ คน โดยที่แต่ละคนนั้นมีอาการหนึ่งในห้าอาการนี้ คือ ไม่เป็นคอพอก (negative), เป็นคอพอกแบบปฐมภูมิ (primary hypothyroid), เป็นคอพอกแบบทุติยภูมิ (secondary hypothyroid) และเป็นคอพอกแบบคอมเพนเสท (compensated hypothyroid) อาการเหล่านี้จัดเป็นกลุ่มของข้อมูล นอกจากนี้ผู้ป่วยแต่ละคนยังมีค่าคุณสมบัติของคุณสมบัติต่างๆ แตกต่างกันไป คุณสมบัติสำหรับแต่ละคนได้แก่ อายุ (age), เพศ (sex) เป็นต้น ตารางที่ 4.1 ด้านล่างแสดงถึงการเก็บข้อมูลของคนสี่คนที่มีกลุ่มและค่าคุณสมบัติต่างๆ เพื่อใช้ในการเรียนรู้

ตารางที่ 4.1 ตัวอย่างการเก็บข้อมูลที่ใช้ในการเรียนรู้

คุณสมบัติ	Case1	Case2	Case3	Case4
age	41	23	46	70
sex	F	F	M	F
on thyroxine	f	f	f	t
query on thyroxine	f	f	f	f
on antithyroid medication	f	f	f	f
sick	f	f	f	f
pregnant	f	f	f	f
thyroid surgery	f	f	f	f
I131 treatment	f	f	f	f
query hypothyroid	f	f	f	f
query hyperthyroid	f	f	f	f
lithium	f	f	f	f
goitre	f	f	f	f
tumor	f	f	f	f
hypopituitary	f	f	f	f
psych	f	f	f	f
TSH measured	t	t	t	t
TSH	1.3	4.1	0.98	0.16
T3 measured	t	t	f	t
T3	2.5	2	?	1.9

TT4 measured	t	t	t	t
TT4	125	102	109	175
T4U measured	t	f	t	f
T4U	1.14	?	0.91	?
FTI measured	t	f	t	f
FTI	109	?	120	?
TBG measured	f	f	f	f
TBG	?	?	?	?
referral source	SVHC	other	other	other
diagnosis	negative	negative	negative	negative

รูปแบบของไฟล์ข้อมูลที่ใช้ในการเรียนรู้

ข้อมูลแต่ละชุดที่ใช้ในการเรียนรู้จะมีชื่อชุดข้อมูล อย่างเช่นในกรณีข้างต้นชุดข้อมูลมีชื่อว่า hypothyroid ไฟล์ต่างๆ ที่ใช้ในการเรียนรู้จะมีชื่ออยู่ในรูปแบบ ชื่อชุดข้อมูล.นามสกุล ชื่อชุดข้อมูลที่เป็นชื่อไฟล์จะบ่งบอกถึงงานที่ทำ ขณะที่นามสกุลของไฟล์บ่งบอกถึงชนิดของไฟล์และเนื้อหาของข้อมูลที่อยู่ภายใน โดยในการเรียนรู้ของโปรแกรม ได้แบ่งชนิดของไฟล์ออกเป็นสามประเภท คือ ไฟล์ชื่อ ไฟล์ข้อมูลการเรียนรู้ และไฟล์ข้อมูลการทดสอบ

ไฟล์ชื่อ

เป็นไฟล์ที่ใช้สำหรับประกาศชื่อกลุ่ม คุณสมบัติ และค่าคุณสมบัติที่เป็นไปได้ของข้อมูล ไฟล์ชื่อมีลักษณะเป็นเท็กซ์ไฟล์ มีนามสกุล names เช่น hypothyroid.names เป็นต้น ไฟล์ชื่อจะประกอบด้วยส่วนต่างๆ ดังนี้

- **ส่วนประกาศชื่อกลุ่ม** จะอยู่ที่บรรทัดแรกสุดของไฟล์ บอกว่ากลุ่มต่างๆ ที่เป็นได้มีชื่ออะไรบ้าง
- **ส่วนประกาศชื่อคุณสมบัติและชื่อค่าคุณสมบัติ** ชื่อของแต่ละคุณสมบัติและชื่อของค่าคุณสมบัติที่เป็นได้ของแต่ละคุณสมบัตินั้นจะอยู่ที่แต่ละบรรทัดของไฟล์

ตัวอย่างรูปแบบข้อมูลในไฟล์ชื่อเป็นดังตารางที่ 4.2 ต่อไปนี้ (จาก hypothyroid.names)

ตารางที่ 4.2 รูปแบบข้อมูลในไฟล์ชื่อ

hypothyroid, primary hypothyroid, compensated hypothyroid, secondary hypothyroid, negative	
age:	continuous
sex:	M, F
on thyroxine:	f, t
query on thyroxine:	f, t
on antithyroid medication:	f, t
sick:	f, t
pregnant:	f, t
thyroid surgery:	f, t
I131 treatment:	f, t
query hypothyroid:	f, t
query hyperthyroid:	f, t
lithium:	f, t
goitre:	f, t
tumor:	f, t
hypopituitary:	f, t
psych:	f, t
TSH measured:	f, t
TSH:	continuous
T3 measured:	f, t
T3:	continuous
TT4 measured:	f, t
TT4:	continuous
T4U measured:	f, t
T4U:	continuous
FTI measured:	f, t
FTI:	continuous
TBG measured:	f, t
TBG:	continuous
referral source:	WEST, STMW, SVHC, SVI, SVHD, other

ข้อกำหนดในการเขียนไฟล์ชื่อ

เพื่อที่โปรแกรมสามารถอ่านไฟล์ชื่อได้ถูกต้อง การเขียนไฟล์ชื่อจำเป็นต้องเป็นไปตามข้อกำหนดเหล่านี้

- เขียนชื่อกลุ่มไว้ที่บรรทัดบนสุดโดยที่ชื่อกลุ่มแต่ละชื่อต้องคั่นด้วยเครื่องหมายจุลภาค ',' เมื่อครบชื่อกลุ่มทุกชื่อแล้ว ควรเว้นบรรทัดเพื่อเขียนชื่อคุณสมบัติต่อไป
- ชื่อคุณสมบัติต้องตามด้วยเครื่องหมาย ':' ก่อนที่จะเขียนชื่อค่าคุณสมบัติที่เป็นไปได้ของคุณสมบัตินั้น
- ถ้ามีค่าคุณสมบัติหลายค่า แต่ละค่าต้องคั่นด้วยเครื่องหมายจุลภาค ','
- ก่อนจะถึงคุณสมบัติใหม่ควรทำการเว้นบรรทัด
- ภายในชื่อกลุ่ม ชื่อคุณสมบัติ และชื่อค่าคุณสมบัติ สามารถมีช่องว่างได้ แต่ถ้ามีช่องว่างติดกัน ช่องว่างนั้นจะถูกยุบเหลือช่องว่างเดียว
- สามารถสร้างหมายเหตุโดยใช้เครื่องหมาย '|' ซึ่งจะทำให้ตัวอักษรทั้งหมดหลังเครื่องหมาย '|' กลายเป็นหมายเหตุและไม่ถูกอ่านเข้ามาจนกระทั่งจบบรรทัด
- ตัวอักษรพิเศษ อันได้แก่ '|', ',' และ ':' ถ้าอยู่ในชื่อของกลุ่ม ชื่อคุณสมบัติ หรือชื่อค่าคุณสมบัติ จำเป็นต้องนำหน้าด้วยเครื่องหมาย '\' เพื่อให้สามารถถูกอ่านเข้ามาได้ตามปกติ
- เครื่องหมาย '.' สามารถถูกอ่านเข้ามาได้ถ้าตามหลังด้วยตัวอักษรอื่นๆ ที่ไม่ใช่ช่องว่าง และเครื่องหมาย '|' ซึ่งถ้าตามด้วยอักษรสองตัวนี้ จะทำให้ '.' ไม่ถูกอ่านเข้ามา และตัวอักษรที่อยู่หลัง '.' ทั้งหมดจะไม่ถูกอ่านเข้ามาด้วยเช่นกัน

ชนิดของค่าคุณสมบัติในไฟล์ชื่อ

แต่ละคุณสมบัติมีค่าคุณสมบัติซึ่งมีรูปแบบการเขียนในไฟล์ชื่อดังนี้

- **ค่าคุณสมบัติแบบต่อเนื่อง** สามารถบ่งชี้ว่าคุณสมบัตินี้มีค่าคุณสมบัติแบบต่อเนื่องโดยใช้คำว่า continuous เช่น

  FTI: continuous

- **ค่าคุณสมบัติแบบไม่ต่อเนื่องที่รู้ค่า** สามารถเขียนค่าคุณสมบัติต่างๆ ได้โดยใช้เครื่องหมายจุลภาคคั่น เช่น

  referral source: WEST, STMW, SVHC, SVI, SVHD, other

- **ค่าคุณสมบัติแบบไม่ต่อเนื่องที่ยังไม่รู้ค่า** สามารถเขียนโดยใช้คำว่า discrete พร้อมต่อท้ายด้วยจำนวนค่าคุณสมบัติมากสุดที่เป็นไปได้ของคุณสมบัตินี้ เช่น

  protocol type: discrete20

ไฟล์ข้อมูลการเรียนรู้

เป็นไฟล์ที่ใช้บรรจุข้อมูลที่ใช้ในการเรียนรู้ มีนามสกุล data เช่น hypothyroid.data เป็นต้น ข้อกำหนดสำหรับไฟล์ข้อมูลการเรียนรู้มีดังนี้

- ในหนึ่งบรรทัดคือข้อมูลหนึ่งชุด ซึ่งประกอบด้วยค่าคุณสมบัติของคุณสมบัติต่างๆ เรียงตามลำดับเดียวกันกับคุณสมบัติในไฟล์ชื่อ ค่าคุณสมบัติของคุณสมบัติที่ต่างกันจะถูกคั่นด้วยเครื่องหมายจุลภาค ','

- ท้ายสุดของบรรทัดคือชื่อกลุ่มของข้อมูล
- สามารถสร้างหมายเหตุโดยใช้เครื่องหมาย '|' ซึ่งจะทำให้ตัวอักษรทั้งหมดหลังเครื่องหมาย '|' กลายเป็นหมายเหตุและไม่ถูกอ่านเข้ามาจนกระทั่งจบบรรทัด
- ตัวอักษรพิเศษ อันได้แก่ '|', ',' และ ':' ถ้าอยู่ในชื่อของค่าคุณสมบัติ จำเป็นต้องนำหน้าด้วยเครื่องหมาย '\' เพื่อให้สามารถถูกอ่านเข้ามาได้ตามปกติ
- ใช้เครื่องหมาย '?' สำหรับค่าของคุณสมบัติที่ไม่รู้ว่าค่าเป็นเท่าไร

ตัวอย่างรูปแบบข้อมูลในไฟล์ข้อมูลการเรียนรู้เป็นดังตารางที่ 4.3 ด้านล่างนี้ (จาก hypothyroid.data)

ตารางที่ 4.3 ตัวอย่างไฟล์ข้อมูลการเรียนรู้

```
41,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,t,1.3,t,2.5,t,125,t,1.14,t,109,f,?,SVHC,negative.
23,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,t,4.1,t,2,t,102,f,?,f,?,f,?,other,negative.
46,M,f,f,f,f,f,f,f,f,f,f,f,f,f,f,t,0.98,f,?,t,109,t,0.91,t,120,f,?,other,negative.
70,F,t,f,f,f,f,f,f,f,f,f,f,f,f,f,t,0.16,t,1.9,t,175,f,?,f,?,f,?,other,negative.
70,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,t,0.72,t,1.2,t,61,t,0.87,t,70,f,?,SVI,negative.
18,F,t,f,f,f,f,f,f,f,f,f,f,f,f,f,t,0.03,f,?,t,183,t,1.3,t,141,f,?,other,negative.
59,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,?,f,?,t,72,t,0.92,t,78,f,?,other,negative.
80,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,t,2.2,t,0.6,t,80,t,0.7,t,115,f,?,SVI,negative.
66,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,t,0.6,t,2.2,t,123,t,0.93,t,132,f,?,SVI,negative.
68,M,f,f,f,f,f,f,f,f,f,f,f,f,f,f,t,2.4,t,1.6,t,83,t,0.89,t,93,f,?,SVI,negative.
84,F,f,f,f,f,f,f,f,f,f,f,f,f,f,f,t,1.1,t,2.2,t,115,t,0.95,t,121,f,?,SVI,negative.
67,F,t,f,f,f,f,f,f,f,f,f,f,f,f,f,t,0.03,f,?,t,152,t,0.99,t,153,f,?,other,negative.
```

ไฟล์ข้อมูลที่ใช้ในการเรียนรู้สามารถสร้างได้จากโปรแกรม Microsoft Excel (ดังเช่นในรูปที่ 4.13) แล้วเลือกการบันทึกให้อยู่ในรูปแบบไฟล์ .csv จากนั้นเมื่อทำการเปลี่ยนนามสกุลเป็น .data จะทำให้ได้รูปแบบไฟล์ตามที่ต้องการ

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE
1	41	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.3	t	2.5	t	125	t	1.1	t	109	f	?	SVHC	negative.	
2	23	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	4.1	t	2	t	102	f	?	f	?	f	?	other	negative.	
3	46	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1	f	?	t	109	t	0.9	t	120	f	?	other	negative.	
4	70	F	t	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.2	t	1.9	t	175	f	?	f	?	f	?	other	negative.	
5	70	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.7	t	1.2	t	61	t	0.9	t	70	f	?	SVI	negative.	
6	18	F	t	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0	f	?	t	183	t	1.3	t	141	f	?	other	negative.	
7	59	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	?	f	?	t	72	t	0.9	t	78	f	?	other	negative.	
8	80	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	2.2	t	0.6	t	80	t	0.7	t	115	f	?	SVI	negative.	
9	66	F	f	f	f	f	f	f	f	f	f	f	f	t	f	f	t	0.6	t	2.2	t	123	t	0.9	t	132	f	?	SVI	negative.	
10	68	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	2.4	t	1.6	t	83	t	0.9	t	93	f	?	SVI	negative.	
11	84	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.1	t	2.2	t	115	t	1	t	121	f	?	SVI	negative.	
12	67	F	t	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0	f	?	t	152	t	1	t	153	f	?	other	negative.	
13	71	F	f	f	f	t	f	f	f	f	t	f	f	f	f	f	t	0	t	3.8	t	171	t	1.1	t	151	f	?	other	negative.	
14	59	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	2.8	t	1.7	t	97	t	0.9	t	107	f	?	SVI	negative.	
15	28	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	3.3	t	1.8	t	109	t	0.9	t	119	f	?	SVHC	negative.	
16	44	M	f	f	f	t	f	f	f	f	f	f	f	f	f	f	t	2	t	1.3	t	136	t	0.9	t	145	f	?	SVHD	negative.	
17	65	F	f	f	f	f	f	f	f	t	f	f	f	f	f	f	t	12	f	?	t	99	t	1.1	t	87	f	?	other	isated hypothyroid	
18	42	?	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.2	t	1.8	t	70	t	0.9	t	81	f	?	other	negative.	
19	63	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.5	t	1.2	t	117	t	1	t	121	f	?	SVI	negative.	
20	80	F	f	f	f	f	f	f	f	f	f	f	f	f	f	t	t	6	t	1.6	t	99	t	1	t	104	f	?	SVI	negative.	
21	28	M	f	f	f	f	f	f	f	f	f	f	f	f	f	t	t	2.1	t	2.6	t	121	t	0.9	t	130	f	?	SVHC	negative.	
22	51	F	t	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.1	f	?	t	130	t	0.9	t	151	f	?	other	negative.	
23	46	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.8	t	2.1	t	108	t	0.9	t	119	f	?	other	negative.	
24	81	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.9	t	0.3	t	102	t	1	t	106	f	?	SVI	negative.	

