การพัฒนาโปรแกรมค้นหา Association Rule โดยใช้ DHP Algorithm

The Development of an Association Rule Discovery Software using DHP Algorithm

โดย

นายฉัตร วัฒนศิริเกียรติ

รหัส 43067161

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน
หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
ภาคเรียนที่ 1 ปีการศึกษา 2545
คณะเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาโปรแกรมค้นหา Association Rule โดยใช้ DHP Algorithm
นักศึกษา	นายฉัตร วัฒนศิริเกียรติ
อาจารย์ที่ปรึกษา	ดร.วรพจน์ กรีสุระเดช
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2545

บทคัดย่อ

ปัจจุบันการดำเนินธุรกิจมีการแข่งขันกันสูง ดังนั้นบริษัทที่ประกอบธุรกิจจำนวนมากมักจะทำการวิเคราะห์ข้อมูลที่ผ่านมาในอดีตเพื่อหาข้อมูลที่เกี่ยวกับลักษณะพฤติกรรมการบริโภคของลูกค้า เพื่อนำข้อมูลเหล่านี้มาสร้างแผนกลยุทธ์ทางการตลาด หรือ รายการส่งเสริมการขาย เพื่อทำให้ธุรกิจได้รับกำไรสูงสุด แต่ ณ ขณะนี้ข้อมูลที่ต้องนำมาวิเคราะห์มีแนวโน้มที่จะเพิ่มขึ้นมาก ซึ่งจะทำให้การวิเคราะห์ข้อมูลมีความซับซ้อนมากขึ้นและต้องใช้เวลามากขึ้น ฉะนั้นโครงการนี้จะพัฒนาโปรแกรมค้นหา Association Rule โดยใช้ DHP Algorithm เพื่อช่วยวิเคราะห์หาความสัมพันธ์ของข้อมูล

Title	The Development of an Association Rule Discovery Software using DHP Algorithm
Student	Mr. Chat Wattanasirikiat
Advisor	Dr. Worapoj Kreesuradej
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

Abstract

Nowadays, the contest in the business world is higher than the challenge in the past. Every companies have to analyze the proprietary data to make a customer behavior report used to prepare the strategy plan, tactic plan and sales promotion that the company will gain high profit. To generate the report from a tremendous number of data requires more complex algorithm and processing time. This project is to develop the Association Rule application used to discovery the interest information. The core engine of this application bases on the DHP Algorithm Methodology.

กิตติกรรมประกาศ

การจัดทำโครงการศึกษากรณีพิเศษในหัวข้อเรื่อง การพัฒนาโปรแกรมค้นหา Association Rule โดยใช้ DHP Algorithm สำเร็จลุล่วงได้เนื่องจากการสนับสนุน การให้คำแนะนำปรึกษาในแนวทางต่าง ๆ จึงส่งผลให้การจัดทำโครงการศึกษากรณีพิเศษนี้สำเร็จลุล่วงได้ตามเป้าหมายที่ได้วางไว้ ผู้จัดทำใคร่ขอขอบคุณบุคคลต่าง ๆ ดังนี้

บิดามารดา และ ญาติพี่น้อง ทุกคนที่คอยเป็นกำลังใจ และให้ความช่วยเหลือในด้านต่าง ๆ จนโครงการฯ นี้สำเร็จลุล่วงได้ด้วยดี

ดร.วรพจน์ กรีสุระเดช ผู้ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการศึกษากรณีพิเศษ ที่ให้คำปรึกษาและคำแนะนำต่างๆ อันเป็นประโยชน์ต่อการพัฒนาระบบ และได้สละเวลาในการแก้ไขข้อบกพร่อง

จึงใคร่ขอขอบคุณบุคคลดังกล่าวข้างต้นมา ณ โอกาสนี้

ฉัตร วัฒนศิริเกียรติ

สารบัญ

หน้า

บทที่

สารบัญภาพ

หน้า

ภาพที่

หน้า

ภาพที่

บทที่ 1
บทนำ

ปัจจุบันการดำเนินธุรกิจมีการแข่งขันกันสูง ดังนั้นบริษัทที่ประกอบธุรกิจจำนวนมากมักจะทำการวิเคราะห์ข้อมูลที่ผ่านมาในอดีตเพื่อหาข้อมูลที่เกี่ยวกับลักษณะพฤติกรรมการบริโภคของลูกค้า เพื่อนำข้อมูลเหล่านี้มาสร้างแผนกลยุทธ์ทางการตลาด หรือ รายการส่งเสริมการขาย(การวางสินค้าที่สามารถขายร่วมกันได้ไว้ใกล้ๆกัน, การออกแบบคูปองลดราคาสินค้า และการออกแบบลักษณะการการจัดวางสินค้าบนชั้นสินค้า เป็นต้น) เพื่อทำให้ธุรกิจได้รับกำไรสูงสุด เนื่องจากความก้าวหน้าของเทคโนโลยี Bar-Code ทำให้การจัดเก็บข้อมูลการขายสินค้าได้รับความสะดวกรวดเร็วและข้อมูลที่ได้มีความน่าเชื่อถือสูง ปัจจุบันข้อมูลที่ต้องนำมาวิเคราะห์มีแนวโน้มที่จะเพิ่มขึ้นมากซึ่งจะทำให้การวิเคราะห์ข้อมูลมีความซับซ้อนมากขึ้นและต้องใช้เวลามากขึ้น ฉะนั้นจึงได้มีการนำแนวความคิดของ Association Rule ของ Data Mining มาช่วยในการวิเคราะห์ข้อมูลในลักษณะดังกล่าว

1.1 วัตถุประสงค์

นำเทคนิคของ Data Mining มาใช้ในการวิเคราะห์หาความสัมพันธ์ของข้อมูล เพื่อให้องค์กรสามารถนำสารสนเทศที่ได้จากการทำ Data Mining มาใช้ในการวางแผนกลยุทธ์ต่างๆได้อย่างมีประสิทธิภาพ รวมทั้งยังทำให้เห็นภาพของการนำเทคนิคของ Data Mining มาประยุกต์ใช้กับข้อมูลทางธุรกิจ

1.2 ขอบเขตการดำเนินงาน

โครงการนี้เป็นการศึกษาถึงการนำเทคนิค Link Analysis ของ Data Mining มาใช้ในการวิเคราะห์หาความสัมพันธ์ของข้อมูล โดยจะนำ Direct Hashing and Pruning(DHP) Algorithm และ Perfect Hashing and Pruning(PHP) Algorithm มาใช้ในการหา Association Rule

1.3 ขั้นตอนและวิธีการดำเนินงาน

เพื่อให้การศึกษาบรรลุวัตถุประสงค์ตามที่ได้กำหนดไว้ภายใต้ขอบเขตของการศึกษาจึงได้กำหนดขั้นตอนในการศึกษาไว้ดังนี้

1. ศึกษาทฤษฎีที่เกี่ยวข้อง Data Mining เพื่อนำมาประยุกต์ใช้

2. ศึกษาหลักการทำงานของ Direct Hashing and Pruning(DHP) Algorithm และ Perferct Hashing and Pruning(PHP) Algorithm เพื่อนำหลักการดังกล่าวมาประยุกต์ใช้

3. ออกแบบและพัฒนาระบบงานเพื่อวิเคราะห์ข้อมูล

4. สรุปผลการศึกษา

1.4 ประโยชน์ที่คาดว่าจะได้รับ

จากการศึกษาและพัฒนาระบบงานคาดว่าจะให้ประโยชน์แก่ ผู้ค้นคว้า และ ผู้นำไปปฏิบัติ ดังนี้

1. เข้าใจหลักการและขั้นตอนของการทำ Data Mining
2. เป็นแนวทางในการนำ Data Mining มาประยุกต์ใช้กับธุรกิจ

บทที่ 2

Data Mining และ ทฤษฎีที่เกี่ยวข้อง

Data Mining เป็นขั้นตอนหรือกระบวนการในการค้นหาสารสนเทศที่มีประโยชน์จากฐานข้อมูลที่มีอยู่ โดยสารสนเทศที่ได้มาจากการทำ Data Mining จะมีลักษณะดังต่อไปนี้ คือ

1. ข้อมูลที่ไม่ทราบล่วงหน้ามาก่อน(Unknown) เป็นข้อมูลที่ผู้ทำ Data Mining ยังไม่เคยทราบล่วงหน้ามาก่อน เช่น เจ้าของร้านสะดวกซื้อแห่งหนึ่ง เพิ่งค้นพบพฤติกรรมของผู้บริโภคว่า ผู้บริโภคที่เป็นพ่อบ้านมักจะซื้อเบียร์และผ้าอ้อมพร้อมกันในวันศุกร์ตอนเย็น

2. ข้อมูลที่มีความถูกต้อง(Valid) หลังจากที่ทำ Data Mining แล้วเราจะพบว่า Data Mining จะให้ข้อมูลออกมาจำนวนมาก ฉะนั้นเราจำเป็นต้องนำข้อมูลที่ได้มาพิจารณาว่า ข้อมูลนั้นถูกต้องหรือไม่

3. ข้อมูลที่สามารถนำมาใช้ประโยชน์ได้จริง(Actionable) คือ ข้อมูลที่ได้จะต้องสามารถนำมาใช้ในทางปฏิบัติเพื่อสร้างความได้เปรียบในเชิงธุรกิจได้ บางครั้งข้อมูลที่เราค้นพบเป็นสิ่งที่คู่แข่งได้ทำไปแล้ว หรือเป็นสิ่งผิดกฎหมาย ข้อมูลดังกล่าวก็จะไม่มีประโยชน์อะไร

2.1 ขบวนการในการทำ Data Mining

ถ้ากล่าวถึง Data Mining คนส่วนใหญ่จะให้ความสำคัญกับการ Mining ข้อมูลแต่ที่จริงแล้วการ Mining ข้อมูลเป็นเพียงขั้นตอนหนึ่งในขบวนการ Data Mining เท่านั้น ฉะนั้นเพื่อให้เกิดความเข้าใจที่ชัดเจนเกี่ยวกับขบวนการในการทำ Data Mining ในหัวข้อนี้เราจะกล่าวถึงขบวนการในการทำ Data Mining โดยเราสามารถแบ่งขบวนการในการทำ Data Mining ออกเป็น 5 ขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 : กำหนดวัตถุประสงค์ทางธุรกิจ (Business Objective Determination)

การกำหนดวัตถุประสงค์ทางธุรกิจจะต้องเข้าใจถึงปัญหาและความต้องการทางธุรกิจเพราะการกำหนดปัญหาให้ชัดเจนจะเป็นตัวกำหนดทิศทางในการทำ Data Mining นอกจากนั้นแล้วการกำหนดปัญหาจะเป็นส่วนที่กำหนดว่าเมื่อไรจะใช้ Data Mining ในการแก้ปัญหา เพราะไม่ได้หมายความว่าทุกปัญหาสามารถแก้ไขด้วยเทคนิค Data Mining ดังนั้นการกำหนดปัญหาที่ไม่ถูกต้องย่อมนำไปสู่ความสำเร็จในการแก้ปัญหาได้ยาก ซึ่งในส่วนนี้จะประกอบด้วยการวิเคราะห์ทางธุรกิจ และการวิเคราะห์ข้อมูลเบื้องต้นว่า เรามีข้อมูลใดอยู่บ้าง และต้องการอะไรจากข้อมูล ซึ่งในขั้นตอนนี้จะสามารถมองถึง Algorithm และฐานข้อมูลที่จะใช้ในเบื้องต้นที่สัมพันธ์กับวัตถุประสงค์ทางธุรกิจได้

ขั้นตอนที่ 2 : การเตรียมข้อมูล (Data Preparation)

การเตรียมข้อมูล เป็นขั้นตอนที่สำคัญ และ ใช้เวลามากที่สุดของขบวนการในการทำ Data Mining ซึ่งในขั้นตอนนี้จะแบ่งการทำงานออกเป็น 3 ขั้นตอนย่อย คือ

1. การเลือกข้อมูล(Data Selection)

ในบางครั้งการทำ Data Mining อาจจะต้องนำข้อมูลจากหลายๆแหล่งมารวมกัน จึงทำให้ข้อมูลมีปริมาณมาก ฉะนั้นวัตถุประสงค์ของการเลือกข้อมูล ก็คือ การระบุลักษณะข้อมูล , เลือกข้อมูลที่ต้องการ และนำข้อมูลที่ไม่ต้องการออกไป ซึ่งถือว่าขั้นตอนนี้เป็นขั้นตอนเริ่มต้นในการเตรียมข้อมูลสำหรับการ Mining ในอนาคต การเลือกข้อมูลนั้นจะแตกต่างไปตามวัตถุประสงค์ของแต่ละธุรกิจที่ได้กำหนดไว้ โดยปกติเราสามารถแบ่งลักษณะและรูปแบบของข้อมูลออกเป็น 2 ลักษณะคือ

