การพัฒนาระบบเพื่อวิเคราะห์ความสัมพันธ์ของข้อมูลโดยใช้อัลกอริธึมดีเอชพี

The Development of the System for Link Analysis Using DHP Algorithm

โดย

นางสาว นิสา หงษ์สุรกุล

รหัส 44067072

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 2 ปีการศึกษา 2545

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ การพัฒนาระบบเพื่อวิเคราะห์ความสัมพันธ์ของข้อมูลโดยใช้อัลกอริธึมดีเอชพี
นักศึกษา นางสาวนิสา หงษ์สุรกุล
อาจารย์ที่ปรึกษา ผศ.ดร.วรพจน์ กรีสุระเดช
ระดับการศึกษา วิทยาศาสตรมหาบัณฑิต สาขาเทคโนโลยีสารสนเทศ
แขนงวิชา วิทยาการสารสนเทศ
ปีการศึกษา 2545

บทคัดย่อ

การดำเนินธุรกิจในปัจจุบันมีการแข่งขันกันอย่างรุนแรง เพื่อให้เกิดความได้เปรียบทางธุรกิจ จึงได้มีการนำดาต้าไมนิ่งมาช่วยในการวิเคราะห์ข้อมูลเพื่อช่วยในการตัดสินใจทางธุรกิจ โครงการนี้มีวัตถุประสงค์เพื่อสร้างระบบที่นำเอาดาต้าไมนิ่งมาใช้ในการวิเคราะห์ความสัมพันธ์ของข้อมูล เพื่อนำผลลัพธ์ที่ได้มาช่วยในการวางแผนกลยุทธ์ทางการตลาด รวมถึงการส่งเสริมการขาย โดยใช้อัลกอริธึมดีเอชพีที่พัฒนามาจากอัลกอริธึมอะพริออริซึ่งเป็นอัลกอริธึมพื้นฐานของ Mining Association Rule เพื่อเพิ่มประสิทธิภาพในการหากฎความสัมพันธ์ของข้อมูล

Title	The Development of the System for Link Analysis Using DHP Algorithm
Student	Ms. Nisa Hongsurakul
Advisor	Asst. Prof. Dr. Worapoj Kreesuradej
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2002

ABSTRACT

Nowadays, the challenge in the business area is higher than past, therefore data mining is brought to help the decision-making of business. The objective of this project is to develop the system that applies data mining to analyze the association of data that can use the result to increase benefit of business. This project uses DHP Algorithm develop from Apriori Algorithm develops this project.

กิตติกรรมประกาศ

โครงการพัฒนาระบบเพื่อวิเคราะห์ความสัมพันธ์ของข้อมูลโดยใช้อัลกอริธึมดีเอชพี สามารถที่จะสำเร็จลุล่วงไปได้ เพราะได้รับความช่วยเหลือจากบุคคลหลายฝ่ายดังนี้

บิดามารดา ญาติพี่น้อง และเพื่อนๆทุกคนที่คอยเป็นกำลังใจ และให้ความช่วยเหลือต่างๆ จนโครงการนี้สำเร็จด้วยดี

ผศ.ดร.วรพจน์ กรีสุระเดช ที่กรุณาให้คำปรึกษาและคำแนะนำต่างๆในการทำโครงงาน รวมทั้งช่วยในการวางระบบการทำโครงการให้เป็นไปอย่างมีระเบียบ เพื่อให้โครงการนี้สำเร็จลุล่วงไปด้วยดี

จึงใคร่ขอขอบคุณบุคคลดังกล่าวข้างต้นมา ณ โอกาสนี้

นิสา หงษ์สุรกุล

สารบัญ

หน้า

สารบัญตาราง

หน้า

ตารางที่

สารบัญรูป

หน้า

รูปที่

บทที่ 1

บทนำ

ปัจจุบันนี้มีการทำไมนิ่งกันมากขึ้น เนื่องจากการขยายตัวของอุตสาหกรรมขายปลีก เพื่อปรับปรุงกลยุทธ์การตลาด และความก้าวหน้าในเทคโนโลยีบาร์โค้ดทำให้องค์กรที่เป็นธุรกิจขายปลีกสามารถรวบรวม และจัดเก็บข้อมูลการขายจำนวนมากได้ และบริษัทแคตตาล็อกสามารถรวบรวมข้อมูลการขายจากใบสั่งซื้อสินค้าที่พวกเขาได้รับ เร็กคอร์ดในข้อมูลโดยทั่วๆไปประกอบด้วย transaction date, ไอเท็มที่ซื้อใน ทรานแซกชั่นนั้น และอาจจะมี customer id ถ้าทรานแซกชั่นทำผ่านการใช้บัตรเครดิต หรือบัตรลูกค้าชนิดต่างๆ การวิเคราะห์ข้อมูลทรานแซกชั่นในอดีตทำให้สามารถได้สารสนเทศที่เป็นพฤติกรรมการซื้อของลูกค้าที่มีค่า และช่วยในการปรับปรุงคุณภาพของการตัดสินใจทางธุรกิจ (เช่น ควรลดสินค้าอะไร, ควรวางสินค้าอะไรไว้ใกล้กัน และวิธีในการวางแผนการตลาด) สิ่งสำคัญ คือ การรวบรวมข้อมูลการขายให้เพียงพอ ก่อนที่จะสามารถดึงข้อสรุปที่มีความหมายจากข้อมูลเหล่านั้น ในปัจจุบันนี้ ปริมาณข้อมูลการขายมีแนวโน้มที่จะมีขนาดใหญ่ขึ้นเรื่อยๆ ดังนั้นมันจึงเป็นสิ่งจำเป็นที่ต้องสร้างอัลกอริธึมที่มีประสิทธิภาพ เพื่อนำไปไมนิ่งข้อมูลเหล่านี้

1.1 วัตถุประสงค์ของโครงการ

1) เพื่อศึกษาขั้นตอน และวิธีการในการค้นหาความสัมพันธ์ของข้อมูลโดยใช้อัลกอริธึม DHP
2) เพื่อสร้างระบบที่ใช้ในการวิเคราะห์ความสัมพันธ์ รูปแบบ และแนวโน้มของข้อมูล
3) เพื่อนำสารสนเทศที่ได้จากระบบ ไปช่วยในการวางแผนกลยุทธ์ต่างๆ เช่น นำไปใช้ช่วยในการวางแผนส่งเสริมการขาย เป็นต้น

1.2 ขอบเขตของโครงการ

โครงการนี้เป็นการศึกษาถึง Link Analysis ซึ่งเป็นเทคนิคหนึ่งใน Data Mining เพื่อการวิเคราะห์ความสัมพันธ์ของข้อมูล โดยนำเอาอัลกอริธึม DHP (Direct Hashing and Pruning) มาใช้ในการพัฒนาระบบเพื่อหาความสัมพันธ์ของข้อมูล

1.3 ขั้นตอนการดำเนินงาน

1) กำหนดหัวข้อ เป้าหมาย จุดประสงค์ และขอบเขตของการพัฒนาระบบ
2) ศึกษาทฤษฎีที่เกี่ยวข้อง ได้แก่ ดาต้าไมนิ่ง, ขั้นตอนในการทำดาต้าไมนิ่ง, Link Analysis, อัลกอริธึมดีเอชพี
3) ออกแบบระบบ
4) พัฒนาระบบงานเพื่อวิเคราะห์ข้อมูล
5) ทดสอบระบบ โดยทดสอบกับข้อมูลต่างๆ ได้แก่ ข้อมูลประเภท Text File และ Relational Database
6) ปรับปรุง และแก้ไขข้อผิดพลาดที่เกิดขึ้น
7) สรุปผลการศึกษา

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) เข้าใจขั้นตอนและวิธีในการทำดาต้าไมนิ่ง
2) ได้ระบบที่ใช้ในการวิเคราะห์ความสัมพันธ์ของข้อมูล เพื่อนำไปประยุกต์ใช้ในธุรกิจต่างๆ

บทที่ 2

ดาต้าไมนิ่ง

(Data Mining)

การสืบค้นความรู้ที่เป็นประโยชน์และน่าสนใจบนฐานข้อมูลขนาดใหญ่มาก (Knowledge Discovery from very large Databases: KDD) หรือที่เรียกกันว่า Data Mining เป็นสาขาหนึ่งในวิทยาศาสตร์คอมพิวเตอร์ที่กำลังได้รับความสนใจอย่างสูงในปัจจุบัน ด้วยเทคนิคของ KDD ข้อมูลขนาดใหญ่จะถูกวิเคราะห์และสืบค้นความรู้สิ่งที่สำคัญออกมารวบรวมและจัดเก็บให้อยู่ในรูปฐานความรู้ (Knowledge Base) เพื่อใช้สำหรับการสืบค้นสิ่งที่ต้องการซึ่งไม่สามารถสืบค้นได้จากวิธีการของระบบจัดการฐานข้อมูล (DBMS) โดยทั่วไป เช่น การวิเคราะห์หาความสัมพันธ์ของข้อมูล หรือการทำนายปรากฏการณ์ต่างๆของข้อมูลที่กำลังจะเกิดขึ้น ตลอดจนนำความรู้ที่ได้ไปช่วยในกระบวนการตัดสินใจ เทคนิคต่างๆเหล่านี้ สามารถนำไปใช้ให้เกิดประโยชน์ได้ในหลาย ๆ สาขา

2.1 Data Mining คืออะไร

Data Mining คือ กระบวนการค้นหาและวิเคราะห์ข้อมูลแบบกึ่งอัตโนมัติ โดยจะนำข้อมูลจำนวนมากมาผ่านกระบวนการเพื่อหารูปแบบและข้อมูลที่เป็นประโยชน์จากฐานข้อมูลนั้นๆ เพื่อนำข้อมูลที่ได้มาใช้ในงานต่างๆ เช่น นำไปใช้ช่วยในการตัดสินใจทางธุรกิจ

2.2 เหตุผลที่ต้องมี Data Mining

ปัจจัยที่ทำให้ Data Mining เป็นที่ได้รับความสนใจอย่างสูงมีดังต่อไปนี้

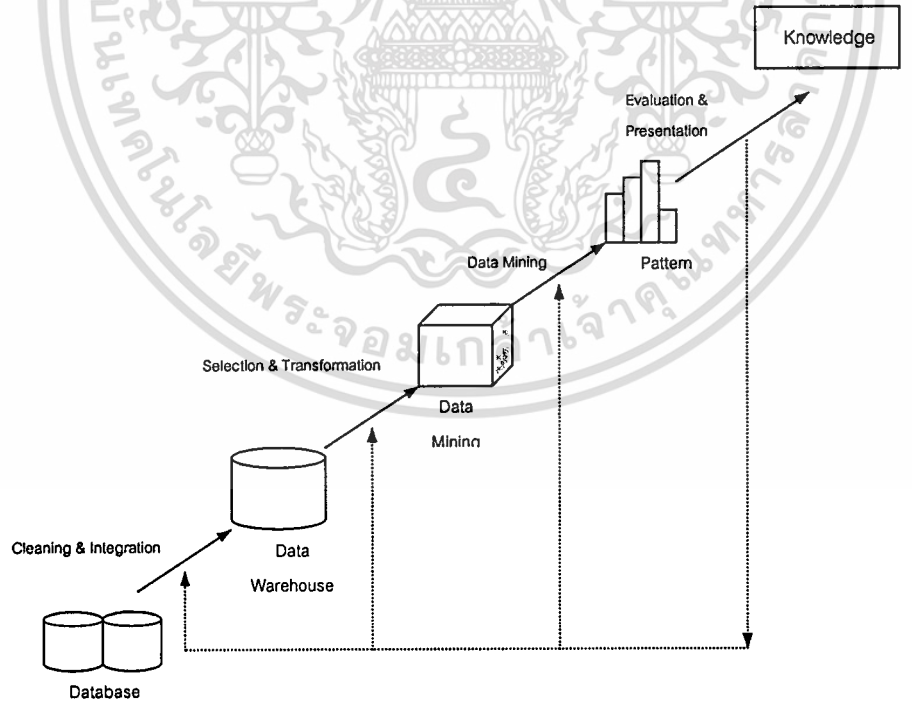
- จำนวนและขนาดข้อมูลขนาดใหญ่ถูกผลิตและขยายตัวอย่างรวดเร็ว การสืบค้นความรู้จะมีความหมายก็ต่อเมื่อฐานข้อมูลที่ใช้มีขนาดใหญ่มาก ปัจจุบันมีจำนวนและขนาดข้อมูลขนาดใหญ่ที่ขยายตัวอย่างรวดเร็ว โดยผ่านทางอินเตอร์เน็ต, ดาวเทียม และแหล่งผลิตข้อมูลอื่น ๆ เช่น เครื่องอ่านบาร์โค้ด, เครดิตการ์ด, อีคอมเมิร์ส เป็นต้น
- ข้อมูลถูกจัดเก็บเพื่อนำไปสร้างระบบการสนับสนุนการตัดสินใจ เพื่อเป็นการง่ายต่อการนำข้อมูลมาใช้ในการวิเคราะห์เพื่อการตัดสินใจ ส่วนมากข้อมูลจะถูกจัดเก็บแยกมาจาก

