เหมืองข้อมูลอัจฉริยะสำหรับการแบ่งกลุ่ม และค้นหาความสัมพันธ์เชิงกฎภายในฐานข้อมูลสารสนเทศ

INTELLIGENT DATA MINER FOR INFORMATION SEGMENTATION AND KNOWLEDGE EXTRACTION

รักษิณา จันทรศิริ

สดุดี โกมลหทัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เหมืองข้อมูลอัจฉริยะสำหรับการแบ่งกลุ่ม และค้นหาความสัมพันธ์เชิงกฎภายในฐานข้อมูลสารสนเทศ

INTELLIGENT DATA MINER FOR INFORMATION SEGMENTATION AND KNOWLEDGE EXTRACTION

ผู้จัดทำ

1. นางสาวรักษิณา จันทรศิริ รหัสนักศึกษา 47010617

2. นายสดุดี โกมลหทัย รหัสนักศึกษา 47010793

(อ.ฟิน)

อาจารย์ที่ปรึกษา

(รศ. ดร. เอื้อน ปิ่นเงิน)

เหมืองข้อมูลอัจฉริยะสำหรับการแบ่งกลุ่ม และค้นหาความสัมพันธ์เชิงกฎ ภายในฐานข้อมูลสารสนเทศ

นางสาวรักษิณา จันทรศิริ 47010617
นายสดุดี โกมลหทัย 47010793
รศ. ดร. เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา
ปีการศึกษา 2550

บทคัดย่อ

แอพพลิเคชั่น ไมเนโร อินเทลิเจนเต้ เป็นแอพลิเคชั่นที่ใช้สำหรับวิเคราะห์ข้อมูลในการหากลุ่มของข้อมูล ในข้อมูล n มิติใดๆ, หาความสัมพันธ์หรือพฤติกรรมของข้อมูลในรูปแบบของกฎ และแสดงผลการวิเคราะห์ออกมา

โครงงานนี้ได้นำเสนอทางเลือกใหม่ ในการทำเหมืองข้อมูล ในส่วนของการแบ่งกลุ่มของรูปแบบข้อมูลและการค้นหาความสัมพันธ์เชิงกฎของแอทริบิวต์ของข้อมูล โดยใช้เทคนิค Neural Clustering แบบ Self organizing map สำหรับการแบ่งกลุ่ม และใช้อัลกอริทึม Apriori ในการค้นหาความสัมพันธ์เชิงกฎของแอทริบิวต์ของข้อมูล ในแง่ของการใช้สองเทคนิคร่วมกันนี้ ทำให้สามารถทำการแบ่งกลุ่มข้อมูลดิบจากแหล่งข้อมูลโดยใช้ SOM เมื่อได้กลุ่มแล้วสามารถนำข้อมูลบางกลุ่มมาหาความสัมพันธ์ หรือพฤติกรรมของข้อมูลในรูปแบบของกฎความสัมพันธ์แบบเจาะลึกลงไปอีกได้ ซึ่งผลลัพธ์ที่ได้จะมีประโยชน์อย่างมากในเชิงธุรกิจ เช่น การทำการตลาดออนไลน์, การวิเคราะห์พฤติกรรมผู้บริโภค เป็นต้น

การใช้เทคนิคร่วมกันนี้ทำให้สามารถทำการแบ่งกลุ่มข้อมูลดิบ, กรองนำข้อมูลแต่ละกลุ่มที่แบ่งแล้วมาหาความสัมพันธ์เชิงภายในแล้วแทนออกมาในรูปแบบของกฎ และนำกฎเหล่านั้นมาแสดงผลรายงาน, แสดงผลกราฟอื่นๆ ได้อย่างดี

Intelligent Data Miner for Information Segmentation and Knowledge Extraction

Ms. Raksina	Jantarasiri	47010617
Mr. Sadudee	Komolhathai	47010793
Assoc. Prof. Dr. Ouen	Pinngern	Advisor

Academic Year 2007

ABSTRACT

Minero Inteligente is the application used in exploring the clusters in the n-dimensional data set, finding the hidden association rules, and visualize them.

In this project, we introduce an alternative method for mining data using multiple techniques together combining between 'Kohonen's self organizing map' which being used in clustering and 'Apriori Algorithm' which being used in association rules discovery.

This techniques is all about obtaining consistence data set from sources, clustering it and put into groups using SOM, construct the association rules from the chosen cluster. The result will be useful mainly on business purposes e.g. e-marketing, customers' behavior extraction.

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา รศ. ดร. เอื้อน ปิ่นเงิน ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบคุณ ห้องปฏิบัติการวิจัยศาสตร์ข้อมูล (Information Science Laboratory) ที่ให้การสนับสนุนการวิจัยนี้ ขอขอบคุณพี่ๆ สมาชิกทุกท่านในห้องปฏิบัติการวิจัย ที่คอยให้คำแนะนำที่ดีแก่ข้าพเจ้ามาโดยตลอด

ขอขอบคุณ คุณเจริญ ตั้งเจริญสมุทร รองผู้อำนวยการโรงพยาบาลเมืองสมุทร ที่ได้กรุณาให้ข้อมูลเชิงธุรกิจโรงพยาบาลที่เป็นประโยชน์อย่างยิ่ง ทำให้ข้าพเจ้าเห็นมุมมองฝ่ายผู้บริหาร และยังเปิดโอกาสให้ข้าพเจ้าได้สัมผัสกับผู้ใช้งานจริง ทำให้ทราบความต้องการที่ชัดเจนของผู้ใช้

สำหรับคุณงามความดีอันใดที่เกิดจากปริญญานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

รักษิณา จันทรศิริ

สดุดี โกมลหทัย

สารบัญ

หน้า

สารบัญ (ต่อ)

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญรูป

รูปที่ **หน้า**

สารบัญรูป (ต่อ)

รูปที่ หน้า

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของโครงงาน

เนื่องจากในโลกยุคปัจจุบันเป็นยุคที่ประชากรเพิ่มจำนวนมากขึ้นอย่างรวดเร็ว เกิดความหลากหลายทางพันธุกรรม เชื้อชาติ ลักษณะนิสัย รวมไปถึงวิถีการดำเนินชีวิต เมื่อมีมนุษย์อยู่ร่วมกันเป็นจำนวนมากบนโลก การเติบโตทางเศรษฐกิจจึงทวีขึ้นเพื่อขับเคลื่อนชีวิตทั้งหลายดำเนินไป ในขณะเดียวกัน เชื้อโรคก็ยังเป็นสิ่งมีชีวิตอีกอย่างหนึ่ง ที่มีการเติบโตไปควบคู่กับมนุษย์เรา มีการกลายพันธุ์ของเชื้อโรคให้พบเห็นเป็นโรคภัยไข้เจ็บชนิดใหม่ๆ อยู่เสมอ แต่โรคภัยเก่าๆ ก็ค่อยๆ จางหายไปเช่นกัน ทำให้สังคมทุกวันนี้ล้วนมีแต่ความสับสนวุ่นวายจากเรื่องราวต่างๆ แต่เมื่อสังเกตสิ่งรอบตัวอย่างละเอียดถี่ถ้วนแล้ว จะพบว่าความยุ่งเหยิงเหล่านั้น ดำเนินไปในลักษณะซ้ำๆ กันอยู่เสมอ คือ มีรูปแบบที่จะต้องวนกลับมาซ้ำเหมือนเดิมอีก หรือที่เรียกว่า แพทเทิร์น (Pattern) ดังนั้นหากนำข้อมูลแต่อย่างรอบตัวมาพิจารณาอย่างรอบคอบ โดยใช้เทคนิคทางด้านปัญญาประดิษฐ์ (Artificial Intelligence) มาวิเคราะห์ จะทำให้มองเห็นความสัมพันธ์บางอย่างของข้อมูลที่ไม่อาจมองเห็นด้วยวิธีทางสถิติทั่วไปได้

การวิเคราะห์ข้อมูลโดยใช้อัลกอริทึมประเภทปัญญาประดิษฐ์ สามารถวิเคราะห์ข้อมูลได้หลากหลายรูปแบบ ขึ้นอยู่กับความต้องการของผู้ใช้งานว่าอยากได้การวิเคราะห์ในลักษณะใด ตัวอย่างเช่น Clustering, Classification, Association Rules Discovery เป็นต้น ซึ่งในโครงงานนี้จะนำเสนอการพัฒนาแอพพลิเคชั่น เพื่อใช้เป็นอุปกรณ์ช่วยในการวิเคราะห์ข้อมูลให้ได้ผลลัพธ์ใน 2 ประเด็น ประเด็นที่หนึ่ง คือ การแบ่งกลุ่มข้อมูลจากแพทเทิร์นข้อมูลหลากหลายมิติ (Data Clustering) ประเด็นที่สอง คือ การค้นหาความสัมพันธ์เชิงกฎที่ซ่อนอยู่ในกลุ่มข้อมูล (Association Rules Discovery) ตัวอย่างเช่น การหาความสัมพันธ์ระหว่างแอทริบิวต์ใดๆ ของชุดข้อมูล

แอพพลิเคชั่นนี้จะเป็นประโยชน์อย่างมากต่อผู้ที่ต้องการการวิเคราะห์ข้อมูลในลักษณะนี้ ตัวอย่างเช่น ธุรกิจที่พฤติกรรมของลูกค้ามีผลมากต่อผลประกอบการ เจ้าของธุรกิจสามารถใช้กระบวนการดังกล่าว หากลุ่มของลูกค้าที่มีพฤติกรรมคล้ายกัน และมีความสนใจไปในทิศทางเดียวกัน เป็นต้น

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาและนำเสนอ อัลกอริทึมที่เหมาะสมสำหรับการแบ่งกลุ่มข้อมูล

1.2.2 เพื่อศึกษาและนำเสนอ อัลกอริทึมที่เหมาะสมกับการค้นหาความสัมพันธ์เชิงกฎ

1.2.3 เพื่อศึกษาและนำเสนอ การจัดการกับเซตข้อมูลที่ไม่สมบูรณ์ให้มีมาตรฐานเพื่อนำข้อมูลไปทำการวิเคราะห์ต่อไป

1.2.4 เพื่อศึกษาและนำเสนอ การออกแบบการแสดงผลรายงานและ กราฟแสดงความสัมพันธ์ของแอทริบิวต์ใดๆ ของข้อมูลในรูปแบบต่างๆ เพื่อให้ตรงความต้องการหลักและเป็นประโยชน์ต่อผู้ใช้งานเชิงธุรกิจมากที่สุด

1.2.5 เพื่อพัฒนาแอพพลิเคชั่น ที่สามารถจัดการกับข้อมูลที่ไม่สมบูรณ์ นำเสนอรายงานและกราฟแสดงความสัมพันธ์ที่ค้นพบ แบ่งกลุ่มข้อมูลได้อย่างแม่นยำและค้นหาความสัมพันธ์เชิงกฎในข้อมูลหลายมิติใดๆ ได้

1.3 ขอบเขตของโครงงาน

การค้นคว้าข้อมูลเชิงเทคนิค ทฤษฎีที่เกี่ยวข้อง และวิธีการพัฒนาแอพพลิเคชั่นมีขอบเขตแบ่งเป็นหมวดหมู่ดังต่อไปนี้

1.3.1 Data Clustering

- สามารถนำแอพพลิเคชั่นมาช่วยวิเคราะห์เพื่อแบ่งกลุ่มข้อมูลที่มีจำนวนแอทริบิวต์มากๆ หรือมีหลายมิติ โดยทำให้มีจำนวนมิติลดลง แล้วแสดงด้วยระนาบการกระจุกตัว 2 มิติ
- สามารถบอกแนวโน้มของกลุ่มข้อมูลที่เปลี่ยนไปเพื่อแสดงให้เห็นถึงพฤติกรรมและแนวโน้มการเปลี่ยนแปลงจากอดีตถึงปัจจุบันได้
- สามารถเลือกดูข้อมูลจำเพาะของกลุ่มย่อยใดๆ ได้จากกลุ่มที่พบทั้งหมด
- สามารถแสดงโครงสร้างกลุ่มย่อยต่างๆ ที่มีการกระจุกตัวได้โดยใช้ Feature Map

1.3.2 Association Rules Discovery

- สามารถหาความสัมพันธ์เชิงกฎหรือแพทเทิร์น ระหว่างแอทริบิวต์หรือกลุ่มของแอทริบิวต์ใดๆ ได้
- สามารถบอกแนวโน้มของความสัมพันธ์เชิงกฎที่เปลี่ยนแปลงไปเพื่อแสดงให้เห็นถึงพฤติกรรมและแนวโน้มการเปลี่ยนแปลงจากอดีตถึงปัจจุบันได้

1.3.3 Data Preprocessing

- สามารถเลือกชุดข้อมูลเฉพาะส่วนที่ต้องการได้ หรือกรองข้อมูลส่วนที่ไม่ต้องการออกได้ โดยใช้ตัวกรองแบบต่างๆ รวมทั้งสุ่มข้อมูลเป็นบางอินสแตนท์เพื่อนำไปเข้าอัลกอริทึมในการวิเคราะห์ข้อมูลต่อ
- สามารถจัดการกับชุดข้อมูลที่ไม่มีคุณสมบัติ consistency ได้
- สามารถรองรับการติดต่อกับฐานข้อมูล เพื่อ Query ข้อมูลออกมาสร้างเป็นชุดข้อมูลได้

- สามารถแปลงชุดข้อมูลออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

1.3.4 Visualizing, Graphing and Reporting

- สามารถนำแอพลิเคชั่นมาช่วยวิเคราะห์ความสัมพันธ์ของข้อมูลดิบจากชุดข้อมูลที่มีแอทริบิวต์หลายตัว โดยแสดงความสัมพันธ์ของแอทริบิวต์ 2 ตัวใดๆ ในรูปแบบกราฟประเภทต่างๆ

- สามารถแสดงข้อมูลแบบจำเพาะของข้อมูลกลุ่มย่อยกลุ่มใดกลุ่มหนึ่งจากกลุ่มย่อยทั้งหมดที่ค้นพบได้ ในรูปแบบกราฟความสัมพันธ์ระหว่างแอทริบิวต์ต่างๆ

- สามารถแปลงกราฟใดๆ ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

- สามารถแปลงสารสนเทศของข้อมูลกลุ่มย่อย ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

- สามารถแปลงอินสแตนท์ที่มีแพทเทิร์นคล้ายข้อมูลกลุ่มย่อย ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

- สามารถแปลงกฎ และดึงชุดข้อมูลที่สอดคล้องกับกฎนั้นๆ ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

1.4 ขั้นตอนและวิธีการดำเนินการ

โครงงานนี้แบ่งการดำเนินงานออกเป็น 2 ระยะ คือ ระยะค้นคว้าข้อมูล สำรวจธุรกิจ และออกแบบขั้นต้น ซึ่งจะดำเนินงานในภาคเรียนที่ 1 และอีกระยะหนึ่งคือ ระยะออกแบบเชิงลึก เขียนโปรแกรม และทดสอบแอพลิเคชั่น ซึ่งดำเนินงานในภาคเรียนที่ 2 โดยรายละเอียดในแต่ละส่วน มีดังต่อไปนี้

1.4.1 ระยะค้นคว้าข้อมูล สำรวจธุรกิจ และออกแบบขั้นต้น (ภาคเรียนที่ 1)

- ศึกษาและหาข้อมูล โครงสร้างของอัลกอริทึมในการทำเหมืองข้อมูลและโมเดลของการทำเหมืองข้อมูลแบบต่างๆ ที่นิยมใช้กันในปัจจุบัน เพื่อเป็นประโยชน์ต่อการใช้งานทางธุรกิจ

- ออกสำรวจและสัมภาษณ์สถานประกอบการต่างๆ ในกลุ่มอุตสาหกรรมที่แตกต่างกันเพื่อความเข้าใจในลักษณะการทำธุรกิจและการประเมินประโยชน์ของแอพพลิเคชั่นที่จะพัฒนาจริง ทำให้สรุปความต้องการออกมาได้อย่างตรงประเด็น

- สรุปความต้องการและออกแบบโมเดลความต้องการขั้นต้นตามกระบวนการทางซอฟต์แวร์เอ็นจิเนียริ่ง

- ศึกษาและเลือกอัลกอริทึมหรือโมเดลที่มีประสิทธิภาพในการแบ่งกลุ่มข้อมูลที่มีลักษณะคล้ายกันออกเป็นข้อมูลกลุ่มย่อยต่างๆ

- ทดลองนำชุดข้อมูลมาตรฐานมาทดสอบประสิทธิภาพในการคัดแยกข้อมูลออกเป็นกลุ่มย่อย โดยเขียนโปรแกรมทดสอบ แล้วนำผลการทดลองการแบ่งข้อมูลเป็นกลุ่มย่อย มาเปรียบเทียบกับกลุ่ม

ของข้อมูลที่ถูกแบ่งไว้แล้วตามความเป็นจริง และนำเสนอการทดลองที่ได้

- เตรียมรวบรวมข้อมูล สรุปและนำเสนอ

1.4.2 ระยะออกแบบเชิงลึก เขียนโปรแกรม และทดสอบแอพพลิเคชั่น (ภาคเรียนที่ 2)

- ศึกษาและหาข้อมูล อัลกอริทึมหรือโมเดลที่มีประสิทธิภาพในการค้นหาความสัมพันธ์เชิงกฎที่ซ่อนอยู่ในชุดข้อมูล

- ทดลองนำชุดข้อมูลมาตรฐานต่างๆ และชุดข้อมูลจากฐานข้อมูลจริงของธุรกิจที่สำรวจแล้วมาทดสอบประสิทธิภาพในการค้นหาความสัมพันธ์เชิงกฎในข้อมูล โดยเขียนโปรแกรมทดสอบ แล้วนำผลลัพธ์ที่ได้มาเรียงลำดับความแม่นยำ

- ออกแบบเชิงลึกของแอพพลิเคชั่นทุกส่วน โดยเน้นไปในการนำเสนอข้อมูล (Visualization) เพื่อให้เรียบง่ายและครบตามความต้องการของผู้ใช้งาน และเขียนเอกสารแสดงแผนภาพการออกแบบทั้งหมด

- พัฒนาส่วนต่างๆ ของแอพพลิเคชั่น ได้แก่ Data Preprocess, Association Rules Discovery, Visualization

- นำส่วนของแอพพลิเคชั่นในการทำการคัดแยกกลุ่มข้อมูล (Clustering) เข้ามาประกอบกับส่วนต่างๆ ของแอพพลิเคชั่นหลัก

- ทดลองทดสอบกับเซตข้อมูลจริง

- เตรียมรวบรวมข้อมูล สรุปและนำเสนอ

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 แอพพลิเคชั่นสามารถทำงานได้กับข้อมูลจริงอย่างราบรื่น

1.5.2 แอพพลิเคชั่นสามารถเอื้อประโยชน์แก่เจ้าของธุรกิจได้ ในหลากหลายอุตสาหกรรม

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 การทำเหมืองข้อมูล (Data Mining)

การทำเหมืองข้อมูล หรือ Data Mining มีผู้ให้นิยามเอาไว้ว่า เป็นกระบวนการของการกลั่นกรองสารสนเทศ (Information) ที่ซ่อนอยู่ในฐานข้อมูลใหญ่ เพื่อนำมาวิเคราะห์ทำนายหาแนวโน้มและพฤติกรรม โดยอาศัยข้อมูลในอดีต และเพื่อใช้สารสนเทศเหล่านี้ในการสนับสนุนการตัดสินใจบางอย่าง

การทำเหมืองข้อมูลมีวิวัฒนาการเรื่อยมา ตั้งแต่ ปี ค.ศ. 1960 มีการทำ Data Collection เป็นการนำข้อมูลมาจัดเก็บอย่างเหมาะสมในอุปกรณ์ที่น่าเชื่อถือ และป้องกันการสูญหายได้เป็นอย่างดี ต่อมามีการพัฒนาเป็น Data Access ในปี ค.ศ. 1980 เป็นการนำข้อมูลที่จัดเก็บมาสร้าง ความสัมพันธ์ต่อกันในข้อมูล เพื่อประโยชน์ในการนำไปวิเคราะห์ และการตัดสินใจอย่างมีคุณภาพ จากนั้นในปี ค.ศ. 1990 มีการพัฒนาเป็น Data Warehouse & Decision Support ขึ้น สำหรับรวบรวมข้อมูลมาจัดเก็บลงไปในฐานข้อมูลขนาดใหญ่โดยครอบคลุมทุกแง่ทุกมุมขององค์กร เพื่อช่วยสนับสนุนการตัดสินใจ จนกระทั่งในปี ค.ศ. 2000 ได้เกิด Data Mining ขึ้น เป็นการนำข้อมูลจากฐานข้อมูลมาวิเคราะห์และประมวลผล โดยการสร้างแบบจำลองและความสัมพันธ์ ทางสถิติ

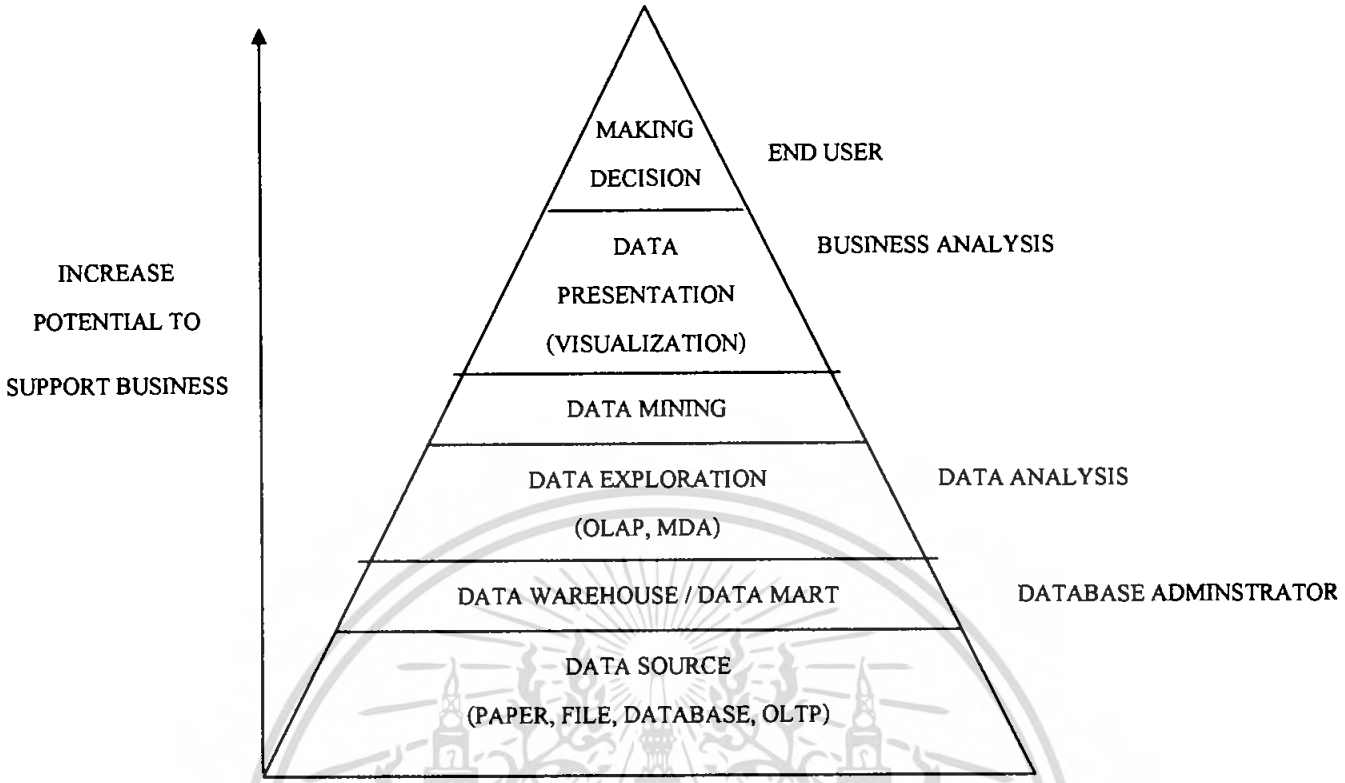
จากคำจำกัดความ Data Mining อาจมีความหมายรวมไปถึง การที่ผู้ใช้สังเคราะห์ และตรวจสอบข้อมูลอย่างละเอียด โดยการสังเคราะห์ดังกล่าวอาจจะเป็นการเรียนรู้ข้อมูลในอดีตหรือข้อมูลในปัจจุบัน ผลลัพธ์ที่ได้ออกมาต้องมีลักษณะของข้อมูลที่เป็นข้อมูลแบบ Unknown ข้อมูลแบบ Valid และข้อมูลแบบ Actionable มาจากฐานข้อมูลขนาดใหญ่ ซึ่งอาจจะมาจากรายการ Transaction ฐานข้อมูลของฝ่ายขาย E-Mail เพื่อนำข้อมูลดังกล่าวไปใช้เป็นพื้นฐานประกอบการตัดสินใจในเชิงธุรกิจ ทำให้เข้าใจแนวโน้มและรูปแบบของตลาด