- ตัวแปรแบบ Categorical

- Nominal เป็นตัวแปรที่ลำดับของข้อมูลไม่มีผลกับค่า เช่น สถานะการแต่งงาน (โสด , แต่งงาน , หย่าร้าง)

- Ordinal เป็นตัวแปรที่ลำดับของข้อมูลมีผลกับค่า เช่น เกรดของนักศึกษา(A , B , C , D , F)

- ตัวแปรแบบ Quantitative

- Continuous ค่าที่เก็บเป็นเลขจำนวนจริง(Real number) หรือ เป็นค่าที่ต่อเนื่อง เช่น น้ำหนักของสินค้า

- Discrete ค่าที่เก็บเป็นเลขจำนวนเต็ม(Integer) เช่น จำนวนนักศึกษาในมหาวิทยาลัย

2. การกลั่นกรองข้อมูล(Data Preprocessing)

วัตถุประสงค์ของการกลั่นกรองข้อมูล ก็เพื่อทำให้ข้อมูลที่ถูกเลือกมานั้นมีคุณภาพเหมาะที่จะนำไปทำ Data Mining เนื่องจากข้อมูลที่ถูกเลือกมาจากขั้นตอนการเลือกข้อมูลอาจมีข้อมูลไม่ถูกต้อง ดังนั้นในขั้นตอนนี้มีประเด็นที่จะต้องพิจารณาเพิ่มเติมอีก 2 ประเด็น คือ

- Noisy Data เป็นข้อมูลที่มีลักษณะต่างจากข้อมูลที่คาดการไว้ ซึ่งอาจมีความหมายได้ทั้งแง่ดีและร้าย ในแง่ดี คือมันจะแสดงชัดเจนถึงสิ่งที่เรากำลังมองหาอยู่ หรือในแง่ร้าย คือมันอาจเป็นข้อมูลที่ไม่สมบูรณ์ สาเหตุอาจจะเกิดจากความผิดพลาดในการบันทึกข้อมูลของคอมพิวเตอร์ หรือความผิดพลาดในการบันทึกข้อมูลของมนุษย์ หรือ ความผิดพลาดที่เกิดขึ้นระหว่างการรับส่งข้อมูลผ่านทาง Network เป็นต้น ดังนั้นจึงควรมีขั้นตอนที่ใช้ในการจัดการกับข้อมูลเหล่านี้ก่อนนำไปใช้งาน เช่น การแก้ไขข้อมูลเหล่านี้โดยอาจจะใช้เทคนิค Regression ทางสถิติ หรือ การตัดข้อมูลเหล่านี้ทิ้งไม่นำมาวิเคราะห์(ในกรณีที่ข้อมูลเหล่านี้มีจำนวนน้อย) เป็นต้น

● Missing Value เป็นข้อมูลที่มีลักษณะบางส่วนของข้อมูลขาดหายไป สาเหตุของการเกิด Missing Value ก็จะคล้ายกับสาเหตุของการเกิด Noisy Data ดังนั้นจึงควรมีขั้นตอนที่ใช้ในการจัดการกับข้อมูลเหล่านี้ก่อนนำไปใช้งาน เช่น การแก้ไขข้อมูลเหล่านี้ด้วยค่าเฉลี่ย หรือ การแก้ไขข้อมูลเหล่านี้ด้วยค่าที่เป็นไปได้มากที่สุด หรือ การแก้ไขข้อมูลเหล่านี้ด้วยค่าคงที่(เช่น Unknown เป็นต้น) หรือ การตัดข้อมูลเหล่านี้ทิ้งไม่นำมาวิเคราะห์(ในกรณีที่ข้อมูลเหล่านี้มีจำนวนน้อย) เป็นต้น

3. การแปลงข้อมูล(Data Transformation)

เป็นการแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสมกับการทำ Data Mining เช่น การแปลงตัวแปร Quantitative ให้เป็น Categorical โดยแบ่งค่าของตัวแปรให้เป็นช่วงๆ เช่น การแปลงข้อมูลของเงินเดือน นอกจากนี้แล้วยังมีเทคนิคของการแปลงตัวแปร Categorical ให้เป็น Numeric เช่น ยี่ห้อรถ Honda , Toyota ให้เป็น 001 , 002 เป็นต้น

ขั้นตอนที่ 3 : การทำ Data Mining(Data Mining)

เป็นการประมวลผลข้อมูลตาม Algorithm ที่ได้กำหนดไว้ ในขั้นตอนนี้จะมีความสัมพันธ์กับขั้นตอนที่ผ่านมา โดยเมื่อทำในขั้นตอนนี้แล้วอาจทำให้ต้องย้อนกลับไปทำในขั้นตอนการกลั่นกรองข้อมูลใหม่ก็ได้ ถ้าผลลัพธ์ที่ได้จากขั้นตอนนี้ไม่ได้เป็นไปอย่างที่คาดหวังไว้ ในขั้นตอนนี้จะเกี่ยวข้องกับการเลือกใช้ Algorithm ซึ่งในปัจจุบันมีการพัฒนา Algorithm ขึ้นมาหลายๆแบบและแต่ละแบบจะมีข้อดี และข้อเสียที่แตกต่างกันไป ดังนั้นการเลือกใช้ Algorithm ในการทำ Data Mining จะต้องมีการศึกษาให้รอบคอบก่อน

ขั้นตอนที่ 4 : การวิเคราะห์ผลลัพธ์ที่ได้จากการทำ Data Mining(Analysis of Results)

ขั้นตอนนี้จะเป็นการแปลความและตีความผลลัพธ์ที่ได้จากการทำ Data Mining การทำงานในส่วนนี้จำเป็นต้องใช้ทักษะในการวิเคราะห์ข้อมูลทางธุรกิจเข้ามาช่วย

ขั้นตอนที่ 5 : การปรับความรู้ที่ได้เข้ากับธุรกิจ (Assimilation of Knowledge)

เป็นการรวบรวมความเข้าใจทางธุรกิจที่ได้จากการแปลความ และ ตีความผลลัพธ์ที่ได้จากการทำ Data Mining เข้าไปเป็นส่วนความรู้เพื่อนำไปใช้ในโอกาสต่อไป โดยในขั้นตอนนี้จะมีหลักอยู่ 2 ประการ คือ

1. การนำเสนอแนวคิดทางธุรกิจที่ค้นพบใหม่
2. หาแนวทางที่จะนำความรู้ที่ค้นพบใหม่ไปใช้เพื่อก่อให้เกิดประโยชน์สูงสุด

2.2 โอเปอเรชั่นของ Data Mining (Data Mining Operation)

Data Mining จะประกอบด้วย 4 โอเปอเรชั่นที่ใช้สำหรับประยุกต์ใช้งานทางธุรกิจ ดังต่อไปนี้

1. Predictive Modeling เป็นโมเดลที่คล้ายกับการเรียนรู้ของมนุษย์ คือ จำเป็นต้องเป็นเข้า

ใจถึงลักษณะของสิ่งที่จะศึกษาอย่างแท้จริง เราจะใช้โมเดลนี้ในการวิเคราะห์ฐานข้อมูลที่มีอยู่เพื่อหาลักษณะที่สำคัญของข้อมูล และนำลักษณะที่สำคัญของข้อมูลออกมาสร้างเป็นโมเดลที่ใช้ในการทำนายต่อไปในอนาคต โดย Predictive Modeling ประกอบด้วย 2 ลักษณะ

● Classification เป็นการแบ่งข้อมูลออกเป็นกลุ่มๆ และนำโมเดลนี้มาทำนายว่าข้อมูลควรอยู่ในกลุ่มใด เช่น การนำข้อมูลของลูกค้ามาทำนายว่าลูกค้าคนใดควรอยู่ในกลุ่มที่จะส่งจดหมายแนะนำสินค้าและบริการใหม่ไปให้หรือไม่ โดยดูจากประวัติและพฤติกรรมการบริโภค เป็นต้น

● Value Prediction จะนำโมเดลมาทำนายค่าที่สัมพันธ์กับข้อมูลที่มีอยู่ เช่น ต้องการทำนายช่วงเวลาที่จะได้ลูกค้าใหม่ในการขายรถ โดยจะต้องทำการ Mining ข้อมูลลูกค้าเก่าและข้อมูลอื่นๆ เช่น ความสามารถทางด้านการเงิน , อายุของลูกค้า , รายได้ , จำนวนคนในครอบครัว และระดับการศึกษา เป็นต้น

2. Database Segmentation เป็นการแยกข้อมูลที่ลักษณะเหมือนกันในฐานข้อมูลออกเป็นส่วนๆหรือเป็นกลุ่มๆ โดยแบ่งตามคุณสมบัติที่เหมือนกัน ซึ่งจะได้ผลลัพธ์ คือ กลุ่มของข้อมูล เช่น การแบ่งกลุ่มนักศึกษาออกตามอายุ , เพศ เป็นต้น

3. Link Analysis เป็นการวิเคราะห์หาความสัมพันธ์ระหว่างข้อมูลว่าข้อมูลแต่ละรายการมีความสัมพันธ์กันหรือไม่ อย่างไร โดย Link Analysis จะประกอบด้วย 3 ลักษณะ

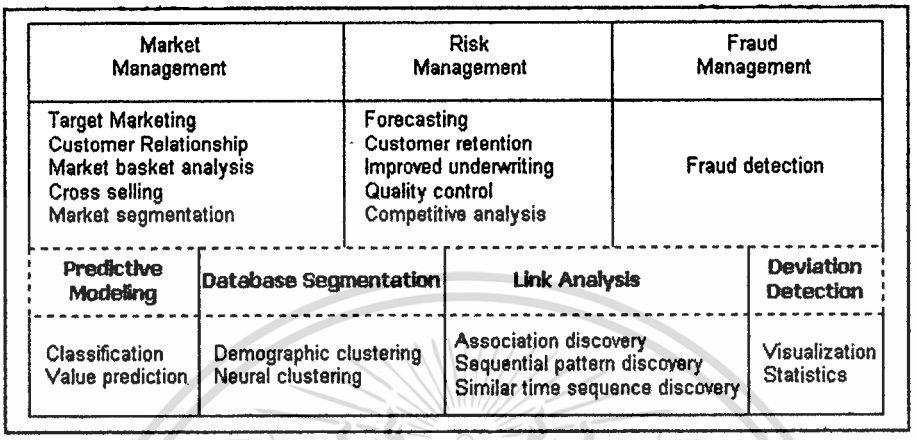
● Association Discovery เป็นการค้นหาความสัมพันธ์ของข้อมูล เช่น การ Mining ข้อมูลการซื้อสินค้าของลูกค้าเพื่อศึกษาพฤติกรรมการซื้อสินค้าของลูกค้า เพื่อจะได้ทราบว่ามีรายการสินค้าประเภทใดบ้างที่ลูกค้ามักซื้อไปพร้อมๆกัน

● Sequential Pattern Discovery คือการชี้เฉพาะถึงลักษณะหรือรูปแบบความสัมพันธ์ที่สนใจทั้งหมดและดึงสารสนเทศที่เกี่ยวข้องกับลำดับเหตุการณ์ต่างๆ เพื่อทำความเข้าใจถึงพฤติกรรมในระยะยาว ซึ่งจะใช้ระบุความสัมพันธ์ของการซื้อสินค้าอย่างหนึ่ง และ จะซื้อสินค้าอีกอย่างหนึ่งในเวลาต่อมา เช่น ผู้ขายอาจพบว่า ลูกค้าที่ซื้อโทรทัศน์มีแนวโน้มที่จะซื้อวีดีโอในเวลาต่อมา

● Similar Time Sequence Discovery ใช้ค้นหาความสัมพันธ์ระหว่างกลุ่มข้อมูล 2 กลุ่มซึ่งมีการขึ้นต่อกันทางด้านเวลา แทนข้อมูลในแนวแกน X ด้วยค่าของเวลา เช่น วันหรือเดือน แทนข้อมูลในแกน Y ด้วยค่าของสิ่งที่เราสนใจ เช่น การขายสินค้า ราคาสินค้า แล้วนำรูปแบบความสัมพันธ์นั้นที่เวลาเดียวกันมาเปรียบเทียบเพื่อหารูปแบบหลักๆ ไว้ใช้ในการทำนายในอนาคต