ระบบปฏิบัติงาน (operational systems) โดยจัดอยู่ ในรูปของคลังหรือเหมืองข้อมูล (Data Warehouse) ซึ่งเป็นการง่ายต่อการนำเอาไปใช้ในการสืบค้นความรู้

- ระบบคอมพิวเตอร์สมรรถนะสูงมีราคาต่ำลง เทคนิคดาตาไมน์นิงประกอบไปด้วยอัลกอริทึมที่มีความซับซ้อนและความต้องการการคำนวณสูง จึงจำเป็นต้องใช้งานกับระบบคอมพิวเตอร์สมรรถนะสูง ปัจจุบันระบบคอมพิวเตอร์สมรรถนะสูงมีราคาที่ต่ำลง พร้อมด้วยเริ่มมีเทคโนโลยีที่นำเครื่องไมโครคอมพิวเตอร์จำนวนมากมาเชื่อมต่อกันโดยเครือข่ายความเร็วสูง (PC Cluster) ทำให้ได้ระบบคอมพิวเตอร์สมรรถนะสูงในราคาถูก
- การแข่งขันอย่างสูงในด้านอุตสาหกรรมและการค้า เนื่องจากปัจจุบันมีการแข่งขันอย่างสูงในด้านอุตสาหกรรมและการค้า มีการผลิตข้อมูลไว้อย่างมากมายแต่ไม่ได้นำมาใช้ให้เกิดประโยชน์เท่าที่ควร จึงจำเป็นอย่างยิ่งที่จะต้องควบคุมและสืบค้นความรู้ที่ถูกซ่อนอยู่ในฐานข้อมูล เพื่อที่จะนำความรู้ที่ได้รับไปวิเคราะห์ประกอบการตัดสินใจการจัดการในระบบต่าง ๆ ให้เกิดประโยชน์สูงสุด

2.3 ขั้นตอนการสืบค้นความรู้ (knowledge) จากฐานข้อมูล

จากรูปที่ 2.1 ประกอบไปด้วยลำดับขั้นตอนดังต่อไปนี้



รูปที่ 2.1 ขั้นตอนการสืบค้นความรู้

- **data cleaning** คือการนำข้อมูลที่ไม่สำคัญ, ข้อมูลที่ใช้ไม่ได้ และส่วนที่มีข้อมูลไม่ครบออกไป เช่น ในคอลัมน์ที่มีข้อมูลทั้งโค้ก และเป๊ปซี่ ควรทำให้เป็นน้ำอัดลมเหมือนกัน
- **data integration** คือการนำข้อมูลจากหลายแหล่งมารวมกัน ซึ่งสองขั้นแรกนี้อาจได้มาจาก data warehouse
- **data selection** คือการวิเคราะห์เลือกสิ่งที่สำคัญและสนใจจากฐานข้อมูลเพื่อมาไมน์
- **data transformation** คือการแปลงข้อมูล หรือรวบรวมข้อมูลเข้าด้วยกัน เพื่อให้เหมาะสมสำหรับการไมน์ต่อไปแล้วจึงแปลงข้อมูลดังกล่าว ซึ่งทั้งขั้น selection และ transformation นี้อาจสลับกันได้ตามลักษณะข้อมูล
- **data mining** เป็นกระบวนการที่สำคัญในการเลือกเทคนิค Data Miningที่เหมาะสม เพื่อหารูปแบบ (pattern) ต่าง ๆ ออกมา
- **pattern evaluation** เป็นกระบวนการเลือกรูปแบบ (pattern) ที่เหมาะสมจากขั้น Data Mining เพื่อนำไปสู่การค้นพบความรู้
- **knowledge presentation** เป็นเทคนิคที่ใช้ในการแสดงความรู้ที่ได้จากการไมน์สู่ผู้ใช้

จะเห็นได้ว่า Data Mining เป็นเพียงแค่ขั้นตอนหนึ่งในกระบวนการทั้งหมด อย่างไรก็ตาม ขั้นตอนนี้ถือได้ว่าเป็นขั้นตอนที่สำคัญมากในการหารูปแบบน่าสนใจที่ซ่อนอยู่ออกมาจากฐานข้อมูลทั้งหมด

2.4 กระบวนการในการทำ Data Mining

กระบวนการหลักในการวิเคราะห์ข้อมูลของ Data Mining แบ่งออกเป็น 4 ประเภท ได้แก่

- **Predictive Modeling** เป็นกระบวนการสร้างโมเดลเพื่อทำนายหาค่าที่ต้องการจากข้อมูลที่มีอยู่
- **Data Segmentation** (Clustering) เป็นวิธีการในการจัดกลุ่มให้กับข้อมูล
- **Link Analysis** เป็นวิธีในการวิเคราะห์ความสัมพันธ์ระหว่างข้อมูล
- **Deviation Detection** เป็นวิธีในการวิเคราะห์สิ่งที่แตกต่างในข้อมูล

บทที่ 3

ลิงค์อนาไลซิส

(Link Analysis)

Link Analysis เป็นกระบวนการในการค้นหาความสัมพันธ์ระหว่างข้อมูลหรือเซ็ตของข้อมูลในฐานข้อมูล ซึ่งเรามักเรียกความสัมพันธ์นี้ว่า association เช่น การหาความสัมพันธ์ระหว่างสินค้าหรือบริการที่ลูกค้ามีแนวโน้มที่จะซื้อพร้อมกัน หรือซื้อในเวลาต่อมา

เทคนิคที่ในการทำ Link Analysis ได้แก่ Association Rule Mining และ Sequential Pattern Mining

สิ่งที่แตกต่างระหว่าง Association Rule Mining และ Sequential Pattern Mining คือ ถ้ากำหนดให้ทรานแซกชั่นหนึ่งเป็นเซ็ตของสินค้าที่ถูกซื้อในการไปร้านของลูกค้าหนึ่งครั้ง association rule mining จะวิเคราะห์การซื้อสินค้าในแต่ละทรานแซกชั่น เพื่อหาความสัมพันธ์ที่ซ่อนอยู่ของสินค้า นั่นก็คือ สินค้าอะไรที่มีแนวโน้มว่าจะขายพร้อมกัน .

Sequential pattern mining จะวิเคราะห์หาความสัมพันธ์ข้ามทรานแซกชั่น การซื้อที่สัมพันธ์กัน เพื่อหาข้อมูลเกี่ยวกับลำดับที่ลูกค้าซื้อสินค้าและบริการ เพื่อที่จะเข้าใจพฤติกรรมการซื้อของลูกค้า

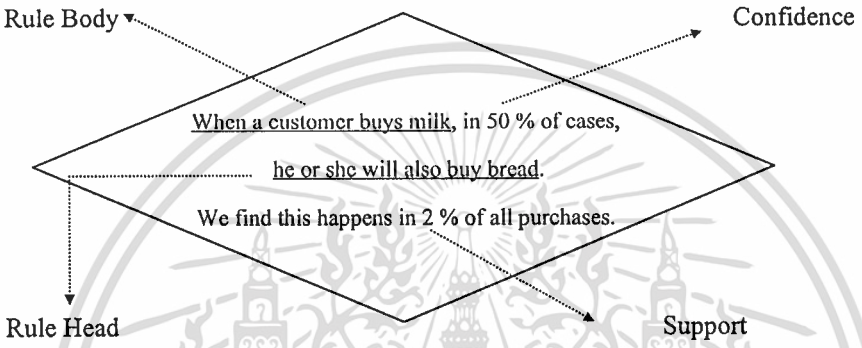
3.1 Association Rule Mining

เป็นเทคนิคในการค้นหาความสัมพันธ์ของข้อมูล เพื่อนำไปใช้ในการวิเคราะห์ หรือทำนายปรากฏการณ์ต่างๆ โดยเทคนิคนี้ใช้กันอย่างแพร่หลายในการขายสินค้า หรือการวิเคราะห์ข้อมูลที่เป็นทรานแซกชั่น

ตัวอย่างของ association rule ได้แก่ market basket analysis (MBA) ซึ่งเป็นเทคนิคที่นำไปใช้ในด้านการตลาด เพื่อวิเคราะห์พฤติกรรมการซื้อของลูกค้า โดยหาความสัมพันธ์ระหว่างสินค้าต่างๆที่ลูกค้าซื้อ การค้นพบความสัมพันธ์สามารถช่วยผู้ขายพัฒนากลยุทธ์ทางการตลาด โดยพิจารณาจากสินค้าที่มักจะถูกซื้อพร้อมกัน เช่น ถ้าลูกค้าซื้อนม เขามักจะซื้อขนมปังพร้อมกัน ข้อมูลนี้สามารถนำไปสู่การเพิ่มยอดขาย โดยการช่วยผู้ขายในการวางแผนการตลาด และการวางแผนการจัดชั้นวางสินค้า ตัวอย่างเช่น การวางนมและขนมปังไว้ใกล้กัน อาจจะเพิ่มยอดการสินค้าทั้งสองนี้

หรืออาจวางไว้คนละมุมของร้าน เพื่อที่ว่า เมื่อลูกค้าซื้อนมแล้ว ต้องการที่จะซื้อขนมปัง ก็จะต้องเดินผ่านสินค้าตัวอื่นๆ ทำให้มีโอกาสที่ลูกค้าจะซื้อสินค้าตัวอื่นเพิ่มขึ้นด้วย

รูปแบบของกฎ คือ "If X, then Y" หรือ "When X then Y" โดย X และ Y เกิดขึ้นพร้อมกันในทรานแซกชั่นเดียวกัน เรียก X ว่า Rule Body และเรียก Y ว่า Rule Head พารามิเตอร์ที่สำคัญในกฎนี้คือ ค่า support (prevalence) และ confidence (predictability) ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 Association Rule

ตารางที่ 3.1 จำนวนทรานแซกชั่นของสินค้าจากจำนวนทรานแซกชั่นทั้งหมด 100,000 รายการ

สินค้า	จำนวนทรานแซกชั่น
นม	4,000
ขนมปัง	6,000
นมและขนมปัง	2,000

- **ค่า Support** คือ ค่าสัดส่วนระหว่างจำนวนทรานแซกชั่นที่สนับสนุนกฎต่อจำนวนทรานแซกชั่นทั้งหมด ทรานแซกชั่นจะสนับสนุนกฎ "When X then Y." ถ้ามีไอเท็ม X และ Y ในกฎเกิดขึ้นในทรานแซกชั่นเดียวกัน จากตัวอย่างในตารางที่ 3.1 ค่า support ที่ได้ คือ 2 % ซึ่งได้มาจากจำนวนรายการที่ขายนมและขนมปังคู่กันเทียบกับจำนวนทรานแซกชั่นทั้งหมด

- **ค่า Confidence** คือ ค่าสัดส่วนระหว่างจำนวน ทรานแซกชั่นที่สนับสนุนกฎต่อจำนวนทรานแซกชั่นที่สนับสนุนส่วน rule body ในตัวอย่างค่า confidence ที่ได้ คือ 50 % ซึ่งได้มาจากจำนวนรายการที่ขายนมและขนมปังคู่กันเทียบกับจำนวนรายการของนม

- **ค่า Expected Confidence** คือ ค่าสัดส่วนระหว่างจำนวนทรานแซกชั่นที่สนใจต่อจำนวนทรานแซกชั่นทั้งหมด เช่น Expected Confidence ของนมคือ 4 % และของขนมปังคือ 6 %

- **ค่า Lift** คือ ค่าที่แสดงความน่าเชื่อถือของความสัมพันธ์ระหว่างเหตุการณ์ โดยเป็นค่าสัดส่วนระหว่างค่า Confidence กับค่า Expected Confidence ของจำนวนชุดข้อมูลของ "Y" เช่น จากข้อมูลในตารางที่ 3.1 ค่า confidence มีค่า 50 % และค่า Expected Confidence ของการซื้อขนมปังคือ 6 % ดังนั้นจะได้ค่า lift เป็น 8.33 ซึ่งจะได้กฎว่า "คนที่ซื้อนม จะซื้อขนมปังด้วยคิดเป็น 8 % " และถ้าค่า Confidence ยิ่งมีค่ามาก ความสัมพันธ์ระหว่างนมและขนมปังก็จะยิ่งมากขึ้น

3.1.1 อัลกอริธึม Apriori

Apriori เป็นอัลกอริธึมพื้นฐานในการไมนิ่งชุดข้อมูลของ Association Rule โดย Apriori ใช้หลักการการทำซ้ำ ที่ซึ่งชุดข้อมูล k+1 ได้มาจากชุดข้อมูล k (k-itemset) แรกสุด เซ็ตของ 1-itemset จะถูกหาออกมาก่อน โดยเซ็ตนี้จะเป็น L_1 และ L_1 ถูกใช้ในการหา L_2 ซึ่งใช้ในการหา L_3 ต่อไปเรื่อยๆ จนกระทั่งไม่พบ itemset ใหม่