ข้อมูลแบบ Unknown ข้อมูลที่ถูกใช้จะต้องเป็นข้อมูลที่ผู้ใช้งานไม่รู้มาก่อนและไม่ชัดเจนไม่สามารถตั้งสมมติฐานล่วงหน้าได้ว่าควรจะเป็นแบบใด ตัวอย่างเช่น เจ้าของห้างสรรพสินค้าแห่งหนึ่งเพิ่งจะค้นพบว่าพฤติกรรมของผู้บริโภคใหม่ที่เป็นพ่อบ้านมักจะซื้อสินค้าเบียร์และผ้าอ้อมในวันศุกร์ตอนเย็น ดังนั้นเป็นสัญญาณให้เจ้าของกิจการควรจะเตรียมสินค้าไว้เพื่อจำหน่าย ซึ่งในขณะเดียวกันห้างสรรพสินค้าคู่แข่งอาจจะไม่รู้เรื่องนี้ก็ได้ แต่ลองสังเกตดูอีกหนึ่งตัวอย่างว่า เจ้าของร้านขายรถยนต์พบว่ารถยนต์ขนาดใหญ่ราคาแพง ผู้ซื้อส่วนมากมักเป็นผู้สูงอายุ ซึ่งเจ้าของไม่รู้มาก่อนเช่นกัน แต่ข้อมูลดังกล่าวไม่เป็นลักษณะ Unknown เนื่องจากสมมติฐานดังกล่าวมีอยู่ คือผู้สูงอายุส่วนใหญ่มักมีฐานะที่ดีขึ้นเมื่อเทียบกับคนในวัยที่อายุน้อยกว่า

ข้อมูลแบบ Valid เมื่อผู้ใช้ได้เริ่มใช้เทคนิค Data Mining จะค้นพบสิ่งที่น่าสนใจตลอดเวลา แต่ว่าต้องพิจารณาด้วยว่าสิ่งนั้น Valid หรือไม่ ตัวอย่างเช่น ผู้ใช้มักจะพบว่ามีความสัมพันธ์ของการซื้อของ 2 สิ่งเสมอ เมื่อจำนวนความหลากหลายของสินค้ามีมากขึ้น แต่นั่นไม่ได้หมายความว่าจะต้องให้ห้างสรรพสินค้าจำหน่ายสินค้ามากขึ้น เพราะข้อมูลที่ได้อาจเกิดความคลาดเคลื่อน ฉะนั้นจะต้องทำการ Validation และ Checking ความถูกต้องของข้อมูลและวิเคราะห์ความถูกต้องอีกครั้งก่อน

ข้อมูลแบบ Actionable ข้อมูลจะต้องถูกแปลงออกมาและนำมาตัดสินใจให้เป็นความได้เปรียบเชิงธุรกิจ บางครั้งข้อมูลที่เราค้นพบเป็นสิ่งที่คู่แข่งได้ทำไปเสียแล้วหรือเป็นสิ่งผิดกฎหมาย ซึ่งจะต้องมีวิจารณญาณในการใช้ด้วย บางครั้งข้อมูลดังกล่าว อาจจะไม่มีประโยชน์อะไร คำว่า Data Mining นั้นมีความหมายแตกต่างกันใน 2 แง่มุม คือ ในมุมมองทางวิชาการ และในมุมมองเชิงธุรกิจ ในมุมมองเชิงวิชาการนั้น นักวิจัยจะอ้างถึงกระบวนการทั้งหมดในการทำ Data Mining ว่า “Knowledge discovery in ฐานข้อมูล” หรือ “KDD” และใช้คำว่า “Data Mining” แทนขั้นตอนหนึ่งของกระบวนการที่เกี่ยวข้องกับการค้นหารูปแบบความสัมพันธ์ของข้อมูลเท่านั้น อย่างไรก็ตาม เมื่อกล่าวถึงแง่มุมเชิงธุรกิจแล้ว จะใช้คำว่า “Data Mining” แทนความหมายของขั้นตอนทั้งหมด เดิมงานค้นคว้าทางด้าน Data Mining นั้น มีการทำการค้นคว้ากันอยู่แล้วในหลายๆ สาขาวิชา แต่มีชื่อเรียกแตกต่างกันไปในแต่ละด้าน เช่น นักวิจัยในด้านสถิติ (statistics), ฐานข้อมูล (database), Neural Networks, Pattern Recognition, Machine Learning, Econometrics และอีกหลายๆ ด้าน ต่างก็มีการค้นคว้าเกี่ยวกับปัญหาในลักษณะเดียวกันนี้ แต่ยังไม่ค่อยมีการใช้ประโยชน์ของการค้นคว้าของอีกฝ่ายหนึ่ง คือ ต่างฝ่ายต่างทำการค้นคว้าของตนเอง ไม่ค่อยมีการแลกเปลี่ยนความรู้กัน ทำให้การค้นคว้าและการเผยแพร่ผลงานดำเนินไปอย่างไม่รวดเร็วเท่าที่ควร ต่อมาจึงมีการใช้ “Data Mining” เป็นชื่อรวมของวิธีแก้ปัญหาในลักษณะนี้ ซึ่งทำให้การเผยแพร่ความรู้ในการแก้ปัญหาในลักษณะนี้ทำได้รวดเร็วและสามารถอ้างอิงได้สะดวกขึ้น

การทำเหมืองข้อมูลถือได้ว่าเป็นการนำข้อมูลไปใช้ที่ให้ประโยชน์สูงกว่า Data Warehouse และ Data Mart การทำเหมืองข้อมูลเป็นแนวคิดในการนำเอาข้อมูลมาใช้เพื่อวิเคราะห์ให้เกิดประโยชน์สูงสุด โดยเฉพาะอย่างยิ่งการตัดสินใจของฝ่ายบริหาร ซึ่งระบบนี้เป็นขั้นตอนต่อไปของ Data Warehouse ซึ่งเป็นระบบที่ทำงานโดยอัตโนมัติ สามารถตัดสินใจแทนผู้ใช้ได้ โดยอาศัยกฎเกณฑ์ต่างๆ ที่กำหนดขึ้นมา แล้วป้อนให้คอมพิวเตอร์คิด จะเห็นว่า เครื่องมือทางธุรกิจ และเทคนิคต่างๆ ที่เราใช้เพื่อสนับสนุนการตัดสินใจทางธุรกิจนั้น ล้วนมีพื้นฐานมาจากเทคโนโลยีสารสนเทศทั้งสิ้น



รูปที่ 2.1 แสดงการทำเหมืองข้อมูลและประโยชน์ทางธุรกิจ

จากรูปที่ 2.1 เริ่มต้นตั้งแต่ตารางข้อมูลธรรมดาไปจนถึงการตัดสินใจระดับสูง จะเห็นได้ว่าการทำเหมืองข้อมูล หรือ Data Mining เป็นส่วนประกอบอันใหม่ที่มีความสำคัญของเครื่องมือทางธุรกิจอย่างหนึ่ง คุณค่าของข้อมูลที่ใช้สนับสนุนการตัดสินใจจะเพิ่มขึ้นจากด้านล่างไปจนถึงด้านบนสุดของรูปปิรามิด จำนวนของข้อมูล ขนาด และระดับการตัดสินใจในข้อมูลที่ลักษณะต่างๆ กัน จึงมีระดับของผู้ตัดสินใจต่างกัน ผู้ดูแลฐานข้อมูลจะตัดสินใจบนระดับของ Data Warehouse และแหล่งข้อมูลเท่านั้น ส่วนนักวิเคราะห์ธุรกิจและผู้บริหารจะตัดสินใจบนส่วนเหนือของปิรามิด

2.1.1 เครื่องมือและเทคโนโลยีที่ใช้ทำเหมืองข้อมูล

เครื่องมือและเทคโนโลยีที่ใช้ทำเหมืองข้อมูลมีอยู่หลายประเภท ได้แก่ Neural Networks, Decision Trees, Memory Based Reasoning (MBR), Cluster Detection, Link Analysis, Genetic Algorithm, Rule Induction, K-Nearest Neighbor, Association and Sequence Detection, Logistic Regression, Discriminant Analysis, Generalized Additive Models (GAM), Multivariate Adaptive Regression Splits (MARS) เป็นต้น ซึ่งเทคโนโลยีแต่ละประเภทมีคุณสมบัติ และกระบวนการในการวิเคราะห์แตกต่างกันไป ทำให้ได้ผลลัพธ์ออกมาแตกต่างกัน ดังนั้นการเลือกใช้เทคโนโลยีแต่ละประเภทจึงขึ้นอยู่กับวัตถุประสงค์ในการวิเคราะห์ข้อมูลของแต่ละชุดข้อมูล

2.2 การเตรียมข้อมูล (Data Preparation)

การเตรียมข้อมูล หรือ Data Preparation เป็นหัวใจของขั้นตอนในการทำทั้งหมด เป็นช่วงที่ใช้เวลามากที่สุดในขั้นตอนโดยปกติแล้วต้องการเวลาประมาณ 60% ของเวลาทั้งหมดในการเตรียมข้อมูล ในขั้นตอนนี้อาจสามารถแบ่งออกได้เป็นขั้นตอนย่อยดังต่อไปนี้

2.2.1 การเลือกข้อมูล (Data Selection)

การเลือกข้อมูล หรือ Data Selection มีจุดประสงค์เพื่อระบุแหล่งของข้อมูลที่มา และดึงข้อมูลออกมาใช้สำหรับการวิเคราะห์เบื้องต้น เพื่อการเตรียมตัวสำหรับทำการ Mining ในขั้นต่อๆ ไป การเลือกข้อมูลนั้นจะแตกต่างไปตามวัตถุประสงค์ของแต่ละธุรกิจที่ได้กำหนดไว้ตั้งแต่ต้น และการเลือกข้อมูลก็ยังถูกกำหนดโดยลักษณะงานที่จะถูกนำมาใช้อีกด้วย ตัวแปรที่ถูกเลือกมาแต่ละตัวนั้นจะต้องถูกทำความเข้าใจว่าตัวแปรแต่ละตัวหมายความว่าอะไร ประกอบด้วยอะไร ไม่เพียงแต่คำจำกัดความทางธุรกิจเท่านั้น แต่จะต้องมีคำอธิบายอย่างชัดเจนเกี่ยวกับชนิดของข้อมูล ค่าที่เป็นไปได้ แหล่งกำเนิดของข้อมูล รูปแบบของข้อมูล และลักษณะอื่นๆ

ตัวแปรแบ่งเป็น 2 ชนิด ได้แก่

1. ตัวแปรแบบ Categorical
   - 1.1 Nominal Variable กล่าวถึงชนิดนี้ของ Object ที่มันอ้างถึงแต่ไม่มีลำดับในค่าที่เป็นไปได้ (Possible Value) ตัวอย่างเช่น สถานะการแต่งงาน (โสด, แต่งงาน, หย่า, ไม่ทราบ), เพศ (ชาย, หญิง), ระดับการศึกษา (ปริญญาโท, ปริญญาตรี, ม. ปลาย, ปวช) เป็นต้น
   - 1.2 Ordinal Variable มีลำดับสำหรับค่าที่เป็นไปได้ ตัวอย่างเช่น ลำดับของ ลูกค้า (ดี, ปานกลาง, ไม่ดี) เป็นต้น
2. ตัวแปรแบบ Quantitative ซึ่งมีการวัดความแตกต่างระหว่างค่าที่เป็นไปได้
   - 2.1 Continuous คือ ค่าที่มีความต่อเนื่อง ตัวอย่างเช่น รายได้, จำนวนครั้งที่ซื้อ เป็นต้น
   - 2.2 Discrete คือ ค่าที่เป็นจำนวนเต็ม ตัวอย่างเช่น จำนวนพนักงาน, ระยะเวลาปี (เดือน, ไตรมาส) เป็นต้น

ข้อมูลชุดหนึ่งๆ จะมีตัวแปรอยู่หลายตัว แต่ตัวแปรที่ถูกเลือกสำหรับทำ Data Mining นั้นถูกเรียกว่า “Active Variable” เนื่องจากตัวแปรเหล่านั้นจะถูกนำมาใช้สร้างความแตกต่างของกลุ่มย่อยต่างๆ และสามารถนำมาทำนายผลได้ ดังนั้นการเลือกชุดข้อมูลเพื่อนำมาวิเคราะห์จะต้องพิจารณาถึงอายุของข้อมูลด้วย เนื่องจากสถานการณ์ภายนอกมีการเปลี่ยนแปลงอยู่ตลอดเวลาซึ่งจะทำให้ประสิทธิภาพของการทำ Mining ลดลง ตัวอย่างเช่น รสนิยมในการใช้ชีวิต การเปลี่ยนงาน แฟชั่น เป็นต้น

2.2.2 การกลั่นกรองข้อมูล (Data Preprocessing)

การกลั่นกรองข้อมูล หรือ Data Preprocessing มีจุดประสงค์เพื่อให้มีความมั่นใจว่าคุณภาพของข้อมูลที่ถูกเลือกนั้นเหมาะสม ข้อมูลที่มีความสมบูรณ์จะเป็นเครื่องประกันได้ว่าสามารถทำ Data Mining ได้สำเร็จ ซึ่งในขั้นตอนการกลั่นกรองข้อมูลนี้เป็นขั้นตอนที่มีปัญหามากกว่า ขั้นตอนการ

เตรียมข้อมูล เนื่องจากข้อมูลส่วนใหญ่ที่มีในองค์กร ไม่ได้ถูกเตรียมมาเพื่องาน Data Mining โดยเฉพาะ ข้อมูลจะถูกนำมาจากแหล่งต่างๆ ถูกจัดเก็บไม่ดี ข้อมูลที่ถูกนำมาจากภายนอก แล้วนำมาเพื่อให้เข้ากับข้อมูลภายในที่มีอยู่ ดังนั้นปัญหาหลักของการนำข้อมูลไปใช้ คือ คุณภาพของข้อมูล และความถูกต้องของข้อมูล

ในขั้นตอนนี้ก่อนอื่นจะต้องทำการทบทวนโครงสร้างของข้อมูลใหม่ และวัดคุณภาพของข้อมูลโดยวิธีทางสถิติ หรือสุ่มตัวอย่าง

เครื่องมือที่ใช้ในการทำการกลั่นกรองข้อมูลมีดังต่อไปนี้

1. ค่าตัวแปรแบบ Categorical จะใช้การแบ่งความถี่ของค่าตัวแปร ซึ่งเป็นวิธีที่ทำให้เกิดความเข้าใจใน Data Content มากที่สุด โดยเครื่องมือทางด้านกราฟฟิกจะเป็นตัวช่วยให้เห็นภาพและกำหนดค่าที่หายไปได้
2. ค่าตัวแปรแบบ Quantitative โดยส่วนมากมักใช้ค่าต่างๆ ในการวัด ตัวอย่างเช่น ค่าสูงสุด ค่าต่ำสุด ค่าเฉลี่ย ค่ากลาง ค่ามัธยฐาน และค่าทางสถิติอื่นๆ เมื่อนำค่าเหล่านี้มาเข้าสูตรคำนวณก็จะบอกถึงค่าที่ไม่สมบูรณ์หรือค่าที่มีปัญหาได้

ระหว่างการทำขั้นตอนการกลั่นกรองข้อมูลจะเกิดปัญหาที่พบบ่อยๆ ได้แก่ Noisy Data คือ ตัวแปรตัวหนึ่งหรือมากกว่า มีค่าเกินกว่าค่าที่คาดไว้ ซึ่งอาจมีความหมายในแง่ดีหรือแง่ร้ายก็ได้ ในแง่ดี คือ ตัวแปรจะแสดงถึงโอกาสที่เรากำลังมองหาอยู่อย่างชัดเจน ส่วนในแง่ร้าย คือ ตัวแปรอาจเป็นข้อมูลที่ไม่สมบูรณ์ สาเหตุที่เกิดขึ้น เนื่องมาจากความเลินเล่อของผู้ใช้งานระบบ ตัวอย่างเช่น ผู้ใช้กรอกข้อมูลอายุคนมีค่า 300 ปี หรือกรอกค่าของรายได้เป็นจำนวนติดลบ เป็นต้น ค่าเหล่านี้ควรจะถูกแก้ไขหรือเอาออกจากการวิเคราะห์ และควรมีขั้นตอนการเช็คข้อมูลก่อนนำข้อมูลมาใช้ ส่วนค่าที่หายไป (Missing Value) คือ ค่าที่ไม่ได้แสดงในข้อมูลที่ถูกเลือก หรือค่าที่ไม่สมบูรณ์ที่ซึ่งถูกลบออกไประหว่างการทำ Noise Detection ดังนั้นค่าของข้อมูลอาจจะหายไป เนื่องจากความเลินเล่อของผู้ใช้งานระบบ เมื่อไม่มีข้อมูลเหล่านั้นระหว่างการทำ Input ข้อมูล การจัดการกับค่าที่หายไปจึงสามารถจัดการได้ด้วยเทคนิคที่แตกต่างกัน

2.2.3 การสำรวจและตรวจสอบข้อมูล (Data Exploration and Cleansing)

เมื่อทำการเก็บข้อมูลเรียบร้อยแล้ว ขั้นตอนต่อไปที่ควรปฏิบัติ คือการตรวจสอบข้อมูล เหตุที่ต้องทำการตรวจสอบข้อมูลมี 2 ข้อ ข้อแรกคือ นักวิเคราะห์ควรมีความคุ้นเคยกับตัวข้อมูล ไม่ใช่เพียงทราบชื่อของแอทริบิวต์ และความหมายเท่านั้น แต่ควรทราบเนื้อหา หรือความมุ่งหมายที่แท้จริงของข้อมูลด้วย ข้อสองคือ อาจมีความผิดพลาดของการเก็บสะสมข้อมูลเกิดขึ้น ขณะทำการรวบรวมข้อมูลจากฐานข้อมูลหลายๆ แหล่ง เข้ามาเป็นหนึ่งเดียวเพื่อใช้ในการวิเคราะห์ ซึ่งนักวิเคราะห์ที่ดีจะต้องทำการตรวจสอบข้อมูลเหล่านี้ให้ถูกต้องเสียก่อน ตัวอย่างของความผิดพลาดที่อาจเกิดขึ้นได้แก่ ความผิดพลาดในการเก็บข้อมูลจากแอทริบิวต์ที่ไม่ต้องการ ซึ่งเกิดจากความสับสนในการตั้งชื่อแอทริบิวต์นั้น (Mislabeling of field) ตัวอย่างเช่น ต้องการเก็บค่าระดับการศึกษาของผู้สมัครเข้าศึกษาต่อ ซึ่งควรจะเก็บไว้ในแอทริบิวต์ ชื่อ "LEVEL_EDU" แต่ในฐานข้อมูลบังเอิญมีแอทริบิวต์อีกตัวหนึ่ง ชื่อ

"EDUCATION" สำหรับเก็บข้อมูลระดับการศึกษาของผู้สมัครต้องการเข้าศึกษา เป็นต้น ฉะนั้น หากไม่มีการตรวจสอบความสัมพันธ์และความมุ่งหมายที่แท้จริงของแต่ละแอทริบิวต์แล้ว จะทำให้เกิดความสับสนของข้อมูลในแต่ละแอทริบิวต์ และเมื่อนำข้อมูลไปทำ Data Mining ทำให้ผลลัพธ์ที่ได้มีความผิดพลาดไปด้วย

2.2.4 การแปลงข้อมูล (Data Transformation)

ระหว่างขั้นตอนการแปลงข้อมูล ข้อมูลที่ผ่านการกลั่นกรองแล้ว จะถูกแปลงให้อยู่ในรูปแบบของข้อมูลที่พร้อมจะนำไปวิเคราะห์ นั่นคือรูปแบบของข้อมูลที่ไม่มีความขัดแย้ง ถูกจัดระเบียบมาอย่างเรียบร้อยแล้ว กลั่นกรองมาจากแหล่งข้อมูลภายนอกและภายใน ขั้นตอนนี้เป็นอีกขั้นตอนหนึ่งที่มีความสำคัญ เนื่องจากมีผลต่อความถูกต้องและความสมบูรณ์ของผลลัพธ์สุดท้าย ซึ่งขึ้นอยู่กับว่านักวิเคราะห์ข้อมูลตัดสินใจกำหนดโครงสร้างและเสนอลักษณะของ Input อย่างไร การแปลงข้อมูลยังรวมไปถึงการทำ Data Recording และ Data Format Conversion เช่น การแปลงวันที่ เป็นต้น ทางสถิติการทำการแปลงข้อมูลยังมีเทคนิคของ Data Reduction จุดประสงค์เพื่อลดตัวแปรสำหรับทำการ Process โดยการนำเอาตัวแปรตั้งแต่ 2 ตัวขึ้นไปมารวมกัน แล้วทำการ Process ข้อดี คือสามารถลดจำนวนของตัวแปรลง และยังสามารถจัดการได้ง่ายขึ้น

เทคนิคอีกอย่างหนึ่งเรียกว่า Discretization โดยการแปลงตัวแปรแบบ Quantitative ให้เป็นแบบ Categorical โดยการแบ่งค่าของตัวแปรที่จะเป็น Input ให้เป็นช่วงๆ เช่น การแปลงเงินเดือน อายุ เป็นต้น และยังมีอีกเทคนิคหนึ่งเรียกว่า One of N โดยการแปลงตัวแปรแบบ Categorical ให้เป็น Numeric ตัวอย่างเช่น การแปลงชนิดของรถยนต์ได้แก่ Ford, Lincoln, Nissan ให้เป็น 100, 010, 001 เป็นต้น โดยส่วนมากการแปลงในลักษณะดังกล่าวนี้มักจะแปลงเพื่อเป็นค่า Input ของกระบวนการ Neural Network

2.2.5 การปรับแต่งข้อมูล (Data Engineering)

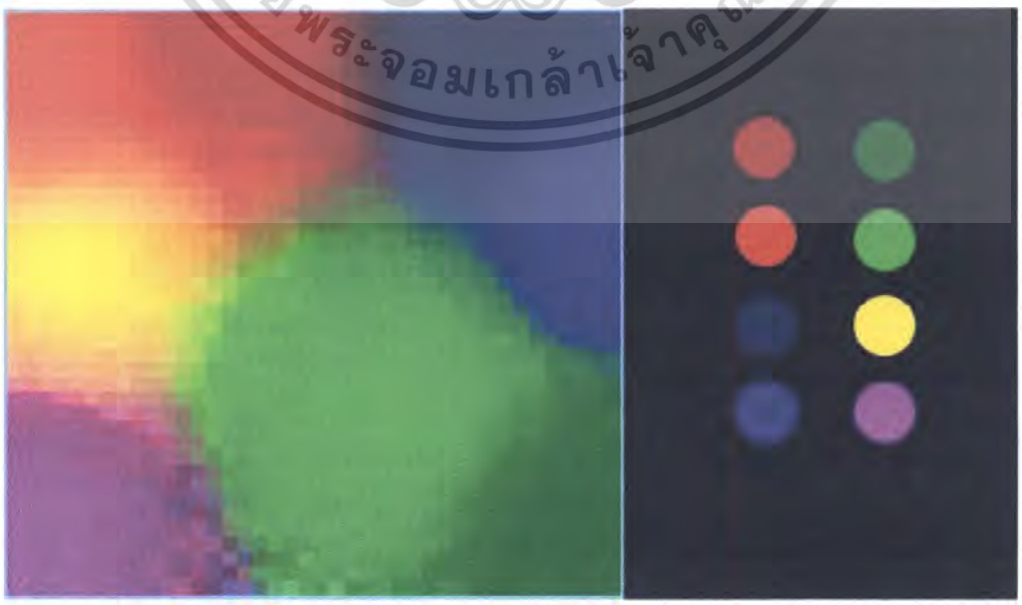
ขั้นตอนในช่วงที่กล่าวมาแล้วจัดอยู่ในส่วนของการสร้าง และการตรวจสอบความถูกต้องของข้อมูลที่จะนำไปวิเคราะห์ ส่วนในขั้นตอนนี้ สิ่งที่จะต้องทำ คือ การปรับแต่งฐานข้อมูล ซึ่งในขั้นตอนนี้จะมีปัญหาหลักๆ ที่สำคัญอยู่ 3 ข้อด้วยกัน ได้แก่ ข้อที่หนึ่ง ฐานข้อมูลที่ได้มาอาจมีแอท ทริบิวจำนวนมากที่ไม่สามารถใช้ประโยชน์ได้ แต่ไม่ได้ถูกตัดทิ้งไป จึงทำให้การเลือกกลุ่มของ แอทริบิวต์ที่สำคัญเกิดความผิดพลาดไป ข้อที่สอง ฐานข้อมูลที่ได้มาอาจมีจำนวนเรคคอดมากเกินกว่าที่จะสามารถนำมาวิเคราะห์ให้จบได้ในเวลาที่เหมาะสม ซึ่งในกรณีนี้จะต้องใช้วิธีการสุ่มข้อมูลตัวอย่างขึ้นมาแทน ข้อที่สาม ข้อมูลบางชนิดอาจเหมาะสมสำหรับการวิเคราะห์แบบเฉพาะเจาะจง เพื่อให้เกิดประโยชน์สูงสุด ดังนั้น การทำ Data Engineering จึงมีการทำซ้ำขึ้นมาหลายๆ ครั้ง เพื่อทดสอบการใช้แอทริบิวต์ที่แตกต่างกัน หรือขนาดของกลุ่มตัวอย่างที่แตกต่างกัน เช่น การทำนายอนาคตเมื่อเวลาผ่านไป 1, 2, 3, หรือ 4 เดือน สามารถทำนายได้โดยใช้เพียงแอทริบิวต์เป็นตัวทำนาย หรืออาจใช้ข้อมูลทุกอย่างที่มีเป็นตัวทำนายเลยก็ได้ เป็นต้น