4. Deviation Detection เป็นโมเดลที่พยายามหาสิ่งแปลกปลอมออกจากกลุ่มของมัน ส่วนใหญ่แล้วการวิเคราะห์ในลักษณะนี้จะใช้เทคนิคทางสถิติ และ visualization เข้ามาช่วย ซึ่งจะทำให้เห็นถึงข้อผิดพลาด หรือ ส่วนที่ไม่เกี่ยวข้องได้ชัดเจน ในทางธุรกิจมักจะใช้โมเดลนี้ในการหาข้อผิด

พลาด หรือ ป้องกันการทุจริตในกรณีต่างๆ เช่น การใช้บัตรเครดิตปลอม , การตรวจสอบคุณภาพต่างๆ เป็นต้น



Market Management	Risk Management	Fraud Management
Target Marketing Customer Relationship Market basket analysis Cross selling Market segmentation	Forecasting Customer retention Improved underwriting Quality control Competitive analysis	Fraud detection

Predictive Modeling	Database Segmentation	Link Analysis	Deviation Detection
Classification Value prediction	Demographic clustering Neural clustering	Association discovery Sequential pattern discovery Similar time sequence discovery	Visualization Statistics

ภาพที่ 2.1 แสดงการนำโมเดลต่างๆของ Data Mining กับการประยุกต์ใช้งาน[1]

ในภาพ 2.1 จะแสดงการนำโมเดลต่างๆ ของ Data Mining ไปประยุกต์ใช้งาน แต่เราจะไม่สามารถเจาะจงได้ว่าธุรกิจประเภทใดต้องใช้โมเดลแบบไหน เพียงแต่เป็นการแนะนำว่าลักษณะงานทางธุรกิจประเภทใดควรเลือกใช้โมเดลแบบใด

จากที่กล่าวมาข้างต้นจะพบว่า โอเปอเรชั่นของ Data Mining มีมากมาย สำหรับโครงการนี้จะนำเสนอการนำ Association Discovery ของ Link Analysis มาใช้วิเคราะห์หาความสัมพันธ์ของข้อมูล โดยจะนำ Direct Hashing and Pruning(DHP) Algorithm และ Perfect Hashing and Pruning (PHP) Algorithm มาใช้ในการหา Association Rule โดยจะกล่าวถึงรายละเอียดในบทถัดไป

บทที่ 3

Association Discovery

Association Discovery เป็นการค้นหาความสัมพันธ์ของข้อมูลที่เกิดขึ้นในฐานข้อมูลที่มีอยู่ การวิเคราะห์แบบนี้บางครั้งเรียกว่า "Market Basket Analysis" คือ อะไรมีการกระทำด้วยกัน แนวคิดดังกล่าวนำไปใช้ในร้านซุปเปอร์มาร์เก็ต เพื่อกำหนดว่าสินค้าประเภทใดมักจะถูกซื้อควบคู่กันในการซื้อแต่ละครั้ง ทำให้ร้านสามารถกำหนดแผนกลยุทธ์ในการส่งเสริมการตลาดได้อย่างมีประสิทธิภาพ เช่น ควรจัดเรียงสินค้าอย่างไร , ควรจัดทำแคตตาล็อคเพื่อขายสินค้าอย่างไร และ การวางแผนเพื่อจัดโปรโมชั่นสนับสนุนการขายอย่างไร

โดยทั่วๆไปข้อมูลการขายสินค้าที่จัดเก็บมักจะประกอบไปด้วย วันที่เอกสาร , รายการสินค้าและปริมาณของสินค้าที่ขาย และ บางเอกสารก็จะสามารถจัดเก็บรายละเอียดของลูกค้าได้ถ้าลูกค้าชำระเงินด้วยบัตรเครดิต หรือ บัตรอื่นที่มีความเกี่ยวข้องกับลูกค้า(บัตรเดบิต หรือ บัตรสมาชิก เป็นต้น) แต่ในการทำ Mining Association Rule จะไม่นำข้อมูลที่เป็น วันที่เอกสาร , ปริมาณของสินค้าที่ขาย และ รายละเอียดของของลูกค้า มาใช้ในการคำนวณ โดยภาพที่ 3.1 จะแสดงตัวอย่างของฐานข้อมูลการขายสินค้าที่นำมาใช้สำหรับการทำ Mining Association Rule

Database *D*

TID	Items
100	A C D
200	B C E
300	A B C E
400	B E

ภาพที่ 3.1 แสดงตัวอย่างฐานข้อมูลการขายสินค้าที่นำมาใช้สำหรับการทำ Mining Association Rule[2]

กำหนดให้ $I = \{ i_1 , i_2 , ... , i_n \}$ เป็นเซตของรายการสินค้า(Items) , $D = \{ T_1 , T_2 , ... , T_m \}$ เป็นเซตของเอกสาร(Transactions) โดยที่แต่ละเอกสาร T เป็นเซตของรายการสินค้า($T \subseteq I$)

และแต่ละเอกสาร T จะต้องมี Unique Identifier เพื่อใช้แยกแยะความแตกต่างของแต่ละเอกสารซึ่งก็คือ TID สามารถที่จะกล่าวได้ว่าเอกสาร T มีรายการสินค้า X ก็ต่อเมื่อ $X \subseteq T$ โดยปกติแล้วกฎของ Association Rule สามารถจัดให้อยู่ในรูปของ $X \Rightarrow Y$ โดยที่ $X \subset I$, $Y \subset I$ และ $X \cap Y = \varnothing$

กฎของ Association Rule มีตัววัดหลักๆ 2 ตัว คือ Support Factor และ Confidence Factor เราสามารถกล่าวได้ว่ากฎ $X \Rightarrow Y$ ที่สร้างจากเซตของเอกสาร D มี Confidence Factor เท่ากับ c ก็ต่อเมื่อ (จำนวนเอกสาร T_i ที่มีทั้งรายการสินค้า X และ Y อยู่ด้วยกัน) / (จำนวนเอกสาร T_i ที่มีรายการสินค้า X) เท่ากับ c และเราสามารถกล่าวได้ว่ากฎ $X \Rightarrow Y$ ที่สร้างจากเซตของเอกสาร D มี Support Factor เท่ากับ s ก็ต่อเมื่อ (จำนวนเอกสาร T_i ที่มีทั้งรายการสินค้า X และ Y อยู่ด้วยกัน) / (จำนวนเอกสาร T_i ทั้งหมดที่อยู่ในเซตของเอกสาร D) เท่ากับ s

จากนิยามข้างต้นนี้ D สามารถเป็นได้ทั้ง ไฟล์ข้อมูล , ระบบฐานข้อมูลเชิงสัมพันธ์ หรือ ผลลัพธ์ที่ได้จากการสอบถามข้อมูลในระบบฐานข้อมูลเชิงสัมพันธ์

ปัญหาของการทำ Mining Association Rule ก็คือ การสร้าง Association Rule ทั้งหมดที่มีค่าของ Support Factor ที่มากกว่าหรือเท่ากับค่าต่ำสุดของ Support Factor ที่ผู้ใช้กำหนดขึ้น (minsup) และมีค่าของ Confidence Factor ที่มากกว่าหรือเท่ากับค่าต่ำสุดของ Confidence Factor ที่ผู้ใช้กำหนดขึ้น(minconf) ดังนั้นเราสามารถแบ่งขั้นตอนของการค้นหา Association Rule ออกเป็นขั้นตอนย่อยๆได้ 2 ขั้นตอนดังต่อไปนี้

1. ค้นหา Itemsets(เซตของรายการสินค้าที่ถูกซื้อไปพร้อมๆกัน) ทั้งหมดที่มีค่า Support Factor มากกว่าหรือเท่ากับ minsup ซึ่งเราสามารถเรียก Itemsets เหล่านี้ว่า Large Itemsets หรือ Frequent Itemsets โดยที่ขนาดของ Itemset จะถูกนำเสนอด้วยจำนวนรายการสินค้าที่อยู่ใน Itemset เช่น ถ้าขนาดของ Itemset เท่ากับ k ดังนั้นจะสามารถเรียก Itemset นี้ว่า k-Itemset

2. การสร้าง Association Rule จาก Large Itemsets สิ่งแรกที่ต้องทำก็คือ หาเซตย่อยที่ไม่ใช่เซตว่างทั้งหมดออกมาจาก Large Itemsets(l) ก่อน หลังจากได้เซตย่อย a ทั้งหมดแล้ว เราสามารถที่จะนำเซตย่อย a แต่ละชุดมาสร้างให้อยู่ในรูปของกฎใน Association Rule ได้เท่ากับ $a \Rightarrow (l - a)$ ก็ต่อเมื่ออัตราส่วนของ Support Factor(l – a) / Support Factor(a) มีค่ามากกว่าหรือเท่ากับ minconf

จะพบว่าขั้นตอนที่หนึ่งของการค้นหา Association Rule จะเป็นงานที่ใช้ทรัพยากรในการคำนวณที่สูงมาก ดังนั้นหัวข้อนี้จึงเป็นงานวิจัยที่ได้รับความสนใจกันมากของ Data Mining ฉะนั้นจึงทำให้เกิด Algorithm ต่างๆขึ้นมากมายเพื่อนำมาช่วยเพิ่มประสิทธิภาพในการค้นหา Association Rule เช่น AIS , SETM , Apriori , Direct Hashing And Pruning (DHP) Algorithm[2,3] เป็นต้น แต่ในโครงการนี้จะนำเสนอ Direct Hashing And Pruning(DHP) Algorithm และ Perfect Hashing

And Pruning(PHP) Algorithm ที่จะนำมาใช้เพื่อช่วยเพิ่มประสิทธิภาพในการหา Large Itemsets ของการหา Association Rule โดยรายละเอียดของ Algorithm ทั้งสองจะกล่าวถึงในหัวข้อต่อไป

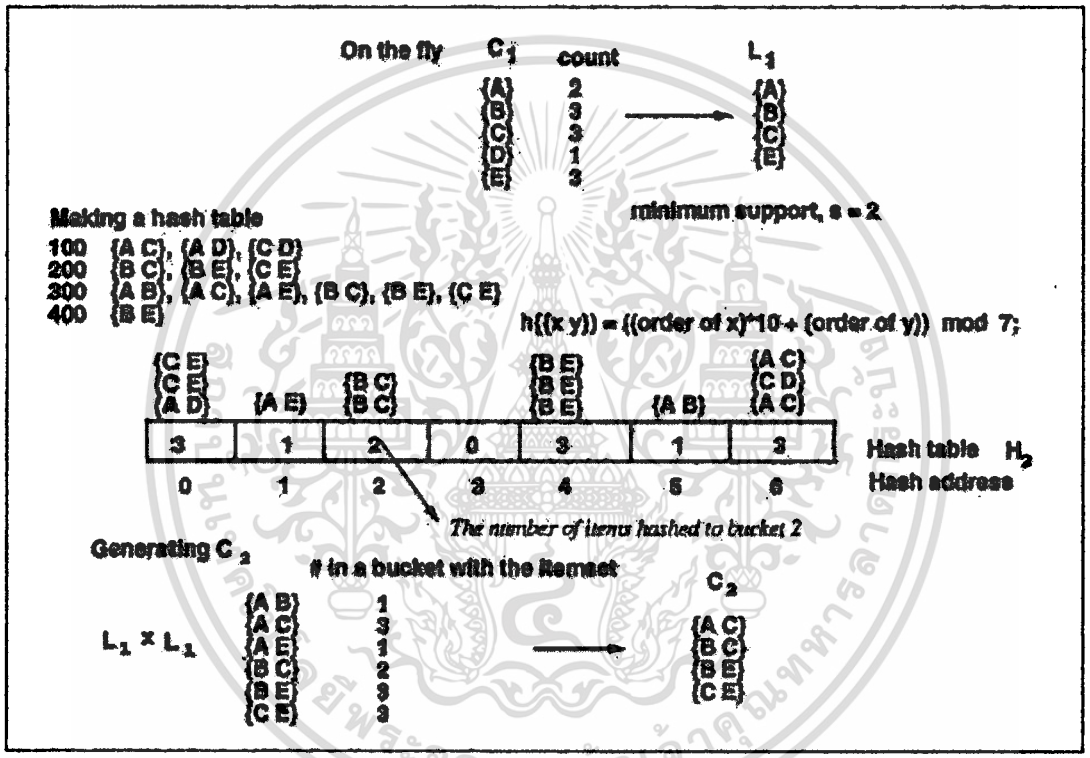
3.1 Direct Hashing And Pruning(DHP) Algorithm

```
Input: Database
Output: Frequent k-itemset
/* Database = set of transactions;
   Items = set of items;
   transaction = <TID, {x ∈ Items}>;
   F1 is a set of frequent 1-itemsets */

F1 = ϕ;
/* H2 is the hash table for 2-itemsets
Read the transactions, and count the
occurrences of each item, and
generate H2 */

for each transaction t ∈ Database do begin
    for each item x in t do
        x.count++;
    for each 2-itemset y in t do
        H2.add(y);
end
//Form the set of frequent 1-itemsets

for each item i ∈ Items do
    if i.count/|Database| ≥ minsup
    then F1 = F1 ∪ i;
end

/*Remove the hash values without the
minimum support */

H2.prune(minsup);
/*Find Fk, the set of frequent k-
itemsets, where k ≥ 2 */

for each (k := 2; Fk-1 ≠ ϕ; k++) do begin

    // Ck is the set of candidate k-itemsets
    Ck = ϕ;
/* Fk-1 * Fk-1 is a natural join of
Fk-1 and Fk-1 on the first k - 2 items
Hk is the hash table for k-itemsets */

    for each x ∈ {Fk-1 * Fk-1} do
        if Hk.hasupport(x)
        then Ck = Ck ∪ x;
    end
/*Scan the transactions to count candidate k-
itemsets and generate Hk+1 */

    for each transaction t ∈ Database do begin
        for each k-itemset x in t do
            if x ∈ Ck
            then x.count++;
        for each (k+1)-itemset y in t do
            if ∃z | z = k-subset of y
                ∧ ¬Hk.hasupport(z)
            then Hk+1.add(y);
    end
    // Fk is the set of frequent k-itemsets

    Fk = ϕ;
    for each x ∈ Ck do
        if x.count/|Database| ≥ minsup
        then Fk = Fk ∪ x;
    end

    /* Remove the hash values without the
    minimum support from Hk+1 */

    Hk+1.prune(minsup);
end
Answer = ∪k Fk;
```