รูปที่ 4.13 ไฟล์ข้อมูลการเรียนรู้ในโปรแกรม Excel

ไฟล์ข้อมูลการทดสอบ

เป็นไฟล์ที่ใช้บรรจุข้อมูลที่ใช้ในการทดสอบ มีนามสกุล test เช่น hypothyroid.test เป็นต้น และมีรูปแบบเดียวกับไฟล์ข้อมูลที่ใช้ในการเรียนรู้

การปรับแต่งข้อมูลและจินตทัศน์ของชุดข้อมูล

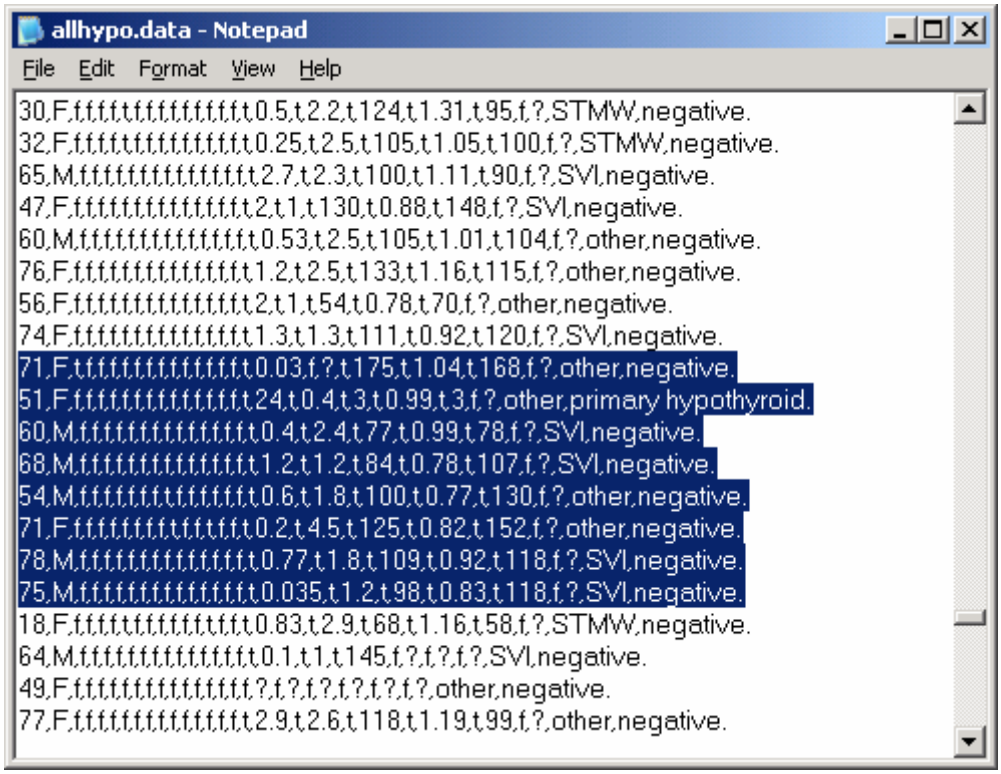
การปรับแต่งข้อมูล

ก่อนจะทำการเรียนรู้ด้วยอัลกอริทึมใดๆ อาจจะต้องมีการปรับแต่งข้อมูลเพื่อให้เหมาะกับกระบวนการการเรียนรู้ของอัลกอริทึมนั้น หรือในบางกรณีข้อมูลที่เก็บมาอาจจะมีส่วนที่ไม่เกี่ยวข้องกับลักษณะงานที่ต้องการจะทำ จึงจำเป็นต้องปรับแต่งข้อมูลก่อนที่จะทำการเรียนรู้ ซึ่งการปรับแต่งข้อมูลก่อนการเรียนรู้สามารถทำได้ดังนี้

การตัดข้อมูลที่ไม่จำเป็น

ข้อมูลสำหรับการเรียนรู้ที่เก็บมานั้นอาจจะมีจำนวนมากไป หรืออาจจะมีข้อมูลที่ซ้ำๆ กันเป็นจำนวนมาก ซึ่งจำนวนข้อมูลที่มากขึ้นอาจจะทำให้กระบวนการเรียนรู้ทำได้ช้าลง ขณะที่ประสิทธิภาพในการเรียนรู้ยังคงเท่าเดิม การตัดข้อมูลที่ไม่จำเป็นออกจึงเป็นการเตรียมข้อมูลวิธีการหนึ่งก่อนที่จะทำการเรียนรู้

การตัดข้อมูลที่ไม่จำเป็นออกสามารถทำได้โดยลบแถวของข้อมูลที่ไม่ต้องการในไฟล์ข้อมูล (ไฟล์ .data) ดังแสดงในรูปที่ 4.14

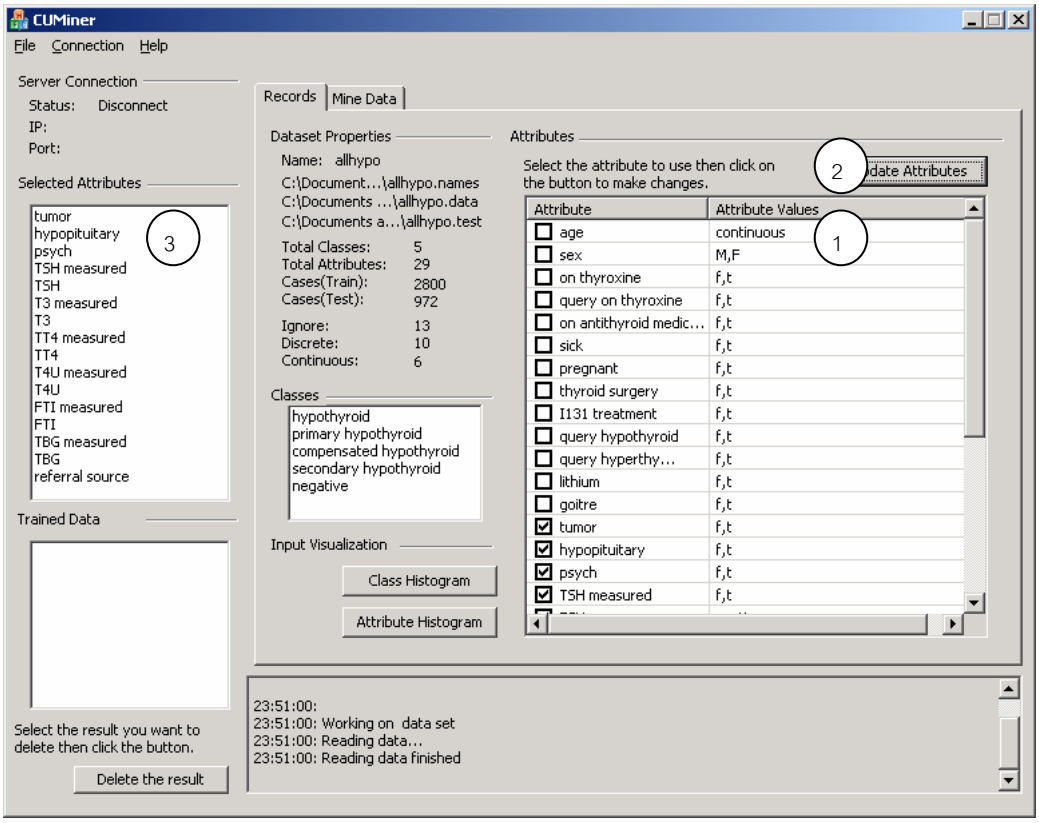


รูปที่ 4.14 การเลือกและลบข้อมูลที่ไม่จำเป็นในไฟล์ข้อมูล

การตัดคุณสมบัติที่ไม่เกี่ยวข้อง

คุณสมบัติบางตัวของข้อมูลที่เก็บมานั้น เมื่อพิจารณาตามสามัญสำนึกแล้วจะเห็นว่าคุณสมบัติเหล่านี้ไม่ส่งผลหรือเกี่ยวข้องกับการแยกแยะเลย เช่น เลขประจำตัวในบัตรประชาชนจะไม่เกี่ยวข้องกับการวินิจฉัยโรคคอพอกของผู้ป่วย เป็นต้น การตัดคุณสมบัติที่ไม่เกี่ยวข้องออกจะเพิ่มความเร็วในกระบวนการการเรียนรู้ รวมทั้งช่วยเพิ่มความแม่นยำ และประสิทธิภาพในการเรียนรู้ด้วย

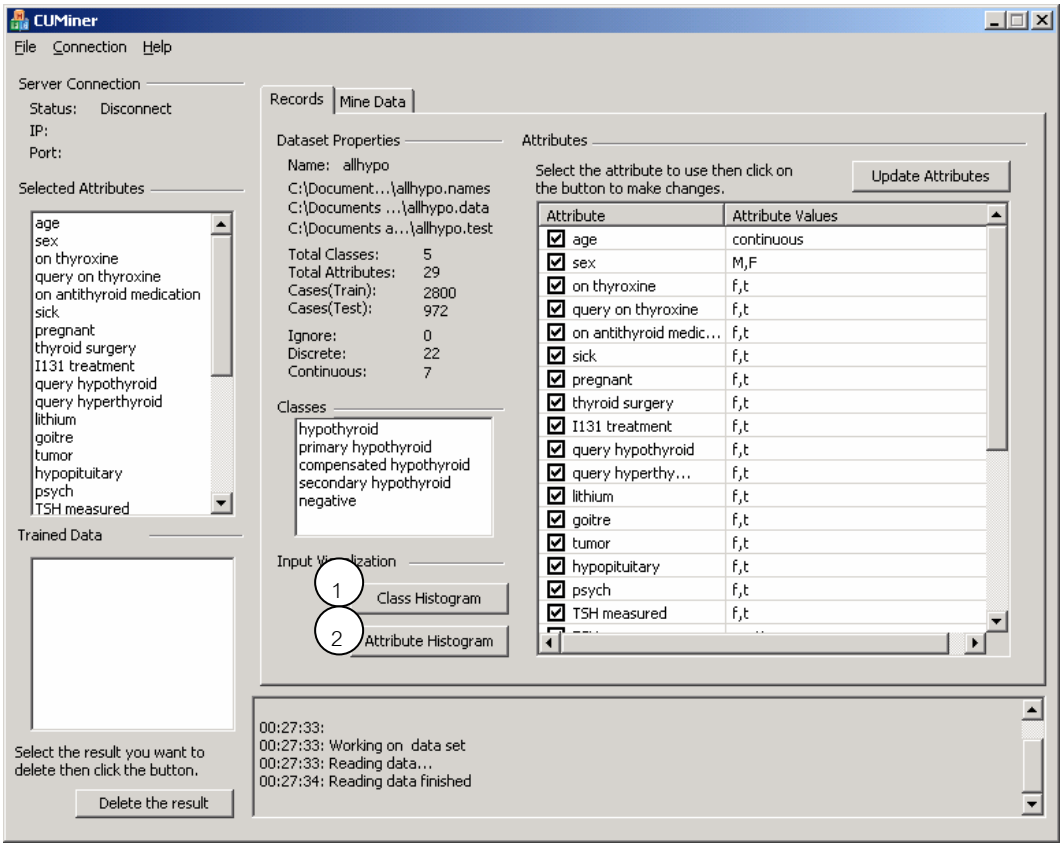
คุณสมบัติทั้งหมดของข้อมูลจะถูกแสดงให้เห็นที่แท็บ Records ของโปรแกรม ซึ่งในตอนต้นจะกำหนดไว้ว่าคุณสมบัติทุกคุณสมบัติจะถูกใช้ในการเรียนรู้ การตัดคุณสมบัติที่ไม่เกี่ยวข้องทำได้โดยคลิกที่กล่องเช็คหน้าชื่อคุณสมบัตินั้นเพื่อให้เครื่องหมายเช็คหายไป (1) และกดปุ่ม Update Attributes (2) เพื่อทำการเลือกใช้เฉพาะคุณสมบัติที่ต้องการ คุณสมบัติที่ถูกใช้ในการเรียนรู้จะปรากฏอยู่ที่กล่องรายการด้านซ้ายมือของหน้าจอ (3) ดังแสดงในรูปที่ 4.15



รูปที่ 4.15 การตัดคุณสมบัติที่ไม่เกี่ยวข้อง

จินตทัศน์ของชุดข้อมูลที่ใช้ในการเรียนรู้

ชุดข้อมูลที่ใช้ในการเรียนรู้ประกอบด้วยค่าทางสถิติต่างๆ เช่น จำนวนข้อมูลที่มีกลุ่มหนึ่งๆ หรือจำนวนข้อมูลที่คุณสมบัติใดๆ มีค่าเป็นค่าคุณสมบัติที่ต้องการนั้น เป็นต้น เพื่อให้ผู้ใช้สามารถทราบลักษณะของชุดข้อมูลด้วยค่าทางสถิติเหล่านี้ จึงมีการนำค่าทางสถิติของชุดข้อมูลมาแสดงให้เห็นด้วยการทำจินตทัศน์

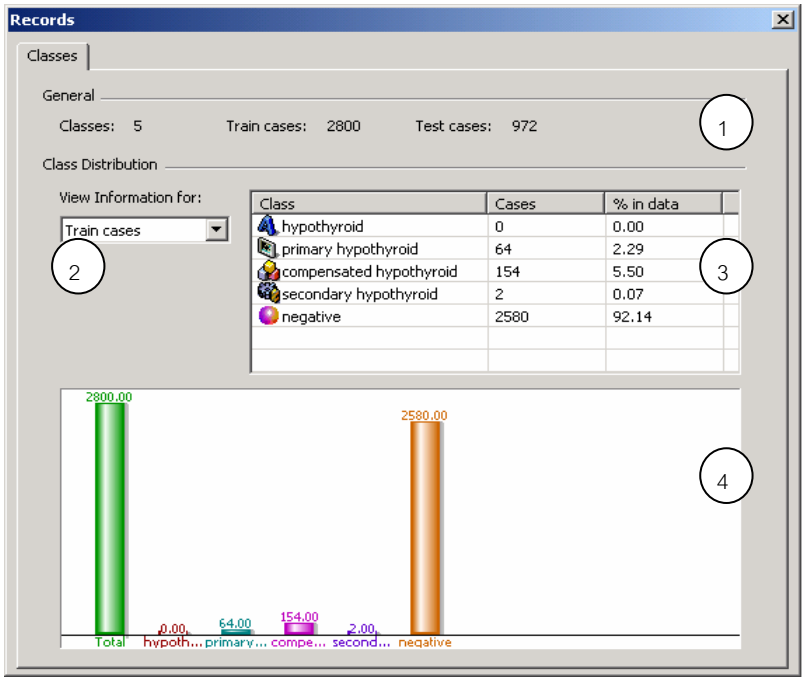


รูปที่ 4.16 ปุ่มที่ใช้เพื่อทำจินตทัศน์ของชุดข้อมูล

ผู้ใช้สามารถดูรายละเอียดเกี่ยวกับกลุ่มของชุดข้อมูลได้โดยกดปุ่ม Class Histogram (1) และสามารถดูรายละเอียดเกี่ยวกับคุณสมบัติของชุดข้อมูลได้โดยกดปุ่ม Attribute Histogram (2) ซึ่งทั้งสองปุ่มจะอยู่ทางซ้ายล่างของแท็บ Records (ดูรูปที่ 4.16 ประกอบ)

รายละเอียดของกลุ่มในชุดข้อมูล

เมื่อผู้ใช้กดปุ่ม Class Histogram จะปรากฏหน้าจอแสดงรายละเอียดของกลุ่มในข้อมูลดังรูปที่ 4.17 ด้านล่างนี้



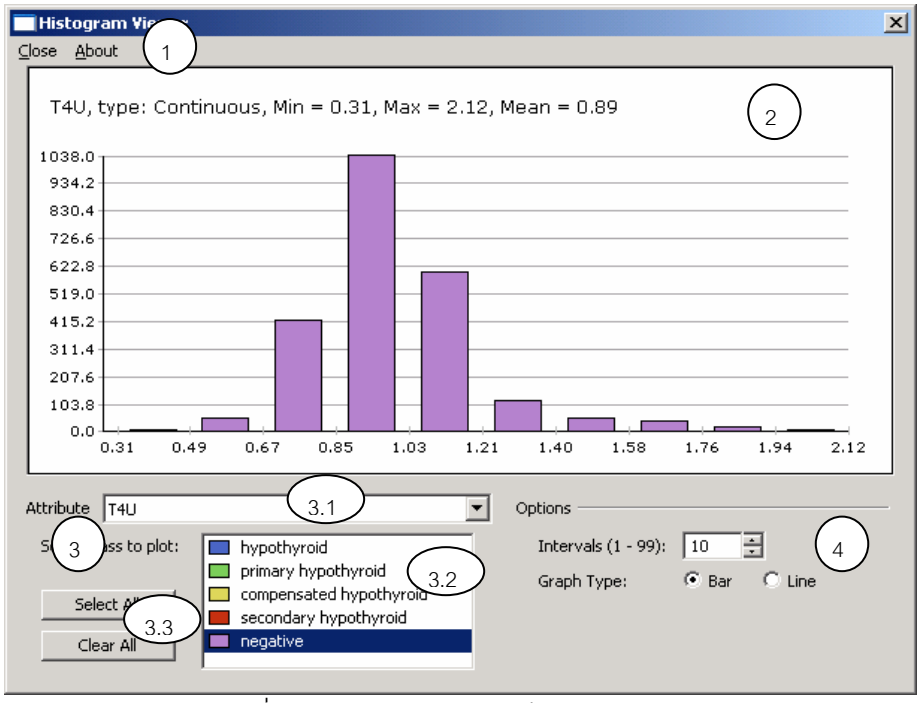
รูปที่ 4.17 หน้าจอแสดงรายละเอียดของกลุ่ม

หน้าจอแสดงรายละเอียดของกลุ่มประกอบด้วยส่วนต่าง ๆ คือ

- ส่วนแสดงรายละเอียดทั่วไป (1) บอกจำนวนกลุ่มในชุดข้อมูล จำนวนข้อมูลที่ใช้ในการเรียนรู้ และจำนวนข้อมูลที่ใช้ทดสอบ
- กล่องคอมโบสำหรับเลือกการแสดงผล (2) ว่าจะแสดงรายละเอียดกลุ่มของข้อมูลที่ใช้ในการเรียนรู้ หรือรายละเอียดกลุ่มของข้อมูลที่ใช้ในการทดสอบ
- ตารางแสดงรายละเอียดของกลุ่ม (3) ว่าข้อมูลที่สังกัดอยู่ในกลุ่มแต่ละกลุ่มมีจำนวนเท่าใด และคิดเป็นกี่เปอร์เซ็นต์เมื่อเทียบกับจำนวนข้อมูลทั้งหมด
- แผนภูมิแท่ง (4) แสดงจำนวนข้อมูลที่สังกัดอยู่ในกลุ่มแต่ละกลุ่มเปรียบเทียบกัน และเปรียบเทียบกับจำนวนข้อมูลทั้งหมด

รายละเอียดของคุณสมบัติในชุดข้อมูล

เมื่อผู้ใช้กดปุ่ม Attribute Histogram จะปรากฏหน้าจอแสดงรายละเอียดของคุณสมบัติในข้อมูลดังรูปที่ 4.18