2.3 อัลกอริทึมที่ใช้ในการคัดแยกกลุ่มข้อมูล (Clustering)

ในการคัดแยกกลุ่มข้อมูลออกจากชุดข้อมูล หรือ Data Clustering จะเลือกใช้อัลกอริทึม Self-Organizing Feature Maps Neural Network หรือ SOM ซึ่งคิดค้นโดย Teuvo Kohonen ศาสตราจารย์ของมหาวิทยาลัยในประเทศฟินแลนด์ จุดประสงค์เพื่อแสดงผลข้อมูลที่มีหลายมิติ (Multidimensional Data) ให้อยู่ในรูปที่เข้าใจง่ายขึ้นโดยการลดจำนวนมิติให้เหลือน้อยลง โดยกระบวนการนี้จะใช้เทคนิคการบีบอัดข้อมูล ที่มีชื่อเรียกว่า "Vector Quantisation" ซึ่งเป็นที่รู้จักกันอย่างกว้างขวางในชื่อ "โคโฮเนนเทคนิค (Kohonen Technique)" ลักษณะการทำงานจะเป็นการสร้างเครือข่ายนิวรอน (Neural Networks) ที่สามารถเรียนรู้และเก็บข้อมูลเชิงความสัมพันธ์ทางโครงสร้างได้อย่างสมบูรณ์ในขณะที่ชุดข้อมูลทดสอบ (Training Set) ยังคงอยู่

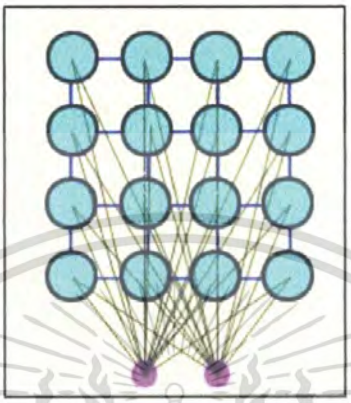
2.3.1 ลักษณะพื้นฐานของอัลกอริทึม SOM

ตัวอย่างอย่างง่ายเพื่อให้เข้าใจการทำงานพื้นฐานของ SOM คือการเรียนรู้สีต่างๆ แต่ละสี จากส่วนประกอบของสีทั้ง 3 สี ได้แก่ สีแดง สีเขียว และสีน้ำเงิน หรือ ค่า RGB เปรียบเทียบได้กับการรับอินพุต 3 มิติ แล้วลดรูปลงให้เหลือ 2 มิติ เพื่อแสดงเป็นแผนภาพในระนาบ 2 มิติ ดังแสดงในรูปที่ 2.2 ให้ SOM เรียนรู้สีที่แตกต่างกัน 8 สีในรูปขวา ซึ่งแต่ละสีจะส่งค่า RGB ซึ่งเป็นแม่สีหลัก ไปเป็นอินพุต กล่าวคือ ส่งค่า สีแดง สีเขียว และสีน้ำเงิน ในลักษณะเวกเตอร์ 3 มิติ ไปยังชั้นอินพุตของ SOM โดยแต่ละมิติจะมีค่าแม่สีหลักอยู่ 1 สี จากนั้น โคโฮเนนเน็ทเวิร์คจะถูกกระตุ้นให้ทำงาน และเรียนรู้การแทนข้อมูลเหล่านั้นลงในระนาบ 2 มิติ ดังรูปที่ 2.2 ทางซ้าย เป็นผลลัพธ์จากการให้ SOM จำสีต่างๆ จะเห็นว่าแทนที่จะแบ่งกลุ่มของสีให้อยู่ในขอบเขตที่แยกออกจากกันชัดเจน แต่ SOM จะทำการแสดงกลุ่มของสีที่มีค่า RGB ใกล้เคียงกันอยู่ติดกันเสมอ ทำให้ Kohonen Feature Map แสดงผลการคัดแยกออกมาในลักษณะคล้ายการไล่เฉดสี



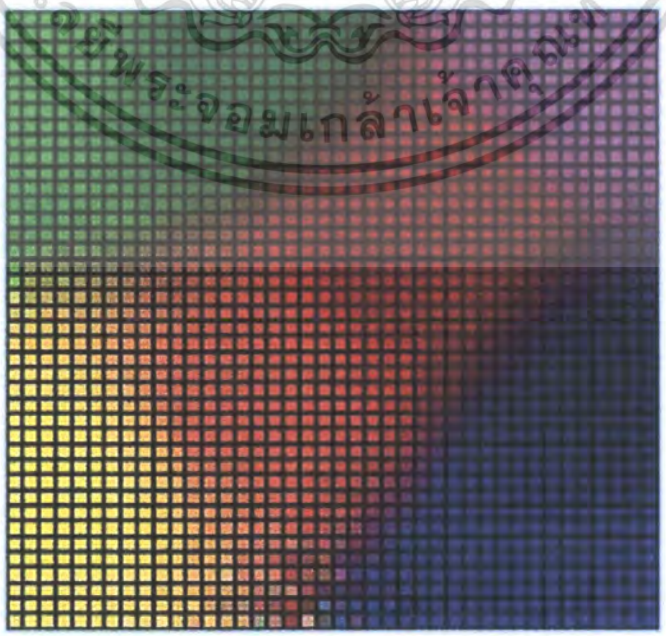
รูปที่ 2.2 แสดงผลการคัดแยกสี (ขวา) และสีที่ต้องคัดแยก (ซ้าย)

สิ่งหนึ่งที่เป็นที่น่าสนใจสำหรับเทคนิค SOM คือ โครงข่ายสามารถเรียนรู้การแบ่งกลุ่มข้อมูลได้ แม้ไม่มีการบอกล่วงหน้ามาก่อนว่าชุดข้อมูลนั้นมีทั้งหมดกี่กลุ่มย่อย ซึ่งแตกต่างจากการทำงานของอัลกอริทึมแบบอื่น เช่น Backpropagation Neural Network ที่จะต้องกำหนดเวกเตอร์เป้าหมายและจำนวนกลุ่มข้อมูลก่อนการทำการเรียนรู้



รูปที่ 2.3 แสดงโครงข่ายทั่วไปของ SOM

จากรูปที่ 2.3 แสดงโครงข่ายทั่วไปของ SOM แทนชั้นของ SOM ด้วยโหนดสีน้ำเงิน 16 โหนด ซึ่งถูกกำหนดตำแหน่งไว้ตายตัว แทนชั้นของอินพุตด้วยโหนดสีชมพู ทุกโหนดของชั้นอินพุตจะมีเส้นเชื่อมกับทุกโหนดของชั้น SOM แต่ละเส้นการเชื่อมต่อจะมีค่าน้ำหนักกำกับอยู่ และจำนวนโหนดในชั้นอินพุตจะมีค่าเท่ากับจำนวนมิติของชุดข้อมูลที่นำมาคัดแยกกลุ่ม กล่าวคือ แต่ละโหนดในชั้น SOM จะเก็บค่าน้ำหนัก ในรูปเวกเตอร์ w ของทุกโหนดในชั้นอินพุต

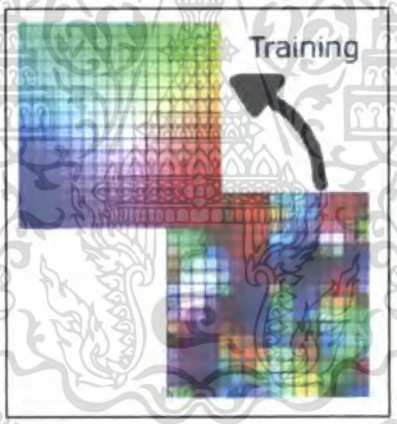


รูปที่ 2.4 แสดงโครงข่าย SOM ขนาด 40 x 40

รูปที่ 2.4 แสดงโครงข่ายขนาด 40 x 40 หลังจากให้โครงข่าย SOM เรียนรู้สีทั้งหมดแล้ว เมื่อพิจารณารูปที่ 2.4 โครงข่าย SOM มีการกำหนดขนาดเท่ากับ 40 x 40 โหนด ในแต่ละโหนดจะเก็บค่าเวกเตอร์ของน้ำหนัก w อยู่ 3 มิติ เพื่อรับค่า RGB ได้แก่ สีแดง สีเขียว และสีน้ำเงิน จากโหนดในชั้นอินพุต

2.3.2 การเรียนรู้ของอัลกอริทึม SOM

ดังที่กล่าวมาแล้วว่า SOM ไม่ต้องการการกำหนดเวกเตอร์ของผลลัพธ์ล่วงหน้า หมายความว่า SOM ไม่จำเป็นต้องรู้จำนวนกลุ่มข้อมูลที่จะแบ่ง SOM ใช้การเปรียบเทียบเวกเตอร์ของอินพุตกับเวกเตอร์ของ w ในแต่ละโหนดในชั้น SOM แล้วหาโหนดที่มีค่าความแตกต่างน้อยที่สุด ส่งผลให้พื้นที่รอบข้างโหนดนั้น เป็นพื้นที่ที่มีแพทเทิร์นของข้อมูลคล้ายกับเวกเตอร์ของโหนดในชั้นอินพุตมากที่สุด จากนั้น SOM จะปรับค่าเวกเตอร์ w ของโหนดในพื้นที่นั้น ให้มีลักษณะใกล้เคียงกับเวกเตอร์ของโหนดในชั้นอินพุตมากขึ้น จนในที่สุด จะเห็นว่ามีหลายพื้นที่ที่จับกันเป็นกลุ่มก้อน ดังในรูป 2.4 จะเห็นได้อย่างชัดเจน และเมื่อมีอินพุตทดสอบใดๆ ผ่านเข้ามาในโครงข่าย SOM จะสามารถบอกได้ว่าอินพุตนั้นอยู่ในพื้นที่ส่วนใด หรืออยู่ในกลุ่มข้อมูลใด นั่นเอง



รูปที่ 2.5 แสดงผลที่ได้จากการเรียนรู้ด้วย SOM

สำหรับอัลกอริทึมในการเรียนรู้ มี 6 ขั้นตอนดังนี้

1. ทำการกำหนดค่าน้ำหนักเริ่มต้นให้แต่ละโหนดในชั้น SOM
2. เลือกข้อมูลเพื่อเรียนรู้ 1 อินสแตนท์จากชุดข้อมูล ด้วยวิธีการสุ่ม
3. คำนวณความแตกต่างระหว่างค่าเวกเตอร์อินพุต กับ ค่าเวกเตอร์ w ของทุกโหนด โดยใช้ Euclidean Distance Function พร้อมหาโหนดที่มีค่าระยะห่าง (Distance) น้อยที่สุด โหนดนั้นถูกเรียกว่า Best Matching Unit หรือ BMU
4. คำนวณรัศมีรอบข้างโหนด BMU ในโครงข่าย เพื่อหาโหนดที่ประชิดกับ BMU โดยรัศมีในการเรียนรู้รอบแรกๆ จะมีค่ามาก และค่าจะลดลงไปเรื่อยๆ ตามฟังก์ชันเวลาที่กำหนด โหนดใดที่อยู่ในวงรัศมีของ BMU จะเรียกว่า Neighbourhood

5. ในแต่ละโหนด Neighbourhood จะมีการปรับค่าเวกเตอร์ w ด้วย เพื่อให้โหนดเหล่านั้นมีค่าคล้ายเวกเตอร์อินพุตมากขึ้น โหนด Neighbourhood ที่อยู่ใกล้กับ BMU มากกว่าก็จะถูกปรับ w ด้วยค่ามากกว่า
6. ทำซ้ำข้อ 2-6 จำนวน n รอบ

ต่อไปเป็นรายละเอียดของอัลกอริทึมในการเรียนรู้ของ SOM แต่ละขั้นตอน ก่อนอื่นจะต้องมีการสร้างโหนดขึ้นมาก่อน จากนั้นก่อนการเรียนรู้ จะต้องกำหนดค่าเริ่มต้นให้กับเวกเตอร์ w โดยวิธีการสุ่มค่าน้ำหนักให้กับเวกเตอร์ w ตาม Source Code แสดง Class ของการสร้างโหนด

```
class CNode
{
private:
  //this node's weights
  vector<double>      m_dWeights;

  //its position within the lattice
  double              m_dX,
                      m_dY;


  //the edges of this node's cell. Each node, when draw to the client
  //area, is represented as a rectangular cell. The colour of the cell
  //is set to the RGB value its weights represent.
  int                 m_iLeft;
  int                 m_iTop;
  int                 m_iRight;
  int                 m_iBottom;

public:
  CNode(int lft, int rgt, int top, int bot,
        int NumWeights):m_iLeft(lft),
                        m_iRight(rgt),
                        m_iBottom(bot),
                        m_iTop(top)
  {
    //initialize the weights to small random variables
    for (int w=0; w<NumWeights; ++w)
    {
      m_dWeights.push_back(RandFloat());
    }

    //calculate the node's center
    m_dX = m_iLeft + (double)(m_iRight - m_iLeft)/2;
    m_dY = m_iTop  + (double)(m_iBottom - m_iTop)/2;
  }
   ...
};
```

รูปที่ 2.6 Source Code แสดง Class ของการสร้างโหนด

```
double CNode::GetDistance(const vector<double> &InputVector)
{
  double distance = 0;

  for (int i=0; i<m_dWeights.size(); ++i)
  {
    distance += (InputVector[i] - m_dWeights[i])
                  * (InputVector[i] - m_dWeights[i]);
  }

  return sqrt(distance);
}
```

รูปที่ 2.7 Source Code แสดง Class ของการหาระยะห่าง Euclidean

หลังจากนำชุดข้อมูลทดสอบที่สุ่มขึ้นมาส่งเข้าไปยังโหนดในชั้นอินพุตแล้ว กระบวนการหา BMU ดังที่กล่าวไปแล้ว จะใช้ Euclidean Distance Function เป็นฟังก์ชันที่ใช้ในการหาระยะห่างระหว่างค่าเวกเตอร์ w และค่าเวกเตอร์อินพุตปัจจุบัน โหนดที่มีค่าเวกเตอร์ w ใกล้เคียงกับค่าเวกเตอร์อินพุตมากที่สุด จะเป็น BMU ในการเรียนรู้รอบนั้นๆ ซึ่งสามารถคำนวณหาระยะห่างระหว่างค่าเวกเตอร์ w และค่าเวกเตอร์อินพุตได้จากสมการที่ 2.1 สมการ Euclidean Distance ดังต่อไปนี้

$$Dist = \sqrt{\Sigma (V_i - W_i)^2} \qquad (2.1)$$

จากสมการ Euclidean Distance

Dist แทนระยะห่างระหว่างค่าเวกเตอร์ w และค่าเวกเตอร์อินพุต

V_i แทนค่าเวกเตอร์อินพุตปัจจุบันที่ถูกป้อนเข้ามาทางโหนดในชั้นอินพุต

W_i แทนค่าเวกเตอร์ w ของโหนดในชั้น SOM

สมการ Euclidean Distance สามารถแปลงเป็นโปรแกรมได้ตาม Source code แสดง Class ของการหาระยะห่าง Euclidean

ตัวอย่างการคำนวณระยะห่าง Euclidean ระหว่างค่าเวกเตอร์ w และค่าเวกเตอร์อินพุต ดังนี้ ให้เวกเตอร์อินพุตแทนค่า RGB เป็นสีแดง คือ (1, 0, 0) ส่วนค่าเวกเตอร์ w ของโหนดในชั้น SOM มีค่าน้ำหนัก คือ (0.1, 0.4, 0.5) สามารถคำนวณหาระยะห่างได้ ดังนี้

Distance = $\text{sqrt}((1 - 0.1)^2 + (0 - 0.4)^2 + (0 - 0.5)^2)$

= $\text{sqrt}((0.9)^2 + (0.4)^2 + (0.5)^2)$

= $\text{sqrt}(0.81 + 0.16 + 0.25)$

= $\text{sqrt}(1.22)$

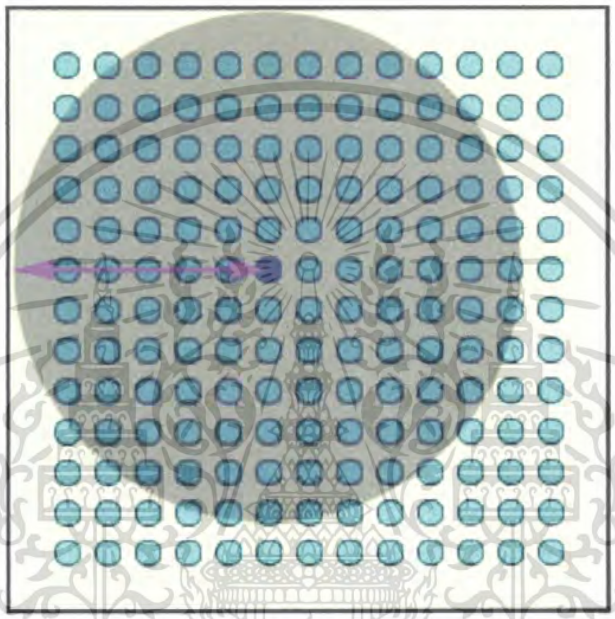
Distance = 1.106

ได้ค่าระยะห่าง Euclidean ระหว่างค่าเวกเตอร์ w และค่าเวกเตอร์อินพุต มีค่า 1.106

2.3.3 การหารัศมีของโหนดใกล้เคียง (BMU's Neighbourhood)

ในแต่ละรอบการเรียนรู้ เมื่อหาโหนดที่เป็น BMU ได้แล้ว ต่อไปจะต้องหาโหนดอยู่ในขอบเขตใกล้เคียงกับ BMU (BMU's Neighbourhood) เพื่อให้โหนดใกล้เคียงเหล่านั้นได้ปรับค่าเวกเตอร์ w ด้วย

ในขั้นตอนแรกจะต้องคำนวณหารัศมีที่โอบล้อมโหนดใกล้เคียง BMU โดยรัศมีจะกวาดพื้นที่รอบโหนด BMU ในระนาบวงกลม



รูปที่ 2.8 แสดงรัศมีของพื้นที่ที่ใกล้เคียงกับโหนด BMU

จากรูปที่ 2.8 แสดงรัศมีของพื้นที่ที่ใกล้เคียงกับโหนด BMU โดยแทนโหนด BMU ด้วยโหนดสีน้ำเงิน และพื้นที่ใกล้เคียง คือ ส่วนที่เป็นสีเทา ซึ่งมีรัศมีจากโหนด BMU กว้างเท่ากับลูกศรสีชมพู จะเห็นโหนดใกล้เคียง (BMU's Neighbour) คือ โหนดสีฟ้าที่อยู่ภายในพื้นที่สีเทา

คุณสมบัติที่ดีของอัลกอริทึมการเรียนรู้แบบโคโฮเนนนี้ คือ รัศมีที่บ่งบอกพื้นที่ของโหนดใกล้เคียงนั้นจะต้องลดลงเมื่อเวลาผ่านไป โดยใช้สูตรการคำนวณหารัศมี ณ เวลา t ใดๆ ได้จากสมการที่ 2.2 สมการหารัศมี ณ เวลา t ใดๆ ดังต่อไปนี้

$$\sigma(t) = \sigma_0 \exp(-t / \lambda) \quad ; t = 1, 2, 3, \ldots \tag{2.2}$$

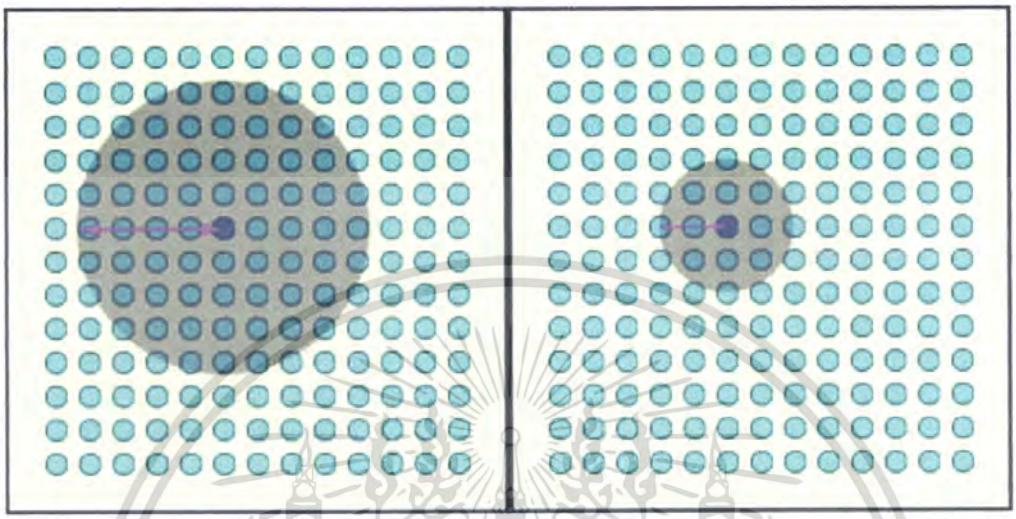
จากสมการหารัศมี ณ เวลา t

σ_0 แทนความยาวของรัศมีที่เวลา t = 0

λ แทนค่าคงที่เวลา (Time Constant)

σ (t) แทนความยาวของรัศมีที่เวลา t ใดๆ

เมื่อสร้างโปรแกรมการคำนวณตามสมการนี้ ความยาวของรัศมีจะค่อยๆ ปรับลดลงเมื่อเวลาผ่านไป



รูปที่ 2.9 แสดงการลดลงของรัศมีเมื่อเวลาผ่านไป

จากรูปที่ 2.9 แสดงการลดลงของรัศมีเมื่อเวลาผ่านไป ภาพซ้ายแสดงความยาวของรัศมีเมื่อเวลาเริ่มต้น ส่วนภาพขวาแสดงความยาวของรัศมีเมื่อเวลาผ่านไป จนกระทั่งรัศมีสุดท้ายลู่เข้าสู่ค่าคงที่ ซึ่งมีการกำหนดค่าคงที่สำหรับรัศมีสุดท้ายเอาไว้แล้ว ทำให้โปรแกรมจะหยุดการเรียนรู้

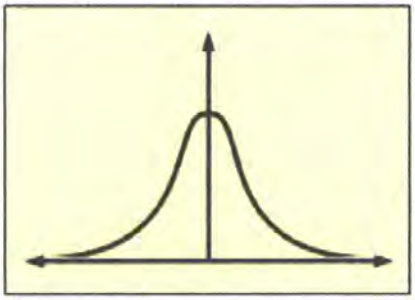
2.3.4 การปรับค่าเวกเตอร์น้ำหนัก w

เมื่อหาระยะรัศมีของโหนดใกล้เคียง (BMU's Neighbourhood) ได้แล้ว จะต้องปรับค่าของเวกเตอร์ w ของทุกๆ โหนดในพื้นที่ ซึ่งคำนวณหาการปรับค่าเวกเตอร์ w ได้จากสมการที่ 2.3 ดังต่อไปนี้

$$W(t+1) = W(t) + \Theta(t)\, L(t)\, (V(t) - W(t)) \qquad (2.3)$$

จากสมการการปรับค่าเวกเตอร์ w ค่าน้ำหนัก w ใหม่ที่ถูกปรับสำหรับโหนดใดๆ นั้น มีค่าเท่ากับ ค่าน้ำหนัก w เดิม รวมกับค่าความแตกต่างที่คำนวณได้จาก w เดิมและค่าเวกเตอร์อินพุต ส่วนค่า Θ (t) คือ ค่า Gaussian Decay ซึ่งสามารถคำนวณหาค่าได้จากสมการที่ 2.4 ดังต่อไปนี้

$$\Theta(t) = \exp(-\mathrm{dist}^2 / 2\sigma^2(t)) \quad ; t = 1, 2, 3, \ldots \qquad (2.4)$$



รูปที่ 2.10 แสดงกราฟระฆังคว่ำสำหรับค่า Θ (t)

จากรูปที่ 2.10 แสดงกราฟระฆังคว่ำสำหรับค่า Θ (t) ซึ่งจากสมการการหาค่า Θ (t) ตัวแปร dist แทนระยะห่างจากโหนดใกล้เคียงไปยังโหนด BMU ส่วน Σ แทนความกว้างของรัศมี คำนวณจากสมการการหารัศมี ณ เวลา t ใดๆ จากสมการการหาค่า Θ (t) นี้แสดงให้เห็นว่า Θ จะมีค่าลดลงเมื่อเวลาผ่านไป การหาค่า Θ (t) เพื่อนำไปใช้ในการคำนวณสมการการปรับค่าเวกเตอร์ w