ภาพที่ 3.2 แสดง Direct Hashing And Pruning Algorithm[2]

ทั้ง Apriori และ DHP Algorithm สามารถสร้าง Candidate k+1-Itemsets(C_{k+1}) ได้จากการนำ Large k-Itemsets(L_k) มา join กับ L_k ที่มีขนาดรายการสินค้า k – 1 รายการ($L_k * L_k$ หรือ $L_k \times L_k$) จากนั้นก็จะทำการสแกนฐานข้อมูลเพื่อนับค่า Support Factor ของแต่ละ Itemset ที่อยู่ใน C_{k+1} โดยจะนำ Itemset ที่มีค่า Support Factor ที่มากกว่าหรือเท่ากับ minsup มาสร้างเป็น L_{k+1}

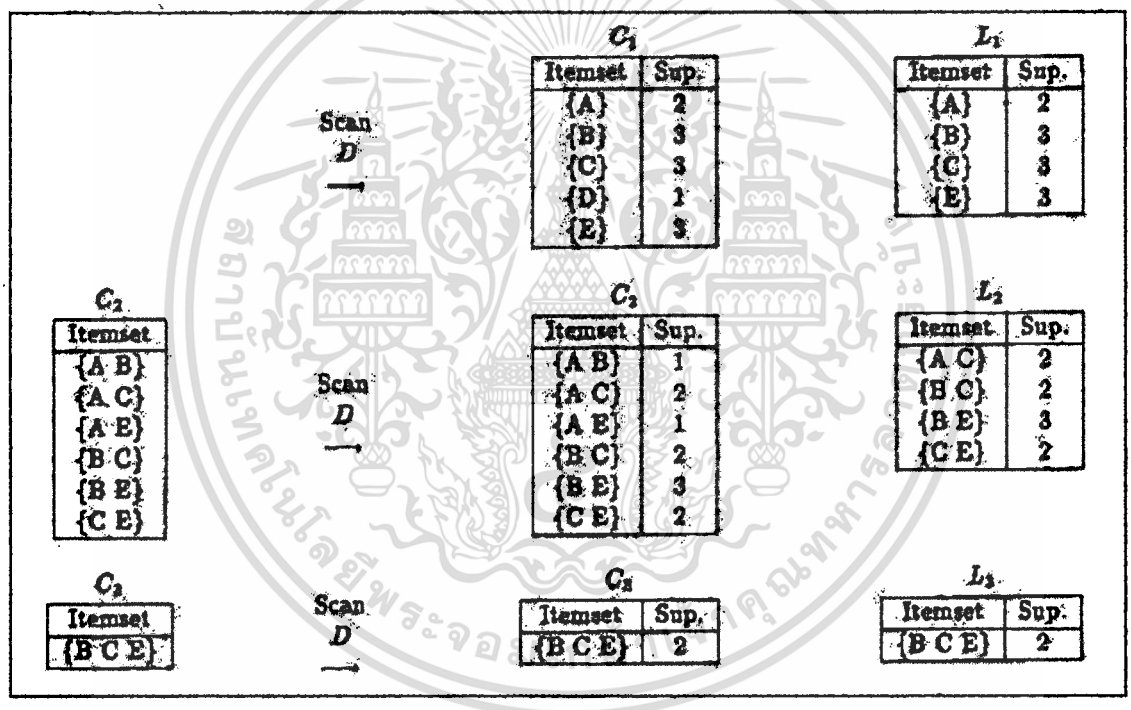
ต่อไปจะเป็นการอธิบายลักษณะการทำงานของ DHP Algorithm ที่อยู่ในภาพที่ 3.2 ในขั้นตอนแรก DHP Algorithm จะสแกนฐานข้อมูลรอบแรกเพื่อนับค่า Support Factor ของ 1-Itemsets และก็จะสร้างเซตย่อยของ 2-Itemsets ทั้งหมดที่เป็นไปได้ในแต่ละเอกสาร(ดูภาพที่ 3.3 ประกอบความเข้าใจ) จากนั้น DHP Algorithm ก็จะนำเซตย่อยของ 2-Itemsets มาผ่าน Hash Function ทีละชุดเพื่อหา Hash Address แล้วก็จะบวก 1 เพิ่มเข้าไปใน Bucket ของ Hash Table จากนั้นก็จะสร้าง L_1 จากค่า Support Factor ของ 1-Itemsets ที่มีค่ามากกว่าหรือเท่ากับ minsup



ภาพที่ 3.3 แสดงการใช้ Candidate Itemsets และ Large Itemsets โดย DHP Algorithm[2]

ในขั้นตอนถัดมา DHP Algorithm ก็จะสร้าง C_k จากการนำ L_{k-1} มา join กันแล้วใช้ Hash Table ที่สร้างไว้ในรอบก่อนหน้านี้ช่วยในการกำหนดว่า Itemsets ชุดใดควรจะนำไปสร้างเป็น C_k จากนั้นก็จะทำการสแกนฐานข้อมูลเพื่อหาว่ามี Itemsets ใดใน C_k ที่มีค่า Support Factor มากกว่าหรือเท่ากับ minsup เพื่อนำ Itemsets เหล่านั้นมาสร้างเป็น L_k และสร้าง Hash Table ที่จะนำไปใช้ในการสร้าง C_k ในรอบถัดไป โดยขั้นตอนนี้จะทำงานวนลูปไปจนกระทั่งไม่สามารถสร้าง L_k ได้อีกแล้ว

จะเห็นว่า DHP Algorithm จะใช้เทคนิคของ Hashing มาช่วยถอน Itemsets ที่ไม่มีความจำเป็นสำหรับการสร้าง C_k ในรอบถัดไปออก ซึ่งจะทำให้ขนาดของ C_k ที่ได้จากการใช้ DHP Algorithm มีขนาดเล็กเมื่อเปรียบเทียบกับขนาดของ C_k ที่ได้จากการใช้ Apriori โดยสามารถสังเกตได้จากภาพที่ 3.3 และ ภาพที่ 3.4 จะพบว่า C_2 ที่ได้จากการใช้ DHP Algorithm จะมี Itemsets ทั้งหมด 4 ชุด คือ {A C} , {B C} , { B E} และ {C E} แต่ C_2 ที่ได้จากการใช้ Apriori จะมี Itemsets ทั้งหมด 6 ชุด คือ {A B} , {A C} , {A E} , {B C} , { B E} และ {C E} ดังนั้น DHP Algorithm จะมีประสิทธิภาพในการสร้าง L_k ที่ดีกว่า Apriori เพราะ Overhead ของการสแกนฐานข้อมูลเพื่อนับค่า Support Factor ของแต่ละ Itemset ที่อยู่ใน C_k จะลดลง



ภาพที่ 3.4 แสดงการใช้ Candidate Itemsets และ Large Itemsets โดย Apriori Algorithm[2]

ประสิทธิภาพของ DHP Algorithm ในการลดจำนวนของ Candidate Itemsets จะขึ้นอยู่กับจำนวนของการเกิด False Positives โดยที่ False Positives จะเกิดขึ้นก็ต่อเมื่อ กลุ่มของ Candidate Itemsets ที่แตกต่างกันได้ค่า Hash Address ที่เหมือนกัน และแต่ละกลุ่มของ Candidate Itemsets มีค่า Support Factor น้อยกว่า minsup แต่เมื่อนำค่า Support Factor ของแต่ละกลุ่มมารวมกันก็จะได้ค่า Support Factor ที่มากกว่าหรือเท่ากับ minsup ซึ่งทำให้ DHP Algorithm ต้องเสียเวลามานับหาค่า

Support Factor ของ Candidate Itemsets ใหม่ เพราะใน Hash Table ไม่ได้เก็บค่าของ Support Factor ที่แท้จริงของแต่ละ Itemset ไว้

จะพบว่าขนาดของ Hash Table จะมีความสัมพันธ์ที่ผกผันกับโอกาสการเกิด False Positives เช่น ถ้ากำหนดขนาดของ Hash Table ไว้น้อยก็จะทำให้จำนวน Candidate Itemsets ที่แตกต่างกันแต่ได้รับค่า Hash Address ที่เหมือนกันมีจำนวนมากขึ้น(โอกาสการเกิด False Positives สูงขึ้น) ฉะนั้นอุปสรรคของการใช้ DHP Algorithm จะอยู่ที่การเลือกใช้ Hash Function และการกำหนดขนาดของ Hash Table ให้เหมาะสมกับชนิดข้อมูลที่จะนำมาทำ Mining Association Rule

3.2 Perfect Hashing And Pruning(PHP) Algorithm

จาก DHP Algorithm ถ้ากำหนดให้ Hash Table มีขนาดใหญ่พอสมควร เมื่อนำแต่ละ Itemsets ที่แตกต่างกันมาผ่าน Hash Function จะทำให้แต่ละ Itemset ได้ Hash Address ที่ไม่เหมือนกัน จึงทำให้ไม่เกิดปัญหา False Positives ขึ้นและทำให้ Hash Table สามารถเก็บค่า Support Factor จริงๆของแต่ละ Itemset ได้ ซึ่งเป็นผลทำให้ DHP Algorithm ไม่ต้องเสียเวลามานับหาค่า Support Factor ของ Candidate Itemsets ใหม่อีกในขณะที่ทำการสแกนฐานข้อมูล

ในบทความที่[2] ได้นำเสนอวิธีการเพิ่มประสิทธิภาพของการทำ Mining Association Rule ไว้ 2 วิธี คือ

1. การลดจำนวนของเอกสารที่ต้องถูกสแกนในแต่ละรอบลง

2. การตัดรายการสินค้า(Items) ที่ไม่ใช้ออกจากเอกสาร(รายการสินค้าที่มี Support Factor น้อยกว่า minsup)

ในหัวข้อนี้จะนำเสนอ Perfect Hashing And Pruning(PHP) Algorithm สำหรับการการทำ Mining Association Rule โดย PHP Algorithm จะใช้ Perfect Hashing สำหรับการสร้าง Hash Table ในแต่ละรอบ และจะลดขนาดของฐานข้อมูลที่ต้องถูกสแกนในแต่ละรอบลง ต่อไปจะทำการอธิบายลักษณะการทำงานของ PHP Algorithm ที่อยู่ในภาพที่ 3.5

โดยขั้นตอนแรก PHP Algorithm จะกำหนดให้ Hash Table มีขนาดเท่ากับจำนวน Items ที่แตกต่างกันในฐานข้อมูล เมื่อนำแต่ละ Item ที่แตกต่างกันมาผ่าน Hash Function แล้วจะได้ Hash Address ที่ไม่เหมือนกัน ซึ่งวิธีนี้จะเรียกว่า Perfect Hashing โดยที่ add Method ของ Hash Table จะสร้างสมาชิกใน Hash Table ขึ้นมาใหม่ ถ้า Item x ไม่อยู่ใน Hash Table และกำหนดให้ค่า Support Factor เท่ากับ 1 แต่ถ้า Item x มีอยู่ใน Hash Table แล้ว PHP Algorithm จะเพิ่มค่า Support Factor ขึ้นมาทีละ 1 หลังจากผ่านขั้นตอนแรกแล้ว Hash Table จะเก็บค่า Support Factor ของแต่ละ