รูปที่ 4.18 หน้าจอแสดงรายละเอียดของคุณสมบัติ

หน้าจอแสดงรายละเอียดของคุณสมบัติประกอบด้วยส่วนต่างๆ คือ

- รายการ (1) ประกอบด้วยรายการสำหรับปิดหน้าจอ และรายการความช่วยเหลือ
- กรอบแสดงผล (2) สามารถแสดงผลได้ทั้งในรูปแบบแผนภูมิแท่งและกราฟ
- ส่วนเลือกข้อมูลการแสดงผล (3) ประกอบด้วยกล่องคอมโบสำหรับเลือกคุณสมบัติที่ต้องการจะศึกษารายละเอียด (3.1) กล่องรายการสำหรับกำหนดกลุ่มของข้อมูลที่จะมาแสดงรายละเอียด (3.2) และปุ่มสำหรับเลือกว่าจะใช้กลุ่มทั้งหมดเพื่อการแสดงผลหรือลบกลุ่มทั้งหมดทิ้งไป (3.3)
- ส่วนเลือกรูปแบบการแสดงผล (4) สามารถเลือกได้ว่าจะแสดงผลในรูปของแผนภูมิแท่งหรือกราฟ เลือกแสดงผลแบบแผนภูมิแท่งทำได้โดยกดที่ปุ่ม Bar หรือกดที่ปุ่ม Line เพื่อเลือกแสดงผลแบบกราฟ นอกจากนี้ยังสามารถแบ่งช่วงการแสดงผลสำหรับคุณสมบัติแบบต่อเนื่องได้ ว่าจะแบ่งค่าคุณสมบัติของคุณสมบัติแบบต่อเนื่องนั้นตั้งแต่ค่าน้อยสุดไปจนถึงค่ามากสุดเป็นกี่ช่วง โดยการป้อนช่วงตัวเลข หรือเลื่อนค่าที่กล่องสปิน

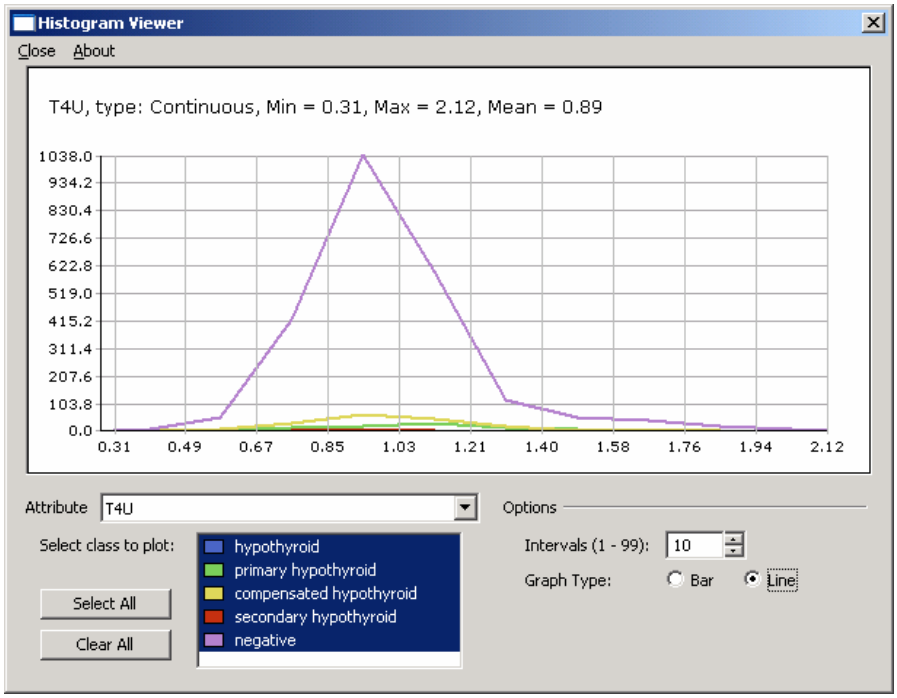
การตีความที่กรอบแสดงผล

ช่วงบนของกรอบแสดงผลจะบอกชื่อของคุณสมบัติที่เลือก (เลือกคุณสมบัติโดยใช้กล่องคอมโบด้านล่าง) ทั้งนี้ถ้าคุณสมบัติมีค่าคุณสมบัติเป็นแบบต่อเนื่อง ช่วงบนของกรอบแสดงผลจะแสดงค่าน้อยสุด ค่ามากสุด และค่าเฉลี่ยของค่าคุณสมบัติของคุณสมบัตินั้นที่มีอยู่ในชุดข้อมูลเอาไว้ด้วย ในรูปที่ 4.19 ช่วงบนของกรอบแสดงผลบอกว่าคุณสมบัตินี้ ชื่อ T4U มีค่าคุณสมบัติแบบต่อเนื่อง (Continuous) มีค่าน้อยสุดคือ 0.31 (Min = 0.31) มีค่ามากสุดคือ 2.12 (Max=2.12) และมีค่าเฉลี่ยคือ 0.89 (Mean=0.89)

ในแนวนอนของกรอบแสดงผลจะแสดงค่าคุณสมบัติของคุณสมบัติที่เลือก ถ้าคุณสมบัติที่เลือกมีค่าคุณสมบัติแบบไม่ต่อเนื่อง แนวนอนของกรอบแสดงผลจะแบ่งแยกเป็นค่าคุณสมบัติแต่ละค่า ถ้าคุณสมบัติที่เลือกมีค่าคุณสมบัติแบบต่อเนื่อง แนวนอนของกรอบแสดงผลจะแบ่งแยกเป็นค่าคุณสมบัติออกเป็นช่วงๆ ตั้งแต่ค่าน้อยสุดไปจนถึงค่ามากสุด ซึ่งผู้ใช้สามารถเลือกจำนวนช่วงที่จะแบ่งได้โดยใช้กล่องสปินด้านล่าง

ในแนวตั้ง หรือความสูงของแผนภูมิแท่งและกราฟนั้น ถ้าคุณสมบัติที่เลือกมีค่าคุณสมบัติแบบไม่ต่อเนื่อง ความสูงจะแสดงถึงจำนวนข้อมูลที่มีค่าคุณสมบัติแต่ละค่านั้น ถ้าคุณสมบัติที่เลือกมีค่าคุณสมบัติแบบต่อเนื่อง ความสูงจะแสดงถึงจำนวนข้อมูลที่มีค่าคุณสมบัติอยู่ในช่วงนั้น

ดังรูปที่ 4.19 สามารถบอกได้คร่าวๆ ว่าในชุดข้อมูลนี้ จำนวนข้อมูลที่มีกลุ่ม negative ที่มีค่าคุณสมบัติของคุณสมบัติ T4U อยู่ในช่วงระหว่าง 0.85 ถึง 1.03 จะมีจำนวนสูงกว่าจำนวนข้อมูลที่มีกลุ่ม neagive ที่มีค่าคุณสมบัติของคุณสมบัติ T4U อยู่ในช่วงอื่น โดยมีข้อมูลประมาณหนึ่งพันตัว



รูปที่ 4.19 กรอบแสดงผลในรูปแบบกราฟ

รายละเอียดการเรียนรู้ของอัลกอริทึมต่าง ๆ

ต้นไม้ตัดสินใจ

การนำไปใช้

- สำหรับสร้างแบบจำลองเพื่อใช้แยกแยะข้อมูลออกเป็นกลุ่มต่าง ๆ
- สำหรับทำนายกลุ่มของข้อมูล

แนวคิด

- ต้นไม้ตัดสินใจนับได้ว่าเป็นอัลกอริทึมสำหรับแยกแยะข้อมูลที่รวดเร็วและให้ผลลัพธ์ที่ตีความได้ง่าย การแยกแยะด้วยต้นไม้ตัดสินใจถูกนำไปใช้อย่างแพร่หลายทั้งทางด้านธุรกิจและวิทยาศาสตร์

ลักษณะของคุณสมบัติ

- เป็นได้ทั้งคุณสมบัติที่มีค่าคุณสมบัติต่อเนื่องและคุณสมบัติที่มีค่าคุณสมบัติไม่ต่อเนื่อง
- ค่าคุณสมบัติขาดหายไปได้

จำนวนข้อมูลที่เหมาะสม

- อย่างน้อย 100 ข้อมูล
- อย่างมาก 500,000 ข้อมูล

ข้อเสนอแนะในการเตรียมข้อมูล

- คุณสมบัติที่มีค่าคุณสมบัติหลากหลายเกินไปอาจทำให้ผลลัพธ์ผิดเพี้ยนและดูยาก

อัลกอริทึมที่ใช้

- C4.5

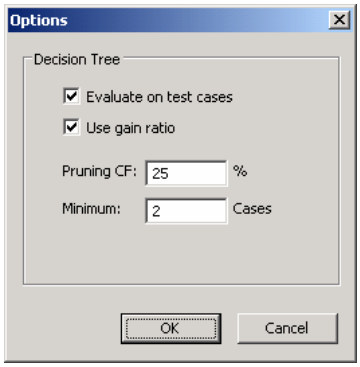
วิธีใช้ซอฟต์แวร์เพื่อการเรียนรู้

- เลือกช่องคอมโบเป็น “Decision Trees”
- กดปุ่ม “Options…” เพื่อปรับแต่งพารามิเตอร์
- กดปุ่ม “Train” เพื่อทำการเรียนรู้

พารามิเตอร์ในการเรียนรู้

พารามิเตอร์ในการเรียนรู้แสดงในรูปที่ 4.20 ด้านล่าง และมีรายละเอียดดังนี้

- Evaluate on test cases บอกว่าจะให้ทำการแยกแยะไฟล์ทดสอบหรือไม่
- Use gain ratio เลือกว่าจะใช้ค่ามาตรฐานอัตราส่วนเกนหรือไม่
- Pruning CF เลือกเปอร์เซ็นต์ความเชื่อมั่นที่ใช้ในการตัดเล็มต้นไม้ตัดสินใจ
- Minimum Cases คือจำนวนข้อมูลน้อยสุดที่ต้องใช้ในการสร้างโนดของต้นไม้ตัดสินใจ



รูปที่ 4.20 กรอบโต้ตอบตัวเลือกการเรียนรู้ของต้นไม้ตัดสินใจ

นิวรอลเน็ตเวิร์ก

การนำไปใช้

- สำหรับสร้างแบบจำลองเพื่อใช้แยกแยะข้อมูลออกเป็นกลุ่มต่าง ๆ
- สำหรับทำนายกลุ่มของข้อมูล

แนวคิด

- การเรียนรู้ของโครงข่ายประสาทเทียมสามารถสร้างระนาบการแบ่งแยกข้อมูลขึ้นมาได้อย่างซับซ้อน โดยทั่วไปมักให้ความถูกต้องแม่นยำมากกว่าการแยกแยะโดยใช้วิธีอื่น

ลักษณะของคุณสมบัติ

- เป็นได้ทั้งคุณสมบัติที่มีค่าคุณสมบัติต่อเนื่องและคุณสมบัติที่มีค่าคุณสมบัติไม่ต่อเนื่อง
- ค่าคุณสมบัติขาดหายไปได้

จำนวนข้อมูลที่เหมาะสม

- อย่างน้อย 100 ข้อมูล
- อย่างมาก 500,000 ข้อมูล

อัลกอริทึมที่ใช้

- Backpropagation Learning

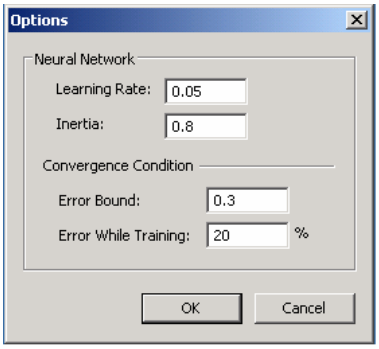
วิธีใช้ซอฟต์แวร์เพื่อการเรียนรู้

- เลือกช่องคอมโบเป็น “Neural Networks”
- กดปุ่ม “Options…” เพื่อปรับแต่งพารามิเตอร์
- กดปุ่ม “Train” เพื่อทำการเรียนรู้

พารามิเตอร์ในการเรียนรู้

พารามิเตอร์ในการเรียนรู้แสดงในรูปที่ 4.21 ด้านล่าง และมีรายละเอียดดังนี้

- Learning Rate หรือค่าอัตราการเรียนรู้ หมายถึงค่าคงที่ซึ่งแสดงให้เห็นว่าเน็ตเวิร์กควรที่จะเรียนรู้ด้วยอัตราความเร็วเท่าใด
- Inertia คือค่าคงที่ซึ่งใช้ในการปรับน้ำหนักของเส้นเชื่อมให้ลู่เข้าค่าที่ถูกต้องได้เร็วยิ่งขึ้น
- Error Bound และ Error While Training หมายความว่า เมื่อนำค่าเอาท์พุตที่คำนวณได้มาเปรียบเทียบกับค่าเอาท์พุตจริงแล้วนั้น มีค่าสัมบูรณ์ของความต่างเกินค่า Error Bound ถึงจำนวนที่คิดเป็นเปอร์เซ็นต์แล้วมากกว่า Error While Training โปรแกรมจะทำการปรับค่าน้ำหนักเส้นเชื่อมใหม่อีกครั้ง เนื่องจากจะพิจารณาว่าค่าเดิมไม่สามารถยอมรับได้ แต่ถ้าคิดแล้วไม่เกิน Error While Training เป็นเปอร์เซ็นต์ จะหยุดการปรับค่าน้ำหนัก เพราะถือว่าค่าน้ำหนักเส้นเชื่อมในขณะนั้นสามารถยอมรับได้



รูปที่ 4.21 กรอบโต้ตอบตัวเลือกการเรียนรู้ของตัวแยกแยะนิวรอลเน็ตเวิร์ก

การเรียนรู้แบบเบย์อย่างง่าย

การนำไปใช้

- สำหรับสร้างแบบจำลองเพื่อใช้แยกแยะข้อมูลออกเป็นกลุ่มต่าง ๆ
- สำหรับทำนายกลุ่มของข้อมูล

แนวคิด

- การเรียนรู้แบบเบย์อย่างง่ายเป็นวิธีการแยกแยะข้อมูลที่มีประสิทธิภาพวิธีหนึ่ง โดยผลลัพธ์ที่ได้นั้นเทียบได้กับผลลัพธ์จากอัลกอริทึมที่มีความซับซ้อนกว่า เช่น C4.5

ลักษณะของคุณสมบัติ

- เป็นได้ทั้งคุณสมบัติที่มีค่าคุณสมบัติต่อเนื่องและคุณสมบัติที่มีค่าคุณสมบัติไม่ต่อเนื่อง
- ค่าคุณสมบัติขาดหายไปได้

จำนวนข้อมูลที่เหมาะสม

- อย่างน้อย 100 ข้อมูล
- อย่างมาก 500,000 ข้อมูล

ข้อเสนอแนะในการเตรียมข้อมูล

- คุณสมบัติที่มีค่าคุณสมบัติแบบต่อเนื่องอาจทำให้การเรียนรู้ช้าลง

อัลกอริทึมที่ใช้

- Naïve Bayes Leaning

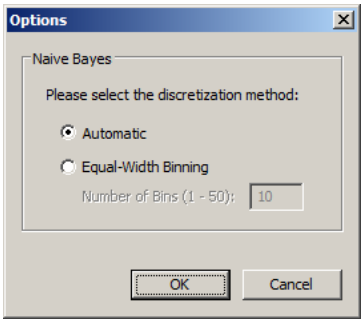
วิธีใช้ซอฟต์แวร์เพื่อการเรียนรู้

- เลือกช่องคอมโบเป็น “Naïve Bayes”
- กดปุ่ม “Options…” เพื่อปรับแต่งพารามิเตอร์
- กดปุ่ม “Train” เพื่อทำการเรียนรู้

พารามิเตอร์ในการเรียนรู้

การเลือกวิธีทำให้ข้อมูลเป็นแบบไม่ต่อเนื่อง ในกรณีที่ข้อมูลมีคุณสมบัติที่มีค่าคุณสมบัติแบบต่อเนื่อง แสดงในรูปที่ 4.22 ซึ่งมีสองวิธี คือ

- Automatic ตัวเรียนรู้จะทำการแบ่งช่วงให้เองโดยอัตโนมัติ
- Equal-Width Binning กำหนดให้แบ่งเป็นช่วงเท่า ๆ กัน โดยสามารถกำหนดจำนวนของช่วงได้ระหว่าง 1 – 50



รูปที่ 4.22 กรอบโต้ตอบตัวเลือกการเรียนรู้ของตัวแยกแยะเบย์อย่างง่าย

การค้นหากฎความสัมพันธ์

การนำไปใช้

- สำหรับหาความสัมพันธ์ของค่าคุณสมบัติต่างๆ ของข้อมูล

แนวคิด

- การค้นหากฎความสัมพันธ์เป็นการค้นหาว่าค่าต่างๆ ที่สนใจมีค่าใดบ้างที่มีโอกาสเกิดขึ้นพร้อมกัน และค่าใดที่เป็นตัวเหนี่ยวนำก่อให้เกิดอีกค่าหนึ่งขึ้น ซึ่งสามารถนำแนวคิดนี้ไปประยุกต์ใช้ได้หลากหลาย โดยมีจุดเริ่มต้นจากการวิเคราะห์การตลาดว่าเมื่อลูกค้าซื้อสินค้าชนิดนี้ไป ลูกค้าจะซื้อสินค้าชนิดใดไปอีกบ้าง นอกจากนี้ยังมีการนำไปใช้ในการหาผลข้างเคียงของตัวยาที่ใช้พร้อมกัน หามาตรากฎหมายที่เหมาะสมจะมาใช้ในคดี และงานต่างๆ ทางด้านสังคม ธุรกิจ และวิทยาศาสตร์

ลักษณะของคุณสมบัติ

- ใช้ได้เฉพาะคุณสมบัติที่มีค่าคุณสมบัติไม่ต่อเนื่อง
- ค่าคุณสมบัติขาดหายไปได้

จำนวนข้อมูลที่เหมาะสม

- อย่างน้อย 100 ข้อมูล
- อย่างมาก 10,000 ข้อมูล

ข้อเสนอแนะในการเตรียมข้อมูล

- ไม่สามารถใช้ได้กับคุณสมบัติที่มีค่าคุณสมบัติไม่ต่อเนื่อง

อัลกอริทึมที่ใช้

- CHARM
- Apriori – GenRule

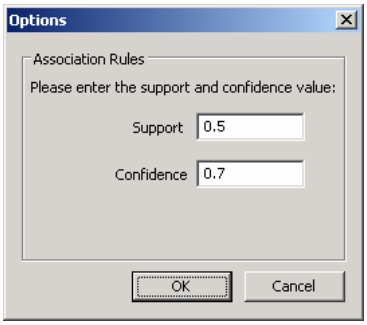
วิธีใช้ซอฟต์แวร์เพื่อการเรียนรู้

1. เลือกช่องคอมโบเป็น “Association Rules”
2. กดปุ่ม “Options…” เพื่อปรับแต่งพารามิเตอร์
3. กดปุ่ม “Train” เพื่อทำการเรียนรู้