เพื่อเพิ่มประสิทธิภาพในการสร้าง large itemset (itemset ที่มีค่ามากกว่าค่าสนับสนุนต่ำสุด) จึงมีคุณสมบัติที่สำคัญ เรียกว่า คุณสมบัติของ Apriori กำหนดไว้ว่า เซ็ตย่อยทั้งหมดของ large itemset ต้องเป็น large ด้วย

Apriori แบ่งออกเป็นสองขั้นตอน ดังนี้

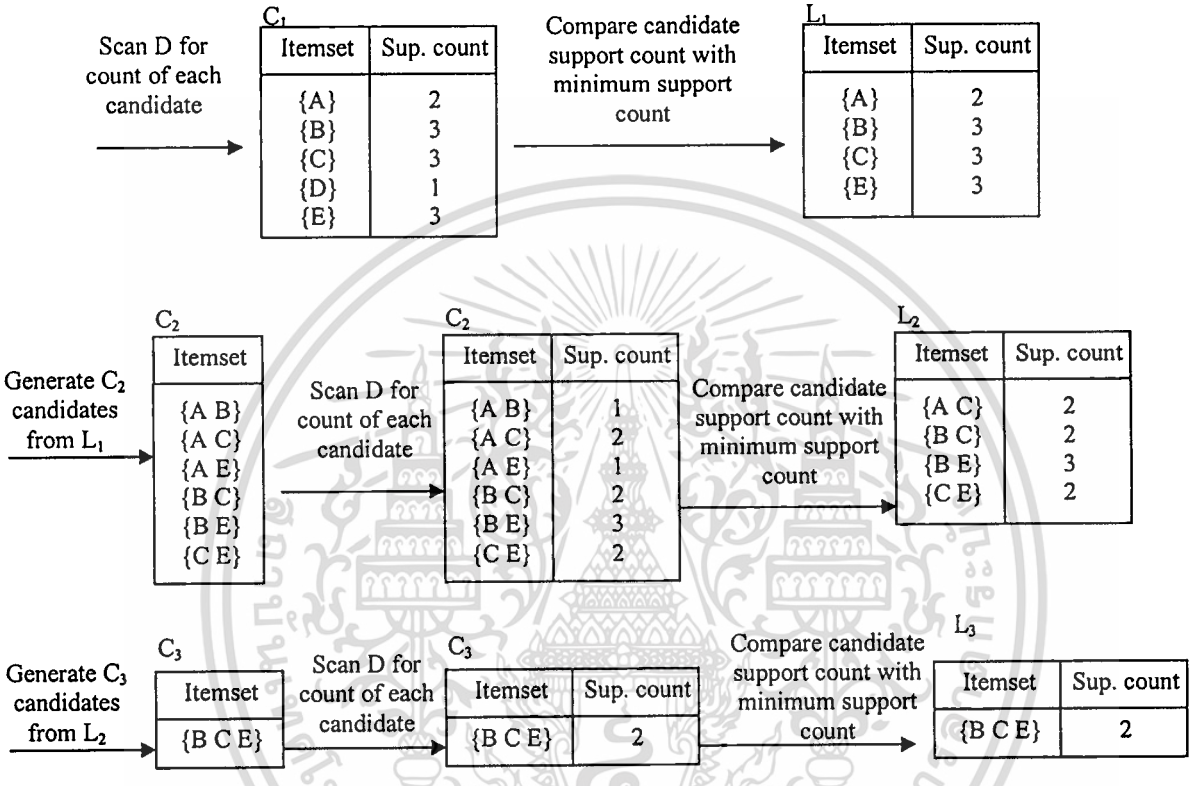
- **ขั้นตอนการ Join:** ในการหา L_k เซ็ตของ candidate k-itemsets จะสร้างขึ้นจากการ join L_{k-1} กับตัว L_{k-1} เอง เซ็ตของ candidate นี้ เรียกว่า C_k
- **ขั้นตอนการ prune:** การหา L_k โดยตัดค่า candidate ใน C_k ที่มีค่าน้อยกว่าค่าสนับสนุนต่ำสุดออก

ตัวอย่างการทำงานของ Apriori โดยใช้ข้อมูลในตารางที่ 3.2 และในรูปที่ 3.2 แสดงอัลกอริธึม Apriori ในการค้นหา itemset (กำหนดให้ตารางที่ 3.2 ชื่อ D)

ตารางที่ 3.2 ตัวอย่างฐานข้อมูลทรานแซกชั่น

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

ขั้นตอนการทำงานของ Apriori เป็นดังต่อไปนี้



รูปที่ 3.2 การสร้าง candidate itemset และ large itemset

1. ในการทำซ้ำรอบแรกของอัลกอริธึม อัลกอริธึมจะนับจำนวนเหตุการณ์ของแต่ละรายการจากทรานแซกชั่นทั้งหมด และได้ตาราง C_1 ออกมา
2. เซ็ตของ large 1-itemset (L_1) จะประกอบด้วย itemset จาก C_1 ที่มีค่าถึง minimum support ซึ่งในตัวอย่างนี้กำหนดให้ minimum support มีค่าเป็น 2
3. ในการหา L_2 อัลกอริธึมใช้ L_1 join L_1 เพื่อสร้าง C_2 ก่อน
4. จากนั้นนับจำนวนของแต่ละ itemset ใน C_2 โดยดูจากตารางที่ 2 และจะได้ผลดังตารางตรงกลางของแถวที่สองในรูปที่ 1
5. หา L_2 ออกมา โดย L_2 ประกอบด้วย itemset ใน C_2 ที่มีค่า >= 2

6. จากนั้น join L_2 เข้าด้วยกันจะได้ {{A B C}, {A C E}, {B C E}} และจากคุณสมบัติของ Apriori ที่ว่าซับเซ็ตทั้งหมดของ large itemset ต้อง large ด้วย ทำให้ candidate 2 ตัวแรกถูกตัดออกไปจาก C_3
7. นับจำนวนแต่ละ itemset เพื่อหา L_3 ซึ่งประกอบด้วย itemset ใน C_3 ที่มีค่า >= 2
8. จาก L_3 พบว่าไม่สามารถหา C_4 ได้ ทำให้อัลกอริธึมสิ้นสุดลง และได้ large itemset ทั้งหมดออกมา

3.1.2 ข้อดีของ Association Rule

ข้อดีของ Association Rule ได้แก่

- ง่าย
- มีพารามิเตอร์เพียง 2 ตัวเท่านั้นที่ต้องถูกกำหนด คือ ค่า support และ ค่า confidence
- กฎที่ได้จะอธิบายตัวมันเองอยู่แล้ว
- สามารถไม่นิ่งกับข้อมูลบางส่วนได้ ทำให้ลดปัญหาในกรณีที่มีข้อมูลไม่สมบูรณ์ได้
- เหมาะสำหรับจัดการข้อมูลที่มีจำนวนทรานแซกชั่นมากๆ
- สามารถกำหนดค่า minimum support และค่า minimum confidence ได้ ซึ่งช่วยให้สามารถควบคุมจำนวนผลลัพธ์ได้

3.1.3 ข้อเสียของ Association Rule

ข้อเสียของ Association Rule ได้แก่

- ไม่มีการคิดค่าทางธุรกิจของความสัมพันธ์ ตัวอย่างเช่น ยอดขายของไวน์ขาวราคาแพงขวดหนึ่งมีค่าเท่าๆกับยอดขายนมทั้งลัง

3.2 Sequential Pattern Mining

Sequential Pattern Mining คือ การไมน์นิงหารูปแบบการเกิดที่สัมพันธ์กับเวลาหรือลำดับ เช่น ใช้ระบุความสัมพันธ์ของการซื้อสินค้าอย่างหนึ่ง แล้วจะซื้อสินค้าอีกอย่างในเวลาต่อมา นอกจากนี้ Sequential Pattern Mining สามารถนำไปใช้ในการวิเคราะห์ข้อมูลเพื่อหาเป้าหมายทางการตลาด, การรักษาลูกค้า, การทำนายสภาพอากาศ เป็นต้น

3.2.1 เทคนิคของ Sequential Pattern Mining

ค่าต่างๆใน Sequential Pattern จะเหมือนกับใน Association Rule ยกเว้นค่า support โดยค่า support ใน Sequential Pattern คำนวณจากอัตราส่วนจำนวนลูกค้าที่มีข้อมูลการซื้อสินค้าเป็นลำดับต่อจำนวนลูกค้าทั้งหมด

ตารางที่ 3.3 แสดงฐานข้อมูลของร้านขายเครื่องดื่ม โดยข้อมูลจะถูกเรียงลำดับตามรหัสของลูกค้า และรหัสของ Sequential Pattern ตัวอย่างเช่น ลูกค้า C. John มาที่ร้านติดต่อกัน 3 วัน เขาซื้อเบียร์ในวันแรก ไวน์และเหล้าผลไม้ในวันถัดไป และบรั่นดีในวันที่ 3

ตารางที่ 3.3 ฐานข้อมูลทรานแซกชั่น

Customer	Transaction Time	Item Bought
B. Adams	Dec 21, 2001 5.27 pm	เบียร์
B. Adams	Dec 22, 2001 10.34 am	บรั่นดี
J. Brown	Dec 20, 2001 10.13 am	น้ำผลไม้, โค้ก
J. Brown	Dec 20, 2001 11.47 am	เบียร์
J. Brown	Dec 21, 2001 9.22 am	ไวน์, น้ำดื่ม, เหล้าผลไม้
J. Mitchell	Dec 21, 2001 3.19 pm	เบียร์, เหล้า, เหล้าผลไม้
C. Lisa	Dec 20, 2001 2.32 pm	เบียร์
C. Lisa	Dec 21, 2001 6.17 pm	ไวน์, เหล้าผลไม้
C. Lisa	Dec 22, 2001 5.03 pm	บรั่นดี
F. Zappa	Dec 20, 2001 11.02 am	บรั่นดี

เทคนิคของ Sequential Pattern จะนับจำนวนความถี่ของทรานแซกชั่นที่มาจากลำดับการซื้อของลูกค้าดังในตารางที่ 3.4 และแสดง Sequential Pattern ของเหตุการณ์ที่มีการเกิดมากกว่าค่า minimum support ดังแสดงในตารางที่ 3.5 ซึ่งจะได้ pattern คือ เมื่อลูกค้าซื้อเบียร์แล้ว เขาจะซื้อบรั่นดีในภายหลัง โดยเหตุการณ์นี้เกิดขึ้นกับลูกค้า 2 คนใน 5 คน

ตารางที่ 3.4 ลำดับการซื้อของลูกค้า

Customer	Customer Sequences
B. Adams	(เบียร์) (บรั่นดี)
J. Brown	(น้ำผลไม้, โค้ก) (เบียร์) (ไวน์,น้ำดื่ม,เหล้าผลไม้)
J. Mitchell	(เบียร์, เหล้า, เหล้าผลไม้)
C. Lisa	(เบียร์) (ไวน์, เหล้าผลไม้) (บรั่นดี)
F. Zappa	(บรั่นดี)

ตารางที่ 3.5 รายการที่มีค่า support มากกว่า 40 %

Sequential Pattern with supports > 40 %	Supporting Customers
(เบียร์)(บรั่นดี)	B. Adams, C. Lisa
(เบียร์)(ไวน์,เหล้าผลไม้)	J. Brown, C Lisa

3.2.2 ข้อดีและข้อเสียของของ Sequential Pattern Mining

ข้อดีและข้อเสียของของ Sequential Pattern Mining จะเหมือนกับ Association Rule Mining โดยมีจุดที่เพิ่มขึ้น ได้แก่

- จำเป็นต้องใช้ข้อมูลจำนวนมากเพื่อหาจำนวน ทรานแซกชั่นของลูกค้าแต่ละคน
- มีเพียงค่า support factor เท่านั้นที่ต้องกำหนด
- ต้องมีฟิลด์ที่เก็บรหัสของลูกค้าในฐานข้อมูล แต่บริษัทส่วนใหญ่ โดยเฉพาะร้านขายปลีก ไม่มีการเก็บรหัสลูกค้า
- เทคนิคนี้จะทำงานได้ดี ก็ต่อเมื่อมีการเรียงข้อมูลตามลำดับทรานแซกชั่นที่เกิดขึ้นของลูกค้าแต่ละราย

บทที่ 4

ดีเอชพี

(DHP)