Item ในฐานข้อมูลไว้ หลังจากนั้นเราจะนำข้อมูลใน Hash Table มาสร้าง Large 1-Itemsets และจะใช้ prune Method ตัดข้อมูลทั้งหมดที่อยู่ใน Hash Table ที่มีค่า Support Factor น้อยกว่า minsup ออก

```
Input: Database
Output: Frequent k-itemset
/* Database = set of transactions;
   Items = set of items;
   transaction = <TID, {x ∈ Items}>;
   F_1 is a set of frequent 1-itemsets */

F_1 = φ ;

/* H_1 is the hash table for 1-itemsets
   Read the transactions, and count the occurrences of
each item, and generate H_1 */

for each transaction t ∈ Database do begin
    for each item x in t do
        H_1.add(x);
end;

// Form the set of frequent 1-itemset

for each itemset y in H_1 do
    if H_1.hasupport(y)
    then F_1 = F_1 ∪ y
end

/* Remove the hash values without the minimum
   support */

H_1.prune(minsup);
D_1 = Database;

// D_k is the pruned database

/* Find F_k, the set of frequent k-itemsets, where k ≥ 2
and prune the database */

k = 2;
repeat
    D_k = φ;
    F_k = φ;
    for each transaction t ∈ D_{k-1} do begin
        // w is k-1 subset of items in t
        if ∀w | w ∉ F_{k-1}
        then skip t;
        else
            items = φ ;
            for each k -itemset y in t do
                if ¬∃x | x = k − 1 subset of y
                    ∧ ¬H_{k-1}.hasupport(x)
                then H_k.add(y);
                items=items ∪ y;
            end
            D_k = D_k ∪ t   // such that t contains
                            // items only in the set items
    end

    for each itemset y in H_k do
        if H_k.hasupport(y)
        then F_k = F_k ∪ y
    end

    /* Remove the hash values without the minimum
       support from H_k */
    H_k.prune(minsup);
    k++;
until F_{k-1} = φ;
Answer = ∪_k F_k;
```

ภาพที่ 3.5 แสดง Perfect Hashing And Pruning Algorithm [2]

ในขั้นตอนถัดมา PHP Algorithm จะตัดเอกสารที่ไม่มีรายการสินค้า(Items) ที่อยู่ใน Large Itemsets ออกจากฐานข้อมูล และจะตัดรายการสินค้าที่ไม่อยู่ใน Large Itemsets ออกจากเอกสาร ในขณะเดียวกัน PHP Algorithm จะสร้าง Candidate k-Itemsets และนับค่า Support Factor ของ k-Itemsets ไว้ใน Hash table ที่ตอนจบการทำงานของแต่ละรอบ PHP Algorithm จะสร้าง D_k, H_k และ F_k ขึ้นมา โดย D_k จะเก็บฐานข้อมูลใหม่ที่ตัดเอกสารและรายการสินค้าที่ไม่จำเป็นต้องใช้ในรอบถัด

ไปออก ในขณะที่ H_k จะเก็บค่า Support Factor ของ Candidate k-Itemset ไว้ และ F_k จะเก็บเซตของ Large k-Itemsets โดยในขั้นตอนนี้จะทำงานวนลูปไปจนกระทั่งไม่สามารถสร้าง Large k-Itemsets (F_k) ได้อีกแล้ว

จะพบว่า PHP Algorithm มีประสิทธิภาพที่ดีกว่า DHP Algorithm เพราะเมื่อนำค่า Support Factor ของ Candidate k-Itemsets ไปเก็บไว้ใน Hash Table แล้ว PHP Algorithm ไม่ต้องเสียเวลามานับค่า Support Factor ของ Candidate Itemsets ใหม่อีกเหมือนในกรณีของ DHP Algorithm และ PHP Algorithm จะมีประสิทธิภาพที่ดีกว่า Apriori Algorithm เมื่อฐานข้อมูลมีขนาดใหญ่ และจำนวน Large Itemsets มีขนาดเล็กเพราะว่าในแต่ละรอบการทำงาน PHP Algorithm จะมีการลดขนาดของฐานข้อมูลที่ต้องสแกนลง

3.3 Direct Hashing And Pruning(DHP) และ Perfect Hashing And Pruning(PHP) Algorithm

จากทฤษฎีข้างต้น จะพบว่าถ้าจะนำ PHP Algorithm มาประยุกต์ใช้งาน จะต้องหา Hash Function ที่มีคุณสมบัติของ Perfect Hashing(เมื่อนำแต่ละ Item ที่แตกต่างกันมาผ่าน Hash Function แล้วจะได้ Hash Address ที่ไม่เหมือนกัน)ให้ได้ก่อน แต่ในทางปฏิบัติไม่มี Hash Function ใดที่สามารถการันตีได้ว่ามีคุณสมบัติดังกล่าว ดังนั้นในโครงการนี้จะนำแนวคิดในเรื่อง การลดจำนวนของเอกสารที่ต้องถูกสแกนในแต่ละรอบลง , การตัดรายการสินค้า(Items) ที่ไม่ใช้ออกจากเอกสาร (รายการสินค้าที่มี Support Factor น้อยกว่า minsup)ของ PHP Alogrithm มาผสมผสานกับ DHP Algorithm เพื่อนำไปประยุกต์ใช้งานต่อไป

3.4 การนำ Large Itemset มาสร้างเป็นกฎ

```
// Simple Algorithm
forall large itemsets l_k, k ≥ 2 do
    call genrules(l_k, l_k);

// The genrules generates all valid rules ã ⟹ (l_k − ã), for all ã ⊂ a_m
procedure genrules(l_k: large k-itemset, a_m: large m-itemset)
1)  A = {(m−1)-itemsets a_{m−1} | a_{m−1} ⊂ a_m};
2)  forall a_{m−1} ∈ A do begin
3)     conf = support(l_k)/support(a_{m−1});
4)     if (conf ≥ minconf) then begin
7)        output the rule a_{m−1} ⟹ (l_k − a_{m−1}), with confidence = conf and support = support(l_k);
8)        if (m − 1 > 1) then
9)           call genrules(l_k, a_{m−1}); // to generate rules with subsets of a_{m−1} as the antecedents
10)    end
11) end
```

ภาพที่ 3.6 แสดง Algorithm ที่นำ Large Itemset มาสร้างเป็นกฎ [4]

นำ Large Itemset(F_k) ที่ได้จากหัวข้อที่แล้วมาสร้างเป็นกฎ โดยนำค่า Large Itemset ตั้งแต่ F_2 เป็นต้นไปมาคำนวณหา Subset ของ Large Itemset ทั้งหมดที่เป็นไปได้ และนำแต่ละ Subset ของ Large Itemset มาสร้างเป็นกฎ โดยภาพที่ 3.6 จะแสดง Algorithm ที่นำ Large Itemset มาสร้างเป็นกฎ และ ภาพที่ 3.7 จะแสดงตัวอย่างกฎที่ได้จากการทำ Mining Association Rule

L_2

Itemset	Sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

Possible Rules →

Rules	Confidence	Support
A -> C	1	2
C -> A	0.67	2
B -> C	0.67	2
C -> B	0.67	2
B -> E	1	3
E -> B	1	3
C -> E	0.67	2
E -> C	0.67	2

L_3

Itemset	Sup
{B, C, E}	2

Possible Rules →

Rules	Confidence	Support
B -> C & E	0.67	2
C -> B & E	0.67	2
E -> B & C	0.67	2
B & C -> E	1	2
B & E -> C	0.67	2
C & E -> B	1	2

ภาพที่ 3.7 แสดงตัวอย่างกฎที่ได้จากการทำ Mining Association Rule

บทที่ 4
การประยุกต์ใช้ Data Mining เพื่อหาความสัมพันธ์ของข้อมูล

เพื่อให้การศึกษาถึงการนำ Direct Hashing and Pruning(DHP) และ Perfect Hashing and Pruning(PHP) Algorithm มาใช้ค้นหา Association Rule ของ Data Mining บรรลุตามวัตถุประสงค์ที่กำหนด จึงได้พัฒนาโปรแกรมที่ใช้ค้นหา Association Rule ซึ่งจะใช้ DHP และ PHP Algorithm ดังนั้นในบทนี้จะกล่าวถึงวิธีการใช้งานโปรแกรมที่พัฒนาขึ้น

4.1 การติดต่อกับข้อมูลที่นำมาวิเคราะห์

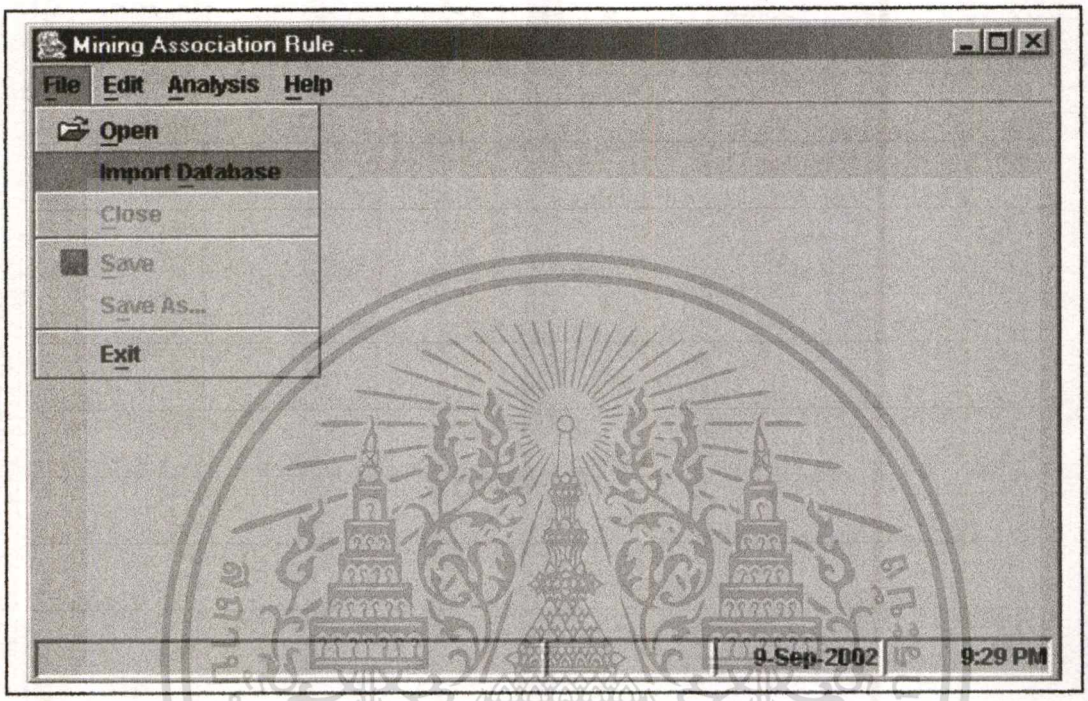
เมื่อเข้าสู่โปรแกรมจะปรากฏเมนูหลัก ดังภาพที่ 4.1

ภาพที่ 4.1 หน้าจอหลักของโปรแกรม

โปรแกรมที่พัฒนาขึ้นมาจะสามารถติดต่อกับข้อมูลที่จะนำวิเคราะห์ได้ 2 รูปแบบ คือ Text File และ Relational Database โดยรายละเอียดจะกล่าวในหัวข้อถัดไป

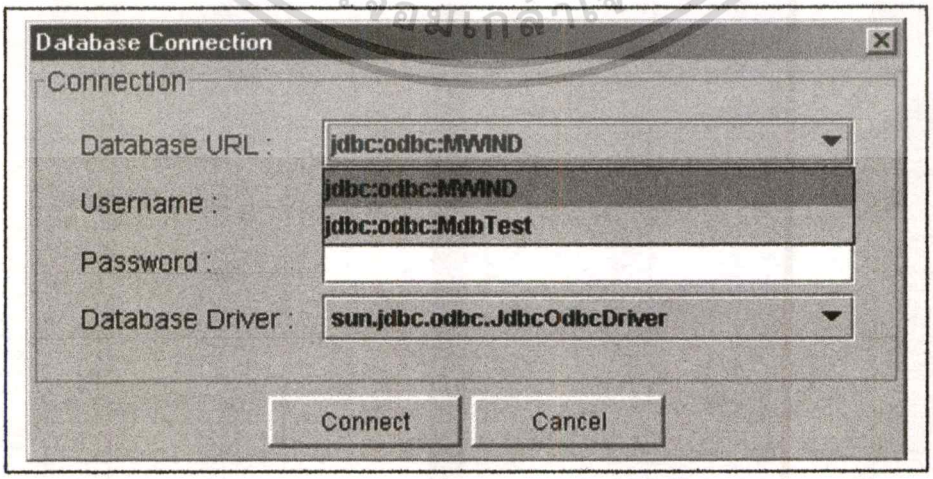
4.2 การติดต่อกับข้อมูลที่อยู่ในรูปของ Relational Database

