การพัฒนาระบบดาต้าไมน์นิ่งเพื่อค้นหาความสัมพันธ์แบบหลายระดับ

Data Mining System for Discovery Multi-Level Association Rules

โดย

นภาพร รุ่งแสงเพชร

รหัส 45061623

อาจารย์ที่ปรึกษา

ผศ.ดร. วรพจน์ กรีสุระเดช

รายงานนี้เป็นส่วนหนึ่งของวิชาโครงการพัฒนาระบบงาน

หลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคเรียนที่ 1 ปีการศึกษา 2548

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อหัวข้อ	การพัฒนาระบบดาต้าไมน์นิ่งเพื่อค้นหาความสัมพันธ์แบบหลายระดับ
นักศึกษา	นางสาว นภาพร รุ่งแสงเพชร
อาจารย์ที่ปรึกษา	ผศ.ดร. วรพจน์ กรีสุระเดช
ระดับการศึกษา	วิทยาศาสตรมหาบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
แขนงวิชา	วิทยาการสารสนเทศ
ปีการศึกษา	2548

บทคัดย่อ

กระบวนการทางด้านดาต้าไมน์นิ่งถูกนำมาใช้ในการวิเคราะห์ข้อมูลได้หลายรูปแบบ ซึ่งวิธีการที่รู้จักกันแพร่หลายที่ในธุรกิจต่างๆ นิยมนำมาใช้ในการวิเคราะห์ข้อมูลการบริโภคของลูกค้าเพื่อค้นหาความสัมพันธ์ของรายการสินค้าที่เป็นที่นิยม นั่นคือการทำดาต้าไมน์นิ่งเพื่อค้นหาความสัมพันธ์ของข้อมูล

โครงงานพัฒนาระบบดาต้าไมน์นิ่งนี้ จะทำการศึกษาและพัฒนาระบบดาต้าไมน์นิ่งเพื่อค้นหาความสัมพันธ์ของข้อมูลในแบบหลายระดับ ซึ่งจะทำการวิเคราะห์รวมไปถึงการจัดกลุ่มของข้อมูลที่สนใจด้วย ทำให้ได้ประโยชน์ในการค้นหากฎความสัมพันธ์ของข้อมูลได้ละเอียดมากขึ้น ซึ่งจะใช้อัลกอริทึมคูมูเลทเป็นต้นแบบในการประมวลผลข้อมูล

Title	Data Mining System for Discovery Multi-Level Association Rules
Student	Miss. Napaporn Roongsangpetch
Advisor	Asst. Prof. Dr. Worapoj Kreesuradej
Level of Study	Master of Science in Information Technology
Major	Information Science
Academic Year	2005

ABSTRACT

The data mining is used to analyze the data in several types. The very popular method that many businesses use to analysis the consumer data for discovery the association of the favorite data item. This method is data mining for discovery data association.

Data Mining System for Discovery Multi-Level Association Rules is used for mining multiple levels of data association. The project use cumulate algorithm that process including of data category so it will give more delicate information.

กิตติกรรมประกาศ

โครงงานนี้ได้รับความสนับสนุนอย่างดีจากหลายๆฝ่ายซึ่งทำให้ผู้จัดทำรู้สึกขอบคุณ และมีความปีติยินดีเป็นอย่างมากที่งานสำเร็จขึ้นมาได้ การที่โครงการนี้สามารถที่จะสำเร็จลุล่วงได้เป็นอย่างดีเนื่องจากได้รับความช่วยเหลือจากบุคคลต่างๆ เหล่านี้

ข้าพเจ้าขอขอบคุณอาจารย์วรพจน์ กรีสุระเดช ซึ่งเป็นอาจารย์ที่ปรึกษาสำหรับโครงการพัฒนาระบบงานนี้ที่ได้กรุณาสละเวลา ให้ความรู้ ให้คำปรึกษา ให้ความเอาใจใส่ และให้คำแนะนำต่างๆ อันเป็นประโยชน์ต่อการพัฒนาระบบของข้าพเจ้าเป็นอย่างมาก

ข้าพเจ้าขอกราบขอบพระคุณบิดา มารดา, อาอึ๊ ตลอดจนขอบใจพี่น้องที่ได้ให้กำลังใจในการทำโครงการนี้ตลอดมา และที่ลืมไม่ได้ ต้องขอขอบคุณเพื่อนๆ ที่มีส่วนให้ความช่วยเหลือ คอยผลักดันและแนะนำ ติเตือนทำให้ผลงานนี้สำเร็จออกมาได้ ขอบคุณค่ะ