หนึ่งในปัญหาที่สำคัญที่สุดของ data mining คือการไมนิ่งแบบ Association Rule ตัวอย่างเช่น การค้นหาความสัมพันธ์ทั้งหมดระหว่างไอเท็มจากฐานข้อมูลของทรานแซกชั่นการขายจะได้ว่า การมีไอเท็มหนึ่งในทรานแซกชั่นหนึ่ง จะสรุปได้ว่า จะมีอีกไอเท็มหนึ่งในทรานแซกชั่นเดียวกันด้วย ปัญหาของการไมนิ่งแบบ Association Rule คือ เราจำเป็นต้องระบุเซ็ตทั้งหมดของไอเท็ม (itemset) ที่อยู่ในทรานแซกชั่นที่มีจำนวนมากกว่าค่าต่ำสุดที่ต้องการ (minimum support) itemset เหล่านี้เรียกว่า large itemset เมื่อได้ large itemset ทั้งหมด ก็จะสามารถหา Association Rule ที่ต้องการได้ หรือกล่าวคือ อัลกอริธึมเหล่านี้ต้องสร้างเซ็ตที่เป็น candidate ของ large itemset ก่อน แล้วจึงหาเซ็ตย่อยที่บรรจุ large itemset จริงๆ กระบวนการนี้เป็นกระบวนการทำซ้ำ (Iteration) ดังที่ได้กล่าวไปในหัวข้อ 3.1.1 โดย large itemset ที่พบในหนึ่งรอบของการทำซ้ำจะถูกใช้ในการสร้างเซ็ต candidate สำหรับใช้ในการทำซ้ำรอบต่อไป ตัวอย่างเช่น ที่การทำซ้ำครั้งที่ k large itemset ทั้งหมดจะมี k ตัว ซึ่งเรียกว่า large k-itemset ถูกสร้างขึ้น ในการทำซ้ำครั้งต่อไปก็จะสร้างเซ็ต candidate ของ large (k+1)-itemset

การสร้างเซ็ต candidate ของ large itemset มีผลต่อสมรรถภาพ เพื่อที่จะให้มีประสิทธิภาพควรสร้าง candidate ด้วย large itemset ที่มีค่ามากเท่านั้น เพราะว่าสำหรับ candidate แต่ละตัว เราจำเป็นต้องนับจำนวนการเกิดของมันในทรานแซกชั่นทั้งหมด ยิ่งเซ็ต candidate มีขนาดใหญ่ ก็จะมี cost ในการค้นหา large itemset มากเท่านั้น ดังนั้นการสร้างเซ็ต candidate อันแรก โดยเฉพาะอย่างยิ่ง large 2-itemset จึงเป็นประเด็นสำคัญในการปรับปรุงสมรรถภาพของการไมนิ่งข้อมูล

ประเด็นที่เกี่ยวข้องกับสมรรถภาพอีกประเด็นหนึ่ง คือ จำนวนของข้อมูลที่ต้องถูกอ่านค่าเข้ามาระหว่างการหา large itemset การประมวลผลแบบ Apriori นั้นจะต้องอ่านค่าฐานข้อมูลของทรานแซกชั่นทั้งหมดสำหรับการทำซ้ำแต่ละรอบ จะสังเกตได้ว่า เมื่อค่า k เพิ่มขึ้น ไม่เพียงแต่จำนวน large k-itemset จะน้อยลง แต่จำนวนทรานแซกชั่น ที่บรรจุ large k-itemset ก็น้อยลงด้วย ดังนั้นการลดจำนวนทรานแซกชั่นที่ต้องอ่านค่า และการลดจำนวนไอเท็มในแต่ละทรานแซกชั่นจะเป็นการปรับปรุงประสิทธิภาพในการทำดาต้าไมนิ่งในขั้นตอนต่อไป

4.1 อัลกอริธึมดีเอชพี

อัลกอริธึมดีเอชพีมี 2 คุณสมบัติใหญ่ๆ คือ การสร้าง large itemset อย่างมีประสิทธิภาพ และการลดขนาดทรานแซกชั่นอย่างมีประสิทธิภาพ ในแต่ละรอบ เราต้องใช้เซ็ตของ large itemset L_i ในการสร้างเซ็ตของ candidate large itemset C_{i+1} โดยการ join L_i กับ L_i ($L_i * L_i$) แล้วอ่านฐานข้อมูลและนับค่าสนับสนุน (support) ของแต่ละ itemset ใน C_{i+1} เพื่อใช้ในการหา L_{i+1} โดยทั่วๆไปยิ่งมี itemset ใน C_i มาก cost ในการประมวลผลหา L_i ก็จะยิ่งสูง ในฐานข้อมูลขนาดใหญ่ การดึงข้อมูลจากฐานข้อมูลในรอบแรกเป็นส่วนที่มี cost สูงที่สุด ในความจริงแล้ว cost ในการประมวลผลของการทำซ้ำสองรอบแรก (เช่น การหา L_1 และ L_2) คิดเป็น cost ในการประมวลผลเกือบจะทั้งหมด

สิ่งนี้สามารถอธิบายด้วยเหตุผลที่ว่า โดยปกติแล้วเราจะมี L_1 ที่ใหญ่มาก ซึ่งทำให้ได้ C_2 ที่มีจำนวน itemset มาก กล่าวคือ ใน apriori ขนาดของ $C_2 = (\binom{|L_1|}{2})$ ขั้นตอนในการหา L_2 จาก C_2 โดยการอ่านค่าฐานข้อมูลทั้งหมด และทดสอบแต่ละทรานแซกชั่นกับ hash tree ที่สร้างโดย C_2 เป็นขั้นตอนที่มี cost สูงมาก แต่ DHPสามารถสร้าง C_2 ที่มีขนาดเล็กกว่านั้น และยังสามารถสร้าง D_3 ที่เล็กกว่าเพื่อใช้ในการหา C_3 ด้วย ขนาดของ L_i ลดลงอย่างรวดเร็วขณะที่ i เพิ่มขึ้น L_i ที่เล็กนำไปสู่ C_{i+1} ที่เล็ก ดังนั้น cost ในการทำงานจึงต่ำ ดังที่ได้กล่าวมาข้างต้น อัลกอริธึมดีเอชพีถูกออกแบบมาเพื่อลดจำนวน itemset ใน C_i ในการทำซ้ำรอบแรกๆ

อัลกอริธึมดีเอชพีในรูปที่ 2 ใช้เทคนิคของการ hash ตัด itemset ที่ไม่จำเป็นออกไป เพื่อการสร้าง candidate itemset ตัวถัดไป เมื่อจำนวนของ candidate k-itemset ถูกนับโดยการอ่านฐานข้อมูล ดีเอชพีจะสะสมข้อมูลเกี่ยวกับ candidate (k+1)-itemset ล่วงหน้า เพื่อหา (k+1)-itemset ทั้งหมดที่เป็นไปได้ของแต่ละทรานแซกชั่น และทำการ prune เสร็จแล้วก็ hash ลงสู่ตาราง hash แต่ละ bucket ในตาราง hash ประกอบด้วยจำนวนของ itemset ที่ถูก hash มาอยู่ใน bucket นั้น และมีค่า bit vector ซึ่งจะถูกเซ็ตเป็นหนึ่ง ถ้าจำนวนของค่าที่อยู่ใน bucket นั้น มีค่ามากกว่าหรือเท่ากับ s ดังนั้น bit vector สามารถช่วยลดจำนวน itemset ใน C_i ลงอย่างมาก

รูปแบบอัลกอริธึมของดีเอชพีในรูปที่ 4.1 เพื่อให้ง่ายต่อการนำเสนอ ได้แบ่งออกเป็น 3 ส่วน ส่วนที่ 1 จะได้เซ็ตของ large itemset และสร้างตาราง hash (เช่น H_2) สำหรับ 2-itemset ส่วนที่ 2 สร้างเซ็ตของ candidate itemset C_k จากตาราง hash (เช่น H_k) ซึ่งได้จากการสร้างจากรอบที่แล้วเพื่อหาเซ็ตของ large k-itemset L_k และสร้างตาราง hash เพื่อหา candidate large (k+1)-itemset (ลักษณะสำคัญของดีเอชพี คือการสร้างตาราง hash เพื่อใช้ในรอบถัดไป) ส่วนที่ 3 มีพื้นฐานเหมือนกับส่วนที่ 2 ยกเว้นว่ามันไม่ได้ใช้ตาราง hash ดีเอชพีมีประสิทธิภาพอย่างมากในการหา large itemset ในรอบแรกๆ จึงเป็นการปรับปรุงปัญหาคอขวด และขนาดของ C_k ลดลงอย่างมากในรอบหลังๆ ซึ่งเป็นเหตุผลที่ใช้ส่วนที่ 2 ในการทำซ้ำรอบแรกๆ และใช้ส่วนที่ 3 ในการทำซ้ำรอบ

หลังๆเมื่อจำนวน hash bucket ที่มีจำนวนของค่าที่อยู่ในนั้นมากกว่าหรือเท่ากับ s (เช่น $|\{x \mid H_k[x] \geq s\}|$ ในส่วนที่ 2 ของรูปที่ 4.1 มีค่าน้อยกว่าค่า threshold (LARGE) ที่กำหนดไว้ และในส่วนที่ 3 Procedure apriori_gen มีหน้าที่สร้าง C_{k+1} จาก L_k

```
/* Part 1 */
s = a minimum support;
set all the buckets of H2 to zero; /*hash table*/
forall transaction t ∈ D do begin
    insert and count 1-items occurrences in a hash tree;
    forall 2-subsets x of t do
        H2[h2(x)]++;
end

/* Part 2 */
k = 2;
Dk = D; /* database for large k-itemsets */
while (|{x|Hk[x] ≥ s}| ≥ LARGE) {
    /* make a hash table */
    gen_candidate(Lk-1, Hk, Ck);
    set all the buckets of Hk+1 to zero;
    Dk+1 = ϕ;
    forall transactions t ∈ Dk do begin
        count_support(t, Ck, k, t');   /* t' ⊆ t */
        if (|t'| > k) then do begin
            make_hasht(t', Hk, k, Hk+1, t'');
            if (|t''| > k) then Dk+1 = Dk+1 ∪{t''};
                end
        end
    Lk = {c ∈ Ck|c.count ≥ s};
    k++;
}

/* Part 3 */
gen_candidate(Lk-1, Hk, Ck);
while (|Ck|>0){
    Dk+1 = ϕ;
    forall transactions t ∈ Dk do begin
        count_support(t, Ck, k, t');   /* t' ⊆ t */
        if (|t'| > k) then Dk+1 = Dk+1 ∪{t'};
    end
    Lk = {c ∈ Ck|c.count ≥ s};
    if (Dk+1 = 0) then break;
    Ck+1 = apriori_gen(Lk);
    k++;
}
```

รูปที่ 4.1 โปรแกรมหลักของอัลกอริธึมดีเอชพี

หลังจากส่วนที่ 1, ส่วนที่ 2 ซึ่งประกอบด้วย 2 เฟส เฟสแรก สร้างเซ็ตของ candidate k-itemset C_k จากตาราง hash H_k ซึ่งอธิบายโดย Procedure gen_candidate ในรูปที่ 4.2 เช่นเดียวกับ Apriori ดีเอชพีสร้าง k-itemset ด้วย L_{k-1} อย่างไรก็ตามดีเอชพีมีการใช้ bit vector ที่สร้างในรอบที่แล้ว และแทนที่ C_k จะประกอบด้วย k-itemset ทั้งหมดจาก L_{k-1}*L_{k-1} ดีเอชพีจะเพิ่ม k-itemset เข้าไปใน C_k ก็ต่อเมื่อ k-itemset นั้นผ่านการกรองจากการ hash คือมีค่ามากกว่าหรือเท่ากับ s k-itemset ทุกตัวที่ผ่านการกรองจากการ hash ถูกรวมเข้าไปใน C_k และจัดเก็บใน hash tree hash tree ที่สร้างจาก C_k จะถูกตรวจสอบโดยแต่ละ ทรานแซกชั่นภายหลัง (เช่น ในส่วนที่ 2) เมื่อมีการอ่านฐานข้อมูล และนับค่าสนับสนุนต่ำสุดของแต่ละ candidate เฟสที่ 2 ของส่วนที่ 2 เป็นการนับค่าสนับสนุนของ candidate itemset และลดขนาดของแต่ละทรานแซกชั่น ดังที่ได้อธิบายไว้ใน Procedure count_support ในรูปที่ 4.2 ขณะที่สแกนทรานแซกชั่นในฐานข้อมูลทีละทรานแซกชั่น เซ็ตย่อย k ของแต่ละทรานแซกชั่นจะถูกใช้ในการนับค่าสนับสนุนของ itemset ใน C_k