วิธีการติดต่อกับข้อมูลที่อยู่ในรูปของ Relational Database จะแสดงไว้ในภาพที่ 4.2



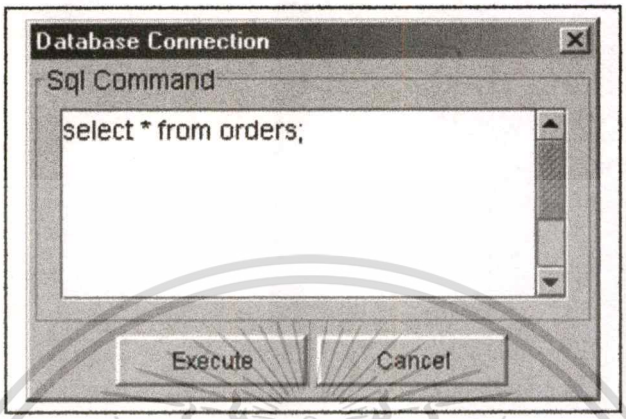
ภาพที่ 4.2 หน้าจอแสดงการติดต่อกับข้อมูลที่อยู่ในรูปของ Relational Database

จากนั้นจะปรากฏหน้าจอให้เลือกฐานข้อมูลที่ต้องการติดต่อ ดังภาพที่ 4.3



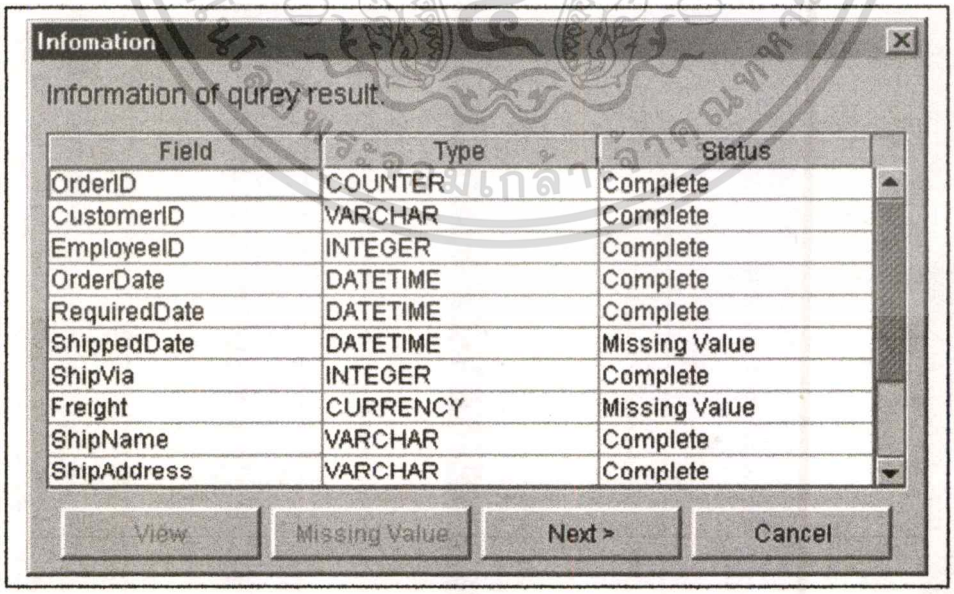
ภาพที่ 4.3 หน้าจอแสดงการเลือกฐานข้อมูลที่ต้องการติดต่อ

จากนั้นจะปรากฏหน้าจอให้พิมพ์คำสั่ง SQL(Structured Query Language) เพื่อเลือกข้อมูลที่จะนำมาวิเคราะห์ ดังภาพที่ 4.4



ภาพที่ 4.4 หน้าจอให้พิมพ์คำสั่ง SQL เพื่อเลือกข้อมูลที่จะนำมาวิเคราะห์

จากนั้นจะปรากฏหน้าจอที่แสดงรายละเอียดโครงสร้างของข้อมูลที่จะนำมาวิเคราะห์ ดังภาพที่ 4.5

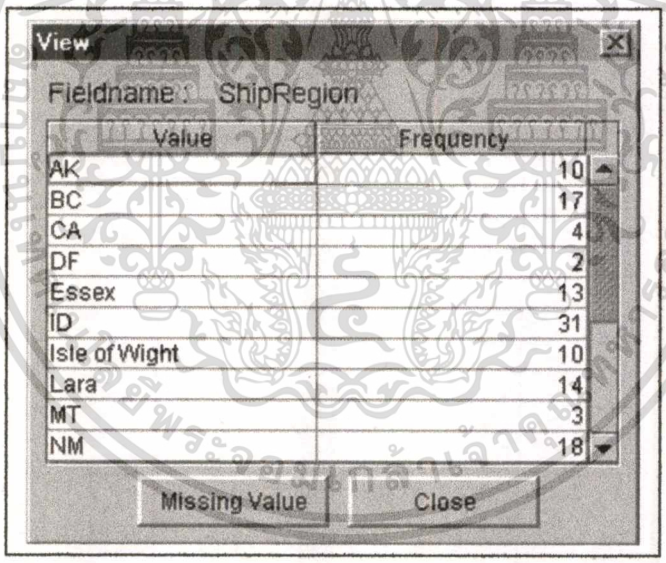


ภาพที่ 4.5 หน้าจอแสดงรายละเอียดโครงสร้างของข้อมูลที่จะนำมาวิเคราะห์

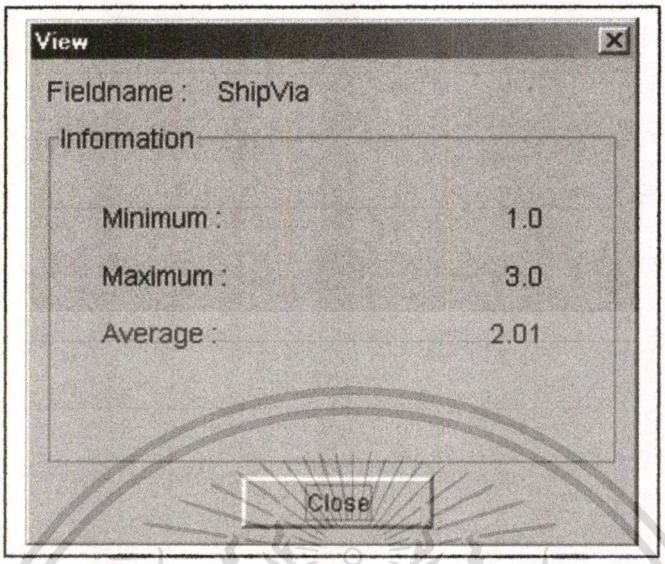
4.3 การตรวจสอบคุณภาพของข้อมูล

ภาพที่ 4.5 โปรแกรมจะแสดงโครงสร้างของข้อมูลในแต่ละแอททริบิว กรณีที่แอททริบิวใดมีข้อมูลที่หายไป(เกิด Missing Value) ช่อง Status ในภาพที่ 4.5 จะแสดงคำว่า Missing Value พร้อมทั้งปุ่มให้จัดการกับ Missing Value แต่ถ้าแอททริบิวใดมีข้อมูลครบสมบูรณ์ช่อง Status ในภาพที่ 4.5 จะแสดงคำว่า Complete

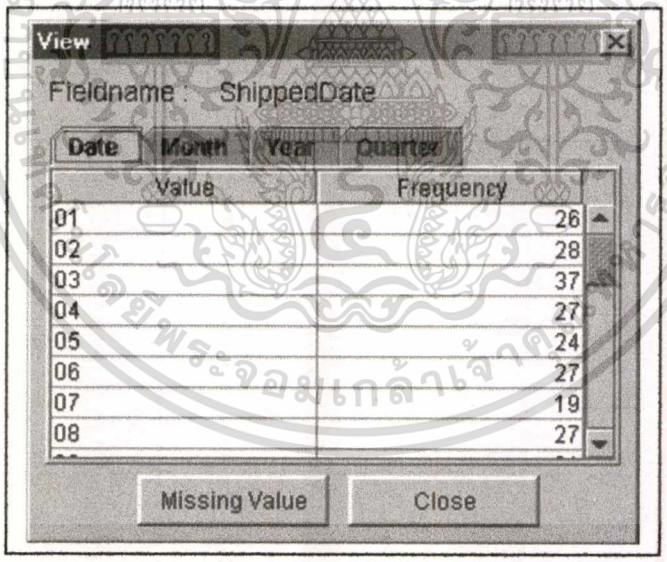
นอกจากนั้น โปรแกรมยังสามารถแสดงรายละเอียดที่เกี่ยวกับข้อมูลในแต่ละแอททริบิวได้ด้วย โดยถ้าแอททริบิวมีข้อมูลชนิดที่เป็นข้อความ(Text) จะแสดงค่าของข้อมูลและค่าความถี่ที่เกิดขึ้นของข้อมูลในแอททริบิวนั้นๆ ดังภาพที่ 4.6 ถ้าแอททริบิวมีข้อมูลชนิดที่เป็นตัวเลข(Number) จะแสดงค่าสูงสุด, ค่าต่ำสุด และ ค่าเฉลี่ยของข้อมูลในแอททริบิวนั้นๆ ดังภาพที่ 4.7 และถ้าแอททริบิวมีข้อมูลชนิดที่เป็นวันที่(Date/Time) จะแสดงค่าของข้อมูลตาม วัน, เดือน, ปี และ ไตรมาส ดังภาพที่ 4.8



ภาพที่ 4.6 หน้าจอแสดงรายละเอียดข้อมูลของแอททริบิวที่ชนิดข้อมูลเป็นข้อความ(Text)



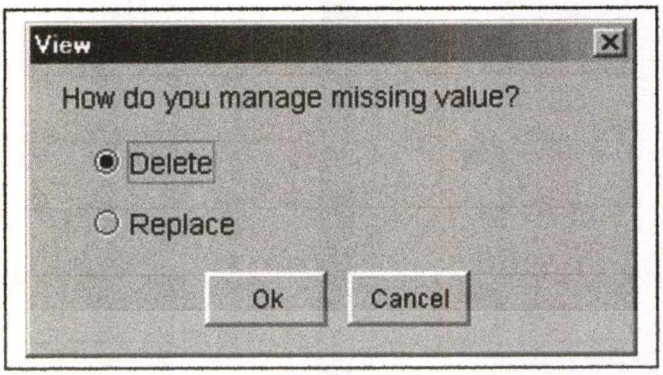
ภาพที่ 4.7 หน้าจอแสดงรายละเอียดข้อมูลของแอททริบิวที่ชนิดข้อมูลเป็นตัวเลข(Number)



ภาพที่ 4.8 หน้าจอแสดงรายละเอียดข้อมูลของแอททริบิวที่ชนิดข้อมูลเป็นวันที่(Date/Time)

4.4 การจัดการกับข้อมูลที่เป็น Missing Value

กรณีที่แอททริบิวใดมีการเกิด Missing Value ขึ้น โปรแกรมได้เตรียมปุ่มชื่อ Missing Value ไว้เพื่อจัดการกับข้อมูลที่เป็น Missing Value โดยสามารถลบ(Delete)เรคอร์ดที่เกิด Missing Value ทุกเรคอร์ด หรือ แทนที่(Replace)ข้อมูลที่เป็น Missing Value ด้วยค่าที่ป้อนเข้าไปใหม่

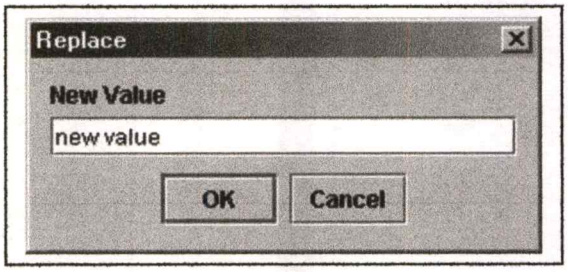


ภาพที่ 4.9 หน้าจอแสดงวิธีที่ใช้จัดการกับแอททริบิวที่มี Missing Value

จากภาพที่ 4.9 ถ้าเลือกวิธีจัดการกับข้อมูลที่เกิด Missing Value ด้วยวิธีการลบ(Delete) จะปรากฏหน้าจอเพื่อให้ยืนยันการลบเรคอร์ดที่ประกอบด้วย Missing Value ทุกเรคอร์ดดังภาพที่ 4.10