สมการการหาค่าอัตราการเรียนรู้ (Learning Rate) ซึ่งจะมีค่าลดตามเวลาที่ผ่านไป สามารถคำนวณได้จากสมหการที่ 2.5 ดังต่อไปนี้

$$L(t) = L_0 \exp(-t/\lambda) \quad ; t = 1, 2, 3, \ldots \tag{2.5}$$

ค่า L(t) คือ อัตราการเรียนรู้ (Learning Rate) ซึ่งจะมีค่าลดตามเวลาที่ผ่านไป การหาค่า L(t) สำหรับนำไปใช้ในการคำนวณสมการการปรับค่าเวกเตอร์ w เช่นกัน

ในแต่ละรอบของเรียนรู้ (Iteration) คือการสุ่มอินสแตนท์จากชุดข้อมูลทั้งหมดป้อนให้ SOM เรียนรู้ โดยการปรับค่าเวกเตอร์น้ำหนัก w เมื่อเรียนรู้ชุดข้อมูลครบทุกอินสแตนท์แล้ว เรียกว่า การเรียนรู้ 1 Epoch ตัวอย่าง Source Code แสดง Class ของการเรียนรู้ 1 Epoch ดังนี้

```
bool Csom::Epoch(const vector<vector<double> > &data)
{
  //make sure the size of the input vector matches the size of each node's
weight vector
  if (data[0].size() != constSizeOfInputVector) return false;

  //return if the training is complete
  if (m_bDone) return true;

  //enter the training loop
  if (--m_iNumIterations > 0)
```

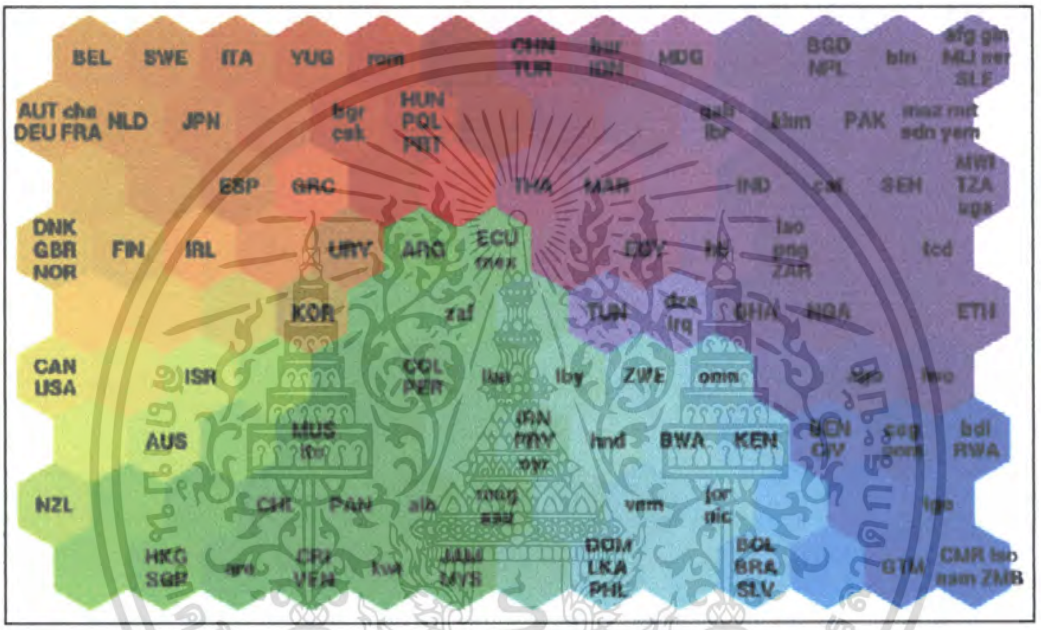
```
{
  //chose a vector at random from the training set to be
  //this time-step's input vector
  int ThisVector = RandInt(0, data.size()-1);

  //present the vector to each node and determine the BMU
  m_pWinningNode = FindBestMatchingNode(data[ThisVector]);

  //calculate the width of the neighbourhood for this timestep
  m_dNeighbourhoodRadius = m_dMapRadius *
          exp(-(double)m_iIterationCount/m_dTimeConstant);

 /* Now to adjust the weight vector of the BMU and its neighbours
 For each node calculate the m_dInfluence (Theta from equation 6 in
 the tutorial. If it is greater than zero adjust the node's
 weights accordingly */
 for (int n=0; n<m_SOM.size(); ++n)
 {
   //calculate the Euclidean distance (squared) to this node
   //from the BMU
   double DistToNodeSq = (m_pWinningNode->X()-m_SOM[n].X()) *
                         (m_pWinningNode->X()-m_SOM[n].X()) +
                         (m_pWinningNode->Y()-m_SOM[n].Y()) *
                         (m_pWinningNode->Y()-m_SOM[n].Y());

   double WidthSq = m_dNeighbourhoodRadius * m_dNeighbourhoodRadius;

   //if within the neighbourhood adjust its weights
   if (DistToNodeSq < (m_dNeighbourhoodRadius *
                                    m_dNeighbourhoodRadius))
   {
     //calculate by how much its weights are adjusted
     m_dInfluence = exp(-(DistToNodeSq) / (2*WidthSq));

     m_SOM[n].AdjustWeights(data[ThisVector],
                            m_dLearningRate,
                            m_dInfluence);
   }

 }//next node

 //reduce the learning rate
 m_dLearningRate = constStartLearningRate *
           exp(-(double)m_iIterationCount/m_iNumIterations);
  ++m_iIterationCount;
}

else
{
  m_bDone = true;
}

return true;
}
```

รูปที่ 2.11 Source Code แสดง Class ของการเรียนรู้ 1 Epoch

2.3.5 การประยุกต์ใช้งานอัลกอริทึม SOM

อัลกอริทึม SOM ถูกใช้อย่างแพร่หลายในการช่วยลดมิติของข้อมูลให้เหลือน้อยลง เพื่อช่วยในการแสดงผล อัลกอริทึม SOM ช่วยมนุษย์ซึ่งไม่สามารถมองเห็นความสัมพันธ์ของข้อมูลจำนวนมากให้เหลือเพียงแผนภาพในระนาบ 2 มิติที่เข้าใจง่าย

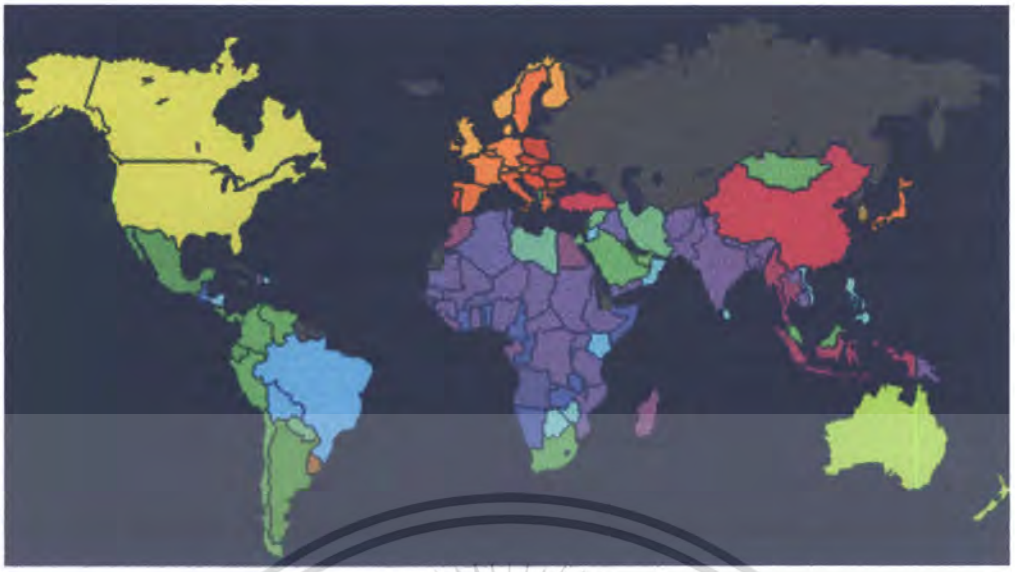
ตัวอย่างสำคัญที่เห็นภาพชัดเจนคือ แผนภาพรายได้ของประชากรโลก มีการนำอัลกอริทึม SOM มาใช้เพื่อคัดแยกข้อมูลเชิงสถิติที่บ่งบอกคุณภาพชีวิตของประชากรในประเทศต่างๆ เช่น ดัชนีสุขภาพ ดัชนีอาหาร ดัชนีการศึกษา และอื่นๆ



รูปที่ 2.12 แสดงแผนภาพรวงผึ้งของประเทศต่างๆ

จากรูปที่ 2.12 แสดงแผนภาพรวงผึ้งของประเทศต่างๆ รูปรวงผึ้งหกเหลี่ยมนี้ มีชื่อเรียกว่า Hexagonal Grid หรือ U-Matrix เป็นการใช้รูปรวงผึ้งหกเหลี่ยมในการแทนโหนดแต่ละโหนดในโครงข่าย SOM

พบว่าประเทศที่มีคุณภาพชีวิตที่ใกล้เคียงกันจะอยู่ในรวงผึ้งเดียวกันหรือใกล้เคียงกัน โดยกำหนดให้รวงผึ้งต่างๆ แทนกลุ่มย่อยแต่ละกลุ่ม ประเทศที่มีคุณภาพชีวิตที่ดีกว่าจะอยู่ทางด้านซ้ายบนของแผนภาพ และประเทศที่มีคุณภาพชีวิตที่แย่ที่สุดจะอยู่ทางด้านซ้ายล่าง ซึ่งแสดงกลุ่มย่อยต่างๆ ตามสีของรวงผึ้งที่แตกต่างกัน เมื่อนำสีที่แสดงใน U-Matrix มาวาดกราฟตามแผนที่โลกจะได้ลักษณะตามภาพต่อไปนี้



รูปที่ 2.13 แสดงประเทศบนแผนที่โลกเมื่อใช้สีจากรวงผึ้ง

อัลกอริทึม SOM มีประโยชน์ต่อการวิเคราะห์ข้อมูลที่มีหลายมิติ และสามารถนำไปประยุกต์ใช้งานได้หลายสาขา เช่น Bibliographic classification, Image browsing systems, Medical Diagnosis, Interpreting seismic activity, Speech recognition, Data compression, Separating sound sources, Environmental modeling, Vampire classification เป็นต้น

2.4 อัลกอริทึมที่ใช้ในการค้นหาความสัมพันธ์เชิงกฎ (Association Rules Discovery)

ในการค้นหาความสัมพันธ์เชิงกฎของข้อมูล หรือ Association Rules Discovery จะนำเทคนิคการค้นหาความสัมพันธ์เชิงกฎมาใช้หากฎความเชื่อมโยงต่างๆ ที่แอบแฝงอยู่ในฐานข้อมูลขนาดใหญ่ อัลกอริทึมที่จะใช้ประมวลผลเพื่อค้นหาความสัมพันธ์เชิงกฎในโครงงานนี้ คือ อัลกอริทึม Apriori ซึ่งเสนอโดย Agrawal ในปี ค.ศ.1993 อัลกอริทึม Apriori เป็นขั้นตอนวิธีที่เหมาะกับการหาความสัมพันธ์ของข้อมูลขนาดใหญ่ โดยกระบวนการในการสร้างกฎความสัมพันธ์จะแบ่งออกเป็น 2 กระบวนการย่อย คือ กระบวนการหา Frequent Itemset และ กระบวนการสร้างกฎความสัมพันธ์

2.4.1 ลักษณะพื้นฐานของอัลกอริทึม Apriori

ตัวอย่างอย่างง่ายเพื่อให้เข้าใจการทำงานของอัลกอริทึม Apriori คือการค้นหาความสัมพันธ์เชิงกฎของชุดตัวอักษร กำหนดคำนิยามของสัญลักษณ์ต่างๆ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 แสดงคำนิยามของสัญลักษณ์ที่ใช้สำหรับอัลกอริทึม Apriori

สัญลักษณ์	นิยาม
k-itemset	ชุดของไอเท็มที่มีสมาชิก k ไอเท็ม โดยที่ k =1, 2, 3, …, n
L_k	ชุดของ Frequent k-itemset นั่นคือชุดของ Frequent Itemset ที่ประกอบด้วยสมาชิก k-Item
C_k	ชุดของ Candidate k-itemset นั่นคือชุดของ Candidate Itemset ที่ประกอบด้วยสมาชิก k-Item

ในการค้นหาความสัมพันธ์เชิงกฎของชุดตัวอักษร กำหนดให้ชุดของไอเท็มที่ต้องการหากฎความสัมพันธ์มีจำนวนสมาชิก 4 ไอเท็ม (4-Itemset) ในที่นี้ คือ อักษรทั้งหมดที่พิจารณา ได้แก่ A, B, C, D และ E ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 แสดงชุดของตัวอักษร 4 ไอเท็ม

TID	Items
1	A C D
2	B C E
3	A B C E
4	B E
5	A B C E

อัลกอริทึม Apriori ประกอบด้วย 2 กระบวนการย่อย ในการประมวลผล คือ กระบวนการหา Frequent Itemset และ กระบวนการสร้างกฎความสัมพันธ์

ในกระบวนการหา Frequent Itemset จะได้ตาราง C_k หรือตาราง Candidate k-itemset ดังแสดงในตารางที่ 2.3 ซึ่งเป็นตารางแสดงผลการนับความถี่ของแต่ละไอเท็มที่ปรากฏในชุดข้อมูลไอเท็ม โดยที่ k จะแทนแต่ละรอบของการนับความถี่ และจะคัดไอเท็มที่มีผลความถี่ต่ำกว่าค่าสนับสนุนที่กำหนดไว้ออกไป เหลือเพียงไอเท็มที่มีค่าสนับสนุนมากพอ ได้เป็นตาราง L_k หรือตาราง Frequent k-itemset ดังแสดงในตารางที่ 2.4

ตารางที่ 2.3 แสดง Candidate 1-itemset

itemset	support
{A}	3
{B}	4
{C}	4
{D}	1
{E}	4

ตารางที่ 2.4 แสดง Frequent 1-itemset

itemset	support
{A}	3
{B}	4
{C}	4
{E}	4

จากนั้นสร้าง Candidate 2-itemset และ Frequent 2-itemset ไปเรื่อยๆ จนกระทั่งไม่สามารถสร้างตารางได้อีก แสดงว่าจบการทำงานในกระบวนการหา Frequent Itemset จะได้ไอเท็มเซ็ทที่มีสมาชิกต่างๆ กัน เช่น {A}, {B}, {C}, {AB}, {AC}, {ABC} เป็นต้น

เมื่อได้ไอเท็มเซ็ททั้งหมดออกมาแล้ว กระบวนการต่อไป คือ กระบวนการสร้างกฎความสัมพันธ์จากไอเท็มเซ็ทเหล่านั้น ตัวอย่างกฎความสัมพันธ์ที่พบได้แก่ A, B □ C ซึ่งจะกล่าวถึงรายละเอียดในหารคำนวณทั้ง 2 กระบวนการนี้ ในหัวข้อต่อไป

2.4.2 กระบวนการหา Frequent Itemset

Frequent Itemset คือ ชุดของข้อมูลที่คาดว่าจะมีการใช้งานบ่อยๆ ซึ่ง Frequent Itemset ในตัวอย่างนี้ได้แก่ตัวอักษร หรือชุดของตัวอักษรที่ปรากฏหลายครั้ง โดยอัลกอริทึม Apriori มีกระบวนการทำงานในการหา Frequent Itemset ดังนี้คือ รอบแรกของการทำงานจะทำการนับจำนวนความถี่ของไอเท็มที่มีในชุดข้อมูลไอเท็ม (k-itemset) ในตัวอย่างนี้ไอเท็มแต่ละไอเท็มแทนตัวอักษร หรือชุดของตัวอักษร ที่ปรากฏในชุดข้อมูลไอเท็ม ดังนั้น เพื่อจะหา Frequent 1-Itemset นั่นคือ ชุดของ Itemset ที่ประกอบด้วยสมาชิกหนึ่งตัวที่เป็น Frequent Itemset ซึ่งความถี่ที่นับได้จะนำมาคำนวณหาค่าสนับสนุนของไอเท็มนั้นๆ โดยสามารถคำนวณค่าสนับสนุนได้จากสมการที่ 2.6 ดังต่อไปนี้

$$\text{Support (S)} = (|U| / |T|) * 100\% \qquad (2.6)$$

จากสมการคำนวณค่าสนับสนุน

Support (S) แทนค่าสนับสนุนของ Item นั้นๆ

|U| แทนจำนวนของรายการที่มี Item ดังกล่าวปรากฏอยู่ในชุดข้อมูลไอเท็ม

|T| แทนจำนวนของรายการทั้งหมดที่เกิดขึ้น

ตัวอย่างเช่น ชุดข้อมูลไอเท็มชุดหนึ่งมีรายการทั้งหมด 4 รายการ คือ {ABC, BD, ABCD, CDE} ค่า Support หรือ ค่าสนับสนุนของไอเท็ม A คือ Support (A) = (2/4)*100% เท่ากับ 50%

จากนั้น พิจารณาเปรียบเทียบค่าสนับสนุนที่คำนวณได้ กับค่าสนับสนุนต่ำสุดที่กำหนดเอาไว้แล้ว เพื่อระบุว่าไอเท็มเซ็ทใดบ้างที่เป็น Frequent 1-Itemset จากนั้น ในรอบถัดมาของการทำงาน จะนำ Frequent Itemset ที่ได้จากการทำงานในรอบที่แล้วมาสร้างชุดไอเท็มเซ็ทที่อาจเป็น

Frequent Itemset เรียกว่า Candidate Itemset แล้วนับความถี่ของ Candidate Itemset ที่สร้างขึ้น แล้วนำมาคำนวณหาค่าสนับสนุน จากนั้นเปรียบเทียบกับค่าสนับสนุนที่กำหนด ถ้า Candidate Itemset ใดที่มีค่าความสนับสนุนมากกว่าค่าสนับสนุนต่ำสุดที่กำหนด ก็จะพิจารณา Candidate Itemset ดังกล่าวเป็น Frequent k-Itemset ส่วน Candidate Itemset ใดที่มีค่าความสนับสนุนน้อยกว่าค่าสนับสนุนต่ำสุดที่กำหนด ก็จะไม่นำมาใช้พิจารณาอีกต่อไป และมีการทำงานเป็นเช่นนี้ซ้ำๆ จนกระทั่งไม่สามารถสร้าง Candidates Itemset ได้อีกจึงจบการทำงานในการหา Frequent Itemset กระบวนการหา Frequent Itemset ของอัลกอริทึม Apriori แสดงดังตัวอย่าง Source Code ต่อไปนี้

```
L1 = {frequent 1-itemsets};
for (k = 2; Lk-1 ≠ ∅; k++) do begin
      Ck = apriori-gen (Lk-1);        // New candidates
      forall transactions t in the database do begin
            Ct = subset (Ck, t);      // Candidates contained in t
                  forall candidates c ∈ Ct do
                        c.count++;
      end
      L1 = { c ∈ Ck | c.count ≥ minsup}
End
Answer = Uk Lk;
```

รูปที่ 2.14 Source Code แสดง กระบวนการหา Frequent Itemset

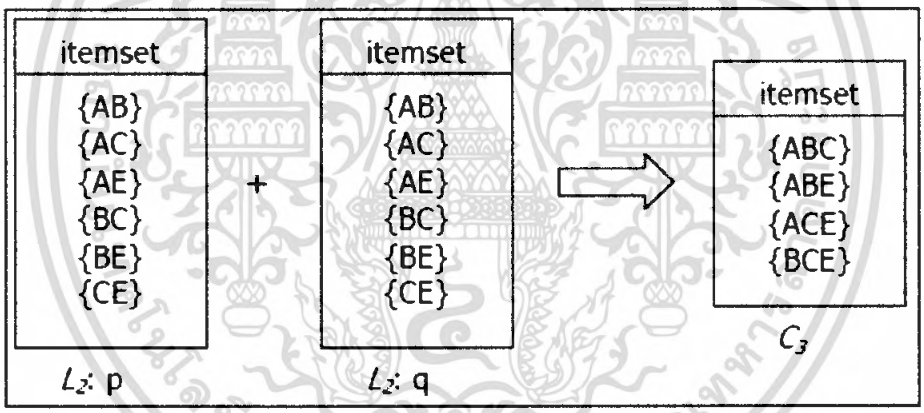
สำหรับขั้นตอนการสร้าง Candidate Itemset จะเป็นขั้นตอนที่นำ Frequent Itemset ในระดับก่อนหน้ามาเชื่อมต่อกันเป็นไอเท็มเซ็ทชุดเดียวกัน โดยพิจารณาเปรียบเทียบจากไอเท็มตัวแรกต้องเหมือนกันจึงจะทำการเชื่อมไอเท็มให้กลายเป็นไอเท็มเซ็ทชุดใหม่ หลังจากนั้นทำการแยกไอเท็มเซ็ทที่ประกอบด้วยไอเท็มที่ไม่ได้เป็นสมาชิกใน Frequent Itemset ระดับก่อนหน้าออกไป ให้เหลือเพียงไอเท็มเซ็ทที่พิจารณา ซึ่งกระบวนการสร้าง และตัวอย่างการสร้าง Candidate Itemset แสดงดัง Source Code ต่อไปนี้ ตามลำดับ รวมทั้งภาพประกอบแสดงการสร้าง Candidate Itemset แสดงในรูปที่ 2.17

```
Step 1:     join L(k-1) with L(k-1)
      insert into Ck
      select p.item1, p.item2, …, p.item(k-1), q.item(k-1)
      from L(k-1) p, L(k-1) q
      where p.item1=q.item1, …, p.item(k-2)=q.item(k-2), p.item(k-1)<q.item(k-1);

Step 2:     prune
      forall itemsets c ∈ Ck do
            forall (k-1)-sunsets s of c do
                  if (s ∉ L(k-1)) then
                        delete c from Ck
```

รูปที่ 2.15 Source Code แสดงกระบวนการสร้าง Candidate Itemset

```
insert into Ck
select c.item1, p.item2, q.item2
from L2p, L2q
where p.item1 = q.item1, p.item2 < q.item2;
```

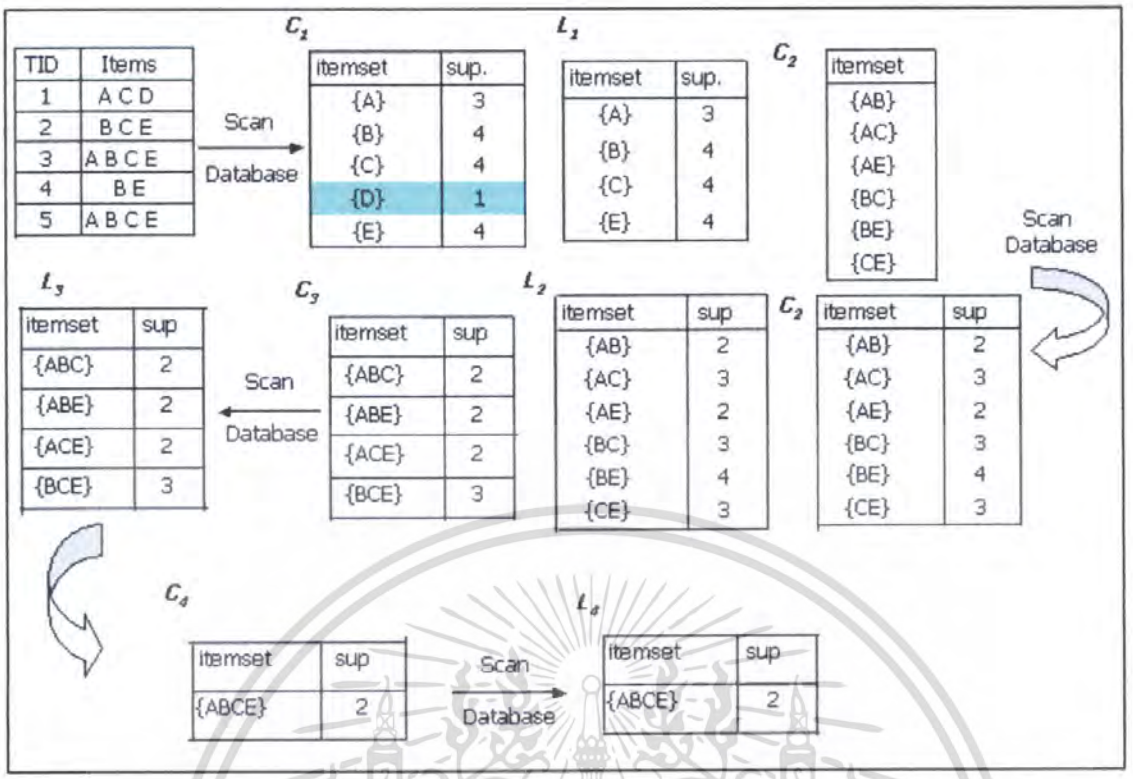
รูปที่ 2.16 Source Code แสดงตัวอย่างการสร้าง Candidate Itemset