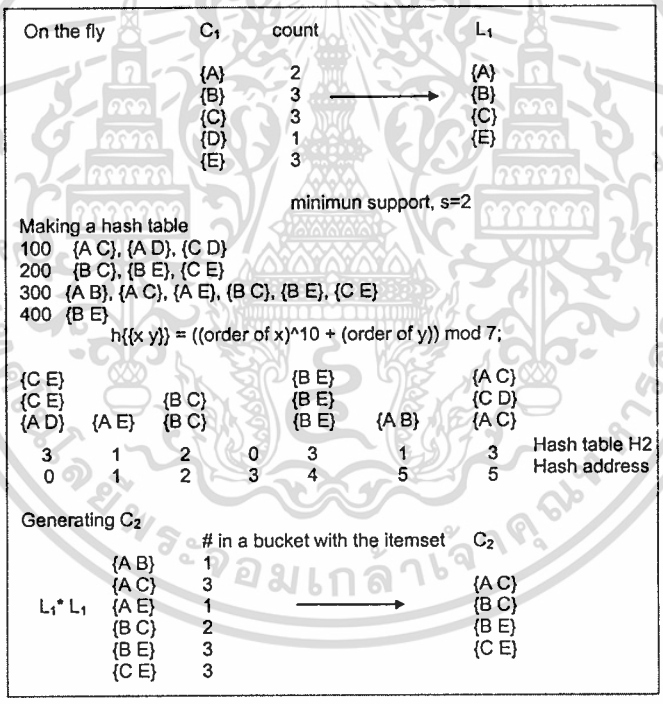
```
Procedure gen_candidate(L_{k-1}, H_k, C_k)
    C_k = φ;
    forall c = c_p[1]. .... c_p[k-2] . c_p[k-1] . c_p[k-1], c_p, c_q ∈ L_{k-1}, |c_p ∩ c_q| = k-2 do
        if (H_k[h_k(c)] ≥ s) then
            C_k = C_k ∪ {c};  /* insert c into a hash tree */
end Procedure

Procedure count_support(t, C_k, k, t')
    forall c such that c∈C_k and c(=t_{i1}...t_{ik}) ∈ t do
    begin
        c.count++;
        for (j=1; j ≤ k: j++) a[i_j]++;
    end
    for (i=0; j=0; i<|t|; i++)
        if(a[i]≥k) then do begin t'_j=t_i; j++; end
end Procedure

Procedure make_hasht(t', H_k, k, H_{k+1}, t'')
    forall (k+1)-subsets x(=t'_{i1} ... t'_{ik})of t' do
        if(for all k-subsets y of x, H_k[h_k(y)] ≥ s) then do
        begin
            H_{k+1}[h_{k+1}(x)]++;
        for (j=1; j ≤ k+1; j++) a[i_j]++;
        end
      for (i=0;j=0; i < |t'|; i++)
        if(a[I]>0) then do begin t''_j=t'_i; j++; end
end Procedure
```

รูปที่ 4.2 โปรแกรมย่อยของอัลกอริธึมดีเอชพี

ตัวอย่างในการสร้าง candidate itemset โดยดีเอชพีแสดงในรูปที่ 4.3 เซ็ต candidate ของ large 1-itemset C_1 = {{A}, {B}, {C}, {D}, {E}} ทรานแซกชั่นทั้งหมดของฐานข้อมูลถูกอ่านค่าเพื่อนับค่าสนับสนุนของ 1-item เหล่านี้ ในขั้นตอนนี้มีการสร้าง hash tree แบบ on the fly เพื่อการนับที่มีประสิทธิภาพ ดีเอชพีตรวจสอบว่าแต่ละไอเท็มมีอยู่ใน hash tree อยู่แล้วหรือไม่ ถ้าใช่ ก็จะเพิ่มตัวนับค่าของไอเท็มนี้อีกครั้ง ไม่เช่นนั้นจะใส่ไอเท็มพร้อมกับตัวนับที่มีค่าเท่ากับหนึ่งลงไปใน hash tree สำหรับแต่ละทรานแซกชั่น หลังจากนับการเกิด 1-subset ทั้งหมดเสร็จ 2-subset ทั้งหมดของทรานแซกชั่นนี้ก็จะถูกสร้าง และ hash เข้าไปในตาราง hash H_2 และเมื่อ 2-subset ถูก hash เข้าไปใน bucket i ค่าของ bucket i จะเพิ่มขึ้นหนึ่ง รูปที่ 4.3 แสดงตาราง hash H_2 สำหรับฐานข้อมูลที่กำหนด และหลังจากฐานข้อมูลถูกสแกน แต่ละ bucket ของตาราง hash จะมีจำนวนของ 2-itemset ที่ถูก hash เข้ามาใน hash จากรูปที่ 4 ที่ค่าสนับสนุนต่ำสุด = 2 ทำให้ได้ bit vector <1, 0, 1, 0, 1, 0, 1> การใช้ bit vector เพื่อกรอง 2-itemset ออกจาก L_1* L_1 ทำให้ได้ C_2 = {{AC}, {BC}, {BE}, {CE}} แทน C_2 = {{AB}, {AC}, {AE}, {BC}, {BE}, {CE}} ซึ่งได้จาก Apriori



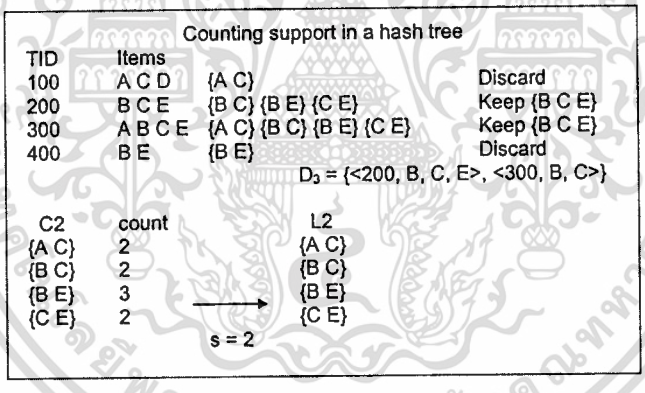
รูปที่ 4.3 ตัวอย่างของตาราง hash และการสร้าง C_2

4.2 การลดขนาดฐานข้อมูลทรานแซกชั่น

ดีเอชพีลดขนาดฐานข้อมูลอย่างมาก ไม่เพียงแต่ด้วยการลดขนาดของทรานแซกชั่นให้เหมาะสม (trimming) แต่ยังมีการ prune จำนวนทรานแซกชั่นในฐานข้อมูลด้วย

ตามกฎของ Association Rule เซ็ตย่อยใดๆของ large itemset ต้องเป็น large itemset ด้วย นั่นคือ $\{B, C, D\} \in L_3$ สรุปได้ว่า $\{B, C\} \in L_2$, $\{B, D\} \in L_2$ และ $\{C, D\} \in L_2$ สิ่งนี้ทำให้ทรานแซกชั่นถูกใช้ในการหาเซ็ตของ large (k+1)-itemset ก็ต่อเมื่อมันประกอบด้วย large k-itemset จำนวน (k+1) ตัวในรอบที่แล้ว ด้วยความคิดนี้ เมื่อนับ k-subset ของแต่ละทรานแซกชั่นที่เป็น candidate k-itemset จะทำให้สามารถทราบว่าทรานแซกชั่นนี้ตรงตามเงื่อนไขของการมี large (k+1)-itemset หรือไม่ ทำให้สามารถลดขนาดทรานแซกชั่นและจำนวนทรานแซกชั่นลง โดยการกำจัดไอเท็มที่พบว่าไม่มีประโยชน์สำหรับการสร้าง large itemset ต่อไป

ต่อไปจะกล่าวถึงการลดขนาดทรานแซกชั่นด้วยดีเอชพี ถ้าทรานแซกชั่นมี large (k+1)-itemset ไอเท็มใดๆที่อยู่ใน (k+1)-itemset จะปรากฏใน candidate k-itemset ใน C_k อย่างน้อยที่สุด k ตัว ผลที่ได้ไอเท็มในทรานแซกชั่น t สามารถถูกลดขนาด ถ้า candidate k-itemset ใน t มีไม่ถึง k ตัว แนวคิดนี้ถูกนำมาใช้ใน Procedure count_support เพื่อลดขนาดของทรานแซกชั่น



รูปที่ 4.4 ตัวอย่างของ L_2 และ D_3

รูปที่ 4.4 แสดงตัวอย่างการลดขนาดและการลดจำนวนของทรานแซกชั่น ค่าสนับสนุนของ k-itemset เพิ่มขึ้น เมื่อ k-itemset นั้นเป็นเซ็ตย่อยของทรานแซกชั่น t และเป็นสมาชิกของ C_k ด้วย ดังที่ได้อธิบายใน Procedure count_support a[i] ถูกใช้ในการเก็บค่าความถี่ในการเกิดของ i-th item ของทรานแซกชั่น t เมื่อ k-subset ที่มี i-th item เป็นสมาชิกของ C_k จะต้องเพิ่มค่า a[i] ขึ้นหนึ่งตามดัชนี (index) ของแต่ละไอเท็มใน k-subset (เช่น ในทรานแซกชั่น 100 a[0] เก็บค่าความถี่ของ A, a[1] เก็บค่าของ C และ a[2] เก็บค่าของ D) จากนั้นใน Proccdurc makc_hasht ก่อนการ hash ของ

(k+1)-subset ของทรานแซกชั่น t' จะตรวจสอบ k-subset ของ t' ทั้งหมด โดยการตรวจสอบค่าของ bucket ที่เกี่ยวข้องของตาราง hash จาก t' ไอเท็ม t' จะถูกทิ้ง ถ้ามันไม่เป็นไปตามที่กล่าวมา

ตัวอย่างในรูปที่ 4.4 ทรานแซกชั่น 100 มี candidate itemset เพียงตัวเดียว คือ AC ความถี่ในการเกิดของไอเท็มทั้ง คือ a[0] = 1, a[1] = 1 และ a[2] = 0 เพราะค่าของ a[i] ทั้งหมดน้อยกว่า 2 ทรานแซกชั่นนี้จึงถือว่าไม่มีประโยชน์ในการสร้าง large 3-itemset ดังนั้นจึงทิ้งไป ส่วนทรานแซกชั่น 300 ในรูปที่ 4.4 มี candidate 2-item สี่ตัว และความถี่ในการเกิดของไอเท็ม คือ a[0] = 1 (เป็นค่าของ A), a[1] = 2 (เป็นค่าของ B), a[2] = 3 (เป็นค่าของ C) และ a[3] = 2 (เป็นค่าของ E) ดังนั้นจึงเก็บไอเท็ม B, C, E และทิ้งไอเท็ม A

อีกตัวอย่างหนึ่งคือ ถ้าทรานแซกชั่น t = ABCDEF และมี 2-subset ห้าตัว (AC, AE, AF, CD, EF) อยู่ใน hash tree ที่สร้างโดย C_2 ค่าของอะเรย์ a[i] เท่ากับ a[0] = 3, a[2] = 2, a[3] = 1, a[4] = 2 และ a[5] = 2 สำหรับ large 3-itemset มี 4 ไอเท็ม คือ A, C, E และ F ซึ่งมีค่ามากกว่าหรือเท่ากับ 2 (ค่าสนับสนุนต่ำสุด) ดังนั้นจึงเก็บไอเท็มเฉพาะ A, C, E, F เช่นเดียวกับทรานแซกชั่น t' ใน Procedure count_support และทิ้งไอเท็ม B และ D เพราะมันไม่มีประโยชน์ในเฟสต่อๆไป จะเห็นได้ว่า ไม่ใช่ไอเท็มทั้งหมดใน t' มีส่วนร่วมในการสร้าง large itemset ในภายหลัง C ไม่ได้อยู่ใน large 3-itemset ใดๆ เพราะมีเพียง AC และ CD เท่านั้นที่เป็น large 2-itemset ส่วน AD ไม่เป็น จาก Procedure make_hasht ไอเท็ม เช่น C จะถูกเอาออกจาก t'' ในฐานข้อมูลที่ถูกลดขนาดลง (D_3) ดังนั้นระหว่างการอ่านค่าทรานแซกชั่น ทรานแซกชั่นจำนวนมากจึงถูก trim หรือทิ้งไป และมีเพียงทรานแซกชั่นซึ่งประกอบด้วยไอเท็มที่จำเป็นต่อการสร้าง large itemset ในภายหลังเท่านั้นที่ถูกเก็บใน D_{k+1} ดังนั้นจึงเป็นการลดขนาดฐานข้อมูลของทรานแซกชั่นลงไปอย่างมาก ความจริงที่ว่า D_k ลดลงอย่างมากในรอบที่ k เป็นเหตุผลที่ดีเอชพีใช้เวลาในการประมวลผลสั้นกว่า Apriori แม้กระทั่งในการทำซ้ำเมื่อทั้งสองอัลกอริธึมใช้ Procedure เดียวกันในการสร้าง large itemset และในรูปที่ 4.4 แสดงตัวอย่างของ L_2 และ D_3