พารามิเตอร์ในการเรียนรู้

พารามิเตอร์ในการเรียนรู้แสดงในรูปที่ 4.23 ด้านล่าง และมีรายละเอียดดังนี้

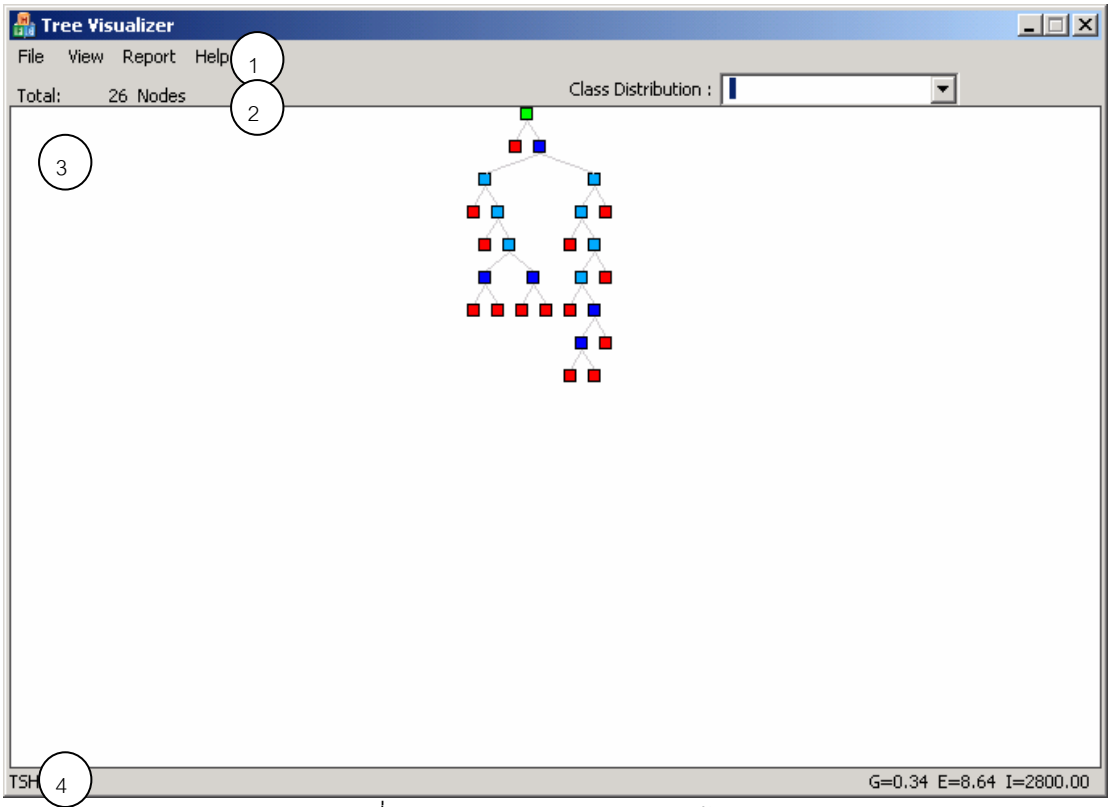
- Support คือค่าสนับสนุนน้อยสุด มีค่าตั้งแต่ 0-1
- Confidence คือค่าความเชื่อมั่นน้อยสุด มีค่าตั้งแต่ 0-1



รูปที่ 4.23 กรอบโต้ตอบตัวเลือกการค้นหากฎความสัมพันธ์

หน้าจอแสดงผล

เมื่อทำการเรียกการแสดงผลของต้นไม้ตัดสินใจ จะปรากฏหน้าจอดังรูปที่ 4.24



รูปที่ 4.24 ส่วนประกอบต่าง ๆ ของหน้าจอ

หน้าจอการแสดงผลของต้นไม้ตัดสินใจประกอบด้วยส่วนต่างๆ ดังนี้

1. **รายการ** ประกอบด้วยรายการหลักคือ รายการสำหรับปิดหน้าจอ รายการสำหรับเลือกรูปแบบและรายละเอียดการแสดงผล รายการรายงานผล และรายการช่วยเหลือ
2. **แถบบน** ประกอบด้วยตัวเลขแสดงจำนวนโนด และกล่องคอมโบสำหรับเลือกกลุ่ม
3. **หน้าจอแสดงผลต้นไม้** แสดงรูปต้นไม้และรายละเอียดต่างๆ
4. **แถบล่าง** แสดงรายละเอียดของโนดในต้นไม้เมื่อผู้ใช้เลื่อนเมาส์ไปถึง

รูปแบบการแสดงผลของต้นไม้ตัดสินใจ

ต้นไม้ตัดสินใจที่เป็นผลลัพธ์จากการเรียนรู้สามารถแสดงผลได้ 2 รูปแบบคือ

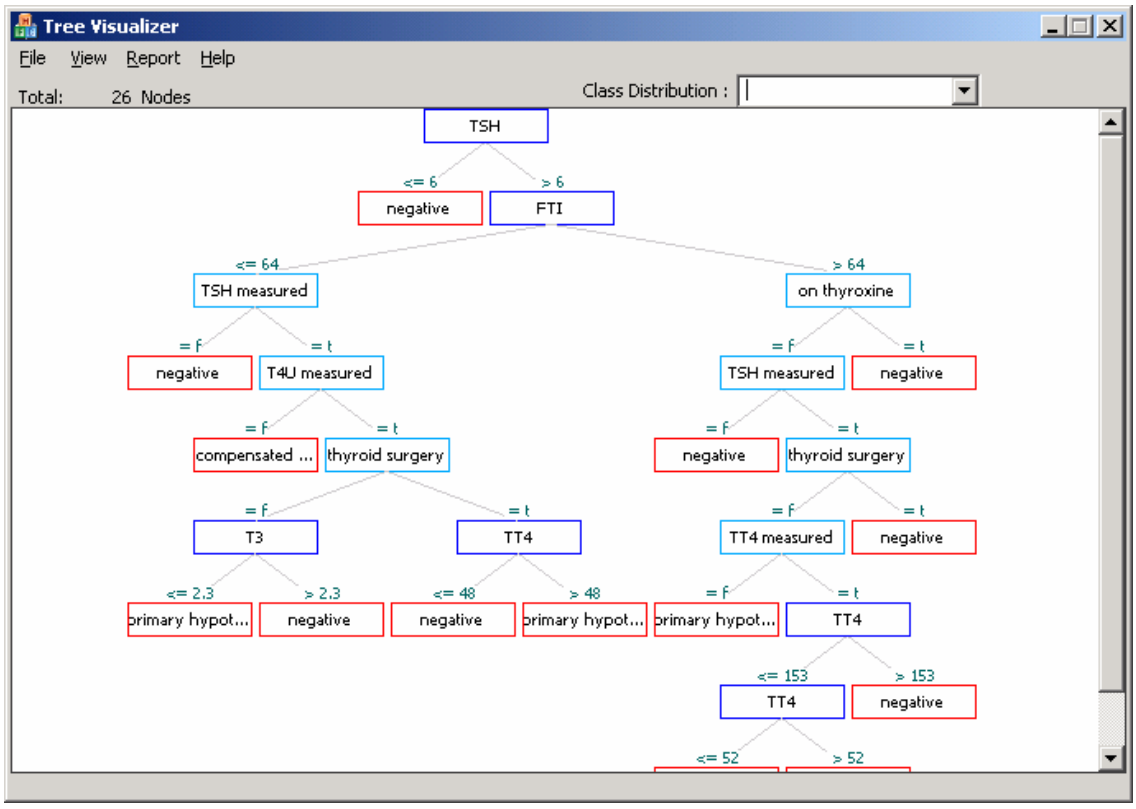
รูปต้นไม้แบบย่อ

รูปต้นไม้แบบย่อจะแสดงผลต้นไม้เพียงโนดและกิ่งโดยที่ไม่บอกรายละเอียดในแต่ละโนด เพื่อให้ความสะดวกในการดูภาพรวมของต้นไม้ทั้งหมดในกรณีที่ต้นไม้มีขนาดใหญ่ ซึ่งรูปต้นไม้แบบย่อนี้จะเป็นรูปต้นไม้ที่แสดงเป็นรูปแรกเมื่อเปิดหน้าจอการแสดงผลของต้นไม้ตัดสินใจ

รูปต้นไม้แบบย่อมีการแสดงผลเป็นดังรูปที่ 4.24

รูปต้นไม้แบบเต็ม

รูปต้นไม้แบบเต็มจะแสดงผลต้นไม้โดยบอกรายละเอียดของโนดและกิ่ง ซึ่งรายละเอียดการแสดงผลสามารถปรับเปลี่ยนได้ตามความต้องการของผู้ใช้ ดังแสดงในรูปที่ 4.25



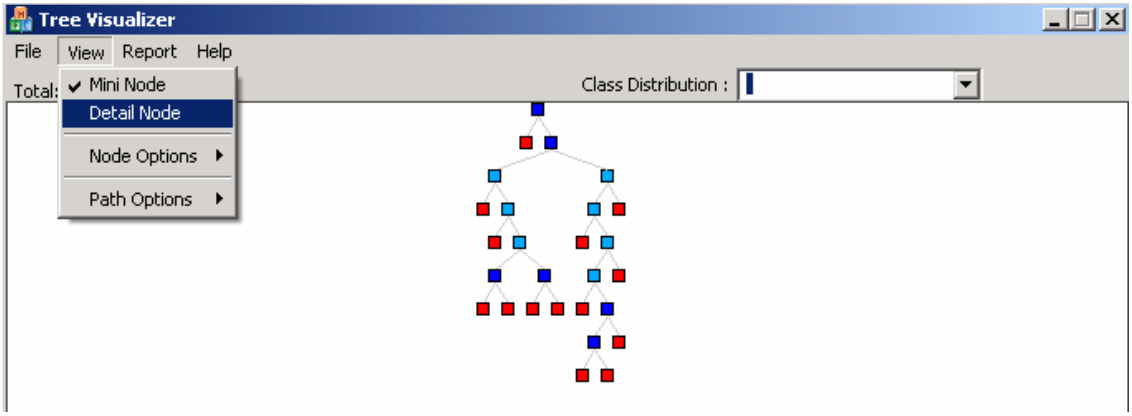
รูปที่ 4.25 รูปต้นไม้ตัดสินใจแบบเต็ม

รายละเอียดการแสดงผลของต้นไม้ตัดสินใจ

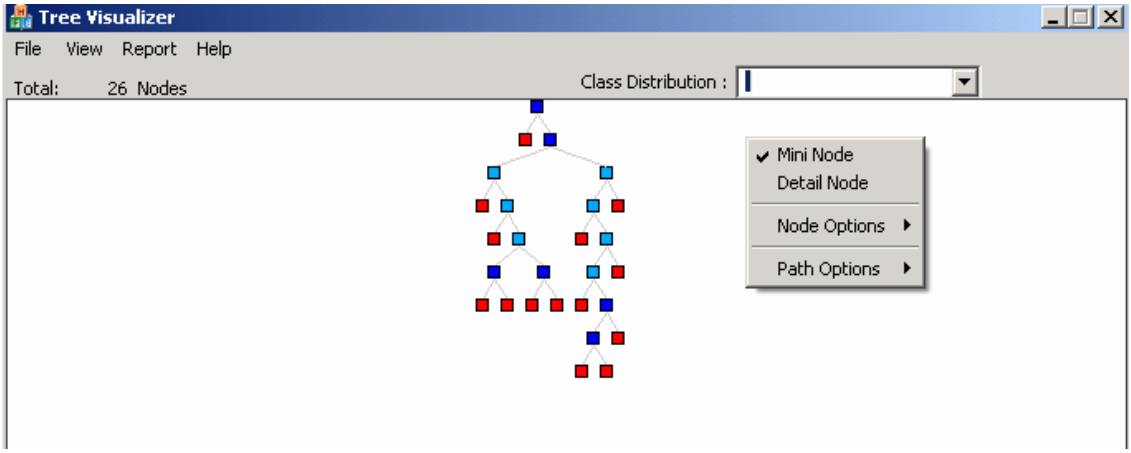
การปรับเปลี่ยนรูปแบบและรายละเอียดการแสดงผล

รูปแบบและรายละเอียดการแสดงผลของต้นไม้ตัดสินใจสามารถปรับเปลี่ยนได้จากรายการมุมมอง หรือจากการคลิกขวาที่หน้าจอ โดยรายละเอียดบางอย่างไม่สามารถแสดงผลได้เมื่อรูปแบบการแสดงผลของต้นไม้ตัดสินใจอยู่ในรูปต้นไม้แบบย่อ

การปรับเปลี่ยนรูปแบบการแสดงผลของต้นไม้ตัดสินใจทำได้โดยเลือกรายการ View และเลือกรูปแบบการแสดงผล ถ้าอยู่ในรูปต้นไม้แบบย่อและต้องการรูปต้นไม้แบบเต็มให้คลิกที่ Detail Node ถ้าอยู่ในรูปต้นไม้แบบเต็มและต้องการรูปต้นไม้แบบย่อให้คลิกที่ Mini Node ซึ่งการเลือกรูปแบบต้นไม้โดยใช้การคลิกขวาสามารถทำได้ในลักษณะเดียวกัน (ดูรูปที่ 4.26 และรูปที่ 4.27 ประกอบ)



รูปที่ 4.26 การใช้รายการมุมมองเพื่อปรับเปลี่ยนรูปแบบและรายละเอียดการแสดงผลของต้นไม้ตัดสินใจ



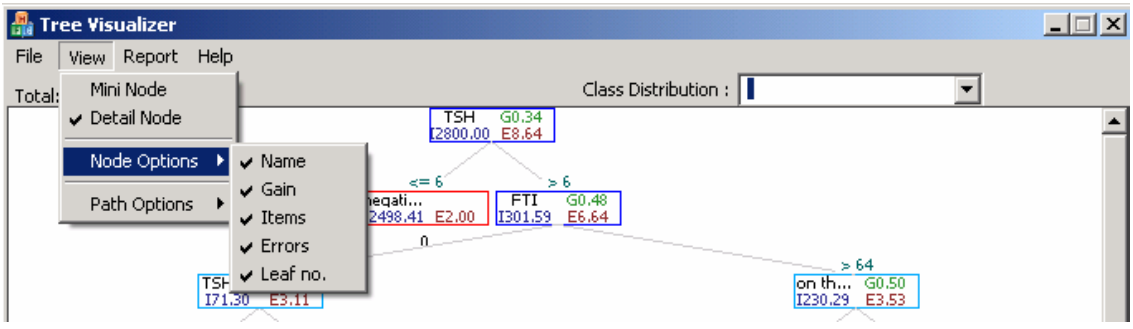
รูปที่ 4.27 การคลิกขวาเพื่อปรับเปลี่ยนรูปแบบและรายละเอียดการแสดงผลของต้นไม้ตัดสินใจ

รายละเอียดการแสดงผลของแต่ละโนด

การแสดงผลของต้นไม้ตัดสินใจในแต่ละโนดมีรายละเอียดดังนี้

1. **สีของโนด** โนดในต้นไม้ตัดสินใจจะมีสีต่างกันเพื่อบอกชนิดของโนด โดยสีที่ใช้มีดังนี้
   - **สีน้ำเงิน** บ่งชี้ว่าโนดเป็นคุณสมบัติที่มีค่าคุณสมบัติแบบต่อเนื่อง
   - **สีฟ้า** บ่งชี้ว่าโนดเป็นคุณสมบัติที่มีค่าคุณสมบัติแบบไม่ต่อเนื่อง
   - **สีแดง** บ่งชี้ว่าโนดเป็นโนดใบ คือกลุ่มที่ได้จากการแยกแยะ
2. **ชื่อของโนด** รูปต้นไม้แบบเต็มสามารถแสดงชื่อของโนดได้
3. **ข้อมูลที่ตกลงมายังโนด** รูปต้นไม้แบบเต็มสามารถแสดงข้อมูลที่ตกลงมายังโนดได้
4. **ความผิดพลาดในแต่ละโนด** รูปต้นไม้แบบเต็มสามารถแสดงความผิดพลาดในการเรียนรู้ของแต่ละโนดได้
5. **ค่ามาตรฐานอัตราส่วนเกน** รูปต้นไม้แบบเต็มสามารถแสดงค่ามาตรฐานอัตราส่วนเกนที่คำนวณได้ของแต่ละโนดภายใน
6. **หมายเลขใบ** แสดงหมายเลขของใบเพื่อนำไปใช้อ้างอิงในส่วนของการสรุปผล

เมื่อต้นไม้ตัดสินใจถูกแสดงผลในรูปแบบเต็ม การปรับเปลี่ยนรายละเอียดการแสดงผลข้างต้นทำได้โดยการเลือกรายการ View->Node Options และเลือกรายละเอียดการแสดงผลของแต่ละโนด ถ้าต้องการให้แสดงชื่อของโนดคลิกที่ Name ถ้าต้องการให้แสดงค่ามาตรฐานอัตราส่วนเกนคลิกที่ Gain ถ้าต้องการให้แสดงข้อมูลที่ตกลงมายังโนดคลิกที่ Items ถ้าต้องการให้แสดงความผิดพลาดในแต่ละโนดคลิกที่ Errors ถ้าต้องการให้แสดงหมายเลขใบคลิกที่ Leaf No. ซึ่งการปรับเปลี่ยนรายละเอียดการแสดงผลโดยใช้การคลิกขวาสามารถทำได้ในลักษณะเดียวกัน (ดูรูปที่ 4.28 ประกอบ)