รูปที่ 2.17 แสดงตัวอย่างการสร้าง Candidate Itemset

กระบวนการหา Frequent Itemset ในแต่ละรอบสามารถอธิบายเป็นขั้นตอนได้จากตัวอย่างชุดตัวอักษรดังกล่าว ดังแสดงในรูปที่ 2.18 โดยกำหนดให้ค่าสนับสนุนต่ำสุด (min_sup) เท่ากับ 40%

จากรูปที่ 2.18 มีการทำงานดังต่อไปนี้คือ เริ่มพิจารณาจากการหา 1-Frequent Itemset ของชุดตัวอักษรซึ่งมีรายการทั้งหมด 5 รายการ พิจาณาไอเท็มทั้งหมดในที่นี้ประกอบด้วย A, B, C, D และ E ซึ่งนับความถี่ของไอเท็มที่ปรากฏในรายการ จะได้ว่า A มีปรากฏในรายการทั้งหมด 4 ครั้ง หรือ 4 รายการ จากนั้นทำการนับความถี่ของไอเท็มทุกตัวผลที่ได้แสดงดังตาราง C1 ซึ่งเรียกไอเท็มเซ็ทที่เกิดขึ้นว่า 1-Candidate Itemset



รูปที่ 2.18 แสดงกระบวนการทำงานของอัลกอริทึม Apriori

หลังจากนั้นเปรียบเทียบความถี่ที่ได้กับค่าสนับสนุนต่ำสุดที่กำหนดเอาไว้ ซึ่งจากตัวอย่างนี้กำหนดให้เท่ากับ 40% ดังนั้น 40% ของรายการชุดตัวอักษร 5 รายการ เท่ากับ 2 เพราะฉะนั้นแต่ละไอเท็มเซ็ทในรายการชุดตัวอักษรจะต้องปรากฏมากกว่า 1 ครั้งจึงจะพิจารณาว่าเป็น 1-Frequent Itemset ดังแสดงในตาราง L1

จากนั้น ทำการสร้างตาราง n-Candidate Itemset ในลำดับต่อไป เพื่อหาไอเท็มเซ็ทสำหรับตาราง n-Frequent Itemset ในลำดับต่อไป โดยในการสร้าง 2-Candidate Itemset เป็นการเชื่อม 1-Frequent Itemset กับ 1-Frequent Itemset โดยเทียบสมาชิกตัวหน้าของแต่ละไอเท็มเซ็ท หากมีสมาชิกเหมือนกันจะนำไอเท็มเซ็ทมาเชื่อมต่อกัน ได้ผลการทำงานดังกล่าวในตาราง C2 หลังจากนั้นนับค่าความถี่ของการเกิดขึ้น โดยต้องนับรายการที่มี Candidate Itemset ปรากฏพร้อมกันในรายการ เช่น ไอเท็มเซ็ท {AB} ปรากฏในรายการชุดตัวอักษรทั้งหมด 2 รายการ เป็นต้น แล้วทำการกรอง 2-Candidate Itemset ที่มีค่าความถี่ต่ำกว่าค่าสนับสนุนต่ำสุด ผลลัพธ์ที่ได้แสดงดังตาราง L2 การทำงานรอบถัดไปให้สร้าง 3-Candidate Itemset แล้วนับค่าความถี่และคัด Candidate Itemset ที่ไม่ผ่านเกณฑ์ผลที่ได้ดังตาราง C2 และ L3 ตามลำดับ จากนั้นทำการสร้าง Candidate Itemset ในลำดับถัดไปคือ 4-Candidate Itemset นับความถี่และกรอง Candidate Itemset ที่ไม่ผ่านออก จะได้ไอเท็มเซ็ทที่เหลืออยู่ ให้สร้าง 4-Frequent Itemset ซึ่งจะพบว่าในขั้นนี้ 4-Frequent Itemset ไม่สามารถสร้าง 5-Candidate Itemset ต่อได้อีกแล้ว เพราะฉะนั้นหยุดการทำงานไว้เพียงเท่านี้ ผลลัพธ์สุดท้ายที่

ได้คือ Frequent Itemset ที่มีสมาชิก ได้แก่ {A}, {B}, {C}, {E}, {AB}, {AC}, {AE}, {BC}, {BE}, {CE}, {ABC}, {ABE}, {ACE}, {BCE} และ {ABCE} ถือว่าจบกระบวนการหา Frequent Itemset ให้ทำกระบวนการสร้างกฎความสัมพันธ์ในลำดับต่อไป

2.4.3 กระบวนการสร้างกฎความสัมพันธ์

เมื่อได้ชุดของ Frequent Itemset จากขั้นตอนข้างต้นแล้ว ให้นำ Frequent Itemset ที่ได้ทั้งหมดมาสร้างกฎความเชื่อมโยง โดยสร้างกฎ (R) ให้อยู่ในรูปของ X □ Y และทำการคำนวณหาค่าความเชื่อมั่นของแต่ละกฎที่สร้างขึ้นมาเพื่อคัดเลือกกฎมาใช้ โดยพิจารณาเปรียบเทียบค่าความเชื่อมั่นของกฎที่ได้กับค่าความเชื่อมั่นต่ำสุดที่กำหนด ถ้ากฎใดมีค่าความเชื่อมั่นมากกว่าค่าความเชื่อมั่นต่ำสุด ก็จะพิจารณาว่ากฎที่ได้นั้นสามารถนำมาใช้งานได้ ซึ่งการคำนวณหาค่าความเชื่อมั่นสามารถคำนวณได้จากสมการที่ 2.7 ดังต่อไปนี้

$$\text{Confidence (R)} = (\text{Support (X}\square\text{Y)} / \text{Support (X)}) * 100\% \qquad (2.7)$$

จากสมการคำนวณค่าความเชื่อมั่น

Confidence (R) แทนค่าความเชื่อมั่นของกฎ

Support (X□Y) แทนค่าสนับสนุนของกฎซึ่งก็คือจำนวนรายการที่ประกอบด้วยไอเท็ม X และไอเท็ม Y อยู่ร่วมกัน

Support (X) แทนค่าสนับสนุนของไอเท็ม X

ตัวอย่างเช่น สมมติให้ Frequent Itemset {A, B, C} มีค่าสนับสนุนเท่ากับ 60% ถ้า {A, B} มีค่าสนับสนุนเท่ากับ 80% และ {C} มีค่าสนับสนุนเท่ากับ 80% เพราะฉะนั้นเมื่อพิจารณากฎความเชื่อมโยง A, B □ C ค่าความเชื่อมั่นของกฎนี้จะเท่ากับ (Support ({A, B, C}) / Support({A, B})) * 100% = (60% / 80%) * 100% = 75% เป็นต้น

ข้อมูลความสัมพันธ์ที่ได้จากกระบวนการหา Frequent Itemset และกระบวนการสร้างกฎความสัมพันธ์ดังที่กล่าวมาทั้งหมดนี้ เรียกว่า ความสัมพันธ์เชิงกฎ

บทที่ 3

การออกแบบและพัฒนา

3.1 แนะนำแอพลิเคชั่น Minero Inteligente

ชื่อแอพลิเคชั่น : เหมืองข้อมูลอัจฉริยะ

Minero Inteligente

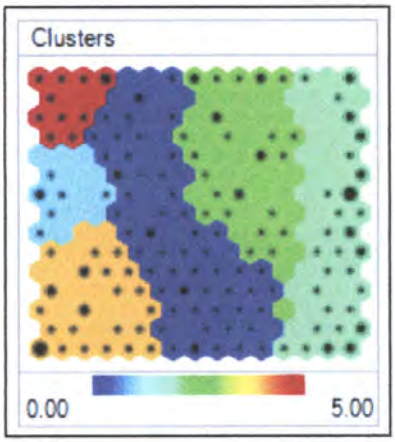
เป็นแอพลิเคชั่นที่ถูกออกแบบมาโดยมีวัตถุประสงค์ดังนี้

1. ช่วยจัดกลุ่มข้อมูลที่ซับซ้อนมากๆ จากข้อมูลที่มีความสัมพันธ์กันมากมายให้สามารถแทนออกมาด้วยแผนภาพที่ชัดเจน เข้าใจง่ายต่อการนำเสนอ อำนวยความสะดวกในการออกรายงานการแสดงผลทั้งเชิงตาราง กราฟฟิก และ กราฟฟิกแอนิเมชั่น ตัวอย่างเช่น แบ่งกลุ่มข้อมูลเพื่อคัดแยกผู้สมัครงานที่ดี ออกจากกลุ่มผู้สมัครทั้งหมดที่สมัครเป็นนักบินผู้ช่วย โดยพิจารณาจากใบประวัติส่วนตัวสมัครเข้าทำงานจำนวนมาก ประกอบด้วยแอทริบิวต์ต่างๆ เช่น 'เพศ' 'การศึกษา' 'เกรดเฉลี่ย' 'อายุ' 'ชั่วโมงบิน (ปี)' 'คะแนนผลการตรวจร่างกาย' 'คะแนนภาษาอังกฤษ' เป็นต้น ดังแสดงในตารางที่ 3.1 แสดงตัวอย่างชุดข้อมูลของผู้สมัครเป็นผู้ช่วยนักบิน

ตารางที่ 3.1 แสดงตัวอย่างชุดข้อมูลของผู้สมัครเป็นผู้ช่วยนักบิน

#	เพศ	ระดับการศึกษา	GPA	อายุ	ชั่งโมงบิน	คะแนนตรวจ	คะแนนภาษาอังกฤษ
1	ชาย	ป.ตรี	3.6	31	2100	230	550
2	หญิง	ป.โท	4	29	3600	230	600
3	ชาย	ป.ตรี	3.9	25	1000	160	450
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

2. ช่วยแสดงผลการจัดกลุ่มจะออกมาในลักษณะกราฟฟิก แสดงการกระจุกตัวในระนาบ 2 มิติ ในลักษณะของการจัดกลุ่มข้อมูลที่อินสแตนท์คล้ายกัน จะอยู่ในกลุ่มย่อย หรือ คลัสเตอร์เดียวกัน ดังแสดงในรูปที่ 3.1 แสดงคลัสเตอร์ของแพทเทิร์นข้อมูลจำนวนมาก สรุปออกมาได้ว่ามี 6 แพทเทิร์นใหญ่ๆ คือ กลุ่มสีแดง สีฟ้า สีเหลือง สีน้ำเงิน สีเขียว และสีเขียวอ่อน



รูปที่ 3.1 แสดงคลัสเตอร์ของแพทเทิร์นข้อมูล

3. ถึงแม้ทำการจัดกลุ่มแล้วผู้ใช้ยังสามารถวิเคราะห์ความสัมพันธ์ภายในของกลุ่มนั้นๆ ได้อีกโดยเลือกกลุ่มย่อยเหล่านั้น จากแผนภาพ รูปที่ 3.1 แล้วหาความสัมพันธ์ภายในที่พบบ่อยๆระหว่างคอลัมน์ใดๆ เช่น

   Rule#1: if (เพศ = ชาย) and (GPA > 3.00) then คะแนนภาษาอังกฤษ > 550

   (confidence 98%)

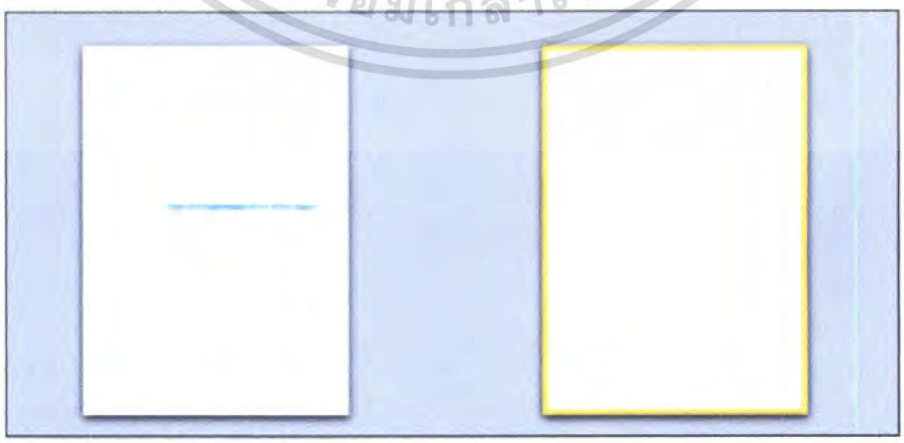
   Rule#2: if (ชั่วโมงบิน > 2500) then กลุ่ม = Red

   (confidence 88%)

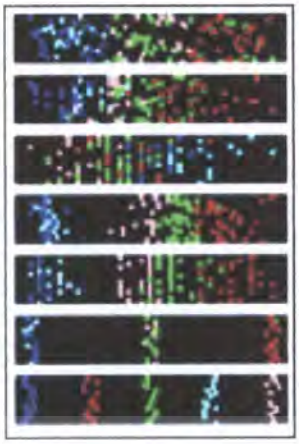
   Rule#3: if (ชั่วโมงบิน < 500) and (GPA < 3.00) then คะแนนภาษาอังกฤษ < 450

   (confidence 75%)

4. ในส่วนของการแสดงผลจะสามารถเลือกแสดงผลได้หลายแบบ มีการเลือกรูปแบบในการออกรายงานและกราฟต่างๆ จากนั้นแปลงผลไปยังไฟล์ ประเภทต่างๆ เช่น PDF, Excel Spreadsheet เป็นต้น ดังแสดงในรูปที่ 3.2 และรูปที่ 3.3



รูปที่ 3.2 แสดงรูปแบบการออกรายงานประเภทต่างๆ



รูปที่ 3.3 แสดงรูป Scatterplot ของแต่ละแอทริบิวต์

3.2 ภาพรวมการทำงานของ Minero Intelligente

Pre-processing Phase

เริ่มการเตรียมข้อมูล นำเข้าข้อมูลจากฐานข้อมูล หรือ ไฟล์ชุดข้อมูลต่างๆ นำมาตรวจสอบมาตรฐานต่างๆ เพื่อการใช้งานต่อใน *Processing Phase* ในขั้นตอนนี้ผู้ใช้งานสามารถดูข้อมูลดิบได้ในระนาบ 2 มิติ โดยแต่ละแกนจะเป็นแอทริบิวต์ต่างๆ ที่ผู้ใช้เลือกขึ้นมาพิจารณา

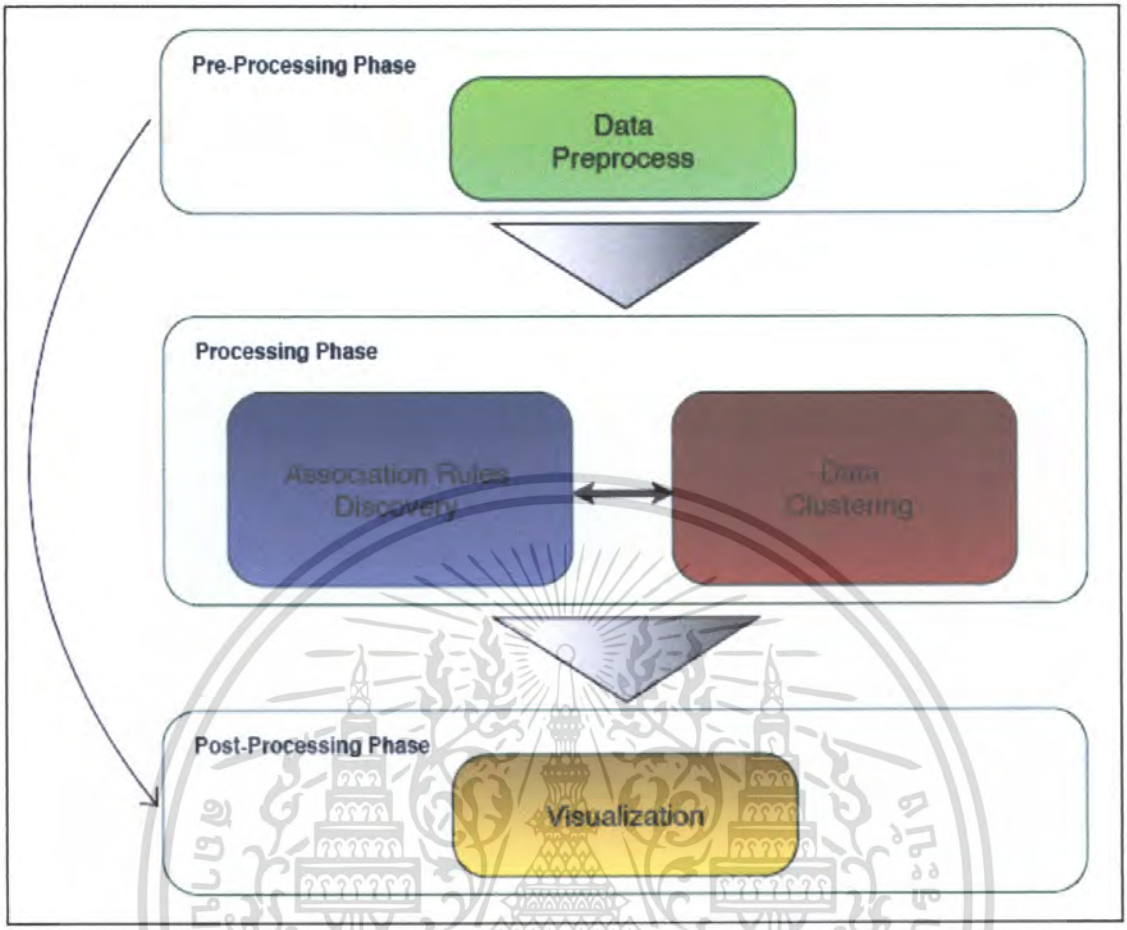
Processing Phase

ในขั้นตอนนี้จะทำการประมวลผลข้อมูลตามลักษณะการใช้งานที่ผู้ใช้เลือก โดยมีการใช้งานอยู่ 2 ประเภท คือ การแบ่งกลุ่มข้อมูล และการหาความสัมพันธ์ที่พบบ่อยของแอทริบิวต์ใดๆ หรือ กลุ่มของแอทริบิวต์ใดๆ ที่แทนออกมาในรูปของกฎได้ ซึ่งขั้นตอนนี้จะมีรายละเอียดการออกแบบค่อนข้างมาก เนื่องจากจะมีการแสดงผลการวิเคราะห์ ทั้งแบบ ตาราง รูปภาพ กราฟ และ แอนิเมชั่นกราฟ

Post-Processing Phase

ขั้นตอนสุดท้ายนี้จะรองรับการออกรายงานของผู้ใช้ในลักษณะต่างๆ เป็นรูปแบบที่สามารถเลือกได้ ในรายงานจะประกอบไปด้วย กราฟ ตาราง ตัวอย่างข้อมูล และอื่นๆ ที่พบในระหว่างขั้นตอนการวิเคราะห์ข้อมูล ขั้นตอนนี้จะสรุปข้อมูลทั้งหมดเพื่อนำเสนอ ดังนั้น จึงสามารถแปลงออกไปเป็นไฟล์ชนิดต่างๆได้ เช่น PDF, Excel หรือ ชุดข้อมูล Extension(.ds) เป็นต้น

ภาพรวมการทำงานทั้ง 3 กระบวนการของแอพลิเคชั่น Minero Intelligente นี้สามารถแสดงออกมาเป็นแผนภาพที่เข้าใจง่ายได้ดังรูปที่ 3.4



รูปที่ 3.4 แสดงภาพรวมการทำงานของ Minero Inteligente

3.3 การออกแบบขั้นต้น

3.3.1 List Candidate Requirement (Application Features)

Data Preprocessing

• สามารถทำการเลือกชุดข้อมูลเพียงบางส่วนได้ โดยกรองออกได้ด้วยตัวกรองแบบต่างๆ แล้วสุ่มข้อมูลก่อนนำไป ทำการวิเคราะห์ใดๆ ได้ เป็นต้น

• สามารถจัดการกับข้อมูลบางส่วนในชุดข้อมูลที่ไม่มีคุณสมบัติ Consistency ได้

• สามารถรองรับการติดต่อกับฐานข้อมูล เพื่อดึงข้อมูลออกจากฐานข้อมูลมาเป็นชุดข้อมูลต่อไป

• สามารถแปลงชุดข้อมูล ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel spreadsheet เป็นต้น

Data Clustering

• สามารถนำแอพพลิเคชั่นมาช่วยวิเคราะห์การแบ่งกลุ่มข้อมูลที่มีจำนวนแอทริบิวต์มากๆ หรือมีจำนวนมิติมากๆ ให้ลดลงจนสามารถแสดงด้วยระนาบการกระจุกตัว 2 มิติเท่านั้น

• สามารถบอกแนวโน้มของกลุ่มข้อมูลที่เปลี่ยนไปเพื่อแสดงให้เห็นถึงพฤติกรรมและแนวโน้มการเปลี่ยนแปลงจากอดีตถึงปัจจุบันได้

• สามารถเลือกดูข้อมูลจำเพาะของกลุ่มย่อยใดๆ ได้จากกลุ่มที่พบทั้งหมด

Association Rules Discovery

• สามารถหาความสัมพันธ์เชิงกฎหรือแพทเทิร์น ระหว่างแอทริบิวต์หรือกลุ่มของแอทริบิวต์ใดๆ ได้

• สามารถบอกแนวโน้มของความสัมพันธ์เชิงกฎที่เปลี่ยนแปลงไปเพื่อแสดงให้เห็นถึงพฤติกรรมและแนวโน้มการเปลี่ยนแปลงจากอดีตถึงปัจจุบันได้

Visualizing, Graphing and Reporting

• สามารถนำแอพลิเคชั่นมาช่วยวิเคราะห์ความสัมพันธ์ของข้อมูลดิบจากชุดข้อมูลที่มีแอทริบิวต์หลายตัว โดยแสดงความสัมพันธ์ของแอทริบิวต์ 2 ตัวใดๆ ในรูปแบบกราฟประเภทต่างๆ

• สามารถแสดงข้อมูลแบบจำเพาะของข้อมูลกลุ่มย่อยกลุ่มใดกลุ่มหนึ่งจากกลุ่มย่อยทั้งหมดที่ค้นพบได้ในรูปแบบกราฟความสัมพันธ์ระหว่างแอทริบิวต์ต่างๆ

• สามารถแปลงกราฟใดๆ ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

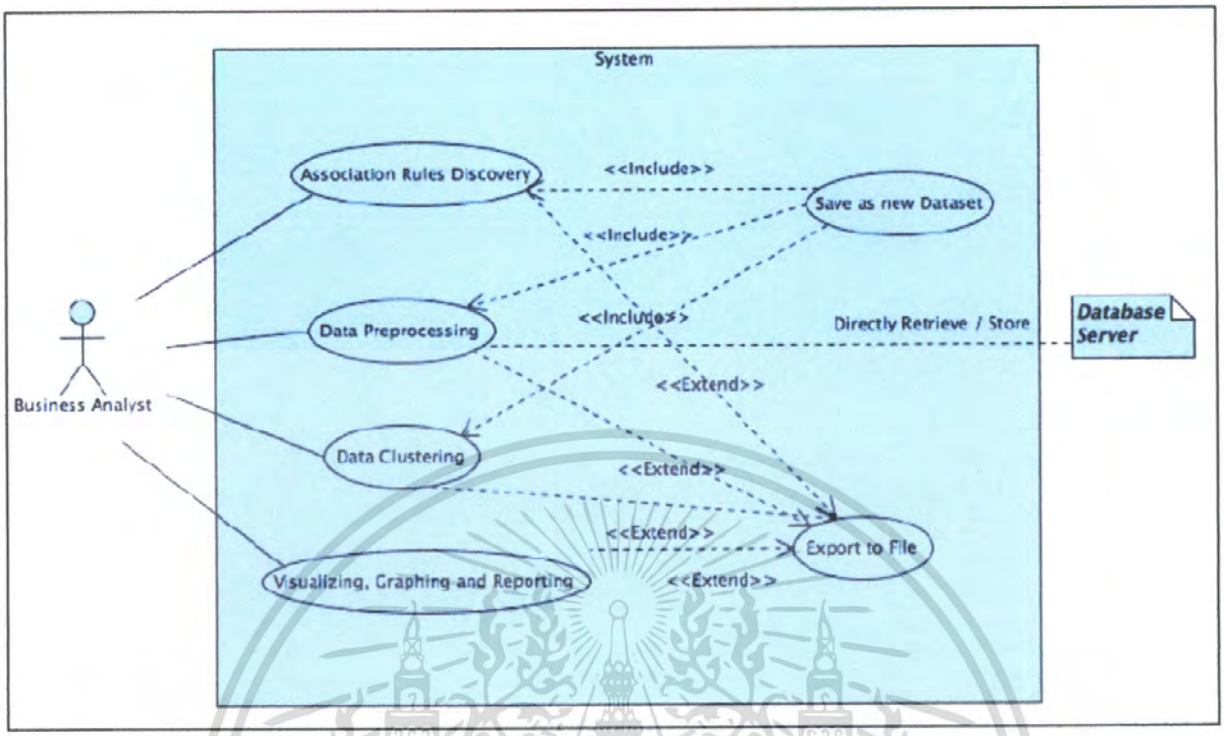
• สามารถแปลงสารสนเทศของข้อมูลกลุ่มย่อย ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