4.3 สรุป

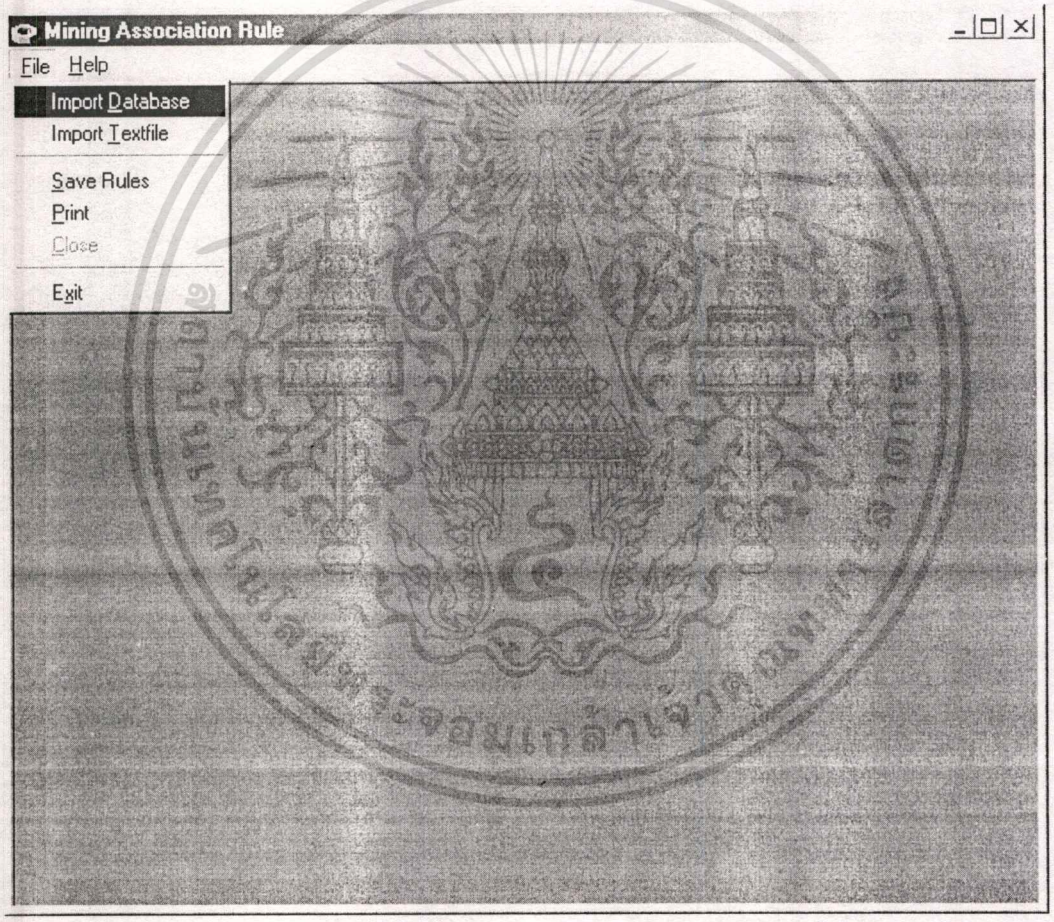
ดีเอชพี เป็นอัลกอริธึมที่ใช้การ hash เพื่อเพิ่มประสิทธิภาพในการสร้างเซ็ต candidate โดยเฉพาะอย่างยิ่งการสร้างเซ็ตของ candidate แรกๆ เช่น การสร้างเซ็ต candidate เพื่อหา large 2-itemset เนื่องจากว่าเซ็ต candidate ที่ได้จากดีเอชพีมีขนาดเล็กกว่าอัลกอริธึมพื้นฐานของ Association Rule เช่น Apriori มาก จึงช่วยแก้ปัญหาคอขวดในการประมวลผลได้ และยังมีการลดจำนวนและลดขนาดทรานแซกชั่นในฐานข้อมูล เพื่อให้สามารถประมวลผลได้เร็วขึ้น ซึ่งเป็นการลด cost ในการประมวลผลลงอย่างมาก

บทที่ 5

การพัฒนาระบบ

5.1 การติดต่อกับข้อมูลที่นำมาวิเคราะห์

เมื่อเข้าสู่โปรแกรมจะปรากฏหน้าจอเมนูหลัก ดังรูปที่ 5.1

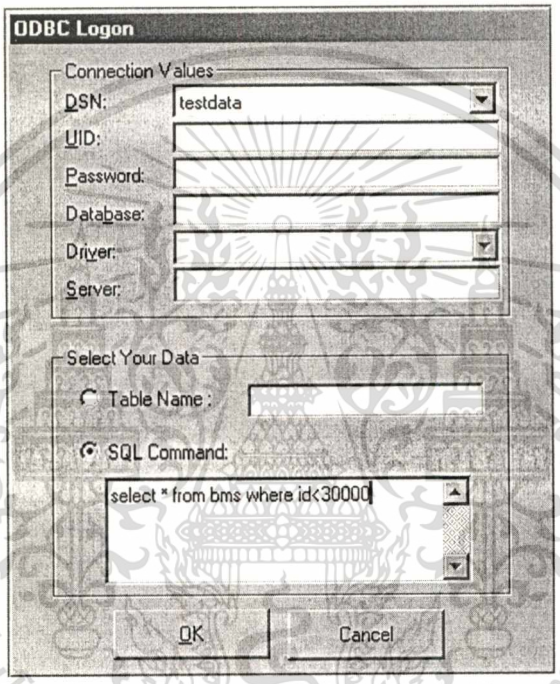


รูปที่ 5.1 หน้าจอหลักของระบบ

ระบบสามารถรับข้อมูลได้ 2 รูปแบบ คือ Relational Database และ Text File โดยเมนูแรก Import Database เป็นการติดต่อกับฐานข้อมูลผ่าน ODBC และเมนูที่ 2 Import Text File เป็นการดึงข้อมูลจาก Text File เพื่อนำข้อมูลที่ได้มาใช้ในการวิเคราะห์ความสัมพันธ์

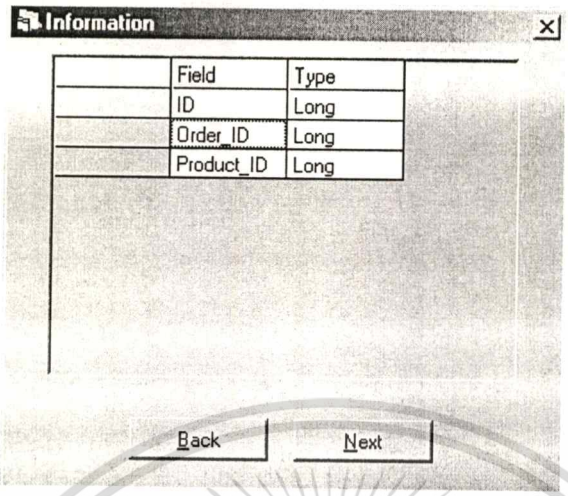
5.1.1 การดึงข้อมูลจากฐานข้อมูล

ทำโดยคลิกที่ Import Database จะปรากฏหน้าจอดังรูปที่ 5.2

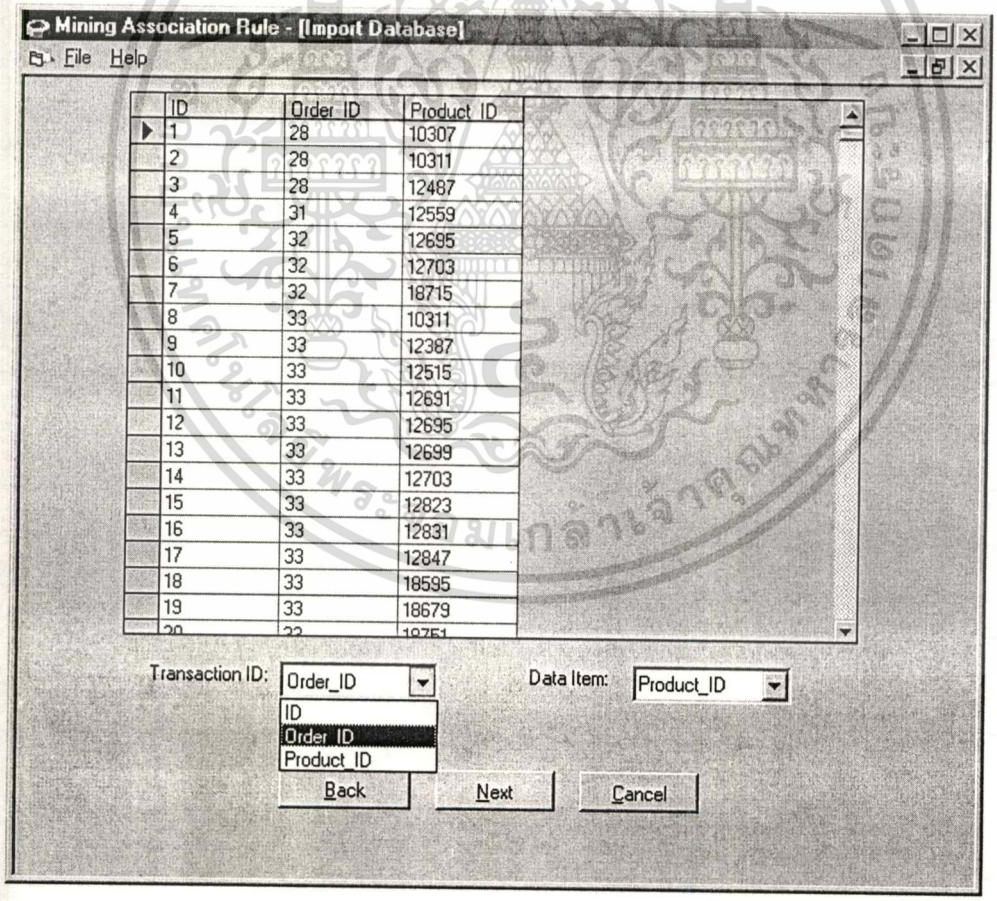


รูปที่ 5.2 หน้าจอแสดงการดึงข้อมูลจากฐานข้อมูล

ผู้ใช้สามารถเลือกข้อมูลที่จะนำมาวิเคราะห์ด้วยการระบุค่าต่างๆที่ใช้ในการเชื่อมต่อผ่านฐานข้อมูลผ่าน ODBC และระบุชื่อตาราง หรือด้วยการพิมพ์คำสั่ง SQL ลงในช่อง SQL Command และเมื่อผู้ใช้คลิกปุ่ม OK จะปรากฏหน้าจอแสดงรายละเอียดโครงสร้างของข้อมูลดังรูปที่ 5.3 จากนั้นกด Next จะปรากฏหน้าจอให้ผู้ใช้เลือกฟิลด์ที่จะนำไปใช้ในการวิเคราะห์ความสัมพันธ์ของข้อมูล ดังรูปที่ 5.4



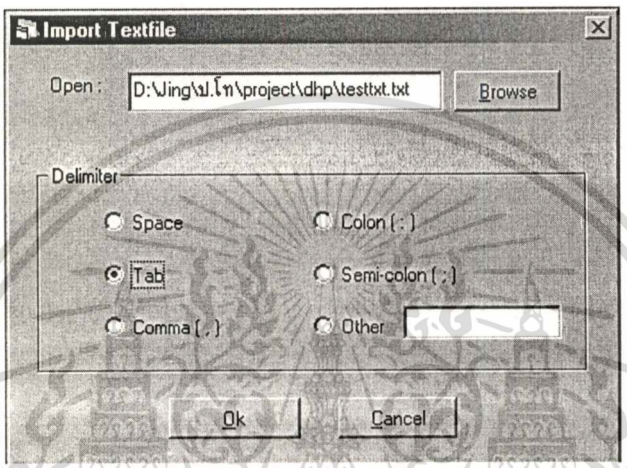
รูปที่ 5.3 หน้าจอแสดงโครงสร้างฐานข้อมูล



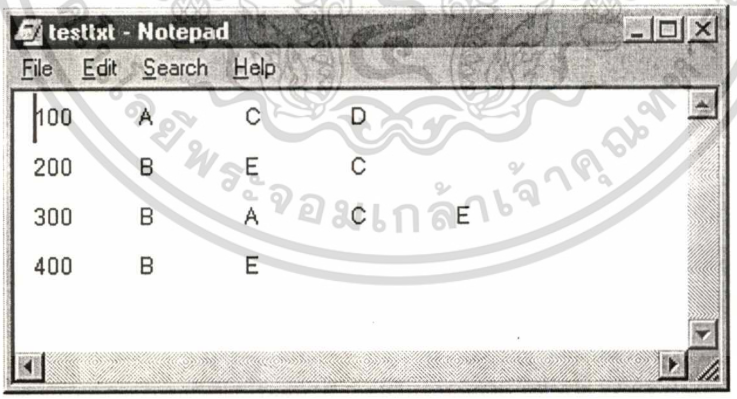
รูปที่ 5.4 หน้าจอแสดงการเลือกฟิลด์ที่จะนำมาวิเคราะห์