รูปที่ 4.28 รายละเอียดการแสดงผลของแต่ละโนดในต้นไม้ตัดสินใจ

เมื่อต้นไม้ตัดสินใจถูกแสดงผลในรูปแบบย่อ รายละเอียดของแต่ละโนด อันได้แก่ ชื่อของโนด ค่ามาตรฐานอัตราส่วนเกน ข้อมูลที่ตกลงมายังโนด และความผิดพลาดในแต่ละโนด จะถูกแสดงให้เห็นที่แถบล่างเมื่อผู้ใช้เลื่อนเมาส์ไปที่โนด ขณะที่หมายเลขใบยังคงแสดงผลอยู่ใต้โนดใบของต้นไม้

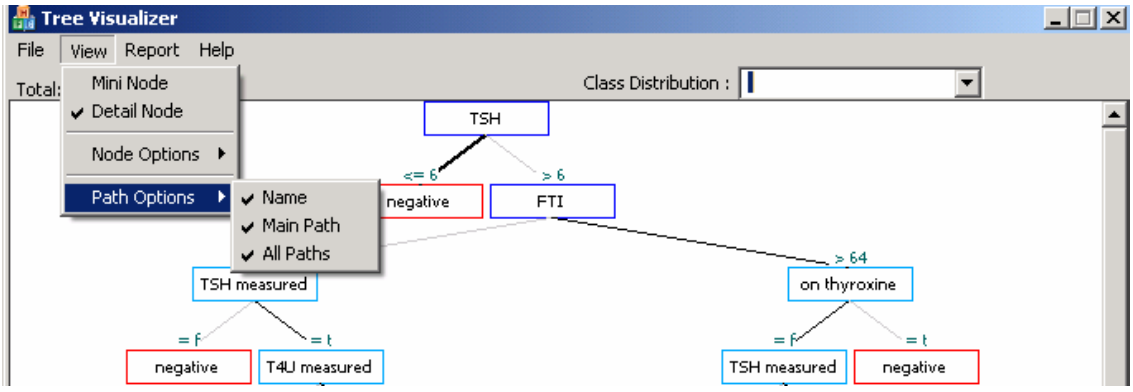
รายละเอียดการแสดงผลของแต่ละกิ่ง

การแสดงผลของต้นไม้ตัดสินใจในแต่ละกิ่งมีรายละเอียดดังนี้

1. **ความสำคัญของกิ่ง** กิ่งในต้นไม้ตัดสินใจสามารถจัดความสำคัญได้ตามจำนวนข้อมูลที่ตกลงมายังกิ่ง โดยแบ่งได้ดังนี้
   - **กิ่งธรรมดา** เป็นกิ่งสีเทา
   - **กิ่งซึ่งเป็นทางที่ข้อมูลส่วนใหญ่ไหลผ่าน** จะเป็นสีดำเข้ม โดยทางหลักที่ข้อมูลส่วนใหญ่ไหลผ่านจะมีทางเดียวคือทางที่ข้อมูลส่วนใหญ่ตกไปเมื่อถึงทางแยกในแต่ละโนด
   - **กิ่งซึ่งเป็นทางที่ข้อมูลตกลงมามากที่สุดในบรรดากิ่งที่สร้างจากโนดเดียวกัน** จะเป็นสีดำ

   ซึ่งค่าความสำคัญของกิ่งสามารถแสดงได้ทั้งในรูปต้นไม้แบบเต็มและรูปต้นไม้แบบย่อ
2. **ชื่อของกิ่ง** รูปต้นไม้แบบเต็มสามารถแสดงชื่อของกิ่งได้

การปรับเปลี่ยนรายละเอียดการแสดงผลข้างต้นทำได้โดยการเลือกรายการ View->Path Options และเลือกรายละเอียดการแสดงผลของกิ่ง ถ้าต้องการให้แสดงชื่อของกิ่งคลิกที่ Name ถ้าต้องการให้แสดงกิ่งที่เป็นทางผ่านของข้อมูลส่วนใหญ่คลิกที่ Main Path ถ้าต้องการให้แสดงกิ่งซึ่งเป็นทางที่ข้อมูลตกลงมามากที่สุดในบรรดากิ่งทั้งหลายที่สร้างจากโนดเดียวกันคลิกที่ All Paths (ดูรูปที่ 4.29 ประกอบ)

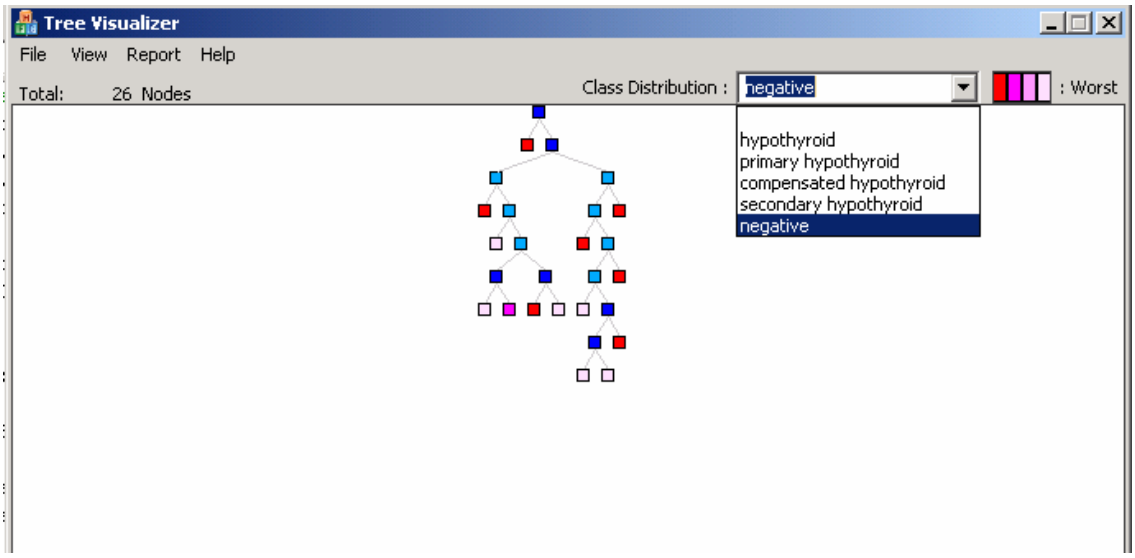


รูปที่ 4.29 รายละเอียดการแสดงผลของกิ่งในต้นไม้ตัดสินใจ

เมื่อต้นไม้ตัดสินใจถูกแสดงผลในรูปแบบย่อ ความสำคัญของกิ่งจะถูกแสดงให้เห็นได้ขณะที่ชื่อกิ่งจะไม่ถูกแสดงให้เห็น

การกระจายของกลุ่ม

รายละเอียดการแสดงผลของต้นไม้ตัดสินใจรวมไปถึงการแสดงการกระจายของกลุ่มในแต่ละโนดใบ ดังรูปที่ 4.30 ซึ่งจะแสดงเป็นสีที่บอกได้อย่างคร่าวๆ ว่าที่โนดใบต่างๆ มีกลุ่มที่เลือกอยู่มากน้อยเพียงใด การดูการกระจายของกลุ่มต่างๆ ทำได้โดยการเลือกกลุ่มที่ต้องการดูการกระจายจากกล่องคอมโบที่อยู่ที่แถบบนของหน้าจอการแสดงผลของต้นไม้ตัดสินใจ



รูปที่ 4.30 รูปต้นไม้ตัดสินใจที่แสดงการกระจายของกลุ่ม

ความมากน้อยของกลุ่มในแต่ละโนดใบจะถูกแสดงตามลำดับสี เริ่มจากสีแดงที่แสดงว่าโนดใบมีกลุ่มนี้อยู่มาก จนถึงมีสีชมพูอ่อนที่แสดงว่าโนดใบมีกลุ่มนี้อยู่น้อย

รายละเอียดในแต่ละโนดของต้นไม้ตัดสินใจ

เมื่อผู้ใช้คลิกที่โนดใดโนดหนึ่งของต้นไม้ รายละเอียดของโนดนั้นจะถูกแสดงขึ้นมา รายละเอียดเหล่านั้นประกอบไปด้วยข้อมูลที่เป็นประโยชน์ต่อการวิเคราะห์ต้นไม้ตัดสินใจดังนี้

ผังการแยกแยะ

ผังการแยกแยะจะแสดงกลุ่มต่างๆ ของโนดที่คลิกเข้าไป และกลุ่มต่างๆ ของโนดที่เป็นลูกของโนดนั้น ทั้งนี้เพื่อให้ผู้ใช้ได้เห็นพฤติกรรมของโนดว่าสามารถแยกแยะกลุ่มใดได้บ้าง และแยกแยะได้ดีเพียงใด

ผังการแยกแยะจะอยู่ที่แท็บแรกของส่วนรายละเอียดในแต่ละโนดของต้นไม้ตัดสินใจ ดังแสดงในรูปที่ 4.31

รูปที่ 4.31 ผังการแยกแยะ

จากรูปเป็นผังการแยกแยะของคุณสมบัติ FTI ซึ่งเห็นได้ว่าคุณสมบัตินี้สามารถแยกแยะกลุ่มที่สาม (compensated hypothyroid) และกลุ่มที่ห้า (negative) ให้ไปตกทางด้านฝั่งขวา และกลุ่มที่สอง (primary hypothyroid) ให้ไปตกทางด้านฝั่งซ้าย

ค่ามาตรฐานอัตราส่วนเกน

แสดงคุณสมบัติต่างๆ ที่มีค่ามาตรฐานอัตราส่วนเกนสูงสุดห้าอันดับแรกเมื่อนำมาคำนวณเพื่อพิจารณาคัดเลือกเป็นโนดนี้ ทำให้ทราบว่านอกจากคุณสมบัติที่เป็นตัวโนดแล้ว ยังมีคุณสมบัติใดอีกบ้างที่ดีพอที่จะถูกคัดเลือกนำมาทำเป็นโนด และเพื่อความสะดวกในการเปรียบเทียบ จึงนำค่ามาตรฐานอัตราส่วนเกนทั้งห้ามาเขียนเป็นกราฟไล่จากมากไปน้อย

กฎ

แสดงกฎต่างๆ ซึ่งคือเส้นทางในต้นไม้ที่ไล่จากรากมาจนถึงโนด

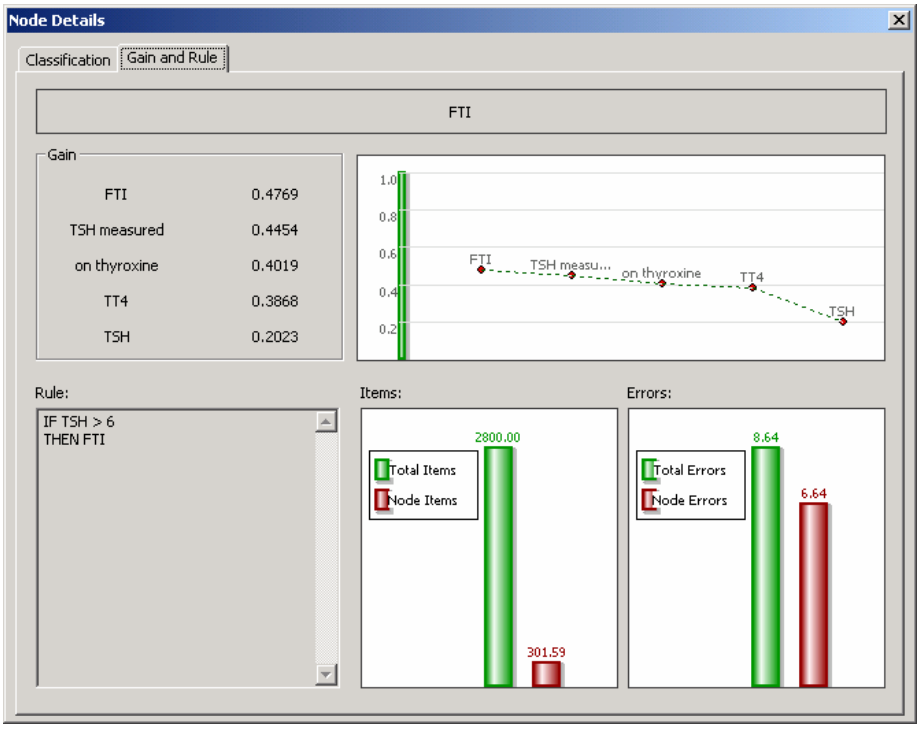
เปรียบเทียบข้อมูลที่ตกลงมายังโนด

เป็นแผนภูมิแท่งเปรียบเทียบระหว่างข้อมูลที่ตกลงมายังโนดกับข้อมูลทั้งหมดที่นำมาใช้เรียนรู้ต้นไม้ตัดสินใจ

เปรียบเทียบความผิดพลาดในการเรียนรู้ของแต่ละโนด

เป็นแผนภูมิแท่งเปรียบเทียบระหว่างความผิดพลาดในการเรียนรู้ของโนดกับความผิดพลาดในการเรียนรู้ทั้งหมดของต้นไม้ตัดสินใจ

ส่วนของค่ามาตรฐานอัตราส่วนเกน กฎ และการเปรียบเทียบข้อมูลและความผิดพลาด จะแสดงอยู่ที่แท็บที่สองของส่วนรายละเอียดในแต่ละโนดของต้นไม้ตัดสินใจ ดังรูปที่ 4.32



รูปที่ 4.32 ค่ามาตรฐานอัตราส่วนเกน กฎ และการเปรียบเทียบข้อมูลและความผิดพลาด

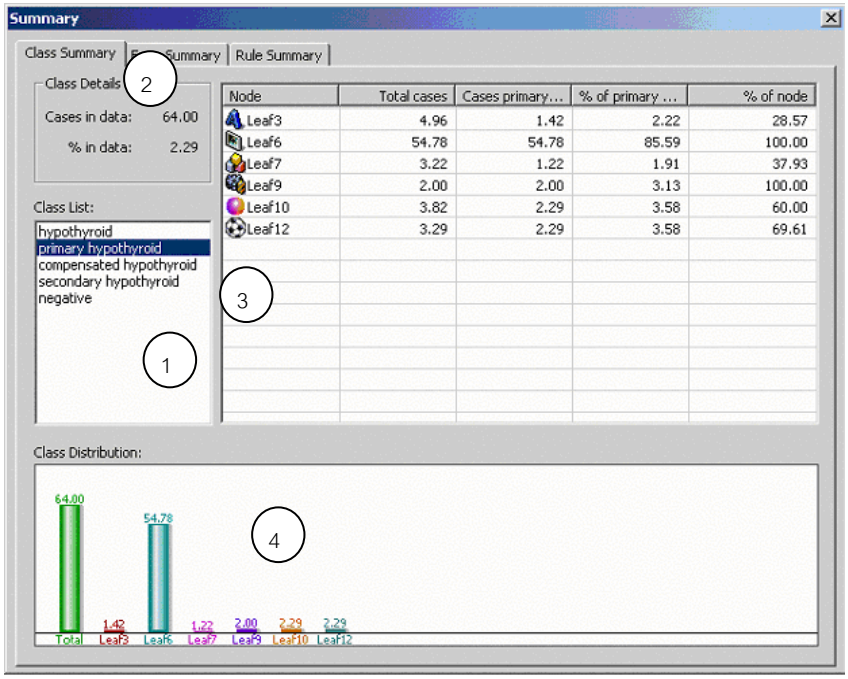
รายงานสรุปรวมของต้นไม้ตัดสินใจ

ผู้ใช้สามารถดูรายงานสรุปรวมของต้นไม้ตัดสินใจได้จากรายการ Report โดยคลิก Report ->Sumary รายงานสรุปรวมจะรวบรวมข้อมูลต่างๆ ของต้นไม้ตัดสินใจที่ได้จากการเรียนรู้มาไว้เป็นหมวดหมู่ ซึ่งมีสามหมวดหมู่ใหญ่ๆ ดังนี้

การสรุปรวมกลุ่ม

เป็นการสรุปกลุ่มทุกกลุ่มของข้อมูลว่าข้อมูลของแต่ละกลุ่มถูกแยกแยะไปที่โนดใบแต่ละโนดเป็นจำนวนเท่าใด

การสรุปรวมกลุ่มจะอยู่ที่แท็บแรกของส่วนรายงานสรุปรวมของต้นไม้ตัดสินใจ (ดูรูปที่ 4.33)



รูปที่ 4.33 การสรุปรวมกลุ่ม

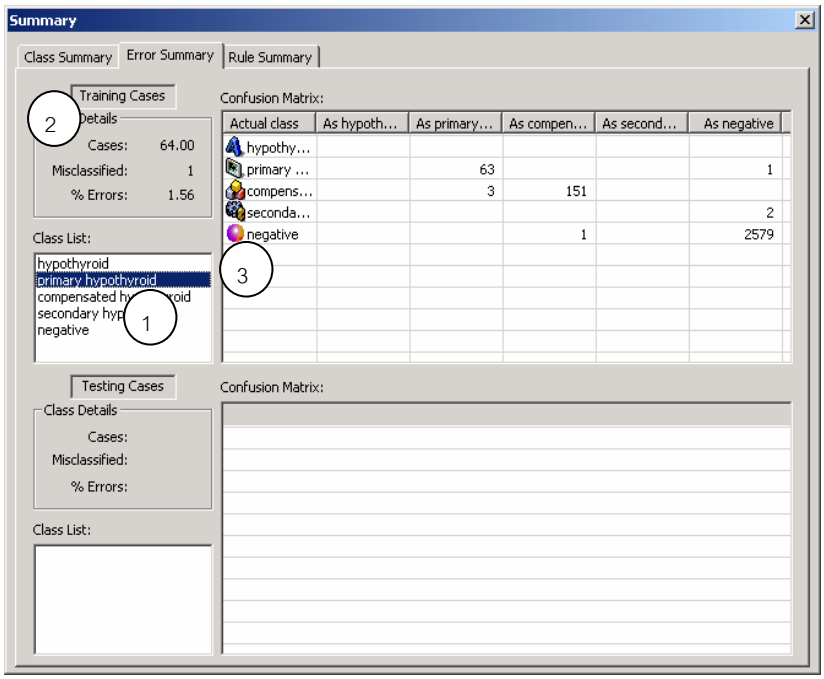
ผู้ใช้สามารถดูรายละเอียดของกลุ่มใดๆ ได้โดยคลิกที่กล่องรายการทางด้านซ้ายมือ (1) เมื่อเลือกกลุ่มในกล่องรายการแล้ว รายละเอียดต่างๆ ของกลุ่มจะปรากฏขึ้นมา ประกอบด้วย