• สามารถแปลงอินสแตนท์ที่มีแพทเทิร์นคล้ายข้อมูลกลุ่มย่อย ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

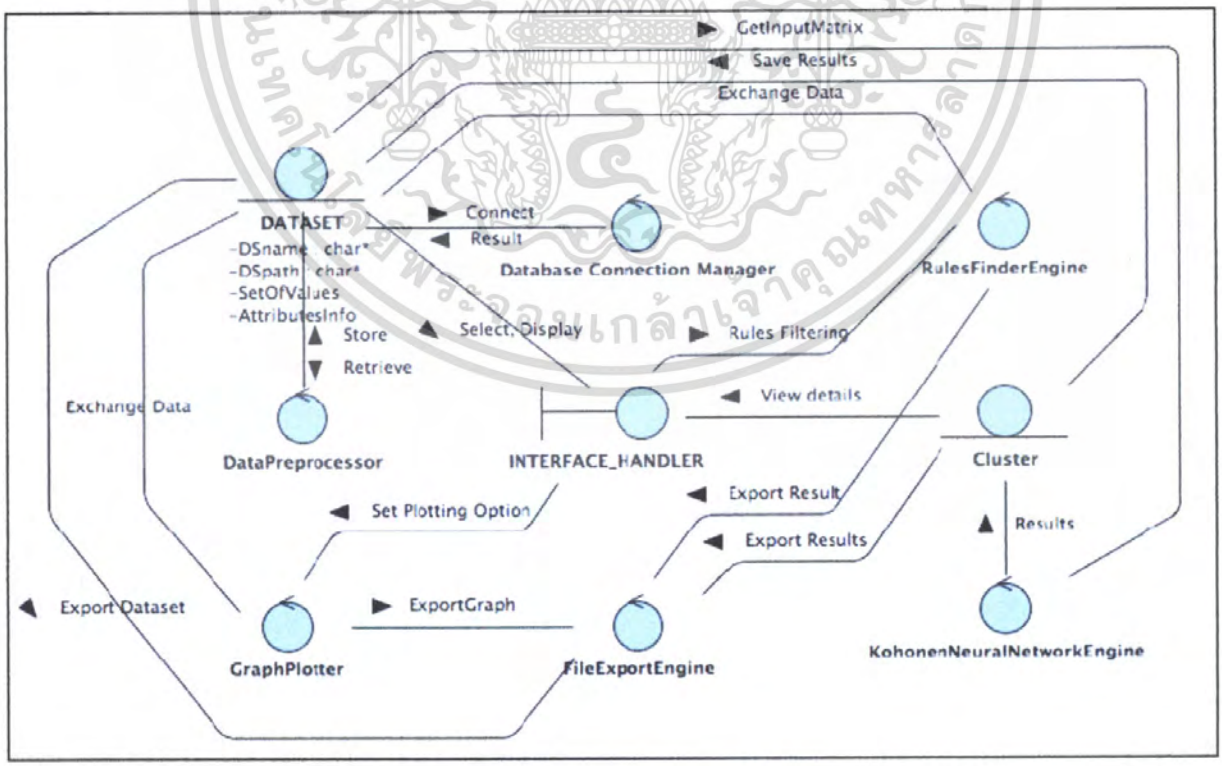
• สามารถแปลงกฎ และดึงชุดข้อมูลที่สอดคล้องกับกฎนั้นๆ ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

3.3.2 System Context

มีการออกแบบ Business Model และ Domain Model ดังแสดงในรูปที่ 3.5 และรูปที่ 3.6 ตามลำดับดังต่อไปนี้



รูปที่ 3.5 แสดง Use Case Diagram ของ Business Model



รูปที่ 3.6 แสดง Domain Model

3.3.3 Functional Requirement

แบ่ง Function Requirement ออกเป็น 4 ส่วน ได้แก่ Data Preprocessing, Data Clustering, Association Rules Discovery และ Visualizing Graphing and Reporting โดยแต่ละส่วนมีรายละเอียดดังต่อไปนี้

Data Preprocessing

• สามารถเลือกชุดข้อมูลเฉพาะส่วนที่ต้องการได้ หรือกรองข้อมูลส่วนที่ไม่ต้องการออกได้ โดยใช้ตัวกรองแบบต่างๆ รวมทั้งสุ่มข้อมูลเป็นบางอินสแตนท์เพื่อนำไปเข้าอัลกอริทึมในการวิเคราะห์ข้อมูลต่อ

- แสดงข้อมูลจำเพาะของชุดข้อมูลได้ เช่น ชื่อชุดข้อมูล จำนวนอินสแตนท์ จำนวนแอทริบิวต์ เป็นต้น
- สามารถสั่งให้สุ่มอินสแตนท์จากชุดข้อมูลได้โดยกำหนดค่าคงที่การสุ่มได้
- สามารถเลือกดูข้อมูลจำเพาะของแอทริบิวต์ได้ประกอบด้วย
  - สำหรับ attribute type = nominal : attribute name, attribute type, total number of missing values (also in percentage), total number of distinct values, unique values (also in percentage) และ ตารางแสดงค่า distinct value ทั้งหมด และ distinct value counts
  - สำหรับ attribute type = numeric : attribute name, attribute type, total number of missing values (also in percentage), total number of distinct values, unique values (also in percentage) และ ตารางแสดง ค่าทางสถิติทั้งหมด คือ minimum, maximum, mean, stdDev
  - สำหรับ attribute type = date : attribute name, attribute type, total number of missing values (also in percentage), total number of distinct values, unique values (also in percentage) และ ตารางแสดงค่าเลือกได้ว่าดูตามวัน เดือน ไตรมาส หรือ ปี และ แสดง frequency ของแต่ละ วัน เดือน ไตรมาส หรือ ปี นั้นๆด้วย
  - สำหรับ attribute type = time : attribute name, attribute type, total number of missing values (also in percentage), total number of distinct values, unique values (also in percentage) และ ตารางแสดงค่าเลือกได้ว่าดูตามนาที (นาทีที่ 1 - 60) หรือ ตามชั่วโมง (ชั่วโมงที่ 0 - 23) และ แสดง frequency ของแต่ละนาทีหรือชั่วโมง นั้นๆด้วย
- สามารถเลือกดู เลือกกรองข้อมูลจาก กราฟแท่งที่แสดงการกระจายตัวของแอทริบิวต์ได้
  - สำหรับ attribute type = nominal : สำหรับกราฟแท่งของ nominal type ให้แต่ละ สีแทน แต่ละ nominal distinct value ที่ด้านบนกราฟแท่งให้บอก จำนวน distinct value และ distinct value counts ด้วยสำหรับการเลือก filter นั้น user สามารถทำ multiple select ยังบนแต่ละกราฟได้ เพื่อ เลือกกรองเอา instance ที่อยู่ใน selection ได้

- สำหรับ attribute type = numeric : สำหรับกราฟแท่งของ numeric type ให้มีแกนนอนเป็นค่า range ตั้งแต่ minimum ถึง maximum และลักษณะของกราฟแท่งจะเป็น กราฟแท่งซ้อนกันหลายสีตามแต่ละ nominal distinct value ที่ถูกเลือกเป็น **target classifier** ของ ชุดข้อมูล นั้นๆ จากนั้น กำหนดตารางช่วง frequency เพื่อ quantization ได้ (ค่าเริ่มต้นการ quantization จะคำนวณจาก ***algorithm*** ที่เหมาะสม) จากนั้นหากต้องการทำ filter ก็สามารถทำ multiple select ยังบนแต่ละกราฟได้ เพื่อ เลือกกรองเอา instance ที่อยู่ใน selectionได้

- สำหรับ attribute type = **date** : สำหรับกราฟแท่งของ date type ให้มีแกนนอนเป็นค่า range ตั้งแต่ minimum เดือน,ไตรมาส หรือ ปี ถึง maximum เดือน,ไตรมาส หรือ ปี และลักษณะของกราฟแท่งจะเป็น กราฟแท่งซ้อนกันหลายสีตามแต่ละ date distinct value ที่ถูกเลือกเป็น **target classifier** ของ ชุดข้อมูล นั้นๆ จากนั้น กำหนดได้ว่าจะ quantization ตาม เดือน,ไตรมาส หรือ ปี (ค่าเริ่มต้นการ quantization จะเป็น 'ตามเดือน') จากนั้นหากต้องการทำ filter ก็สามารถทำ multiple select ยังบนแต่ละกราฟได้ เพื่อ เลือกกรองเอาinstance ที่อยู่ใน selection ได้

- สำหรับ attribute type = time : สำหรับกราฟแท่งของ time type ให้มีแกนนอนเป็นค่า range ตั้งแต่ minimum ถึง maximum และลักษณะของกราฟแท่งจะเป็น กราฟแท่งซ้อนกันหลายสีตามแต่ละ nominal distinct value ที่ถูกเลือกเป็น **target classifier** ของ ชุดข้อมูล นั้นๆ จากนั้น กำหนดการ quantization ได้ ตามนาที (นาทีที่ 1 - 60) หรือ ตามชั่วโมง (ชั่วโมงที่ 0 - 23)(โดยค่าเริ่มต้นการ quantization จะเท่ากับ 'ตามชั่วโมง') จากนั้นหากต้องการทำ filter ก็สามารถทำ multiple select ยังบนแต่ละกราฟได้ เพื่อเลือกกรองเอา instanceที่อยู่ใน selection ได้

- แต่ละแอทริบิวต์สามารถเปลี่ยน attribute type ได้ เป็น { nominal, numeric, date, time }
- แต่ละชุดข้อมูลสามารถเลือก target classifier ได้
- แต่ละชุดข้อมูล จะมี boolean flag status 3 อย่างคือ "consistence" , "date_sequence" , "time_sequence" โดย ถ้า ชุดข้อมูล นั้น consistence , มี field time อยู่ในมาตรฐาน , มี field date อยู่ในมาตรฐาน flag ทั้งสามจะ เป็น 'true'

• สามารถจัดการกับชุดข้อมูลที่ไม่มีคุณสมบัติ consistency ได้

- ปัญหา Inconsistence Case I : ถ้า value ของ attribute type ใด ๆ มีค่าที่เป็น type อื่นปะปนขัดแย้งกับ attribute type นั้นๆ จะต้องทำการ fetch tuples มา edit ได้, เลือก delete tuples ที่มีปัญหาได้

- ปัญหา Inconsistence Case II : ถ้า attribute type = numeric และ มีบาง field เป็น missing value, จะต้องทำการ fill ค่าที่หายไปได้, fetch tuples มา edit ได้, เลือก delete tuples ที่มีปัญหาได้

- ปัญหา Inconsistence Case III : ถ้า attribute type = nominal และ มีบาง field เป็น missing value, จะต้องทำการ fetch tuples มา edit ได้, เลือก delete tuples ที่มีปัญหาได้

• สามารถรองรับการติดต่อกับฐานข้อมูล เพื่อ Query ข้อมูลออกมาสร้างเป็นชุดข้อมูลได้

- จัดการติดต่อกับ database ได้ผ่าน jdbc เพื่อส่งคำสั่ง query ออกไป get ข้อมูลเข้ามาเป็น ชุดข้อมูล ใหม่
- รองรับการเปิดไฟล์ นามสกุล .ds ซึ่งเป็น ชุดข้อมูล file มาตรฐานของโปรแกรม

• สามารถแปลงชุดข้อมูลออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

- มีการ save as ชุดข้อมูล file extension, มีการ save as ในนามสกุลอื่นๆ ด้วยเช่น pdf, excel
- เมื่อมีการ preprocess ข้อมูลต่างๆ เรียบร้อย โปรแกรมจะต้องเรียกให้ save ทุกครั้งก่อนจะนำชุดข้อมูล ไป ยัง process การวิเคราะห์อื่นๆ ต่อไป
- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้

Data Clustering

• สามารถนำแอพพลิเคชั่นมาช่วยวิเคราะห์เพื่อแบ่งกลุ่มข้อมูลที่มีจำนวนแอทริบิวต์มากๆ หรือมีหลายมิติ โดยทำให้มีจำนวนมิติลดลง แล้วแสดงด้วยระนาบการกระจุกตัว 2 มิติ

- สามารถกำหนดค่าเริ่มต้นการทำการ clustering ได้ก่อนทำ operation ใดๆ
- ระหว่างการทำการ clustering ให้แสดงผล feature map ออกมาให้เป็นระหว่าง training epoch ต่างๆเป็นanimation ที่มองเห็นการเปลี่ยนแปลงของระนาบ feature map ตลอดเวลา
- เมื่อทำการ clustering เรียบร้อยแล้วให้แสดงรายละเอียด ผลการ cluster มีกี่ cluster, สามารถเห็น ข้อมูลจำเพาะของแต่ละ cluster ได้ หากต้องการดูรายละเอียด สามารถคลิกเข้าไปดูเพิ่มได้
- แสดงลักษณะจำเพาะของ cluster ที่พบทั้งหมดออกมาเป็นกราฟเมตริกซ์ ในลักษณะ กราฟแท่ง หรือ พายกราฟได้ทุก cluster กำหนดการแสดงผลตามแต่ละ attribute ที่เลือกได้
- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้

• สามารถบอกแนวโน้มของกลุ่มข้อมูลที่เปลี่ยนไปเพื่อแสดงให้เห็นถึงพฤติกรรมและแนวโน้มการเปลี่ยนแปลงจากอดีตถึงปัจจุบันได้

- มีการจัดเรียงและแบ่งชุดของเซตข้อมูลตามช่วงเวลาได้ และต้องสามารถกำหนดได้โดยผู้ใช้
- สามารถนำข้อมูลที่จัดเรียงแล้วมาทำการ train เข้าสู่ self-organizing map อย่างต่อเนื่องเมื่อจบหนึ่งช่วงเวลาให้บันทึกผลไว้ แล้ว train ข้อมูลชุดที่เหลือต่อไป และบันทึก อย่างนี้เรื่อยๆ จนหมด

- นำเสนอการเปลี่ยนแปลงได้โดยพิจารณาดูแพทเทิร์นที่เกิดจากตัวแทนของ cluster ที่พบในช่วงเวลาต่างๆ เป็นตารางสรุป {ช่วงเวลา, cluster #no (ต้องใช้อ้างอิงในกราฟได้ด้วย), ลักษณะ แพทเทิร์น/ค่าใน attribute ทุกตัว}
- นำเสนอการเปลี่ยนแปลงได้ โดยพิจารณาดู feature map ที่พบ ในช่วงเวลาต่างๆ เป็นแอนิเมชั่นและสามารถทำ step-by-step motion ได้ด้วย
- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้

• สามารถเลือกดูข้อมูลจำเพาะของกลุ่มย่อยใดๆ ได้จากกลุ่มที่พบทั้งหมด

- เมื่อเลือกcluster ใดๆ จะต้องแสดงลักษณะจำเพาะทั้งหมดออกมาเป็นกราฟในลักษณะ กราฟแท่ง หรือ พายกราฟได้ แสดงตามแต่ละ attribute ที่เลือกดู
- สามารถ save cluster's ชุดข้อมูล เพื่อนำไปหา association rules ต่อได้
- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้

Association Rules Discovery

• สามารถหาความสัมพันธ์เชิงกฎหรือแพทเทิร์น ระหว่างแอทริบิวต์หรือกลุ่มของแอทริบิวต์ใดๆ ได้

- สามารถกำหนดค่าเริ่มต้นการทำการ discovery ได้ก่อนทำ operation ใดๆ
- สามารถ filter แสดง กฎที่หาได้ หลายรูปแบบ เช่น show top 10, top 50, top 100, all หรือ sort by 'most accurate' / 'least accurate' ได้
- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้
- เมื่อ เลือกดูที่ rules ข้อใด สามารถแสดง ชุดข้อมูล tuples ที่ support rules ข้อนั้นได้
- เมื่อ เลือกดูที่ rules ข้อใด สามารถแสดง scatter plot graph ของ ชุดข้อมูล tuplesที่ support กับrules ข้อนั้นได้
- สามารถ save ชุดข้อมูล tuples ที่ support กับ rules ข้อที่เลือก เพื่อนำไปหา cluster ต่อได้

• สามารถบอกแนวโน้มของความสัมพันธ์เชิงกฎที่เปลี่ยนแปลงไปเพื่อแสดงให้เห็นถึงพฤติกรรมและแนวโน้มการเปลี่ยนแปลงจากอดีตถึงปัจจุบันได้

- มีการจัดเรียงและแบ่งชุดของเซตข้อมูลตามช่วงเวลาได้ และต้องสามารถกำหนดได้โดยผู้ใช้
- สามารถนำข้อมูลที่จัดเรียงแล้วมาทำการหา association rules อย่างต่อเนื่องเมื่อ จบหนึ่งช่วงเวลาให้บันทึกผล rules ranking ไว้ ลบข้อมูลชุดเก่า แล้ว feed ข้อมูลชุดที่เหลือใหม่ และบันทึกผล อย่างนี้เรื่อยๆ จนหมด
- นำเสนอการเปลี่ยนแปลงได้ โดย พิจารณาดู rules ranking (ตามที่เลือกจาก filter option) ที่พบ ในช่วงเวลาต่างๆเป็น ตารางสรุป {ช่วงเวลา, rules #no (ต้องใช้อ้างอิงในกราฟได้ด้วย), rules ในรูปของ if-then statement}

- นำเสนอการเปลี่ยนแปลงได้ โดย ใช้กราฟแท่ง พายกราฟ ในช่วงเวลาต่างๆ ที่กำหนดได้
- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้

Visualizing, Graphing and Reporting

• สามารถนำแอพลิเคชั่นมาช่วยวิเคราะห์ความสัมพันธ์ของข้อมูลคิบจากชุดข้อมูลที่มีแอทริบิวต์หลายตัว โดยแสดงความสัมพันธ์ของแอทริบิวต์ 2 ตัวใดๆ ในรูปแบบกราฟประเภทต่างๆ

- นำทุก combination ของสองแกนใดๆ ของ attribute ทั้งหมด มาสร้าง plot matrix
- สามารถนำ plot matrix มา plot หยาบๆ เพื่อให้เห็นความสัมพันธ์คร่าวๆ พร้อมกันทุก attribute combination ได้
- สามารถเลือกดู กราฟของสองแกนใดๆ แบบขยายส่วนได้ ปรับเลือกสีของแต่ละ cluster ได้ เลือก target attribute ได้ , เลือกแกน x และ y แบบใดก็ได้ , มีภาพย่อส่วน แบบหยาบ 1 มิติ
- แสดงสีและจำนวนของcluster โดยสรุปได้ (หาก target attribute เป็น nominal), แสดงแถบสีของช่วงค่าminimum - maximum ของ target attribute ได้ (หาก target attribute เป็น numeric)
- สามารถแสดง แถบ เลื่อนเปรียบเทียบค่าในเชิงเวลาได้ (ถ้า ชุดข้อมูล มี flag status ที่เหมาะสม) และ สามารถทำ step-by-step motion ได้ด้วย
- สามารถปรับค่า noise ในการแสดงผลให้กับ data ได้ เพื่อให้เห็น data ที่ plot ทับซ้อนกันชัดเจนยิ่งขึ้น

• สามารถแสดงข้อมูลแบบจำเพาะของข้อมูลกลุ่มย่อยกลุ่มใดกลุ่มหนึ่งจากกลุ่มย่อยทั้งหมดที่ค้นพบได้ในรูปแบบกราฟความสัมพันธ์ระหว่างแอทริบิวต์ต่างๆ

- สามารถทำการเลือก cluster ที่ต้องการแสดงผลได้จาก feature map รวมถึงแสดงข้อมูลจำเพาะของ clusterนั้นๆ ได้อีกด้วย
- สามารถนำแอพลิเคชั่นมาช่วยวิเคราะห์ความสัมพันธ์ของข้อมูลจาก ชุดข้อมูล ที่อ้างถึงโดย cluster ที่ถูกเลือกซึ่งมี attribute มากๆ เลือก 2 แกนใดๆมา plot เพื่อใช้ ดูความสัมพันธ์ใดๆ ก็ได้ในรูปแบบของกราฟประเภทต่างๆได้

• สามารถแปลงกราฟใดๆ ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้

• สามารถแปลงสารสนเทศของข้อมูลกลุ่มย่อย ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

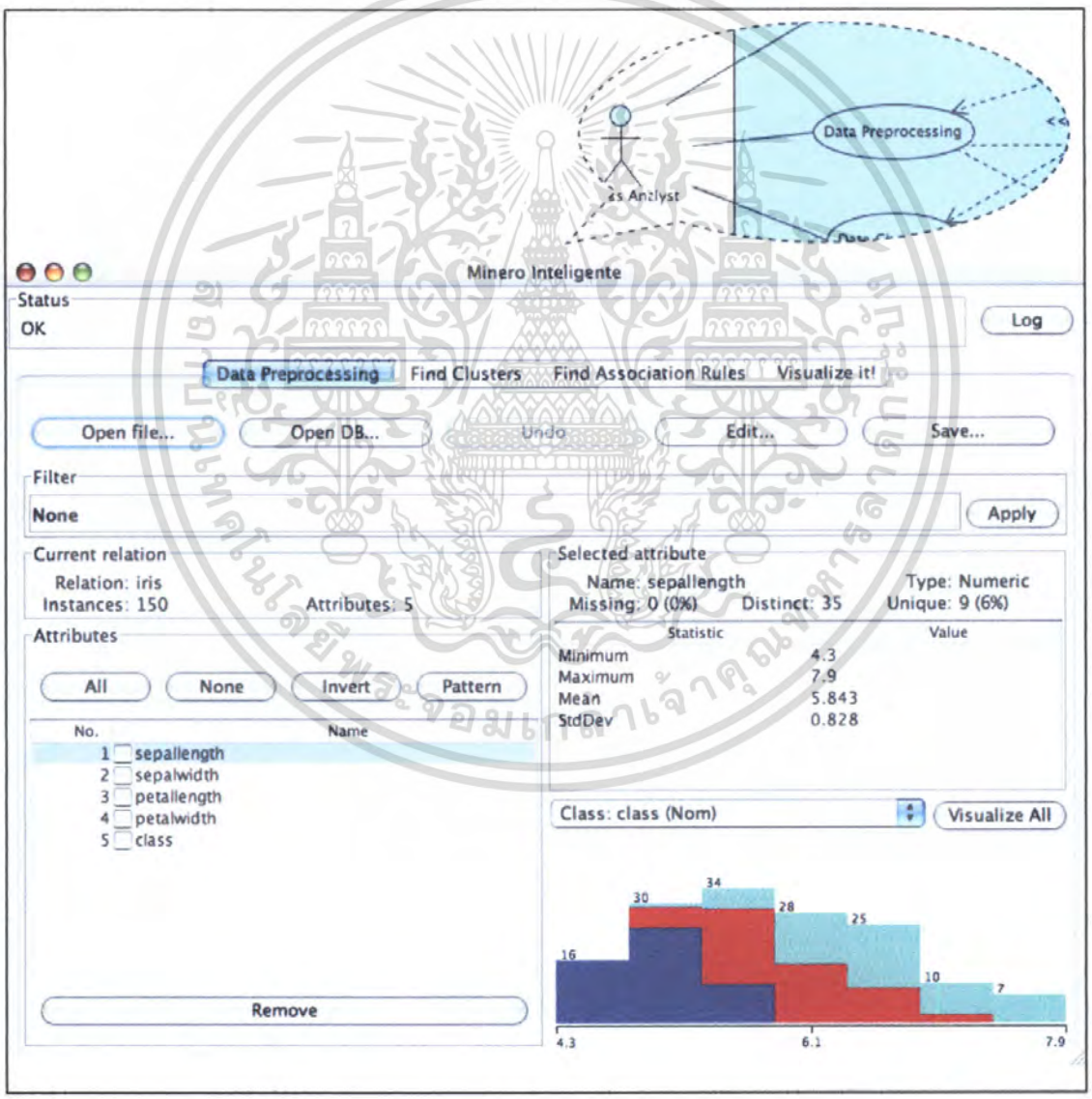
- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้

• สามารถแปลงกฎ และดึงชุดข้อมูลที่สอดคล้องกับกฎนั้นๆ ออกมาเป็นรายงานในรูปแบบต่างๆ ได้ เช่น PDF, Excel, Spreadsheet เป็นต้น