5.1.2 การดึงข้อมูลจาก Text File

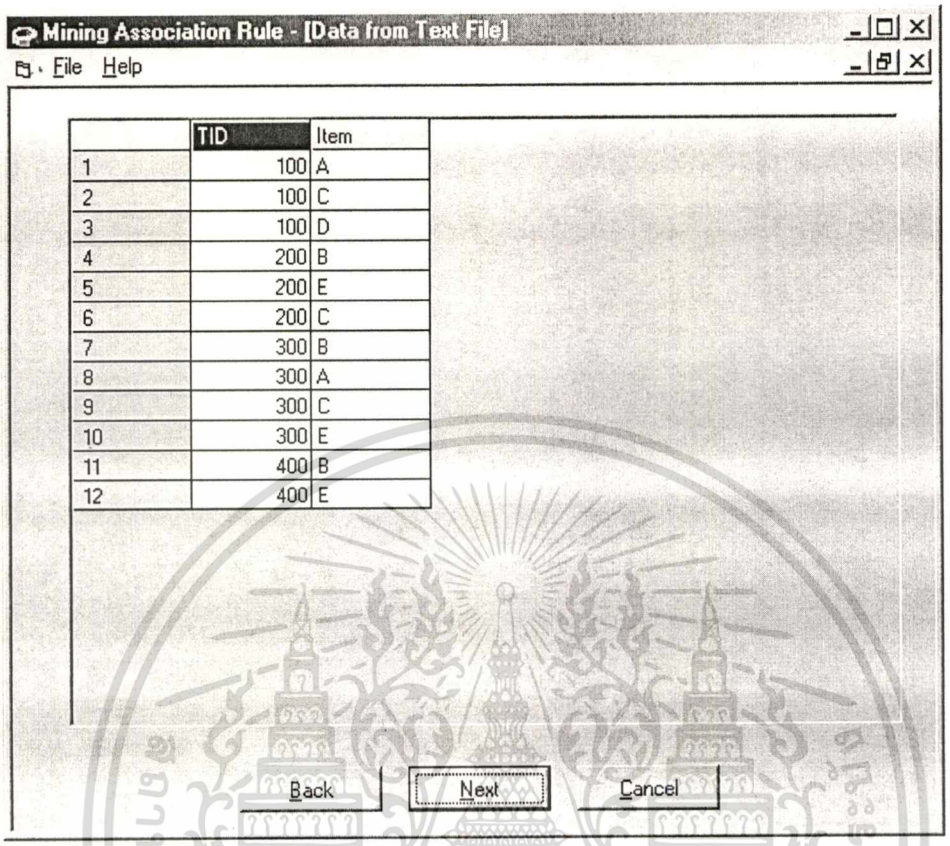
ทำโดยคลิกที่ Import Text File ในเมนู จะปรากฏหน้าจอดังรูปที่ 5.5 ผู้ใช้สามารถเลือกไฟล์ข้อมูลโดยกดปุ่ม Browse ตัวอย่างของ text file แสดงในรูปที่ 5.6 และเลือกสัญลักษณ์ที่เป็นตัวคั่นระหว่างข้อมูล โดยตัวอย่างในรูปที่ 5.6 นั้นมีตัวคั่นเป็น tab จากนั้นกด OK จะปรากฏหน้าจอแสดงข้อมูลจาก Text File ดังรูปที่ 5.7



รูปที่ 5.5 หน้าจอแสดงการดึงข้อมูลจาก Text File



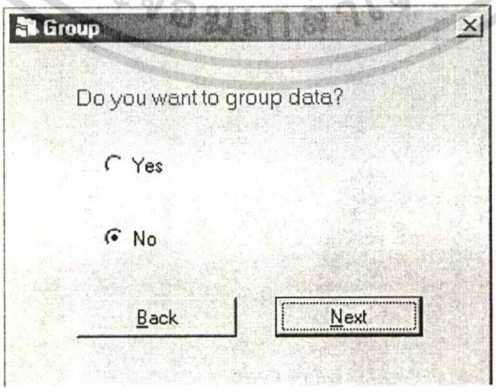
รูปที่ 5.6 ตัวอย่างข้อมูลที่เป็น Text File



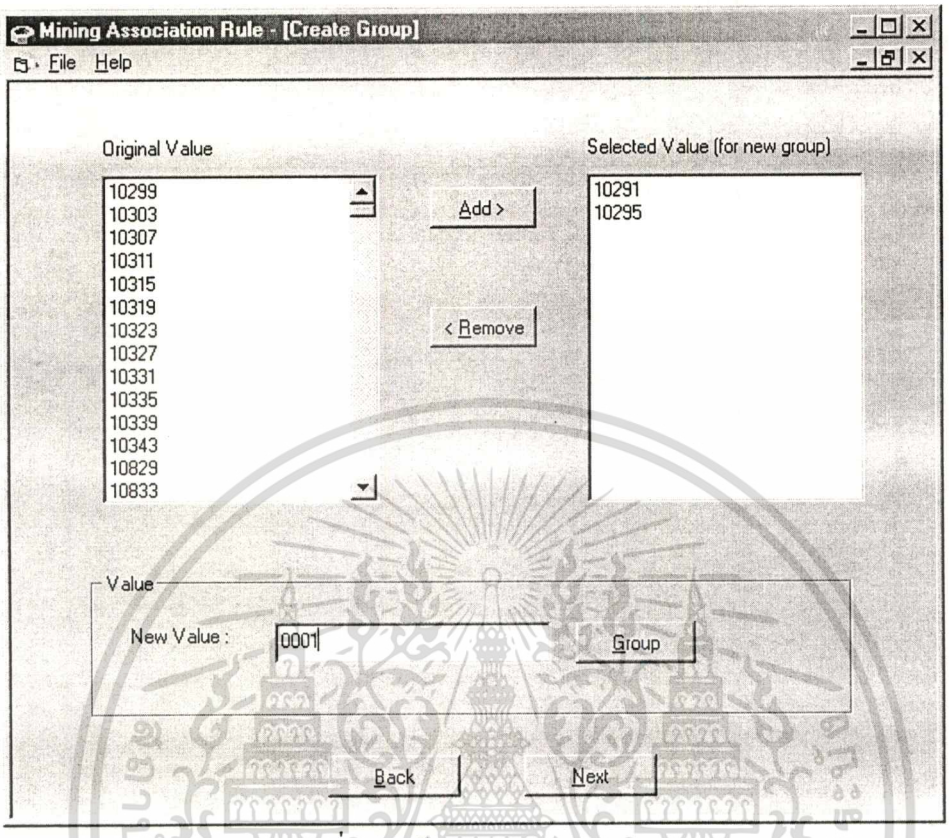
รูปที่ 5.7 หน้าจอแสดงข้อมูลจาก Text File

5.2 การจัดกลุ่มข้อมูล

เมื่อผู้ใช้เลือกข้อมูลที่จะนำมาวิเคราะห์เสร็จแล้ว จะมีหน้าจอให้เลือกว่าต้องการจัดกลุ่มข้อมูลหรือไม่ ดังรูปที่ 5.8 ถ้าเลือก Next จะปรากฏหน้าจอดังรูปที่ 5.9



รูปที่ 5.8 หน้าจอแสดงการเลือกการจัดกลุ่ม



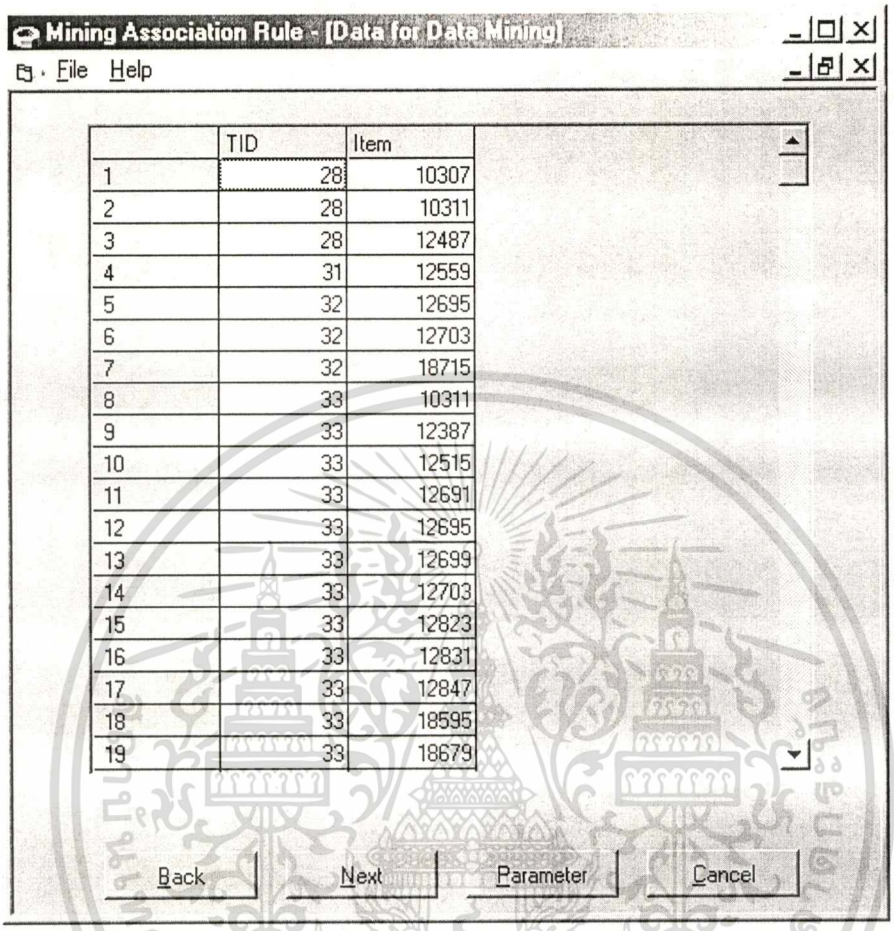
รูปที่ 5.9 หน้าจอแสดงการจัดกลุ่ม

จากรูปที่ 5.9 ให้ผู้ใช้เลือกค่าที่ต้องการจัดให้อยู่กลุ่มเดียวกันจาก Original Value แล้วกดปุ่ม Add จากนั้นตั้งชื่อของกลุ่ม และกดปุ่ม Group เพื่อจัดกลุ่มข้อมูล

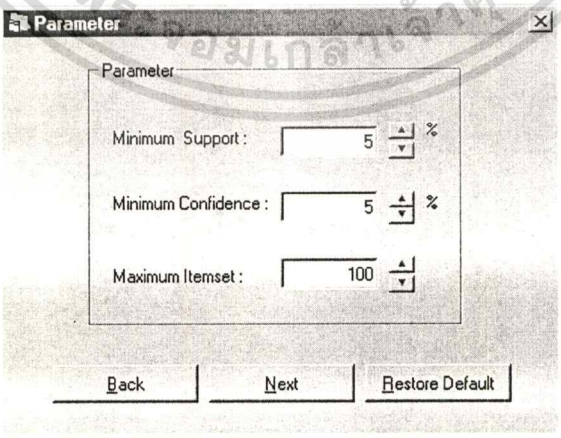
5.3 ข้อมูลที่จะนำไปวิเคราะห์และการกำหนดค่าพารามิเตอร์

เมื่อผู้ใช้เลือกข้อมูลและจัดกลุ่มข้อมูลเสร็จแล้ว ระบบจะแสดงหน้าจอข้อมูลที่จะใช้ในการวิเคราะห์ระบบดังรูปที่ 5.10 และผู้ใช้สามารถกำหนดพารามิเตอร์ที่จะใช้ในการวิเคราะห์โดยคลิกที่ปุ่ม Parameter เพื่อกำหนดค่า Minimum Support, Minimum Confidence และ Maximum Itemset ซึ่งจะปรากฏหน้าจอดังรูปที่ 11 ถ้าผู้ใช้ไม่ได้กำหนด ค่าสองค่าแรกจะมี default อยู่ที่ 5% และค่า Maximum Itemset มี default เป็น 100 ชุดข้อมูล