- จำนวนข้อมูลที่มีกลุ่มที่เลือก และเปอร์เซ็นต์ของจำนวนข้อมูลที่มีกลุ่มที่เลือกต่อจำนวนข้อมูลทั้งหมด (2)
- ตารางแสดงทุกโนดใบที่ประกอบด้วยข้อมูลที่มีกลุ่มนี้อยู่ (3) ช่องแรกของตารางบอกหมายเลขของโนดใบ ช่องที่สองบอกจำนวนข้อมูลทั้งหมดที่ผ่านการแยกแยะมาสู่โนดใบนี้ ช่องที่สามบอกถึงเปอร์เซ็นต์ของจำนวนข้อมูลที่มีกลุ่มที่เลือกในโนดใบนี้ต่อจำนวนข้อมูลที่มีกลุ่มที่เลือกทั้งหมด และช่องสุดท้ายบอกถึงเปอร์เซ็นต์ของจำนวนข้อมูลที่มีกลุ่มที่เลือกในโนดใบนี้ต่อจำนวนข้อมูลทั้งหมดของโนดใบ
- แผนภูมิแท่งเปรียบเทียบจำนวนข้อมูลที่มีกลุ่มที่เลือกในโนดใบต่างๆ (4)

การสรุปรวมความผิดพลาด

เป็นการสรุปรวมความผิดพลาดของต้นไม้ตัดสินใจเมื่อนำไปแยกแยะข้อมูล แบ่งเป็นความผิดพลาดที่ได้จากการแยกแยะข้อมูลที่ใช้เรียนรู้ และความผิดพลาดที่ได้จากการแยกแยะข้อมูลที่ทดสอบ พร้อมแสดงตารางความผิดพลาด เพื่อบอกให้ทราบว่าในแต่ละกลุ่มเกิดความผิดพลาดในการแยกแยะอย่างไร

การสรุปรวมความผิดพลาดจะอยู่ที่แท็บที่สองของส่วนรายงานสรุปรวมของต้นไม้ตัดสินใจ (ดูรูปที่ 4.34)



รูปที่ 4.34 การสรุปรวมความผิดพลาด

ด้านบนของหน้าจอแสดงความผิดพลาดจากการแยกแยะข้อมูลที่ใช้ในการเรียนรู้ ขณะที่ด้านล่างของหน้าจอแสดงความผิดพลาดจากการแยกแยะข้อมูลที่ใช้ในการทดสอบ

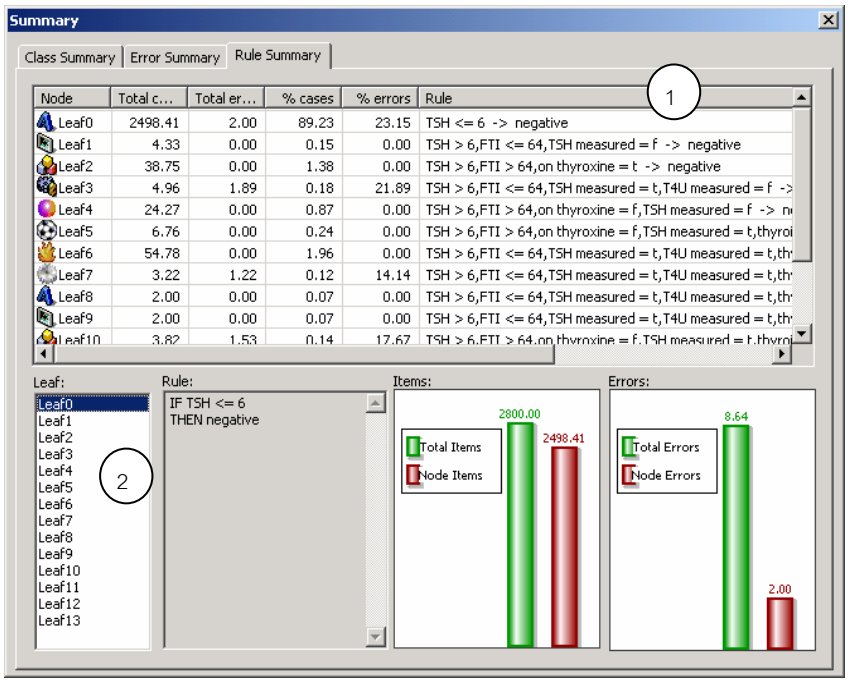
ผู้ใช้สามารถดูความผิดพลาดของกลุ่มใดๆ ได้โดยคลิกที่กล่องรายการทางด้านซ้ายมือ (1) เมื่อเลือกกลุ่มในกล่องรายการแล้ว รายละเอียดความผิดพลาดของกลุ่มจะปรากฏขึ้นมาที่ด้านบน (2) ซึ่งประกอบด้วย จำนวนข้อมูลที่มีกลุ่มที่เลือก จำนวนความผิดพลาดในการแยกแยะกลุ่มที่เลือก โดยแยกแยะกลุ่มที่เลือกผิดไปเป็นกลุ่มอื่น และเปอร์เซ็นต์ของจำนวนความผิดพลาดในการแยกแยะกลุ่มที่เลือกต่อจำนวนข้อมูลที่มีกลุ่มที่เลือกทั้งหมด

ในส่วนด้านขวา (3) เป็นตารางความผิดพลาด บอกให้รู้ว่าต้นไม้ตัดสินใจที่ได้จากการเรียนรู้ทำการแยกแยะกลุ่มผิดพลาดจากกลุ่มใดไปเป็นกลุ่มใดบ้าง

การสรุปรวมกฎ

เป็นการรวมทุกโนดใบของต้นไม้ตัดสินใจ แล้วสรุปว่าเส้นทางจากรากมาถึงโนดใบสามารถเขียนเป็นกฎได้อย่างไรบ้าง

การสรุปรวมกฎจะอยู่ที่แท็บที่สามของส่วนรายงานสรุปรวมของต้นไม้ตัดสินใจ (ดูรูปที่ 4.35)



รูปที่ 4.35 การสรุปรวมกฎ

ด้านบนของหน้าจอเป็นตารางแสดงทุกโนดใบ (1) ช่องแรกของตารางบอกหมายเลขของโนดใบ ช่องที่สองบอกจำนวนข้อมูลทั้งหมดที่ผ่านการแยกแยะมาสู่โนดใบ ช่องที่สามบอกความผิดพลาดในการเรียนรู้ของแต่ละโนดใบ ช่องที่สี่บอกถึงเปอร์เซ็นต์ของจำนวนข้อมูลที่ผ่านการแยกแยะมายังโนดใบต่อจำนวนข้อมูลทั้งหมด ช่องที่ห้าบอกถึงเปอร์เซ็นต์ของความผิดพลาดในการเรียนรู้ของโนดใบต่อความผิดพลาดในการเรียนรู้ทั้งหมด และช่องสุดท้ายแสดงกฎของโนดใบ

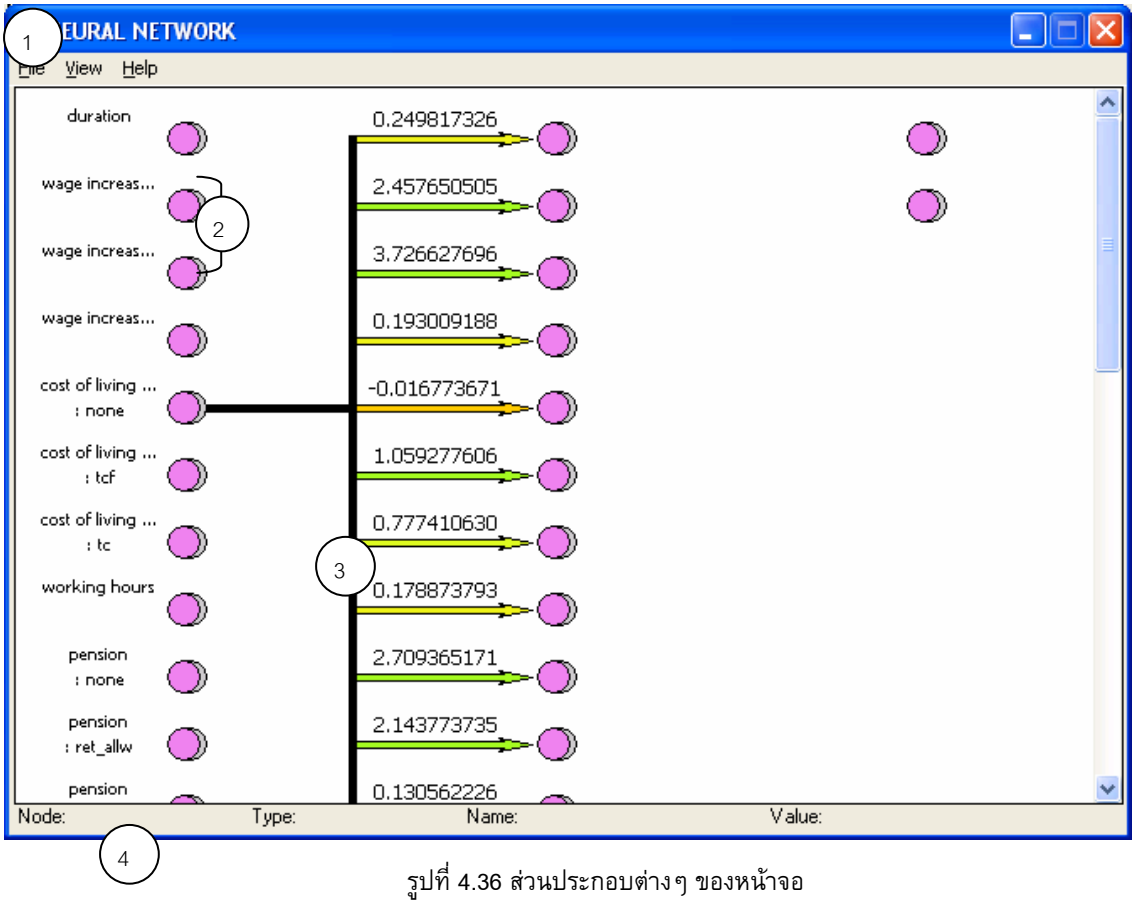
ด้านล่างซ้ายเป็นกล่องรายการแสดงทุกโนดใบ (2) เมื่อเลือกโนดใบจากกล่องรายการแล้วจำนวนข้อมูลที่ผ่านการแยกแยะมาที่โนดใบและความผิดพลาดในการเรียนรู้ของโนดใบจะถูกนำมาแสดงเป็นแผนภูมิแท่งเปรียบเทียบกับจำนวนข้อมูลทั้งหมดและความผิดพลาดในการเรียนรู้ทั้งหมดตามลำดับ รวมถึงแสดงกฎของโนดใบที่เลือก

จินตทัศน์ของนิวรอลเน็ตเวิร์ก

สำหรับการทำจินตทัศน์ของนิวรอลเน็ตเวิร์กนั้น จะมีความซับซ้อนมาก เปรียบได้กับเซลล์สมองที่ซับซ้อนของมนุษย์เราเนื่องจากเส้นเชื่อมต่างๆ ที่เกี่ยวข้องนั้นมีเป็นจำนวนมาก ทำให้ภาพที่ออกมานั้น แลดูไม่สวย ซ้ำยังอาจก่อความสับสนให้กับผู้ใช้ได้อีกด้วย เป็นสาเหตุให้ซอฟท์แวร์ทำเหมืองข้อมูลอื่นๆ ละเลยที่จะนำนิวรอลเน็ตเวิร์กผนวกเข้าอยู่ในซอฟต์แวร์ด้วย แต่เราได้เล็งเห็นถึงความสำคัญของนิวรอลเน็ตเวิร์ก ซึ่งมักจะให้ความถูกต้องที่สูงกว่าเทคนิคการทำเหมืองข้อมูลอื่นๆ จึงได้มีการออกแบบเพื่อให้ง่ายต่อการเข้าใจของผู้ใช้ที่จะเห็นได้จากเนื้อหาข้างล่างนี้

หน้าจอแสดงผล

ทำการเรียกหน้าจอจินตทัศน์โดยการกดปุ่ม Invoke Visualizer ส่วนต่าง ๆ ของหน้าจอจินตทัศน์ แสดงดังในรูปที่ 4.36



รูปที่ 4.36 ส่วนประกอบต่างๆ ของหน้าจอ

1. เมนูบาร์
2. โนดอินพุต โนดฮิดเดน และโนดเอาท์พุต
3. เส้นเชื่อม
4. รายละเอียดของค่าคุณสมบัติที่เลือก แสดงข้อมูลต่อไปนี้
    - Node ประเภทของโนด
    - Name ชื่อของค่าคุณสมบัติหรือค่ากลุ่มนั้น
    - Type ชนิดของโนดว่าเป็นค่าคุณสมบัติหรือค่ากลุ่ม
    - Value ค่าของคุณสมบัตินั้น

แสดงสีของเส้นเชื่อม

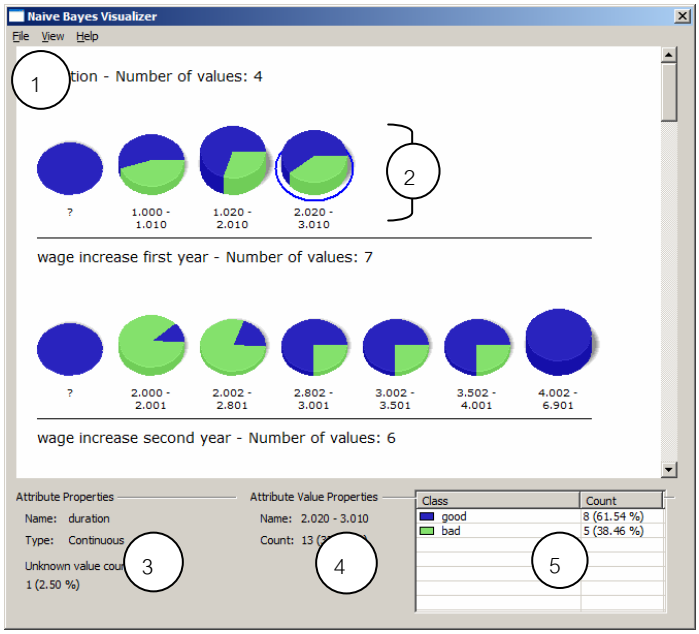
สีของเส้นเชื่อมจะบ่งบอกถึงความสำคัญของคุณสมบัตินั้น ว่ามีความสำคัญมากน้อยเพียงใด ต่อการแยกแยะกลุ่มของคำตอบ ในที่นี้แบ่งออกเป็น 3 ส่วนใหญ่ๆ คือ

- **สีแดง** หมายถึง ค่าน้ำหนักของเส้นเชื่อมนั้นมีค่าติดลบ ซึ่งจะแบ่งลึกลงไปอีกว่ามีผลต่อการแยกแยะมากน้อยเพียงใด โดยดูจากความเข้มของสี
- **สีเหลือง** หมายถึง ค่าน้ำหนักของเส้นเชื่อมนั้นแทบจะไม่มีผลต่อการแยกแยะ ค่าจะเข้าใกล้ 0
- **สีเขียว** หมายถึง ค่าน้ำหนักของเส้นเชื่อมนั้นมีค่าบวก ซึ่งจะแบ่งลึกลงไปอีกว่ามีผลต่อการแยกแยะมากน้อยเพียงใด โดยดูจากความเข้มของสีอีกเช่นเดียวกันกับสีแดง

จินตทัศน์การเรียนรู้เบย์อย่างง่าย

หน้าจอแสดงผล

ทำการเรียกหน้าจอจินตทัศน์โดยการกดปุ่ม Invoke Visualizer ส่วนต่างๆ ของหน้าจอจินตทัศน์ แสดงดังรูปที่ 4.37



รูปที่ 4.37 ส่วนประกอบต่าง ๆ ของหน้าจอ

1. ชื่อของคุณสมบัติ และจำนวนของค่าคุณสมบัติ
2. แผนภูมิแท่งรูปวงกลมของค่าคุณสมบัติต่าง ๆ วงกลมสีน้ำเงินรอบฐานของแท่งแผนภูมิแสดงค่าคุณสมบัติที่เลือกขณะนั้น
3. รายละเอียดของคุณสมบัติ แสดงข้อมูลต่อไปนี้
   - Name ชื่อของคุณสมบัติ
   - Type ชนิดของคุณสมบัติ
   - Unknown value count จำนวนข้อมูลที่ไม่ทราบค่าของคุณสมบัตินั้น
4. รายละเอียดของค่าคุณสมบัติที่เลือก แสดงข้อมูลต่อไปนี้
   - Name ชื่อของค่าคุณสมบัตินั้น
   - Count จำนวนของข้อมูลที่มีค่าคุณสมบัตินั้น
5. ListBox แสดงสัดส่วนกลุ่มต่างๆ ของค่าคุณสมบัติที่เลือก

ดูรายละเอียดของค่าคุณสมบัติ

ใช้เมาส์กดที่รูปแผนภูมิของค่าคุณสมบัติที่ต้องการจะทราบรายละเอียด จะปรากฏวงกลมล้อมที่ฐานของแผนภูมิ และรายละเอียดต่าง ๆ ในส่วนล่างของหน้าจอจินตทัศน์