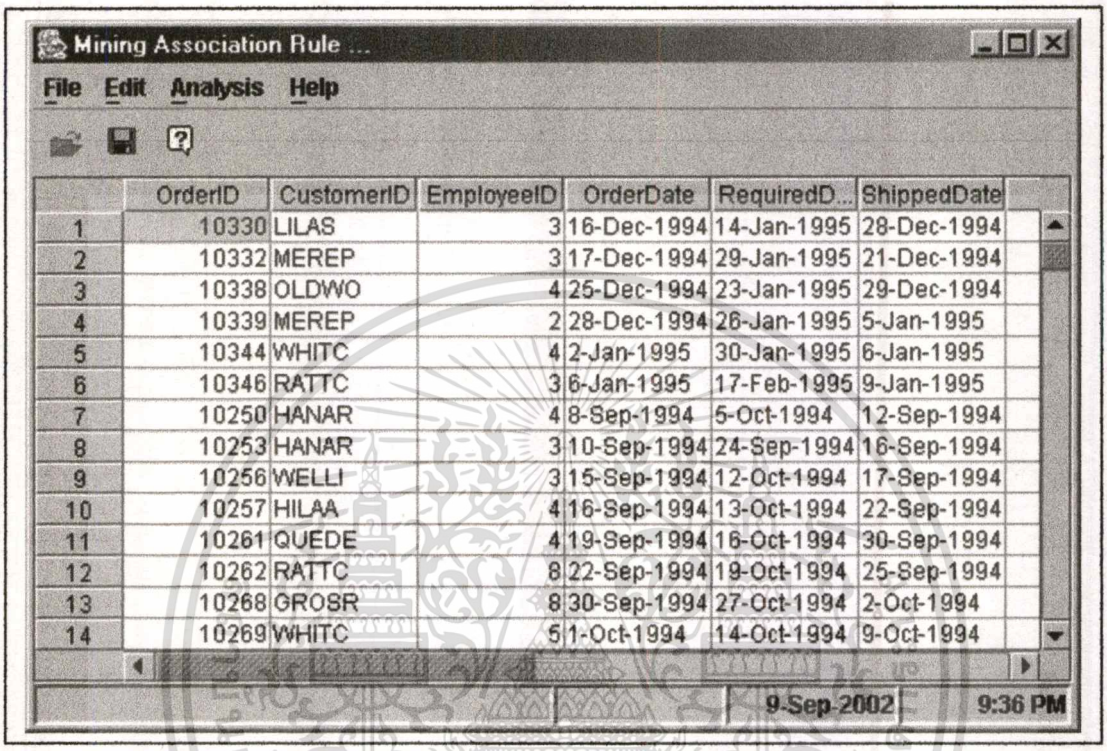
ภาพที่ 4.10 หน้าจอใช้เพื่อยืนยันการลบเรคอร์ดที่ประกอบด้วย Missing Value ทุกเรคอร์ด

จากภาพที่ 4.9 ถ้าเลือกวิธีจัดการกับข้อมูลที่เกิด Missing Value ด้วยวิธีการแทนที่(Replace) จะปรากฏหน้าจอให้ป้อนค่าใหม่ที่จะนำมาแทนที่ข้อมูลที่เกิด Missing Value ดังภาพที่ 4.11



ภาพที่ 4.11 หน้าจอให้ป้อนค่าใหม่ที่จะนำมาแทนที่ข้อมูลที่เกิด Missing Value

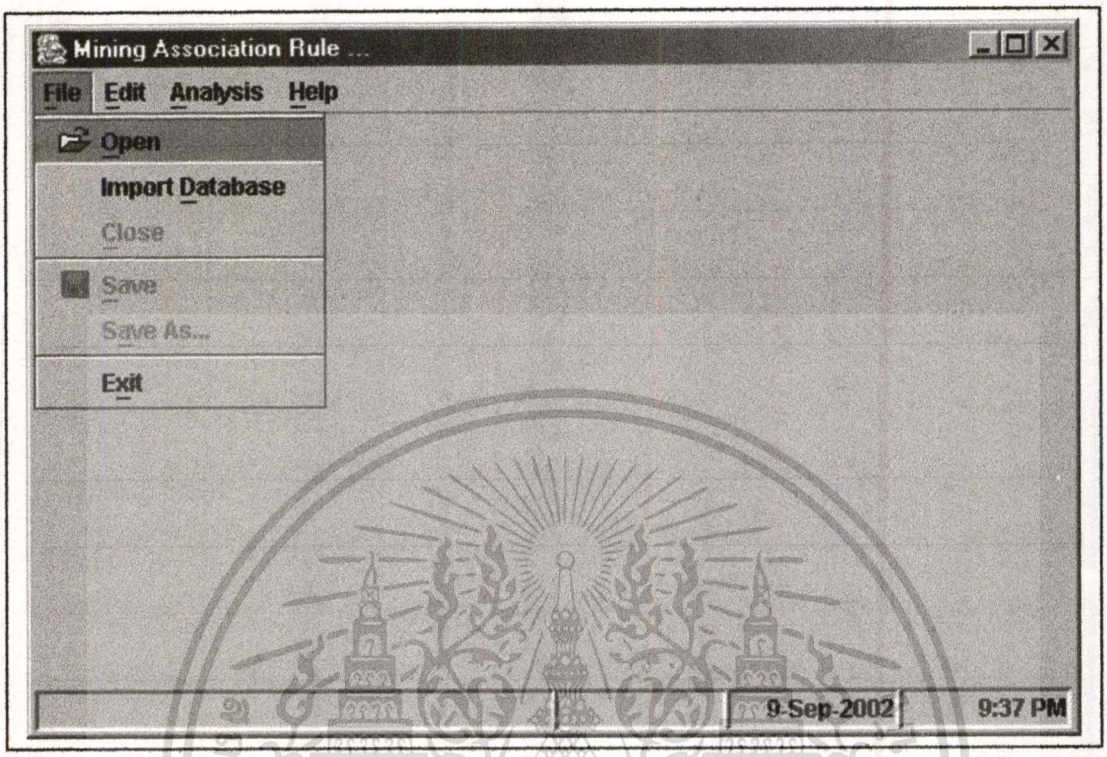
หลังจากผ่านขั้นตอนต่างๆ โปรแกรมจะนำข้อมูลที่ได้จาก Relational Database เข้าสู่ระบบ ดังภาพที่ 4.12



ภาพที่ 4.12 หน้าจอแสดงการนำข้อมูลที่ได้จาก Relational Database เข้าสู่ระบบ

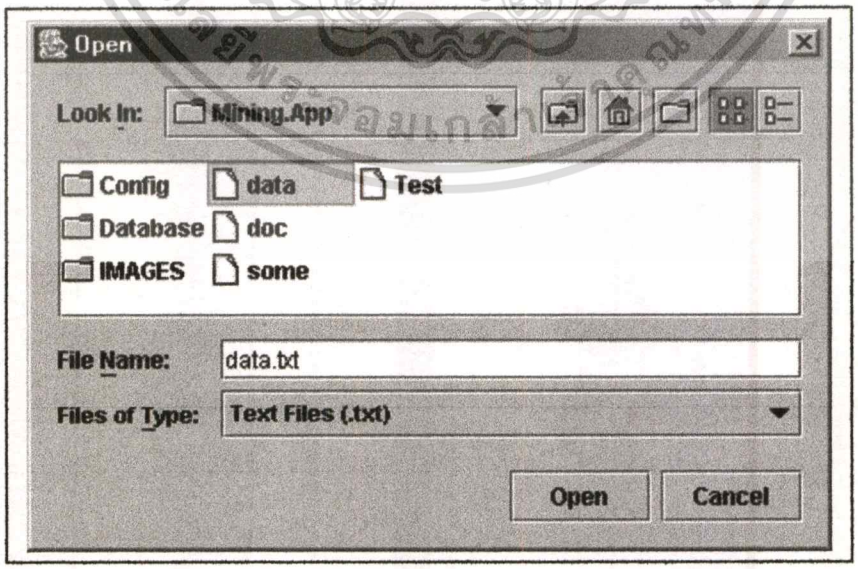
4.5 การติดต่อกับข้อมูลที่อยู่ในรูปของ Text File

วิธีการติดต่อกับข้อมูลที่อยู่ในรูปของ Text File จะแสดงไว้ในภาพที่ 4.13



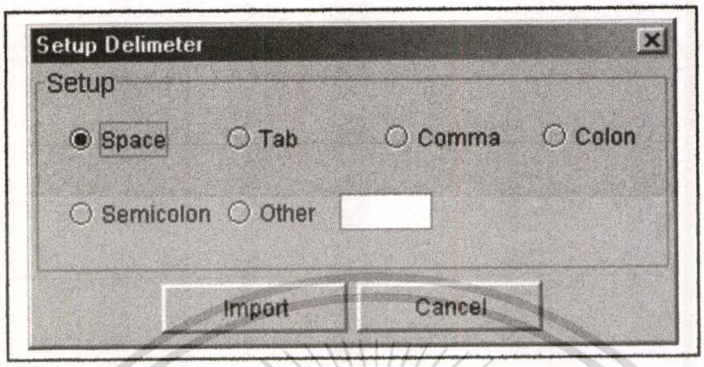
ภาพที่ 4.13 หน้าจอแสดงการติดต่อกับข้อมูลที่อยู่ในรูปของ Text File

จากนั้นจะปรากฏหน้าจอให้เลือก Text File ที่ต้องการติดต่อ ดังภาพที่ 4.14



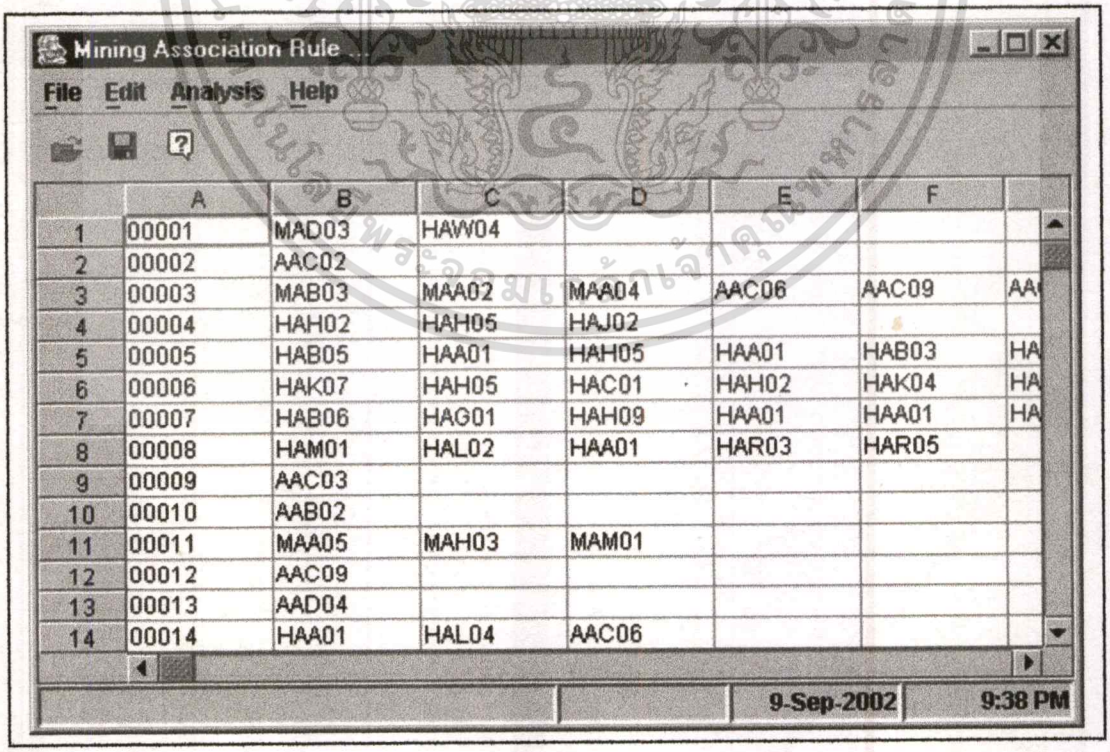
ภาพที่ 4.14 หน้าจอแสดงการเลือก Text File ที่ต้องการติดต่อ

หลังจากเลือก Text File ที่ต้องการติดต่อได้แล้ว จะปรากฏหน้าจอให้เลือก Delimeter ที่ใช้คั่นระหว่างข้อมูลว่าเป็นสัญลักษณ์อะไร ดังภาพ 4.15