เมื่อกำหนดค่าพารามิเตอร์เสร็จแล้ว กดปุ่ม Next



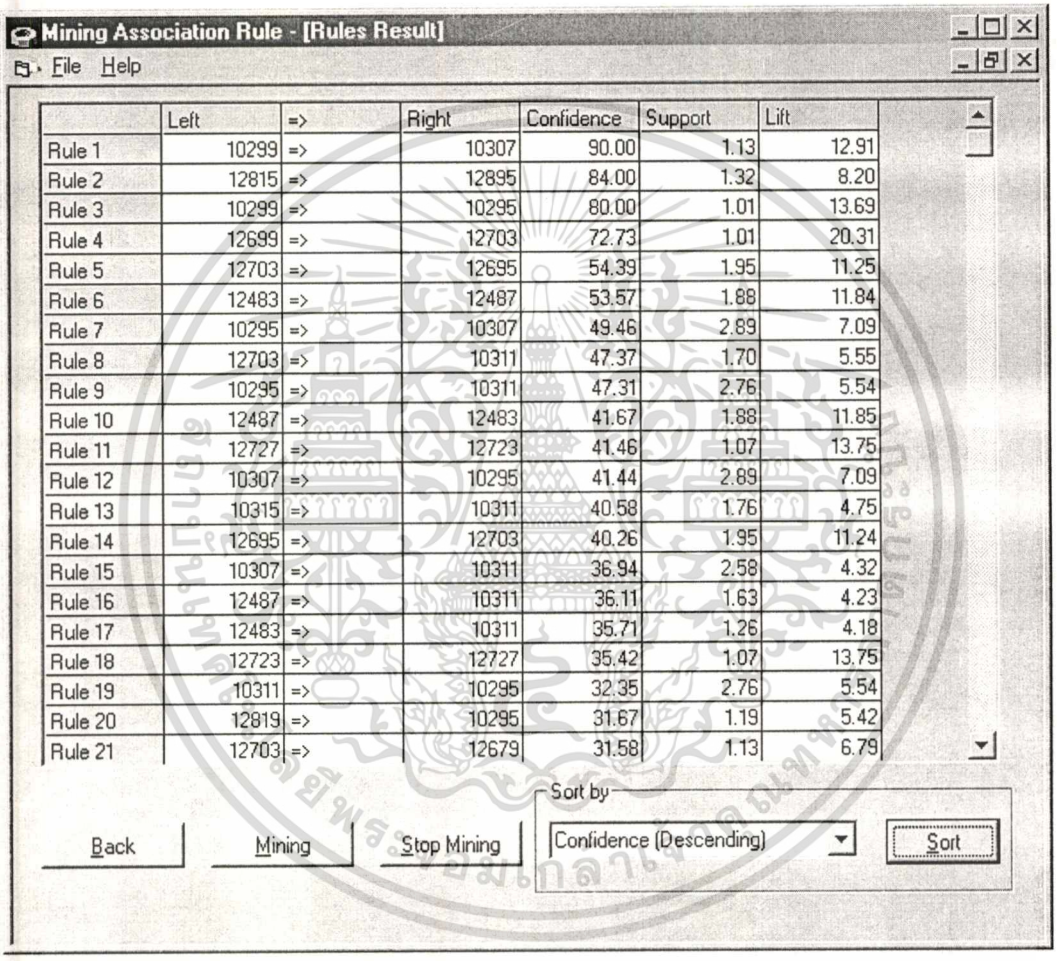
รูปที่ 5.10 หน้าจอแสดงข้อมูลที่จะนำไปวิเคราะห์



รูปที่ 5.11 หน้าจอแสดงการกำหนดพารามิเตอร์

5.4 การ Mining และการแสดงผลลัพธ์

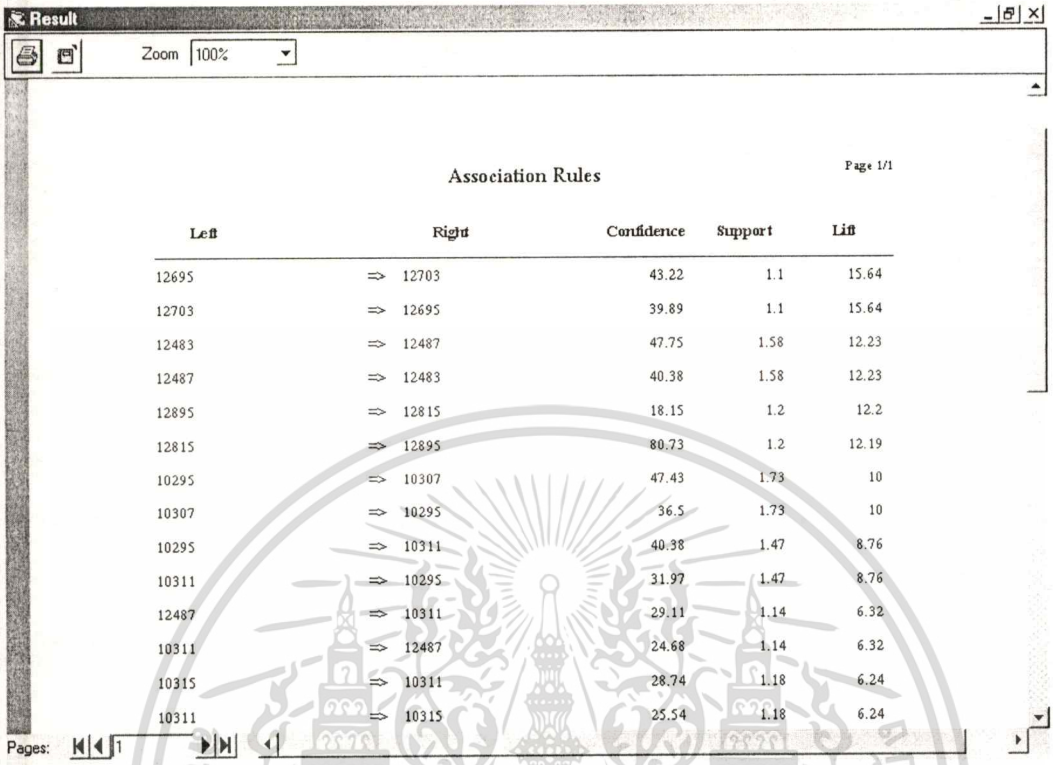
ให้ผู้ใช้กดปุ่ม Mining เพื่อให้โปรแกรมค้นหาความสัมพันธ์ โดยผู้ใช้สามารถหยุดการทำงานโดยการกดปุ่ม Stop Mining หลังจากโปรแกรมวิเคราะห์ข้อมูลเสร็จ จะแสดงหน้าจอผลลัพธ์จากการสร้างกฎ ดังรูปที่ 5.12 ซึ่งแสดงค่า Confidence, ค่า Support และค่า Lift ของแต่ละความสัมพันธ์ โดยผู้ใช้สามารถเลือกวิธีในการเรียงลำดับผลลัพธ์ใน Sort by และกดปุ่ม Sort



	Left	=>	Right	Confidence	Support	Lift
Rule 1	10299	=>	10307	90.00	1.13	12.91
Rule 2	12815	=>	12895	84.00	1.32	8.20
Rule 3	10299	=>	10295	80.00	1.01	13.69
Rule 4	12699	=>	12703	72.73	1.01	20.31
Rule 5	12703	=>	12695	54.39	1.95	11.25
Rule 6	12483	=>	12487	53.57	1.88	11.84
Rule 7	10295	=>	10307	49.46	2.89	7.09
Rule 8	12703	=>	10311	47.37	1.70	5.55
Rule 9	10295	=>	10311	47.31	2.76	5.54
Rule 10	12487	=>	12483	41.67	1.88	11.85
Rule 11	12727	=>	12723	41.46	1.07	13.75
Rule 12	10307	=>	10295	41.44	2.89	7.09
Rule 13	10315	=>	10311	40.58	1.76	4.75
Rule 14	12695	=>	12703	40.26	1.95	11.24
Rule 15	10307	=>	10311	36.94	2.58	4.32
Rule 16	12487	=>	10311	36.11	1.63	4.23
Rule 17	12483	=>	10311	35.71	1.26	4.18
Rule 18	12723	=>	12727	35.42	1.07	13.75
Rule 19	10311	=>	10295	32.35	2.76	5.54
Rule 20	12819	=>	10295	31.67	1.19	5.42
Rule 21	12703	=>	12679	31.58	1.13	6.79

รูปที่ 5.12 หน้าจอแสดงผลลัพธ์

ผู้ใช้สามารถเซฟผลลัพธ์ที่ได้ออกมาเป็น Text File โดยกดปุ่ม File->Save Rules และสามารถพิมพ์ผลลัพธ์ออกมาเป็นรายงานโดยกด File->Print ซึ่งจะได้รายงานดังแสดงในรูปที่ 5.13



รูปที่ 5.13 หน้าจอรายงานผลลัพธ์

บทที่ 6

สรุปผลการศึกษาและข้อเสนอแนะ

โครงการพัฒนาระบบเพื่อวิเคราะห์ความสัมพันธ์ของข้อมูลโดยใช้อัลกอริธึมดีเอชพีนี้ เป็นโครงการที่จัดทำขึ้นเพื่อนำเสนอให้เห็นถึงประโยชน์ของการนำทฤษฎีของดาต้าไมนิ่ง มาใช้ในการหาความสัมพันธ์ของข้อมูล เพื่อนำไปประยุกต์ใช้กับธุรกิจด้านต่างๆ เพื่อให้ได้กำไรสูงสุด

6.1 สรุปผลการดำเนินงาน

โครงการนี้เป็นการพัฒนาระบบเพื่อหาความสัมพันธ์ของข้อมูล โดยใช้อัลกอริธึมดีเอชพี (DHP : Direct Hashing and Pruning) ซึ่งปรับปรุงมาจากอัลกอริธึม Apriori ซึ่งเป็นอัลกอริธึมพื้นฐานของ Association Rule โดยเป็นเทคนิคหนึ่งของ Link Analysis ทำให้สามารถวิเคราะห์ความสัมพันธ์ของข้อมูลได้รวดเร็วขึ้น เนื่องจากมีการใช้เทคนิคการ hash และการ prune มาช่วยในการหาความสัมพันธ์

ระบบสามารถรับข้อมูลได้ 2 รูปแบบ คือ Relational Database และ Text File โดยสามารถนำระบบนี้ไปใช้ในการวิเคราะห์ข้อมูลต่างๆ เช่น ข้อมูลการซื้อสินค้าของลูกค้า เพื่อวิเคราะห์พฤติกรรมการซื้อสินค้าของลูกค้าว่ามักจะซื้อสินค้าใดพร้อมกัน ทำให้สามารถนำข้อมูลที่ได้ ไปใช้ในการวางแผนการตลาด เช่น ควรวางสินค้าที่มักถูกซื้อพร้อมกันไว้ใกล้กัน เพื่ออำนวยความสะดวกให้กับลูกค้า หรือวางไว้คนละมุมของร้าน เพื่อที่ว่าเมื่อลูกค้าซื้อสินค้าชิ้นหนึ่งแล้ว ต้องการซื้ออีกชิ้นก็จะต้องเดินผ่านสินค้าอื่นๆในร้าน เป็นการกระตุ้นยอดขายของสินค้าชนิดอื่นๆในร้านด้วย นอกจากนี้ยังนำมาใช้ในการวางแผนส่งเสริมการขายได้อีกด้วย

ทั้งนี้ผลลัพธ์ของความสัมพันธ์ที่ได้ไม่อาจยืนยันได้ถึงความสำเร็จ 100 เปอร์เซ็นต์ในการกำหนดทิศทางทางการตลาด แต่ก็เป็นสิ่งหนึ่งที่ช่วยให้โอกาสที่ธุรกิจจะประสบความสำเร็จมีมากขึ้น

6.2 ข้อเสนอแนะ

ระบบนี้สามารถนำไปใช้ในการวิเคราะห์ข้อมูลในด้านต่างๆ เนื่องจากถูกออกแบบมาให้สามารถติดต่อกับฐานข้อมูลได้หลายรูปแบบ และผู้ใช้สามารถกำหนดได้ว่า ต้องการวิเคราะห์ข้อมูลอะไร และจากตารางใดได้เอง ทำให้มีความยืดหยุ่นในการใช้งาน

นอกจากนี้ระบบนี้สามารถนำไปประยุกต์ใช้ในการหาค้นหาความสัมพันธ์ของข้อมูลแบบ Sequential Pattern Mining ได้อีกด้วย

บรรณานุกรม

Estelle Brand and Rob Gerriten. **Association and Sequencing.** [Online]. Available: http://www.dbmsmag.com/9807m03.html.

Jiawai Han and Micheline Kamber 2001. **Data Mining: Concepts and Technique.** Morgan Kaufmann Publishers.

Jiawei Han and Micheline Kamber. **Data Mining: Concepts and Techniques.** [Online]. Available: http://www.cs.sfu.ca/~han/dmbook.

Park, J.S. et al 1995. **An Effective Hash-Based Algorithm for Mining Association Rules.** In Proc. of ACM SIGMOD Intl. Conf. on Management of Data.

Peter Cabena 1997. **Discovering Data Mining: From Concept to Implementation** Prentice Hall

R. Agrawal and S. Srikant 1994. **Fast Algorithms for Mining Association Rules in Large Databases.** Proceedings of the 20th International Conference on Very Large Data Bases.

S.Ayse Ozel and H. Altay Guvenir 1996. **An Algorithm for Mining Association Rules Using Perfect Hashing and Database Pruning.** IEEE.

Stonebraker and Hellerstein. **Data Mining.** [Online]. Available: http://redbook.cs.berkeley.edu/lec29.html.

ประวัติผู้เขียน

ชื่อ	นางสาวนิสา หงษ์สุรกุล
วันเกิด	8 ธันวาคม พ.ศ. 2523
ประวัติการศึกษา	ระดับอุดมศึกษา วิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเกษตรศาสตร์
ที่อยู่ปัจจุบัน	637 ซ. รัชดานิเวศน์ 17 ถ.ประชาอุทิศ เขตห้วยขวาง กทม. 10320
E-mail	nisa_jing@yahoo.com