แสดงชื่อกลุ่มของสีต่าง ๆ

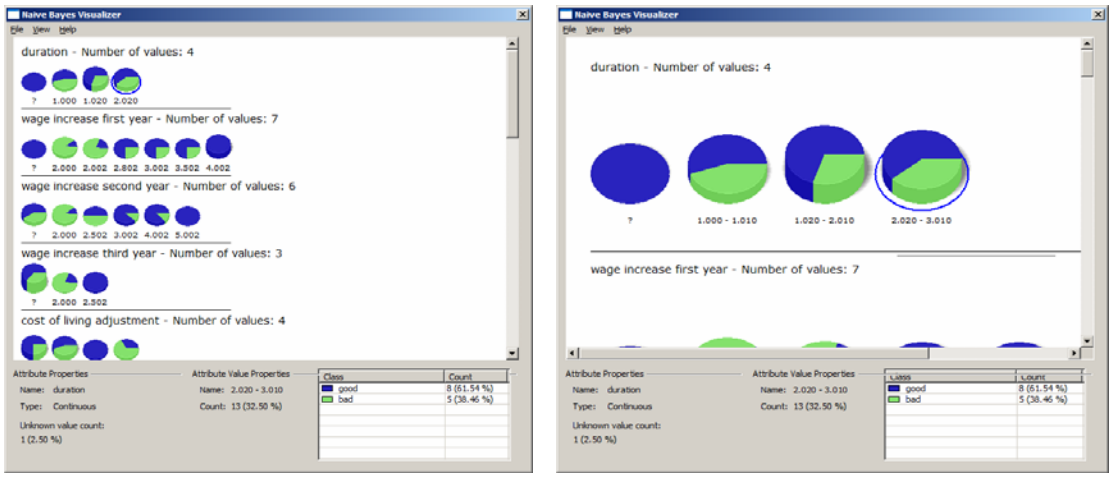
เลือกรายการ View > Legend จะปรากฏหน้าต่าง Legend ขึ้นมา ดังรูปที่ 4.38

รูปที่ 4.38 หน้าต่างแสดงชื่อกลุ่มของสีต่าง ๆ

การขยายมุมมอง

1. เลือกรายการ View > Zoom
2. เลือกขนาดที่ต้องการจาก 50% - 150%

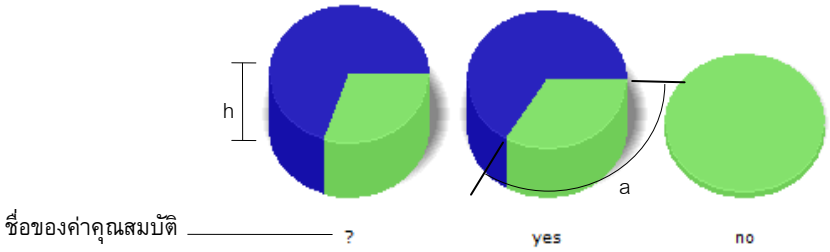
(ดูรูปที่ 4.39 ประกอบ)



รูปที่ 4.39 (ซ้าย) ขยาย 50%, (ขวา) ขยาย 150%

การแปลความหมายของแผนภูมิรูปวงกลมที่แสดง

การแปลความหมายของแผนภูมิแสดงในรูปที่ 4.40



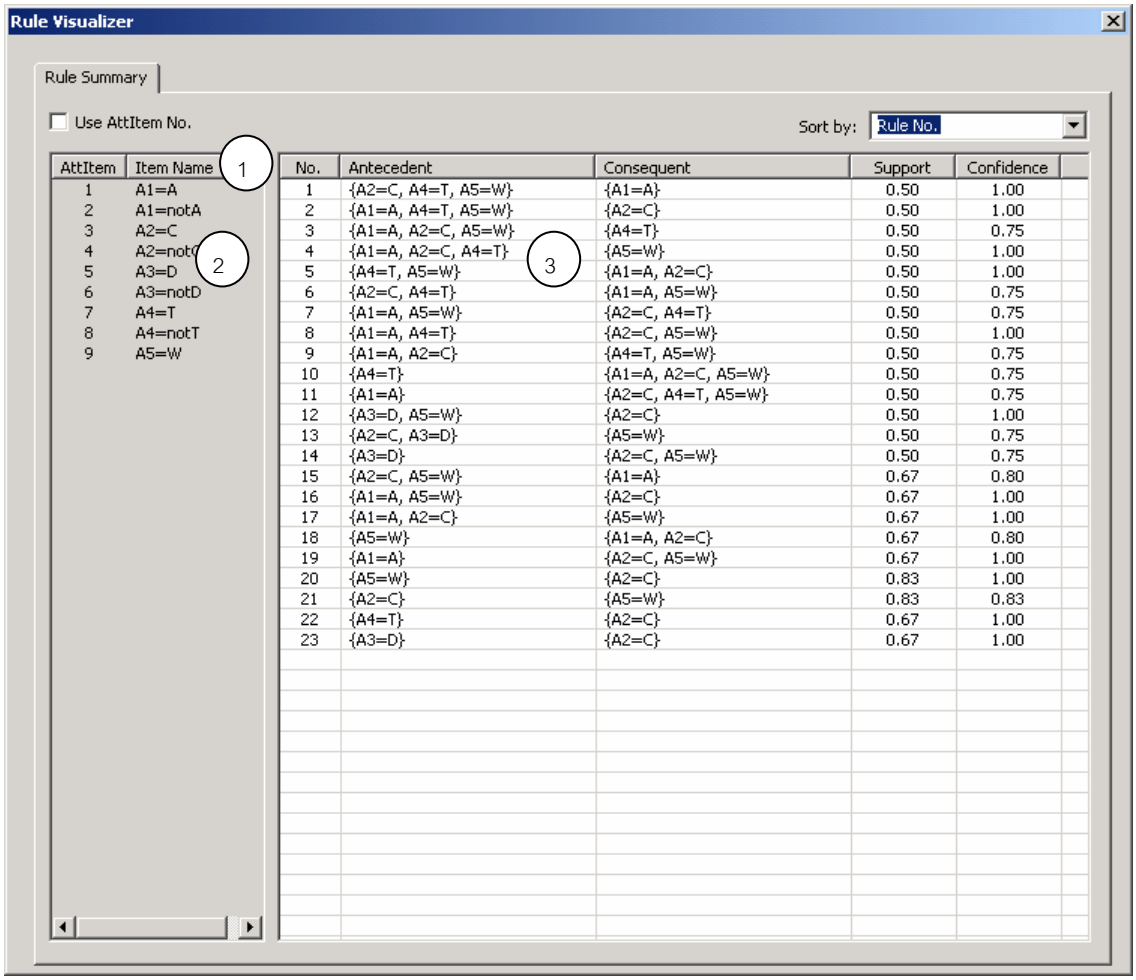
h = ความสูงของแท่งแผนภูมิเท่ากับอัตราส่วนของจำนวนค่าคุณสมบัตินั้นต่อจำนวนข้อมูลทั้งหมด

a = ขนาดของส่วนแบ่งของแผนภูมิเท่ากับเปอร์เซ็นต์ของกลุ่มนั้นจากจำนวนค่าคุณสมบัตินั้นทั้งหมด

รูปที่ 4.40 ส่วนประกอบต่าง ๆ ของแผนภูมิ

หน้าจอแสดงผล

เมื่อทำการเรียกการแสดงผลของระบบจินตทัศน์ของกฎความสัมพันธ์ จะปรากฏหน้าจอดังรูปที่ 4.41 ด้านล่างนี้



รูปที่ 4.41 ส่วนประกอบต่าง ๆ ของหน้าจอ

หน้าจอการแสดงผลของระบบจินตทัศน์ของกฎความสัมพันธ์ประกอบด้วยส่วนต่างๆ ดังนี้

1. **แถบบน** ประกอบด้วยกล่องเช็คสำหรับเลือกรูปแบบการแสดงผล และกล่องคอมโบสำหรับเรียงลำดับกฎความสัมพันธ์
2. **ตารางจับคู่ตัวเลขและชื่อของไอเท็ม** สำหรับดูชื่อของไอเท็มเมื่อเลือกรูปแบบการแสดงผลแบบย่อ
3. **ตารางแสดงกฎความสัมพันธ์** แสดงกฎความสัมพันธ์ทั้งหมดพร้อมด้วยค่าสนับสนุนและค่าความเชื่อมั่น

รูปแบบการแสดงผลกฎความสัมพันธ์

กฎความสัมพันธ์สามารถแสดงผลได้ 2 รูปแบบคือ

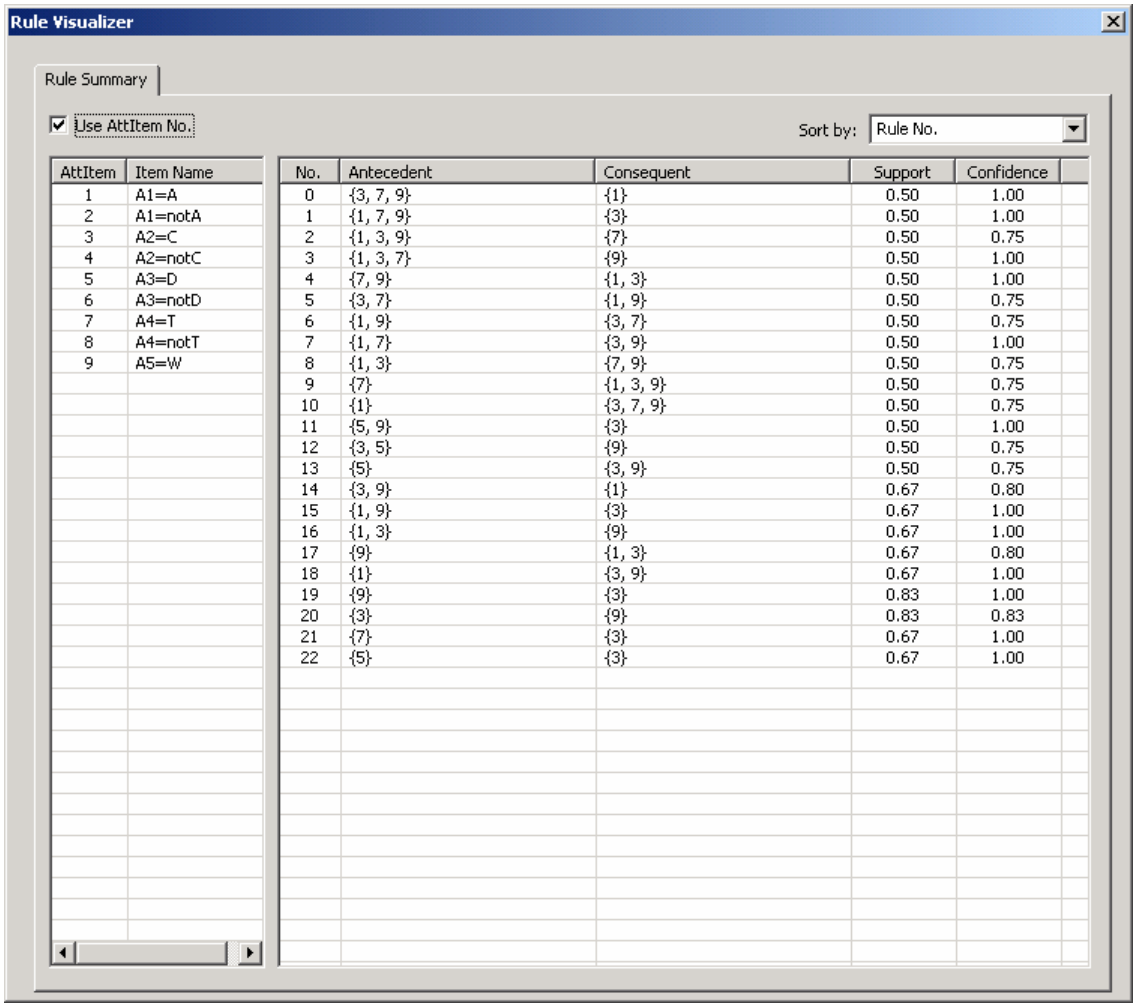
กฎความสัมพันธ์รูปแบบเต็ม

กฎความสัมพันธ์รูปแบบเต็มจะเขียนชื่อไอเท็มต่างๆ ทั้งในส่วนเหตุของกฎความสัมพันธ์และส่วนผลของกฎความสัมพันธ์

กฎความสัมพันธ์แบบเต็มจะมีการแสดงผลดังรูปที่ 4.41

กฎความสัมพันธ์รูปแบบย่อ

ผู้ใช้สามารถเลือกให้กฎความสัมพันธ์มีการแสดงผลรูปแบบย่อได้โดยคลิกที่กล่องเช็ค Use AttItem No. ดังรูปที่ 4.42 กฎความสัมพันธ์รูปแบบย่อทำให้สามารถดูกฎความสัมพันธ์ส่วนเหตุ และกฎความสัมพันธ์ส่วนผลได้ทั่วถึงขึ้น ในกรณีที่กฎความสัมพันธ์ส่วนเหตุ และกฎความสัมพันธ์ส่วนผลประกอบด้วยหลายไอเท็ม และแต่ละไอเท็มมีชื่อยาว



AttItem	Item Name
1	A1=A
2	A1=notA
3	A2=C
4	A2=notC
5	A3=D
6	A3=notD
7	A4=T
8	A4=notT
9	A5=W

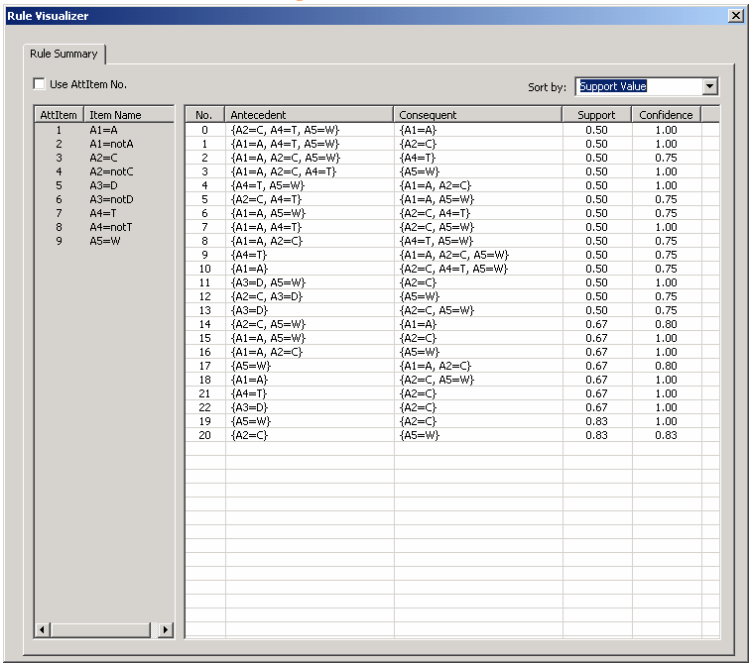
No.	Antecedent	Consequent	Support	Confidence
0	{3, 7, 9}	{1}	0.50	1.00
1	{1, 7, 9}	{3}	0.50	1.00
2	{1, 3, 9}	{7}	0.50	0.75
3	{1, 3, 7}	{9}	0.50	1.00
4	{7, 9}	{1, 3}	0.50	1.00
5	{3, 7}	{1, 9}	0.50	0.75
6	{1, 9}	{3, 7}	0.50	0.75
7	{1, 7}	{3, 9}	0.50	1.00
8	{1, 3}	{7, 9}	0.50	0.75
9	{7}	{1, 3, 9}	0.50	0.75
10	{1}	{3, 7, 9}	0.50	0.75
11	{5, 9}	{3}	0.50	1.00
12	{3, 5}	{9}	0.50	0.75
13	{5}	{3, 9}	0.50	0.75
14	{3, 9}	{1}	0.67	0.80
15	{1, 9}	{3}	0.67	1.00
16	{1, 3}	{9}	0.67	1.00
17	{9}	{1, 3}	0.67	0.80
18	{1}	{3, 9}	0.67	1.00
19	{9}	{3}	0.83	1.00
20	{3}	{9}	0.83	0.83
21	{7}	{3}	0.67	1.00
22	{5}	{3}	0.67	1.00

รูปที่ 4.42 กฎความสัมพันธ์รูปแบบย่อ

การเรียงลำดับกฎความสัมพันธ์

กฎความสัมพันธ์ที่แสดงผลอยู่ในตารางสามารถเรียงลำดับโดยใช้ค่าที่อยู่ในแต่ละกฎความสัมพันธ์ต่างๆ ดังนี้

- **หมายเลขของกฎความสัมพันธ์** ผู้ใช้สามารถเรียงลำดับกฎความสัมพันธ์ตามหมายเลขของกฎความสัมพันธ์ได้โดยเลือก Rule No. ที่กล่องคอมโบ จะทำให้กฎความสัมพันธ์ถูกเรียงตามหมายเลขของกฎความสัมพันธ์จากน้อยไปมาก
- **จำนวนไอเท็มในกฎความสัมพันธ์ส่วนที่เป็นเหตุ** ผู้ใช้สามารถเรียงลำดับกฎความสัมพันธ์ตามจำนวนไอเท็มในกฎความสัมพันธ์ส่วนที่เป็นเหตุ ได้โดยเลือก #Antecedent ที่กล่องคอมโบ จะทำให้กฎความสัมพันธ์ถูกเรียงตามจำนวนไอเท็มในกฎความสัมพันธ์ส่วนที่เป็นเหตุจากน้อยไปมาก
- **จำนวนไอเท็มในกฎความสัมพันธ์ส่วนที่เป็นผล** ผู้ใช้สามารถเรียงลำดับกฎความสัมพันธ์ตามจำนวนไอเท็มในกฎความสัมพันธ์ส่วนที่เป็นผล ได้โดยเลือก #Consequent ที่กล่องคอมโบ จะทำให้กฎความสัมพันธ์ถูกเรียงตามจำนวนไอเท็มในกฎความสัมพันธ์ส่วนที่เป็นผลจากน้อยไปมาก
- **ค่าสนับสนุน** ผู้ใช้สามารถเรียงลำดับกฎความสัมพันธ์ตามค่าสนับสนุน ได้โดยเลือก Support Value ที่กล่องคอมโบ จะทำให้กฎความสัมพันธ์ถูกเรียงค่าสนับสนุนจากน้อยไปมาก
- **ค่าความเชื่อมั่น** ผู้ใช้สามารถเรียงลำดับกฎความสัมพันธ์ตามค่าความเชื่อมั่น ได้โดยเลือก Confidence Value ที่กล่องคอมโบ จะทำให้กฎความสัมพันธ์ถูกเรียงตามค่าความเชื่อมั่นจากน้อยไปมาก ดังรูปที่ 4.43