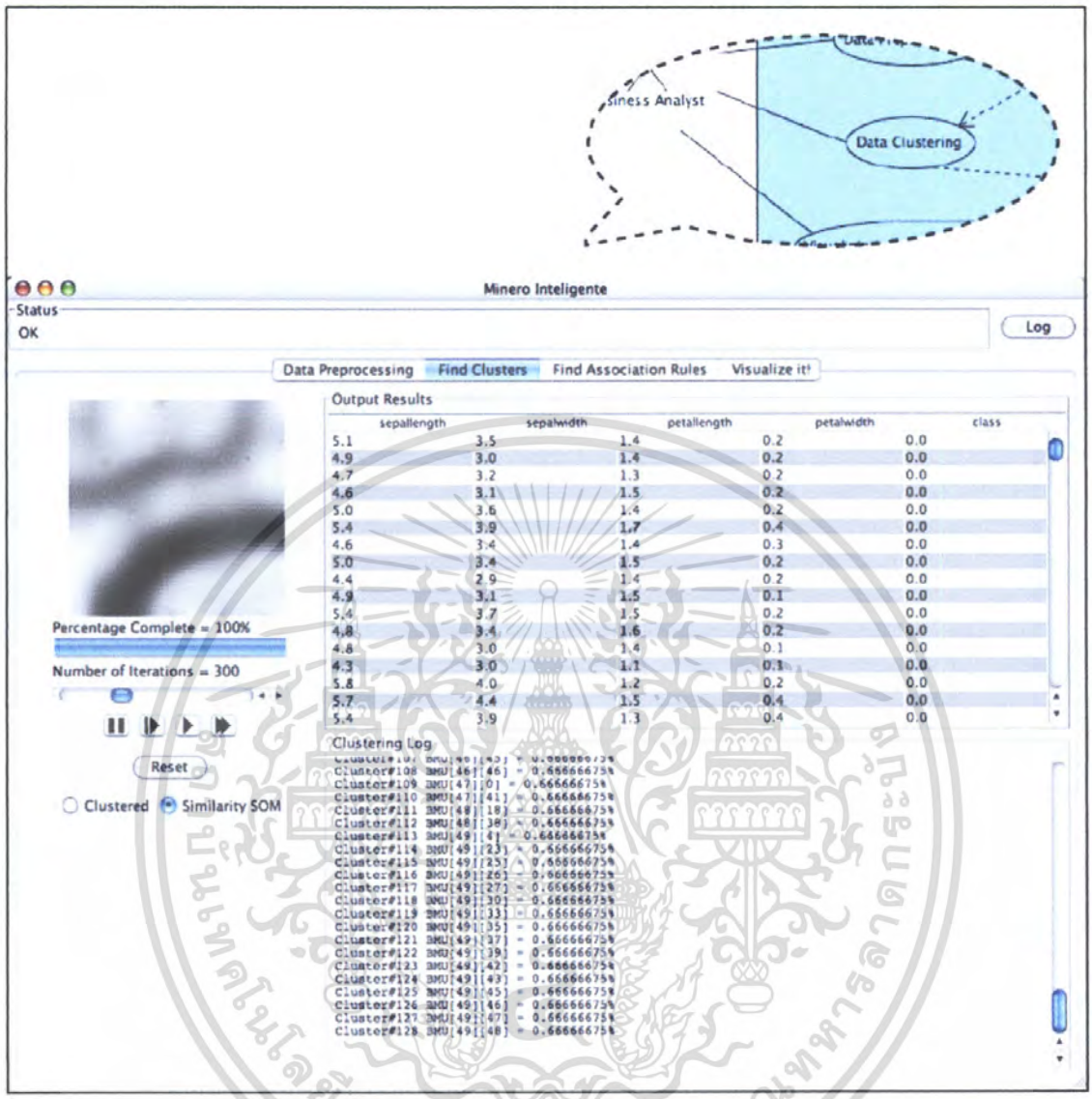
- สามารถเลือกรูปแบบการออกรายงานได้
- สามารถตรวจสอบดูก่อนแปลงไฟล์ได้

3.3.4 Interface Prototype

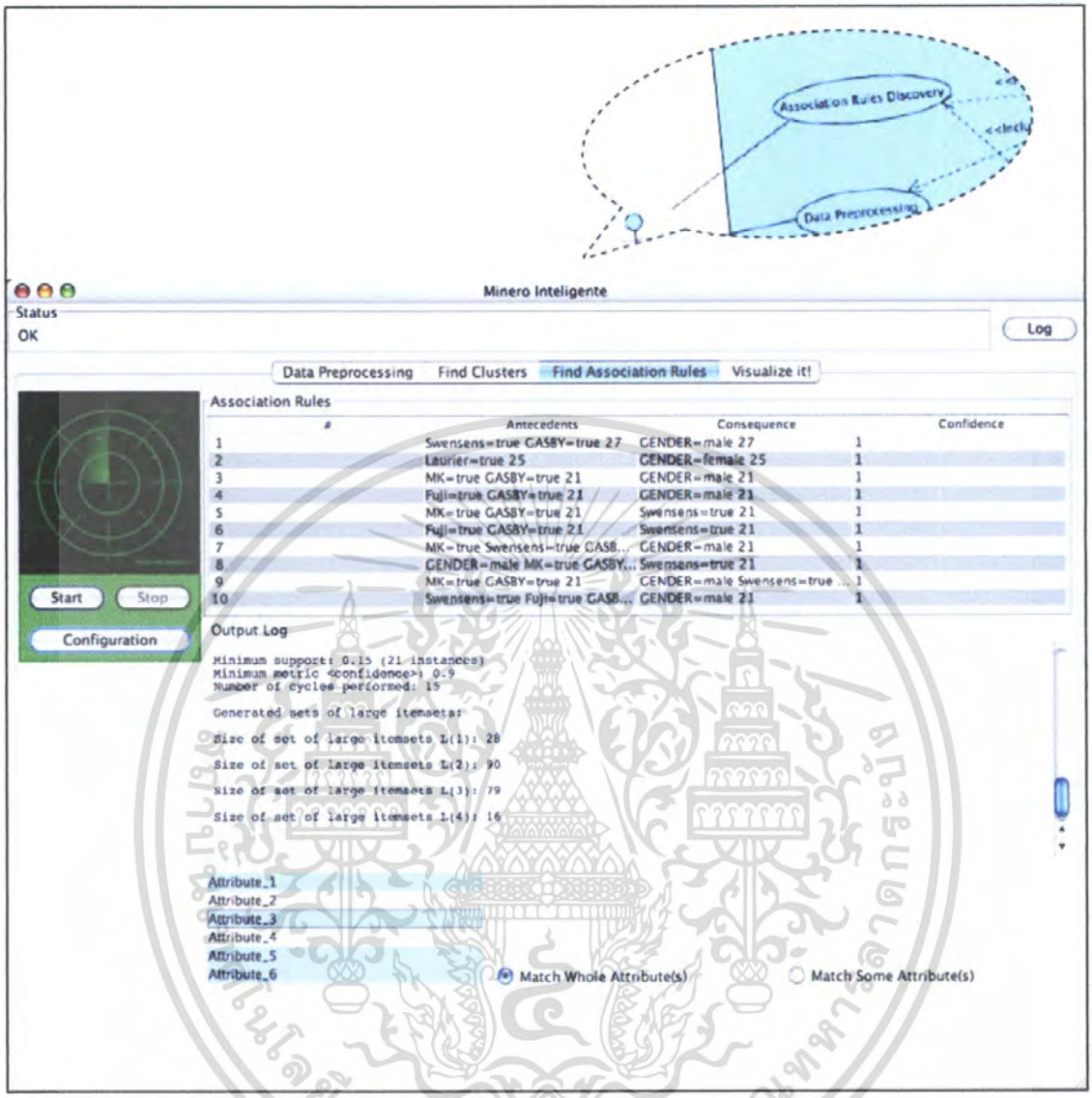
ภาพจำลองโปรแกรม Minero Intelligente ประกอบด้วยส่วนการทำงาน 4 ส่วนหลัก ได้แก่ Data Preprocessing, Data Clustering, Association Rules Discovery และ Visualizing Graphing and Reporting โดยแต่ละส่วนมีรายละเอียดดังแสดงในรูปที่ 3.2, 3.3, 3.4 และ 3.5



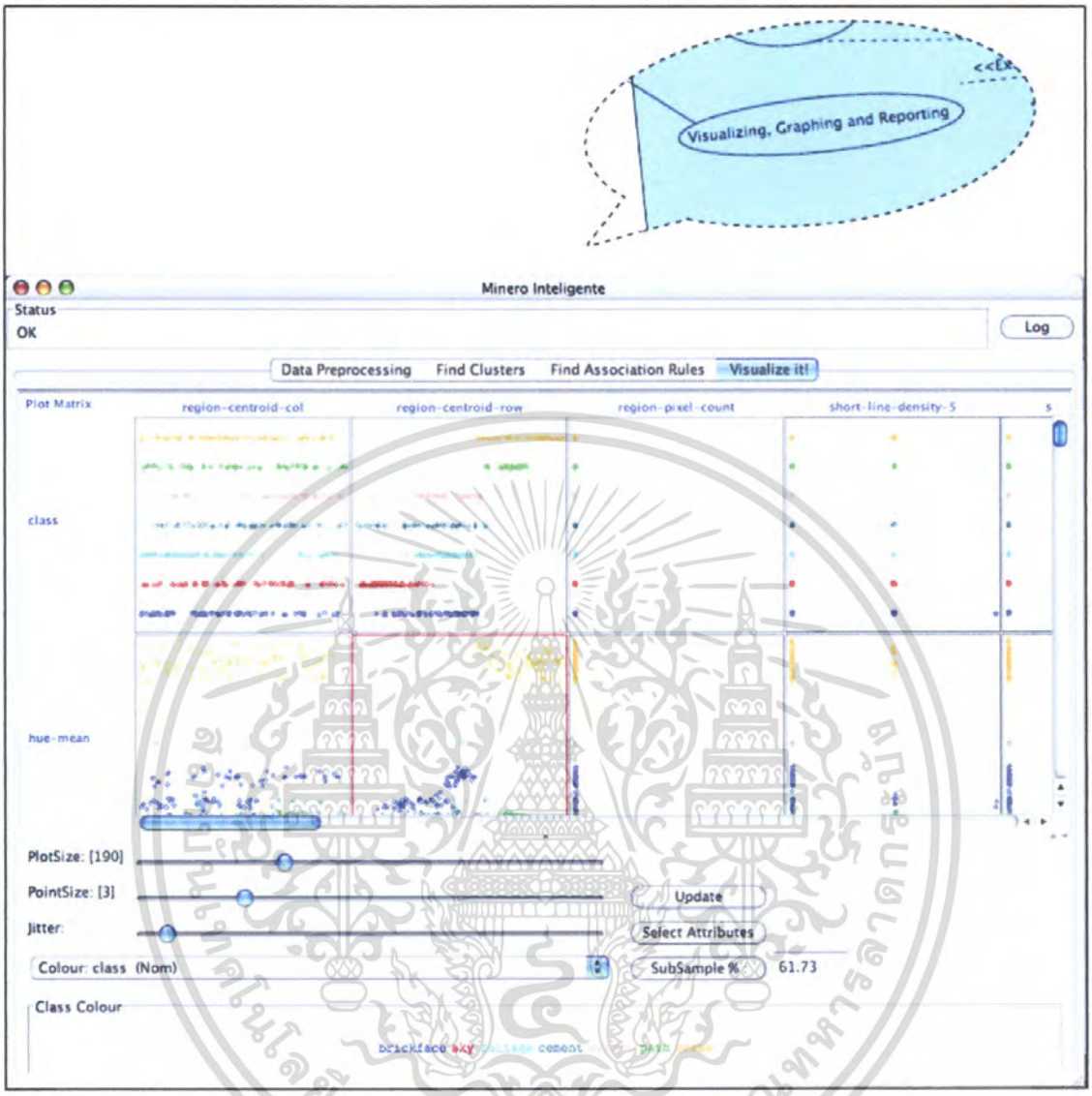
รูปที่ 3.7 แสดงภาพจำลองโปรแกรมในส่วน Data Preprocessing



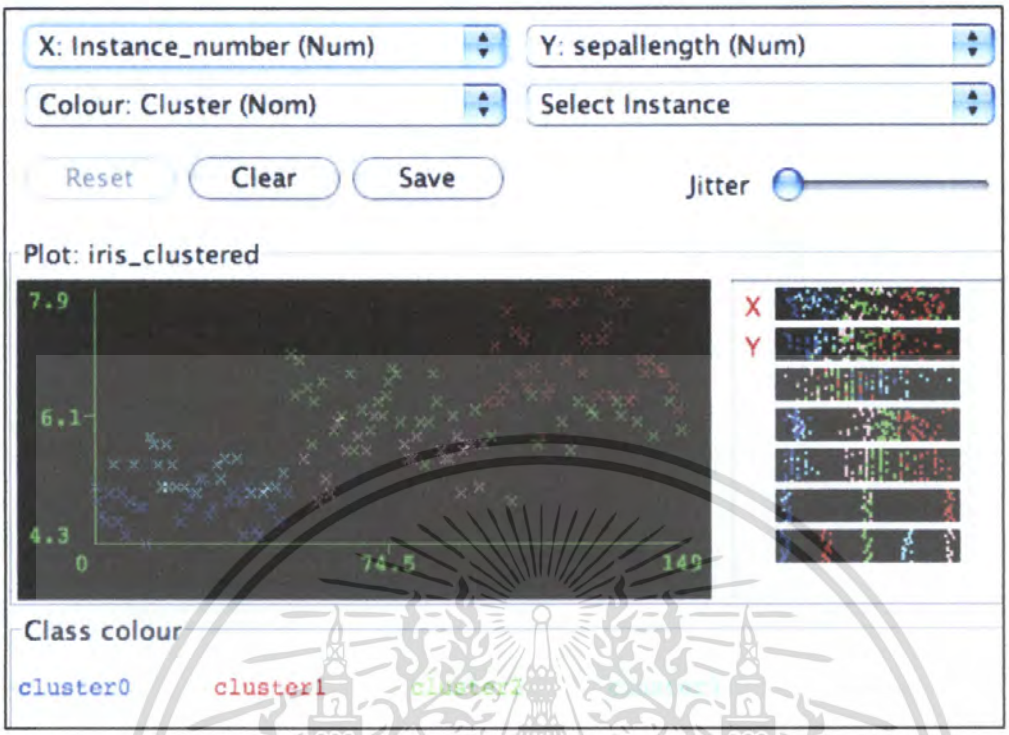
รูปที่ 3.8 แสดงภาพจำลองโปรแกรมในส่วน Data Clustering



รูปที่ 3.9 แสดงภาพจำลองโปรแกรมในส่วน Association Rules Discovery



รูปที่ 3.10 แสดงภาพจำลองโปรแกรมในส่วน Visualizing Graphing and Report



รูปที่ 3.11 แสดงหน้าต่างเพื่อเลือกแสดงผลการวิเคราะห์แบบต่างๆ

รูปที่ 3.11 แสดงหน้าต่างซึ่งมีกราฟหลังการวิเคราะห์ เมื่อเลือกภาพกราฟจากโปรแกรมในรูปที่ 3.10 หน้าต่างนี้สามารถเลือกผลการวิเคราะห์ให้แสดงในรูปแบบกราฟซึ่งมีความหมายของแกน X และ แกน Y ที่แตกต่างกัน ตามความต้องการของผู้ใช้งาน

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดสอบ SOM ด้วยชุดข้อมูลมาตรฐานไอริส

ในการทำทดสอบอัลกอริทึม Self-Organizing Map หรือ SOM จะใช้ชุดข้อมูลมาตรฐานเพื่อทดสอบความสามารถในการแยกข้อมูลออกเป็นกลุ่มย่อยของอัลกอริทึมดังกล่าว ชุดข้อมูลมาตรฐาน หรือ Standard Dataset คือ ชุดข้อมูลทั่วไปซึ่งประกอบด้วยแอทริบิวต์ (Attribute) และอินสแตนท์ (Instance) ที่แตกต่างกันเป็นจำนวนมาก ชุดข้อมูลมาตรฐานเป็นข้อมูลที่มีการใช้กันอย่างแพร่หลายเพื่อการทดลองต่างๆ ตัวอย่างเช่น ใช้ทดสอบอัลกอริทึม ใช้ทดสอบทางสถิติ ใช้เป็นฐานข้อมูลจำลอง เป็นต้น เนื่องจากมีวัตถุประสงค์ในการใช้งานที่แตกต่างกัน ดังนั้นจึงควรเลือกชุดข้อมูลที่เหมาะสมสำหรับงานวิจัยหรือการทดลองต่างๆ โดยพิจารณาจากความหมายของแอทริบิวต์และอินสแตนท์ สำหรับการทดลองนี้จะเลือกใช้ชุดข้อมูลมาตรฐานไอริสเพื่อทำการคัดแยกกลุ่มด้วยอัลกอริทึม SOM

4.1.1 ชุดข้อมูลมาตรฐานไอริส (Iris)

ชุดข้อมูลมาตรฐานไอริส (Iris) คือ ชุดข้อมูลที่เก็บรายละเอียดความกว้าง และความยาวของกลีบดอกไม้ชั้นใน และชั้นนอก ของดอกไอริส 3 สายพันธุ์ และมีการระบุสายพันธุ์ของดอกไอริสเอาไว้แล้วสำหรับแต่ละอินสแตนท์ ชุดข้อมูลประกอบด้วยอินสแตนท์จำนวน 150 อินสแตนท์ แบ่งเป็นคลาสได้ 3 คลาส คลาสละ 50 อินสแตนท์ คลาสแต่ละคลาสแทนชื่อสายพันธุ์ของดอกไอริส ได้แก่ คลาส Iris-Setosa, คลาส Iris-Versicolour และคลาส Iris-Virginica มีจำนวนแอทริบิวต์ทั้งหมด 4 ตัว ได้แก่ sepal length, sepal width, petal length, petal width แทนความกว้าง และความยาวของกลีบดอกไม้ชั้นใน และชั้นนอกในหน่วยเซนติเมตร ชุดข้อมูลมาตรฐานไอริสถือว่าเป็นชุดข้อมูลมาตรฐานที่สมบูรณ์ คือมีข้อมูลครบทุกอินสแตนท์และทุกแอทริบิวต์

No.	sepallength Numeric	sepalwidth Numeric	petallength Numeric	petalwidth Numeric	class Nominal
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa
9	4.4	2.9	1.4	0.2	Iris-setosa
10	4.9	3.1	1.5	0.1	Iris-setosa
11	5.4	3.7	1.5	0.2	Iris-setosa
12	4.8	3.4	1.6	0.2	Iris-setosa
13	4.8	3.0	1.4	0.1	Iris-setosa
14	4.3	3.0	1.1	0.1	Iris-setosa
15	5.8	4.0	1.2	0.2	Iris-setosa
16	5.7	4.4	1.5	0.4	Iris-setosa
17	5.4	3.9	1.3	0.4	Iris-setosa
18	5.1	3.5	1.4	0.3	Iris-setosa
19	5.7	3.8	1.7	0.3	Iris-setosa
20	5.1	3.8	1.5	0.3	Iris-setosa

รูปที่ 4.1 แสดงตัวอย่างชุดข้อมูลมาตรฐานไอริสอินสแตนท์ที่ 1 – 20

จากรูปที่ 4.1 แสดงตัวอย่างของชุดข้อมูลมาตรฐานไอริส ตั้งแต่อินสแตนท์ที่ 1 – 20 ซึ่งประกอบด้วยแอทริบิวต์ 4 ตัว ส่วนแอทริบิวต์ในคอลัมที่ชื่อ class จะแทนคลาสของข้อมูลทั้ง 3 คลาส จากภาพจะเห็นว่าตั้งแต่อินสแตนท์ที่ 1 – 20 เป็นคลาส Iris-Setosa ทั้งหมด ซึ่งชุดข้อมูลจริงๆ คือ อินสแตนท์ที่ 1 – 50 เป็นคลาส Iris-Setosa อินสแตนท์ที่ 51 – 100 เป็นคลาส Iris-Versicolour และอินสแตนท์ที่ 101 – 150 เป็นคลาส Iris-Virginica

ตารางที่ 4.1 แสดงจำนวนของอินสแตนท์และเปอร์เซ็นต์ของข้อมูลแต่ละคลาส

Class	No. of Instances	Percent
Iris-Setosa	50	33%
Iris-Versicolour	50	33%
Iris-Virginica	50	33%

จากตารางที่ 4.1 แสดงจำนวนของอินสแตนท์และเปอร์เซ็นต์ของข้อมูลทั้ง 3 คลาสของชุดข้อมูลมาตรฐานไอริส ซึ่งทั้ง 3 คลาสดังกล่าว มีจำนวนอินสแตนท์เท่ากัน คือ 50 อินสแตนท์ คิดเป็นเปอร์เซ็นต์ได้คลาสละ 33 เปอร์เซ็นต์เท่ากัน

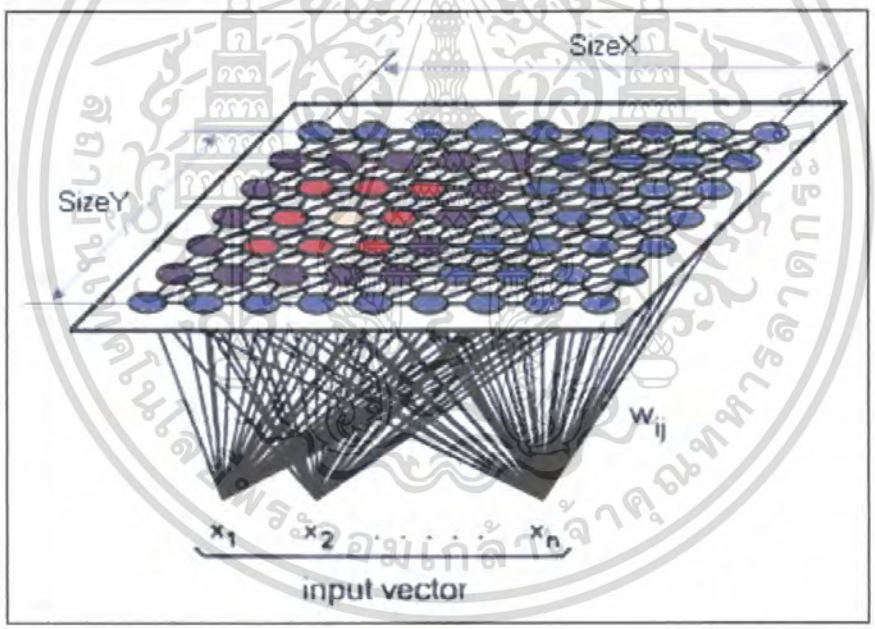
	Minimum	Maximum	Mean	StdDev	Missing	Distinct	Unique
sepallength	4.3	7.9	5.843	0.828	0 (0%)	35	9 (6%)
sepalwidth	2	4.4	3.054	0.434	0 (0%)	23	5 (3%)
petallength	1	6.9	3.759	1.764	0 (0%)	43	10 (7%)
petalwidth	0.1	2.5	1.199	0.763	0 (0%)	22	2 (1%)

รูปที่ 4.2 แสดงรายละเอียดของแต่ละแอทริบิวต์

จากรูปที่ 4.2 แสดงรายละเอียดของข้อมูลในแอทริบิวต์แต่ละตัว ได้แก่ ค่าต่ำสุด ค่าสูงสุด ค่าเฉลี่ย ค่าเบี่ยงเบนมาตรฐาน จำนวนข้อมูลที่ไม่สมบูรณ์ จำนวนค่าที่แตกต่างกันทั้งหมดของแอทริบิวต์ และจำนวนข้อมูลที่ปรากฏครั้งเดียว

4.1.2 โมเดลสำหรับการวิเคราะห์ข้อมูล

ในการวิเคราะห์ข้อมูลแต่ละครั้งจะใช้โมเดลที่แตกต่างกันออกไปตามจุดประสงค์ของผู้วิเคราะห์ ดังนั้นจึงต้องมีการกำหนดโมเดลทุกครั้งก่อนเริ่มทำการวิเคราะห์



รูปที่ 4.3 แสดงแผนภาพโคโฮเนน (Kohonen Featuremap)

จากรูปที่ 4.3 แสดงแผนภาพโคโฮเนน ซึ่งประกอบด้วย ชั้นของโครงข่ายโคโฮเนน และชั้นของอินพุต ชั้นของโครงข่ายมีขนาดความกว้าง เท่ากับ SizeY และความยาว เท่ากับ SizeX ผลคูณระหว่างความกว้างและความยาวจะได้ออกมาเป็นจำนวนนิวรอนทั้งหมดในโครงข่าย ส่วนชั้นของอินพุต มีอินพุตคือ เวกเตอร์ x มีจำนวนอินพุต เท่ากับ n ตัว ซึ่งมีการเชื่อมต่อโหนดของอินพุตทุกโหนดกับโหนดของนิวรอนในชั้นของโครงข่ายโคโฮเนน โดยแต่ละเส้นเชื่อมจะมีค่าน้ำหนัก w_{ij} กำกับอยู่ เพื่อใช้ในกระบวนการทำงานของอัลกอริทึม

ข้อมูลที่จะต้องกำหนดสำหรับสร้างโมเดลในขั้นตอนนี้ได้แก่

- ขนาดของโครงข่าย คือ ความกว้าง กำหนดค่า "10" และ ความยาว กำหนดค่า "10"
- จำนวนอินพุต กำหนดค่า "4" ตามจำนวนแอทริบิวต์
- ช่วงระหว่างค่าอินพุตที่น้อยที่สุดและมากที่สุด กำหนดค่าดังรูปที่ 4.2
- Initial learning rate กำหนดค่า "0.6"
- Initial activation area กำหนดค่า 3.0
- Stop area กำหนดค่า 0.5