ภาพที่ 4.15 หน้าจอแสดงการเลือก Delimeter ที่ใช้คั่นระหว่างข้อมูลว่าเป็นสัญลักษณ์อะไร

หลังจากผ่านขั้นตอนต่างๆแล้ว โปรแกรมจะนำข้อมูลที่ได้จาก Text File เข้าสู่ระบบดังภาพที่ 4.16

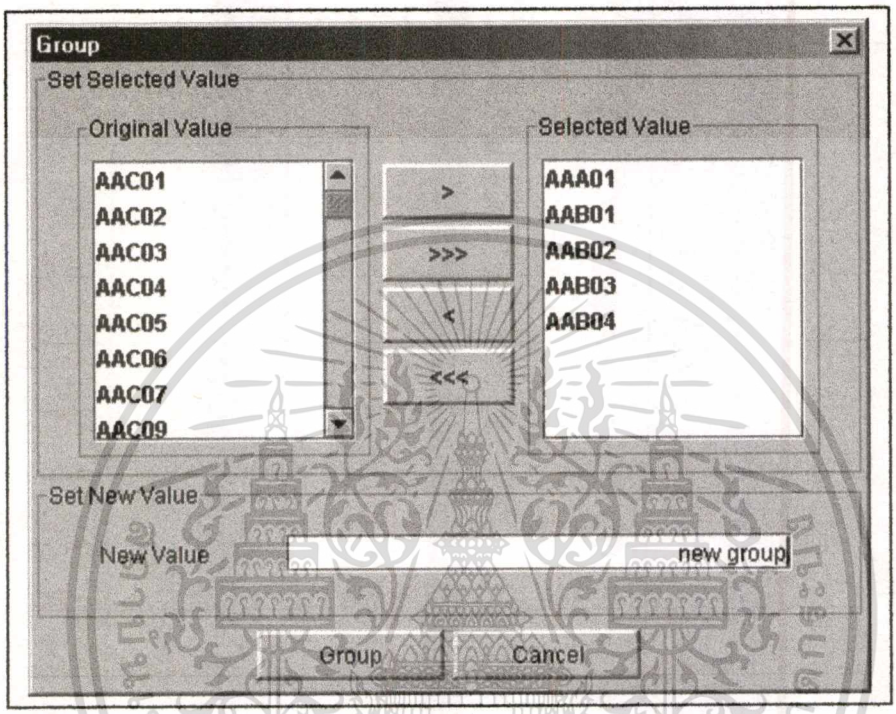


	A	B	C	D	E	F	
1	00001	MAD03	HAW04				
2	00002	AAC02					
3	00003	MAB03	MAA02	MAA04	AAC06	AAC09	AA
4	00004	HAH02	HAH05	HAJ02			
5	00005	HAB05	HAA01	HAH05	HAA01	HAB03	HA
6	00006	HAK07	HAH05	HAC01	HAH02	HAK04	HA
7	00007	HAB06	HAG01	HAH09	HAA01	HAA01	HA
8	00008	HAM01	HAL02	HAA01	HAR03	HAR05	
9	00009	AAC03					
10	00010	AAB02					
11	00011	MAA05	MAH03	MAM01			
12	00012	AAC09					
13	00013	AAD04					
14	00014	HAA01	HAL04	AAC06			

ภาพที่ 4.16 หน้าจอแสดงการนำข้อมูลที่ได้จาก Text File เข้าสู่ระบบ

4.6 การจัดกลุ่มข้อมูล

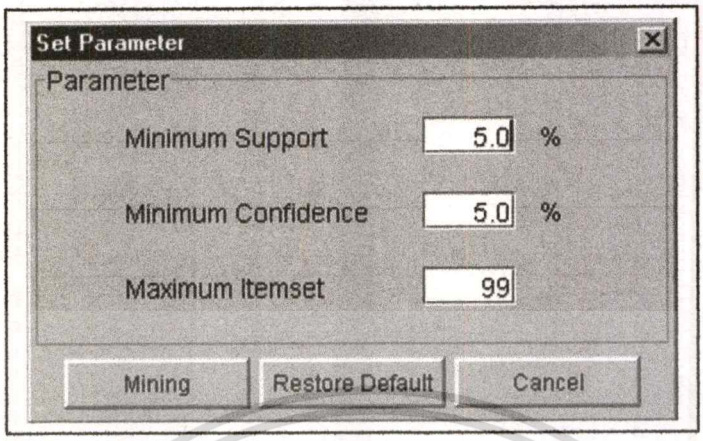
สำหรับข้อมูลที่มีค่าความถี่ที่เกิดขึ้นน้อย สามารถที่จะจัดกลุ่มรวมกับค่าอื่นได้โดยการเลือกเมนู Edit → Group จะปรากฏหน้าจอดังภาพที่ 4.17



ภาพที่ 4.17 หน้าจอแสดงการจัดกลุ่มข้อมูล

4.7 การกำหนดเงื่อนไขให้กับโปรแกรม

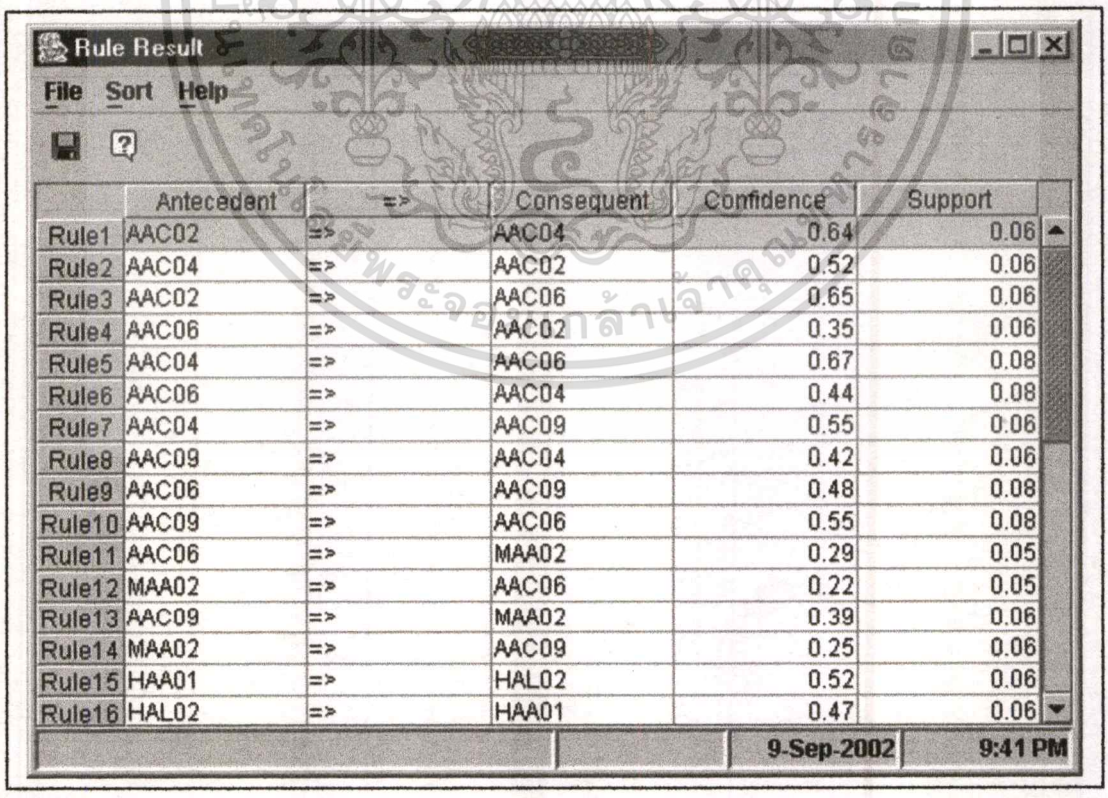
หลังจากตรวจสอบคุณภาพของข้อมูลเสร็จแล้วจะเข้าสู่การกำหนดเงื่อนไขให้กับโปรแกรม เช่น ค่า Minimum Support, Minimum Confidence และ Maximum Itemset เพื่อนำไปใช้สร้างกฎดังภาพที่ 4.18



ภาพที่ 4.18 หน้าจอแสดงการกำหนดเงื่อนไขต่างๆให้กับโปรแกรม

4.8 การแสดงผลลัพธ์

หลังจากโปรแกรมทำการวิเคราะห์ข้อมูลเสร็จแล้วจะแสดงผลลัพธ์ของกฎที่สร้างได้ดังภาพที่ 4.19



	Antecedent	=>	Consequent	Confidence	Support
Rule1	AAC02	=>	AAC04	0.64	0.06
Rule2	AAC04	=>	AAC02	0.52	0.06
Rule3	AAC02	=>	AAC06	0.65	0.06
Rule4	AAC06	=>	AAC02	0.35	0.06
Rule5	AAC04	=>	AAC06	0.67	0.08
Rule6	AAC06	=>	AAC04	0.44	0.08
Rule7	AAC04	=>	AAC09	0.55	0.06
Rule8	AAC09	=>	AAC04	0.42	0.06
Rule9	AAC06	=>	AAC09	0.48	0.08
Rule10	AAC09	=>	AAC06	0.55	0.08
Rule11	AAC06	=>	MAA02	0.29	0.05
Rule12	MAA02	=>	AAC06	0.22	0.05
Rule13	AAC09	=>	MAA02	0.39	0.06
Rule14	MAA02	=>	AAC09	0.25	0.06
Rule15	HAA01	=>	HAL02	0.52	0.06
Rule16	HAL02	=>	HAA01	0.47	0.06

ภาพที่ 4.19 หน้าจอแสดงผลลัพธ์ของกฎที่สร้าง

บทที่ 5
สรุปผลการศึกษาและข้อเสนอแนะ

โครงการนี้จัดทำขึ้นมาเพื่อนำเสนอให้เห็นถึงการนำทฤษฎี Data Mining มาประยุกต์ใช้กับธุรกิจ ในการหาความสัมพันธ์ในรูปแบบต่างๆของข้อมูลที่มีอยู่ เพื่อนำผลลัพธ์ที่ได้ไปใช้ในการสร้างแผนกลยุทธ์ทางการตลาดเพื่อทำให้ธุรกิจได้รับกำไรสูงสุด

5.1 สรุปผลการดำเนินงาน

โครงการนี้เป็นการพัฒนาโปรแกรมเพื่อค้นหา Association Rule โดยจะใช้ Direct Hashing and Pruning(DHP) และ Perfect Hashing and Pruning(PHP) Algorithm ซึ่งโปรแกรมที่พัฒนานี้สามารถติดต่อข้อมูลได้ 2 รูปแบบ คือ Text File และ Relational Database โดยที่ผู้ใช้เป็นคนกำหนดได้ว่าต้องการวิเคราะห์ข้อมูลอะไร ซึ่งผลลัพธ์ที่ได้ไม่อาจยืนยันได้ถึงความสำเร็จ 100 เปอร์เซ็นต์ในการกำหนดทิศทางทางการตลาดแต่เป็นเครื่องยืนยันอย่างหนึ่งถึงโอกาสของความสำเร็จที่จะเกิดขึ้น

5.2 ข้อเสนอแนะ

จากการที่ได้กล่าวมาแล้วว่าโอเปอเรชั่น Link Analysis ของ Data Mining สามารถแบ่งออกมาได้ 3 ลักษณะดังนี้ คือ Association Discovery, Sequential Pattern Discovery และ Similar Time Sequence Discovery ปัจจุบันโปรแกรมที่พัฒนานี้ตอบสนองเพียง Association Discovery เท่านั้น ดังนั้นในอนาคตควรจะมีการพัฒนาโปรแกรมนี้ต่อไปเพื่อให้สามารถที่จะตอบสนองต่อ Sequential Pattern Discovery และ Similar Time Sequence Discovery

บรรณานุกรม

Agrawal, R. and Srikant, R. 1994. "Fast Algorithms for Mining Association Rules." Proc. of the 20th Int'l Conference on Very Large Databases.

Ozel, S.A and Guveniz, H. 1996. "An Algorithm for Mining Association Rule Using Perfect Hashing and Database Prunning." IEEE.

Park, J.S. et al. 1997. "Using a Hash-Based Method with transaction Trimming for Mining Association Rules." IEEE Transactions on Knowledge and Data Engineering.

Simoudis, E. 1998. Discovery Data Mining From Concept to implementation. New Jersey: Prentice Hall.

ประวัติผู้เขียน

ชื่อ : นาย ฉัตร วัฒนศิริเกียรติ

วันเกิด : 14 มีนาคม พ.ศ. 2519

สถานที่เกิด : จังหวัดกรุงเทพมหานคร

วุฒิการศึกษาระดับปริญญาตรี : สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

E-mail Address : chat_zero@hotmail.com