รูปที่ 4.43 กฎความสัมพันธ์เรียงลำดับตามค่าสนับสนุน

รายละเอียดในแต่ละกฎความสัมพันธ์

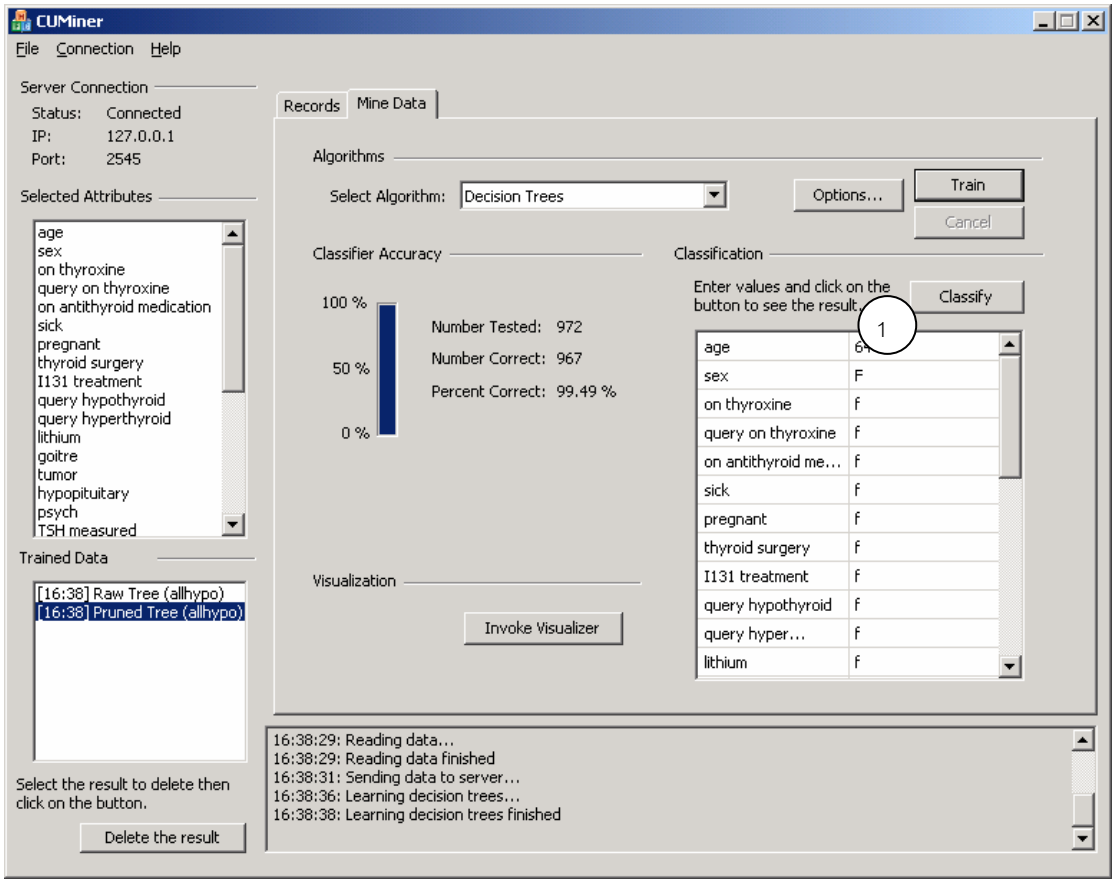
เมื่อผู้ใช้คลิกที่กฎความสัมพันธ์กฎหนึ่งในตาราง รายละเอียดของกฎความสัมพันธ์นั้นจะถูกแสดงขึ้นมาดังรูปที่ 4.44 รายละเอียดเหล่านั้นประกอบไปด้วยข้อมูลดังนี้

- ตารางแสดงกฎความสัมพันธ์พร้อมทั้งค่าสนับสนุนและค่าความเชื่อมั่น (1)
- แผนภูมิแท่งเปรียบเทียบจำนวนข้อมูลที่บรรจุเซตของไอเท็มที่ปรากฏบ่อยของกฎความสัมพันธ์ เปรียบเทียบกับจำนวนข้อมูลทั้งหมด และจำนวนข้อมูลที่บรรจุเซตของแต่ละไอเท็มของกฎความสัมพันธ์ (2)
- แผนภูมิแท่งเปรียบเทียบจำนวนข้อมูลที่บรรจุเซตของไอเท็มที่ปรากฏบ่อยของกฎความสัมพันธ์ เปรียบเทียบกับจำนวนข้อมูลทั้งหมด จำนวนข้อมูลที่บรรจุเซตของไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นเหตุ และจำนวนข้อมูลที่บรรจุเซตของไอเท็มของกฎความสัมพันธ์ส่วนที่เป็นผล (3)

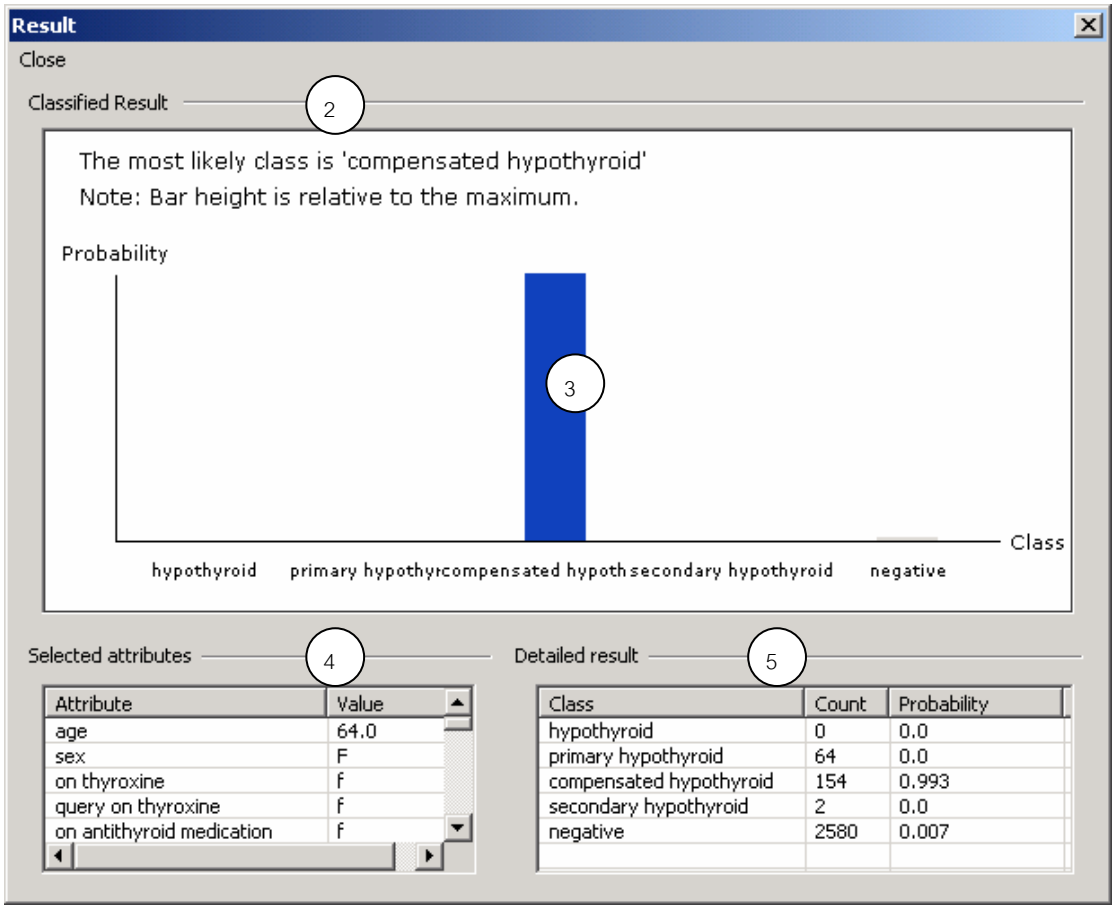
รูปที่ 4.44 หน้าจอแสดงรายละเอียดในแต่ละกฎความสัมพันธ์

การแยกแยะข้อมูล

เมื่อทำการเรียนรู้เสร็จแล้ว ผู้ใช้มีข้อมูลชุดใหม่เข้ามาและต้องการแยกแยะข้อมูลนั้น ผู้ใช้สามารถที่จะป้อนข้อมูลได้จากตารางที่หมายเลข (1) ซึ่งเมื่อผู้ใช้ทำการป้อนเสร็จแล้ว ให้กดปุ่ม Classify ดังรูปที่ 4.45 โปรแกรมจะทำการแยกแยะข้อมูลที่ป้อนเข้าไป พร้อมทั้งแสดงผลออกมา (ดูรูปที่ 4.46 ประกอบ)



รูปที่ 4.45 ตารางการแยกแยะข้อมูล



รูปที่ 4.46 ผลลัพธ์การแยกแยะข้อมูล

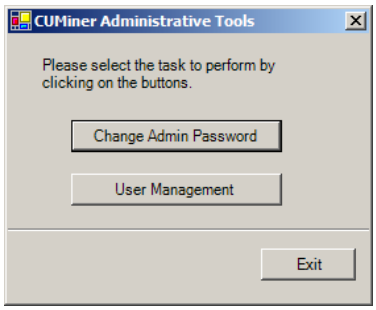
รายละเอียดเกี่ยวกับการแยกแยะข้อมูลมีดังต่อไปนี้

- หมายเลขที่ (2) แสดงคำตอบของการแยกแยะว่าข้อมูลที่ผู้ใช้ป้อนเข้ามาตกอยู่กลุ่มไหน
- หมายเลขที่ (3) แสดงกราฟแท่งของคำตอบของการแยกแยะ
- หมายเลขที่ (4) แสดงค่าคุณสมบัติที่ผู้ใช้ได้ทำการป้อนเข้าไปในตอนแรก
- หมายเลขที่ (5) แสดงค่าความน่าจะเป็นของกลุ่มทุกกลุ่มในการแยกแยะ

การจัดการผู้ใช้

CUMinerAdmin

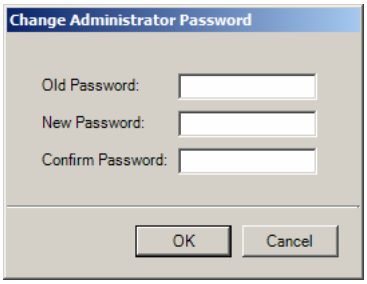
CUMinerAdmin เป็นโปรแกรมสำหรับจัดการบัญชีผู้ใช้ และเปลี่ยนรหัสผ่านของผู้ดูแลระบบ เริ่มโปรแกรมโดยการกดที่ไอคอน CUMinerAdmin ใน Start Menu จะปรากฏหน้าจอหลักของโปรแกรมดังรูปที่ 4.47



รูปที่ 4.47 หน้าจอหลักของโปรแกรม CUMinerAdmin

การเปลี่ยนรหัสผ่านของผู้ดูแลระบบ

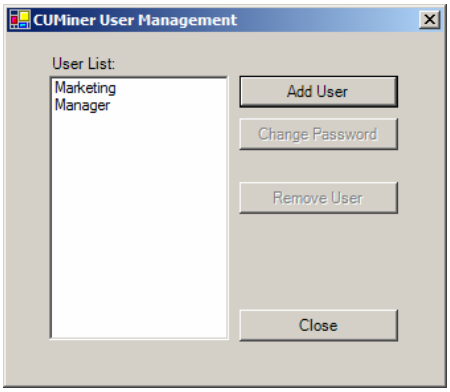
กดปุ่ม “Change Admin Password” ที่หน้าจอหลัก จะปรากฏหน้าจอเปลี่ยนรหัสผ่านผู้ดูแลดังรูปที่ 4.48



รูปที่ 4.48 หน้าจอเปลี่ยนรหัสผ่านผู้ดูแลระบบ

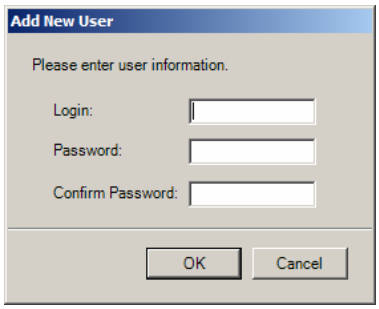
การจัดการบัญชีผู้ใช้

กดปุ่ม “User Management” ที่หน้าจอหลัก จะปรากฏหน้าจอจัดการบัญชีผู้ใช้ดังรูปที่ 4.49

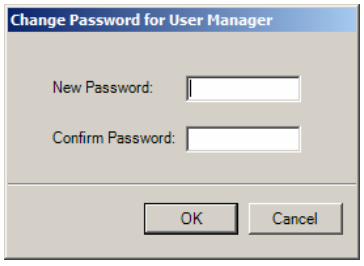


รูปที่ 4.49 หน้าจอจัดการบัญชีผู้ใช้

- เพิ่มบัญชีผู้ใช้โดยการกดปุ่ม “Add User” จะปรากฏหน้าจอดังรูปที่ 4.50
- เปลี่ยนรหัสผ่านผู้ใช้โดยเลือกชื่อผู้ใช้ที่ต้องการเปลี่ยนรหัส แล้วกดปุ่ม “Change Password” จะปรากฏหน้าจอดังรูปที่ 4.50
- ลบผู้ใช้โดยเลือกชื่อผู้ใช้ที่ต้องการลบแล้วกดปุ่ม “Remove User”



(ก)



(ข)

รูปที่ 4.50 (ก) หน้าจอเพิ่มบัญชีผู้ใช้, (ข) หน้าจอเปลี่ยนรหัสผ่านผู้ใช้

อภิธานศัพท์

Architecture	สถาปัตยกรรม
Administrator	ผู้ดูแลระบบ
Association rule	กฎความสัมพันธ์
Attribute	คุณสมบัติ
Attribute Value	ค่าคุณสมบัติ
Bias	ค่าความอคติ
Breadth-first search	การค้นหาแนวกว้าง
Class	คลาส
Classify	แยกแยะ
Client-side program	โปรแกรมทางฝั่งไคลเอนต์
Combinatorial	การจัดหมู่
Combo box	กล่องคอมโบ
Concept	แนวคิดสำคัญ
Connection	การเชื่อมต่อ
Continuous	ต่อเนื่อง
Data structure	โครงสร้างข้อมูล
Data mining	การทำเหมืองข้อมูล
Decision tree	ต้นไม้ตัดสินใจ
Depth-first search	การค้นหาแนวลึก
Dialog	กล่องโต้ตอบ
Discrete	ไม่ต่อเนื่อง
Discretize	ทำให้ข้อมูลเป็นแบบไม่ต่อเนื่อง
Gain Criterion	ค่ามาตรฐานเกน
Gain Ratio Criterion	ค่ามาตรฐานอัตราส่วนเกน
Information	สารสนเทศ
Interface	หน้าจอแสดงผล
IP address	หมายเลขไอพี
Knowledge	ความรู้
List box	กล่องรายการ
Machine learning	การเรียนรู้ของเครื่อง
Menu	รายการ
Model	แบบจำลอง
Noise	ข้อมูลรบกวน
Option	ตัวเลือก
Path	เส้นทาง
Password	รหัสผ่าน

Pattern	รูปแบบ
Prune	ตัดเล็ม
Regression	การถดถอย
Relation	ความสัมพันธ์
Screen	หน้าจอ
Server-side program	โปรแกรมทางฝั่งเซิร์ฟเวอร์
Spin control	กล่องสปิน
Stack	กองซ้อน
Thread	เทรด
Training data	ข้อมูลที่นำมาเรียนรู้
User	ผู้ใช้
User name	ชื่อผู้ใช้
User management	การจัดการผู้ใช้
Visualization	จินตทัศน์
Zoom	ขยาย

บรรณานุกรม

Agrawal, R. and Srikant, R. (1994) Fast Algorithms for Mining Association Rules. *In Proc. of the 20th VLDB Conference*, pp. 487–499, Santiago,Chile.

Becker, B., Kohavi, R. and Sommerfield, D. (1997) Visualizing the Simple Bayesian Classifier, *In Proc. of KDD 1997 Workshop on Issues in the Integration of Data Mining and Data Visualization*.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984) *Classification and Regression Trees*. Wadsworth, Belmont, CA.

Dolsak, B., Bratko, I. and Jezernik, A. (1997) Application of Machine Learning in Finite Element Computation. *In R.S. Michalski, I. Bratko, and M. Kubat (eds.) Machine Learning, Data Mining and Knowledge Discovery: Methods and Applications*, John Wiley and Sons.

Domingos, P. & Pazzani, M. (1996) Beyond Independence Conditions for the Optimality of the Simple Bayesian Classifier, *In Proc. of the 13th International Conference* of *Machine Learning*, Morgan Kaufmann, pp. 105-112.

Dougherty, J., Kohavi, R. and Sahami, M. (1995) Supervised and Unsupervised Discretization of Continuous Features, *In Proc. of the 12th International Conference Machine Learning*, Morgan Kaufmann.

Fayyad, U. M. and Irani, K. B. (1993), Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, *In Proc. of the 13th International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, pp. 1022-1027.

Holte, R. C. (1993) Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, *Machine Learning* 11, 63-90.

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

Mohammed J. Z. and Ching-Jui H. (2002) CHARM: An Efficient Algorithm for Closed Itemset Mining, *In Proc. of the 2nd SIAM International Conference on Data Mining*, Arlington.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1), 81–106.

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

Reingold E. and Tilford, J. (1981) Tidier Drawing of Trees, *IEEE Trans. Softw. Eng.*, SE-7(2):223-228.

Rumelhart, D. E., Hinton G. E. and Williams, R. J. (1986) Learning Internal Representations by Error Propagation, In *D. E. Rumelhart and J. L. McClelland (Eds.), Parallel distributed processing (Vol 1),* Cambridge, MA:MIT Press.