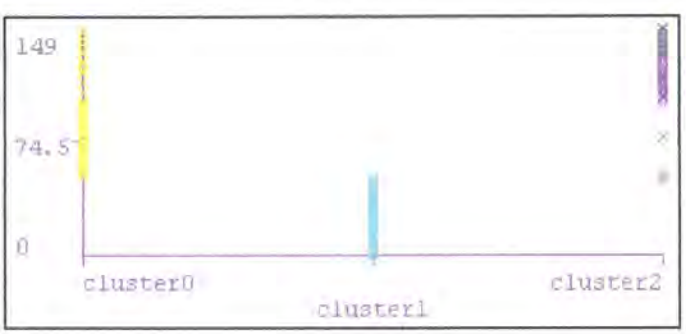
4.1.3 กระบวนการทำงานของอัลกอริทึม

การทำงานของอัลกอริทึม

1. อัลกอริทึมจะกำหนดค่าน้ำหนักเริ่มต้นให้แต่ละโหนดในโครงข่ายโดยวิธีการสุ่ม
2. อัลกอริทึมจะเลือกข้อมูลเพื่อเรียนรู้ 1 อินสแตนท์จากชุดข้อมูลทั้งหมด ด้วยวิธีการสุ่ม
3. คำนวณหาค่าความแตกต่างระหว่างค่าเวกเตอร์อินพุต กับ ค่าเวกเตอร์ w ของทุกโหนด โดยใช้ Euclidean Distance Function แล้วหาโหนด BMU
4. คำนวณหาค่ารัศมีรอบข้างโหนด BMU บนชั้นโครงข่าย เพื่อหาโหนดที่ประชิดกับ BMU ค่ารัศมีในการเรียนรู้จะลดลงไปเรื่อยๆ ตามฟังก์ชันเวลาที่กำหนด โหนดใดที่อยู่ในวงรัศมีของ BMU จะเรียกว่า Neighbourhood
5. ในแต่ละโหนด Neighbourhood จะมีการปรับค่าเวกเตอร์ w ด้วย เพื่อให้โหนดเหล่านั้นมีค่าคล้ายเวกเตอร์อินพุตมากขึ้น โหนด Neighbourhood ที่อยู่ใกล้กับ BMU มากกว่าก็จะถูกปรับ w ด้วยค่ามากกว่า
6. ทำซ้ำข้อ 2-6 จนค่ารัศมีลู่เข้าใกล้ค่าคงที่รัศมี

4.1.4 ผลการวิเคราะห์ข้อมูล

ผลการวิเคราะห์ชุดข้อมูลมาตรฐานไอริส เมื่อให้นิวรอนทุกโหนดได้เรียนรู้อินพุตทั้งหมดหลายๆ รอบแล้ว พบว่ามีการเรียนรู้ทั้งหมด 2,303 รอบ สามารถแบ่งกลุ่มได้เป็น 3 กลุ่มย่อย (3 Clusters)



รูปที่ 4.4 แสดงผลการทดลองในรูปกราฟความสัมพันธ์ระหว่างกลุ่มข้อมูลและหมายเลขอินสแตนท์

จากรูปที่ 4.4 แสดงผลการทดลองในรูปกราฟความสัมพันธ์ระหว่างกลุ่มข้อมูลในแนวแกน X และหมายเลขอินสแตนท์ของชุดข้อมูลทั้งหมดในแนวแกน Y โดยแทนข้อมูลกลุ่มย่อยทั้ง 3 กลุ่ม ที่อัลกอริทึมสามารถคัดแยกได้ด้วยสี 3 สี คือ สีเหลืองแทนกลุ่มย่อยที่ 0 สีน้ำเงินแทนกลุ่มย่อยที่ 1 และสีชมพูแทนกลุ่มย่อยที่ 2

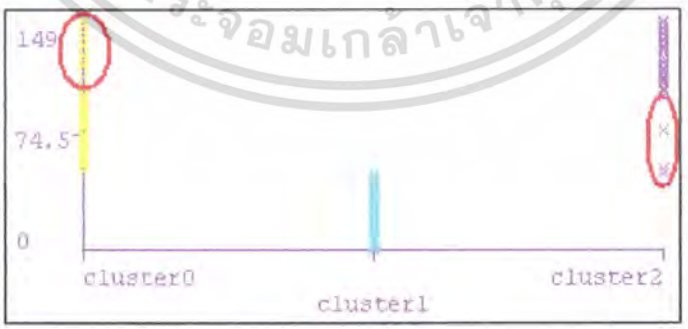
ตารางที่ 4.2 แสดงจำนวนของอินสแตนท์และเปอร์เซ็นต์ของข้อมูลแต่ละกลุ่มย่อย

Cluster	No. of Instances	Percent
Cluster 0	61	41%
Cluster 1	50	33%
Cluster 2	39	26%

จากตารางที่ 4.2 แสดงจำนวนของอินสแตนท์และเปอร์เซ็นต์ของข้อมูลทั้ง 3 กลุ่มย่อยของชุดข้อมูลมาตรฐานไอริสที่อัลกอริทึม SOM สามารถคัดแยกได้ ซึ่งทั้ง 3 กลุ่มย่อยดังกล่าว มีจำนวนอินสแตนท์เท่ากับ 61, 50 และ 39 อินสแตนท์ ตามลำดับ คิดเป็นเปอร์เซ็นต์ได้ 41%, 33% และ 26% ตามลำดับ

จากรูปที่ 4.4 แสดงผลการทดลองในรูปกราฟความสัมพันธ์ระหว่างกลุ่มข้อมูล และหมายเลขอินสแตนท์ โดยส่วนที่ผิดพลาดถูกวงกลมด้วยสีแดงในรูปที่ 4.5

จากผลการทดสอบอัลกอริทึมเพื่อหาข้อมูลกลุ่มย่อยในตารางที่ 4.2 เมื่อเปรียบเทียบกับชุดข้อมูลจริงที่มีการแบ่งกลุ่มเป็นคลาสเอาไว้แล้วในตารางที่ 4.1 พบว่ามีความผิดพลาดในการคัดแยกดังแสดงในตารางที่ 4.3



รูปที่ 4.5 แสดงผลส่วนที่เกิดความผิดพลาดด้วยวงกลมสีแดง

ตารางที่ 4.3 แสดงเปอร์เซ็นต์ความถูกต้องเปรียบเทียบระหว่างผลการคัดแยกและคลาสจริง

Cluster	Equivalent Class	Number of Instance		Percent Accuracy
		Real Classes	Actual Clusters	
Cluster 0	Iris-Versicolour	50	61	78%
Cluster 1	Iris-Setosa	50	50	100%
Cluster 2	Iris-Virginica	50	39	78%

4.2 การทดสอบอัลกอริทึม Apriori ด้วยชุดข้อมูลมาตรฐานซอยบีน (Soybean)

ในการทำทดสอบอัลกอริทึม Apriori จะใช้ชุดข้อมูลมาตรฐานเพื่อทดสอบความสามารถในการค้นหาความสัมพันธ์เชิงกฎของอัลกอริทึมดังกล่าว การเลือกชุดข้อมูลมาตรฐานควรคำนึงถึงความเหมาะสมกับงานวิจัยหรือการทดลองต่างๆ โดยพิจารณาจากความหมายของแอทริบิวต์และอินสแตนท์ สำหรับทดสอบอัลกอริทึม Apriori นี้จะเลือกใช้ชุดข้อมูลมาตรฐานซอยบีน (Soybean) เพื่อทำการค้นหาความสัมพันธ์เชิงกฎ

4.2.1 ชุดข้อมูลมาตรฐานซอยบีน (Soybean)

ชุดข้อมูลมาตรฐานซอยบีน (Soybean) คือ ชุดข้อมูลที่เก็บรายละเอียดเกี่ยวกับต้นถั่วเหลือง ไม่ว่าจะเป็นลักษณะของใบไม้, ผลถั่วเหลือง, เมล็ด, ราก, ดิน หรือแม้กระทั่งการเจริญเติบโต ซึ่งข้อมูลทั้งหมดเก็บอยู่ในรูปแบบที่เป็นชื่อเฉพาะ (Nominal) ไม่ใช่ค่าที่เป็นตัวเลข โดยชุดข้อมูลประกอบด้วยอินสแตนท์จำนวน 683 อินสแตนท์ แบ่งเป็นคลาสได้ 19 คลาส มีจำนวนแอทริบิวต์ทั้งหมด 35 ตัว ยกตัวอย่างเช่น leaves, stem, fruit spots, seed, roots, lodging เป็นต้น ชุดข้อมูลมาตรฐานซอยบีนถือว่าเป็นชุดข้อมูลที่ได้มาตรฐาน เนื่องจากมีเรคอดเป็นจำนวนมาก และมีรายละเอียดสำหรับแต่ละแอทริบิวต์ค่อนข้างมาก ทำให้มีความชัดเจนในแต่ละเรคอดสูง

No.	date Nominal	temp Nominal	plant-growth Nominal	leaves Nominal	stem Nominal	lodging Nominal	seed Nominal	roots Nominal	class Nominal
1	july	norm	norm	abnorm	norm	yes	norm	norm	bacterial-blight
2	august	norm	norm	abnorm	norm	yes	norm	norm	bacterial-blight
3	june	norm	norm	abnorm	norm	yes	norm	norm	bacterial-blight
4	august	gt-norm	norm	abnorm	norm	yes	norm	norm	bacterial-blight
5	june	gt-norm	norm	abnorm	norm	yes	norm	galls-c...	bacterial-pustule
6	july	lt-norm	abnorm	abnorm	norm	yes	abnorm	norm	bacterial-pustule
7	june	lt-norm	norm	abnorm	norm	yes	norm	norm	bacterial-pustule
8	august	norm	norm	abnorm	norm	yes	norm	rotted	bacterial-pustule
9	july	norm	norm	abnorm	norm	yes	abnorm	rotted	bacterial-pustule
10	july	lt-norm	norm	abnorm	norm	yes	norm	rotted	bacterial-pustule
11	july	norm	abnorm	abnorm	norm	yes	abnorm	norm	bacterial-pustule
12	july	norm	norm	abnorm	norm	yes	norm	rotted	bacterial-pustule
13	august	norm	norm	abnorm	norm	yes	abnorm	norm	bacterial-pustule
14	septe...	norm	norm	abnorm	norm	yes	norm	rotted	bacterial-pustule
15	october	lt-norm	norm	norm	abnorm	no	abnorm	norm	purple-seed-stain
16	october	lt-norm	norm	abnorm	abnorm	yes	abnorm	norm	purple-seed-stain
17	august	norm	norm	norm	norm	yes	abnorm	norm	purple-seed-stain
18	august	norm	norm	abnorm	norm	no	abnorm	norm	purple-seed-stain
19	august	lt-norm	norm	abnorm	norm	yes	abnorm	norm	anthracnose
20	october	gt-norm	norm	abnorm	abnorm	yes	abnorm	norm	anthracnose

รูปที่ 4.6 แสดงตัวอย่างชุดข้อมูลมาตรฐานซอยบีนอินสแตนท์ที่ 1 – 20

4.2.2 กระบวนการหา Frequent Itemset

ก่อนเริ่มประมวลผลด้วยอัลกอริทึม Apriori ควรจะตั้งค่าต่างๆ ตามความต้องการของผู้ใช้งานเสียก่อน ค่าขอบเขตในการประมวลผลที่สามารถปรับได้มีหลายค่าด้วยกัน ยกตัวอย่างเช่น numRules หมายถึง จำนวนกฎทั้งหมดที่ต้องการให้ประมวลผลออกมาเป็นต้น

หลังจากปรับตั้งค่าทุกอย่างตามความต้องการเรียบร้อยแล้ว อัลกอริทึมจะทำการคำนวณหา Frequent Itemset ทั้งหมด โดยเริ่มจากการหา Candidate 1-itemset ก่อน แล้วตัดไอเท็มเซ็ทที่มีค่าสนับสนุนน้อยกว่าค่าสนับสนุนต่ำสุดที่กำหนดไว้ออกไป เหลือเพียงไอเท็มเซ็ทที่ต้องการ จะได้เป็น Frequent 1-itemset จากนั้นทำการหา Candidate itemset และ Frequent itemset ลำดับต่อไปอีกเรื่อยๆ จนกระทั่งไม่สามารถหาต่อได้แล้ว แสดงว่าจบกระบวนการหา Frequent Itemset แล้วไอเท็มเซ็ททั้งหมดที่เกิดขึ้นตั้งแต่ Frequent 1-itemset จนถึง Frequent k-itemset ถือเป็น Frequent Itemset

ยกตัวอย่างการหา Frequent 1-itemset ของแอทริบิวต์ seed กำหนดเปอร์เซ็นต์สนับสนุนต่ำสุดมีค่า 40% จากชุดข้อมูลทั้งหมด 683 อินสแตนท์ 40% คิดเป็น 274 อินสแตนท์ ดังนั้นถ้าแอทริบิวต์ seed มีปรากฏมากกว่า 274 อินสแตนท์ แอทริบิวต์ seed ใน Candidate 1-itemset ก็จะสามารถผ่านไปได้ ฉะนั้นแอทริบิวต์ seed ก็ถือเป็นสมาชิกของ Frequent 1-itemset เป็นต้น

4.2.3 กระบวนการสร้างกฎความสัมพันธ์

เมื่อได้ชุดไอเท็มเซ็ททั้งหมดจากกระบวนการหา Frequent Itemset แล้ว ต่อไปเป็นกระบวนการหากฎความสัมพันธ์ โดยการนำ Frequent Itemset ที่ได้มาสร้างกฎความเชื่อมโยง โดย

การสร้างกฎ (R) จะทำให้อยู่ในรูปของ X □ Y และทำการคำนวณหาค่าความเชื่อมั่นของแต่ละกฎที่สร้างขึ้นมาเพื่อคัดเลือกกฎมาใช้

Associator output

```
Best rules found:

    1. class=phytophthora-rot 88 ==> canker-lesion=dk-brown-blk 88    conf:(1)
    2. canker-lesion=dk-brown-blk fruit-pods=diseased 87 ==> stem-cankers=above-sec-nde 87    conf:(1)
    3. roots=rotted class=phytophthora-rot 68 ==> canker-lesion=dk-brown-blk 68    conf:(1)
    4. canker-lesion=dk-brown-blk roots=rotted 68 ==> class=phytophthora-rot 68    conf:(1)
    5. external-decay=firm-and-dry fruit-pods=diseased 76 ==> stem-cankers=above-sec-nde 75    conf:(0.99)
    6. fruit-spots=colored 75 ==> fruit-pods=diseased 73    conf:(0.97)
    7. stem-cankers=above-sec-nde fruit-pods=diseased 103 ==> canker-lesion=dk-brown-blk 87    conf:(0.84)
    8. fruiting-bodies=present 104 ==> stem-cankers=above-sec-nde 84    conf:(0.81)
    9. fruit-pods=diseased 130 ==> stem-cankers=above-sec-nde 103    conf:(0.79)
   10. roots=rotted 86 ==> canker-lesion=dk-brown-blk 68    conf:(0.79)
   11. roots=rotted 86 ==> class=phytophthora-rot 68    conf:(0.79)
   12. roots=rotted 86 ==> canker-lesion=dk-brown-blk class=phytophthora-rot 68    conf:(0.79)
   13. class=phytophthora-rot 88 ==> roots=rotted 68    conf:(0.77)
   14. canker-lesion=dk-brown-blk class=phytophthora-rot 88 ==> roots=rotted 68    conf:(0.77)
   15. class=phytophthora-rot 88 ==> canker-lesion=dk-brown-blk roots=rotted 68    conf:(0.77)
   16. stem-cankers=above-sec-nde external-decay=firm-and-dry 98 ==> fruit-pods=diseased 75    conf:(0.77)
   17. stem-cankers=above-sec-nde fruit-pods=diseased 103 ==> external-decay=firm-and-dry 75    conf:(0.73)
   18. external-decay=firm-and-dry 135 ==> stem-cankers=above-sec-nde 98    conf:(0.73)
   19. stem-cankers=above-sec-nde canker-lesion=dk-brown-blk 120 ==> fruit-pods=diseased 87    conf:(0.73)
   20. canker-lesion=dk-brown-blk 177 ==> stem-cankers=above-sec-nde 120    conf:(0.68)
```

รูปที่ 4.7 แสดงผลการค้นหากฎความสัมพันธ์ด้วยอัลกอริทึม Apriori

ผลลัพธ์ที่ได้ออกมาเป็นกฎความสัมพันธ์ซึ่งเรียงลำดับกฎแต่ละข้อตามค่าความเชื่อมั่น ซึ่งคำนวณจากสูตร Confidence (R) = (Support (X□Y) / Support (X)) *100% ดังที่ได้กล่าวไว้แล้วในบทที่ 2 โดยค่าความเชื่อมั่นนี้มีค่าตั้งแต่ 1.0 ถึง 0.0 แทนความเชื่อมั่นได้มากที่สุดไปถึงน้อยที่สุดตามลำดับ ซึ่งจะเห็นว่ามีวงเล็บบอกค่าความเชื่อมั่นนี้ในตอนท้ายของกฎแต่ละข้อด้วย

ยกตัวอย่าง กฎข้อที่ 2 canker-lesion = dk-brown-blk, fruit-pods = diseased 87 □ stem-cankers = above-sec-nde, 87 conf:(1) จากข้อความดังกล่าว สามารถแปลเป็นภาษาที่เข้าใจง่าย ได้ใจความว่า ถ้า canker-lesion มีค่า dk-brown-blk และ fruit-pods มีค่า diseased ซึ่งมีทั้งหมด 87 เรคอด แล้ว cankers มีค่า above-sec-nde มีจำนวน 87 เรคอด ด้วย ซึ่งคำนวณค่าความเชื่อมั่นออกมาได้ค่าเท่ากับ 1 แสดงว่ากฎข้อดังกล่าวนี้มีความน่าเชื่อถือสูงมาก เป็นต้น

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 ข้อสรุป

การดำเนินงานเป็นไปด้วยดีแม้ว่าจะมีปัญหาในการทำงานบ้าง เช่น การเก็บความต้องการแง่มุมการใช้งานในธุรกิจหลากหลายประเภท ต้องปรึกษาผู้เชี่ยวชาญด้านธุรกิจในสาขาต่างๆ งานบริการ การตลาด งานขาย ธุรกิจที่ได้ไปสำรวจความต้องการมาแล้ว เช่น โรงพยาบาลเมืองสมุทรฯ, เรือนศันสนีย์ เดลิเวอรี่สปา รวมไปถึงการสังเกตุและจดบันทึกศึกษาโครงสร้างการออกแบบทั้งหมดของระบบการประมวลผลข้อมูลต่างๆ ที่ถูกสร้างขึ้นมาใช้ในเชิงพาณิชย์และการวิจัย ซึ่งในสภาวะปัจจุบันเทคโนโลยีทางข้อมูลข่าวสารได้ก้าวไกลอย่างรวดเร็ว และข้อมูลข่าวสารในแต่ละวันได้เกิดขึ้นอย่างมหาศาล หากเรามองลงไปที่องค์กรหนึ่งๆ จะพบว่าในแต่ละวันจะมีข้อมูล ข่าวสารอาทิเช่น ข้อมูลการซื้อขายสิ่งของต่างๆ จะเกิดขึ้นจำนวนมาก ข้อมูลของลูกค้าที่เข้ามาใช้บริการหรือจับจ่ายใช้สอยในห้างสรรพสินค้า ดังที่ยกตัวอย่าง จะเห็นได้ว่าข้อมูลเหล่านี้จะถูกเก็บไว้ในฐานข้อมูล ซึ่งคุณสมบัติของข้อมูลเหล่านี้จะมีความรู้ที่ซ่อนอยู่ให้เราค้นหาออกมาได้อีกมากมายเพื่อนำไปใช้ประโยชน์ เช่น การนำไปวิเคราะห์ คาดการณ์ และประยุกต์ใช้เป็นแนวทางในการทำงานได้อีกด้วย สำหรับการทำเหมืองข้อมูลก็เป็นวิธีการในการนำข้อมูลจำนวนมหาศาลที่เรามีอยู่นั้น มาผ่านการทำกระบวนการของเหมืองข้อมูลเพื่อให้ได้ผลลัพธ์ที่เป็นแบบแผน หรือเป็นกฎเกณฑ์ที่จะใช้อนุมานความเป็นไปได้ของความรู้ที่แฝงอยู่ในข้อมูลนั้นออกมา ซึ่งเราอาจจะต้องเกี่ยวข้องกับหลักการทางสถิติ และปัญญาประดิษฐ์เพื่อนำมาประยุกต์ใช้ให้ได้ซึ่งวิธีการในการดึงความรู้ที่แฝงอยู่นั้นออกมาใช้ให้เกิดประโยชน์ และสมเหตุสมผลมากที่สุด

ในโครงงานเรื่อง เหมืองข้อมูลอัจฉริยะสำหรับการแบ่งกลุ่ม และค้นหาความสัมพันธ์เชิงกฎภายในฐานข้อมูลสารสนเทศ ฉบับนี้นั้น จึงมุ่งเน้นการวิเคราะห์ข้อมูลเชิงธุรกิจ ซึ่งเป็นการใช้เทคโนโลยีทางศาสตร์ปัญญาประดิษฐ์มาวิเคราะห์ข้อมูล และประมวลผลออกมาได้เป็นความรู้ที่สามารถนำไปใช้ประโยชน์ต่อได้ โครงงานนี้มุ่งวิจัยกระบวนการในการทำเหมืองข้อมูล 2 กระบวนการหลัก นั่นคือ การคัดแยกกลุ่มข้อมูล หรือ Data Clustering และ การค้นหาความสัมพันธ์เชิงกฎ หรือ Association Rules Discovery โดยใช้อัลกอริทึมหลักในการประมวลผล 2 อัลกอริทึมด้วยกัน คือ Self-Organizing Map สำหรับการคัดแยกกลุ่มข้อมูล และ Apriori สำหรับการค้นหาความสัมพันธ์เชิงกฎ

5.2 ข้อเสนอแนะ

ในการพัฒนาแอพลิเคชั่นผลลัพธ์ได้ออกมาเป็นข้อมูลรายงานต่อผู้ใช้ ซึ่งหากนำข้อมูลเหล่านี้ไปพัฒนาสร้างเป็นระบบ Recommendation จะก่อให้เกิดประโยชน์อย่างมาก โดยเฉพาะทางเชิงธุรกิจ เช่น พัฒนาระบบแนะนำโฆษณา หรือหนังสือทางสื่อออนไลน์โดยเฉพาะสินค้าแบบที่ลูกค้าสนใจและมีแนวโน้มที่จะซื้อผ่านทางระบบออนไลน์มากที่สุด, พัฒนาระบบแนะนำข่าวสารที่เป็น Digital Content ให้กับลูกค้าทางสื่อเคเบิ้ลทีวี เป็นต้น

ในส่วนของตัวอัลกอริทึมในการประมวลผลหลักทั้งสองตัวนี้ ยังสามารถพัฒนาให้มีความสามารถเพิ่มมากขึ้นได้อีก ให้มีประสิทธิภาพไปจนสามารถทำงานกับแหล่งเก็บข้อมูลที่มีขนาดใหญ่ขึ้น จำพวก Data Warehouse ซึ่งมีขนาดใหญ่มากๆ ได้

บรรณานุกรม

Michael Negnevitsky. 2002. **Artificial Intelligence A Guide to Intelligent Systems.** Harlaw : Addison-Wesley

Dali Wang, Habtom Ressom, Mohamad Musavi, Cristian Domnisoru. 2002. **Double Self-Organizing Maps to Cluster Gene Expression Data**.

Juha Vesanto and Esa Alhoniemi, *Student Member, IEEE*. 2000. **Clustering of the Self-Organizing Maps**

ธนาวินท์ รักธรรมานนท์, ชิดชนก ส่งศิริ, กฤษณะ ไวยมัย. 2545. **ระบบสืบค้นความรู้บนฐานข้อมูลเชิงวัตถุ : การหากฎความสัมพันธ์บนฐานข้อมูลเชิงวัตถุแบบแน่น**

เจริญ ตั้งเจริญสมุทร ให้สัมภาษณ์, 20 สิงหาคม 2550. สดุดี โกมลหทัย ผู้สัมภาษณ์. **การบริหารจัดการข้อมูลผู้ป่วยในโรงพยาบาล.** โรงพยาบาลเมืองสมุทรฯ ปากน้ำ

Neural Networks with JAVA [Online]. Available : http://www.nnwj.de/contents.html

Self Organizing Map Tutorial System. [Online]. Available : http://www.sis.pitt.edu/~ssyn/som/som.html

Kohonen-Netz. [Online]. Available : http://www.lohninger.com/datalab/helpgerm/kohonen_map.htm

Kohonen's Self Organizing Feature Maps. [Online]. Available : http://www.ai-junkie.com/ann/som/som1.html

Association Rules Introductory Overview. [Online]. Available : http://www.statsoft.com/textbook/stathome.html?stassrul.html&1