นภาพร รุ่งแสงเพชร

ตุลาคม 2548

สารบัญ

หน้า

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญตาราง VII
สารบัญภาพ VIII
บทที่
1. บทนำ 1
1.1 ความเป็นมา 1
1.2 วัตถุประสงค์ 1
1.3 ขอบเขตของโครงการ 2
1.4 ขั้นตอนการดำเนินงาน 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2
2. ดาต้าไมน์นิ่ง 3
2.1 ดาต้าไมน์นิ่งคืออะไร 3
2.2 กระบวนการทำงานของดาต้าไมน์นิ่ง 3
2.2.1 การเลือกข้อมูล (Data Selection) 4
2.2.2 การเตรียมข้อมูล (Data Preprocessing) 4
2.2.3 การแปลงข้อมูล (Data Transformation) 5
2.2.4 การทำดาต้าไมน์นิ่ง (Data Mining) 6
2.2.5 การวิเคราะห์ผลลัพธ์ (Interpretation / Evaluation) 7
3. การค้นหากฎความสัมพันธ์จากข้อมูล 8
3.1 กฎความสัมพันธ์ 8
3.2 การคัดเลือกกฎ 11
3.3 ข้อดีและข้อเสียของกฎความสัมพันธ์ 12
3.4 อัลกอริทึมอพริออริ 13

สารบัญ (ต่อ)

หน้า

3.4.1 สัญลักษณ์ที่ใช้ในอัลกอริทึมอพริออริ 14
3.4.2 การทำงานของอัลกอริทึมอพริออริ 14
3.4.3 การสร้างกฎความสัมพันธ์ 17
4. การค้นหากฎความสัมพันธ์จากข้อมูลแบบหลายลำดับชั้น 19
4.1 การค้นหากฎความสัมพันธ์จากข้อมูลแบบหลายลำดับชั้น 19
4.2 อัลกอริทึมเบสิค 20
4.3 อัลกอริทึมคิวมูเลท 22
4.3.1 การค้นหาบรรพบุรุษของข้อมูลเพื่อเพิ่มในทรานเซคชั่น 22
4.3.2 เตรียมข้อมูลบรรพบุรุษก่อนการประมวลผล 22
4.3.3 ลบไอเท็มเซตที่มีตัวไอเท็มและบรรพบุรุษอยู่ด้วยกัน 22
4.4 การทำงานของอัลกอริทึมคิวมูเลท 23
4.4.1 การสร้างกฎความสัมพันธ์ 27
5. การวิเคราะห์และออกแบบระบบ 29
5.1 สภาพแวดล้อมการพัฒนาระบบ 29
5.1.1 ด้านฮาร์ดแวร์ 29
5.1.2 ด้านซอฟต์แวร์ 29
5.2 โครงสร้างตารางที่ใช้ในระบบ 30
5.3 การพัฒนาระบบและหน้าจอการทำงาน 34
6. การประยุกต์ใช้งาน 43
6.1 วัตถุประสงค์ในธุรกิจ 43
6.2 การเตรียมข้อมูล 43
6.3 การไมน์นิ่งข้อมูล 46
6.3.1 ข้อมูลทรานเซคชั่น 46
6.3.2 ข้อมูลไอเท็ม 46
6.4 วิเคราะห์ผลลัพธ์ของการทำไมน์นิ่ง 47
6.5 สรุปผลการทำไมน์นิ่ง และข้อเสนอแนะในการนำไปใช้ 48

สารบัญ (ต่อ)

หน้า

7. สรุปผลการดำเนินงาน 49
7.1 สรุปผลการวิเคราะห์และออกแบบระบบ 49
7.2 ข้อเสนอแนะ 49
7.3 ข้อจำกัด 49
บรรณานุกรม 50
ประวัติผู้เขียน 51

สารบัญตาราง

หน้า

ตารางที่

3.1 ตัวอย่างฐานข้อมูล (D) ที่ใช้ในการทำงานของอัลกอริทึมอพริออริ 15
3.2 ผลลัพธ์ของ L_k ไอเท็มเซต 17
4.1 ตาราง T* ที่สร้างมาจากกราฟบางหมวดหมู่ 23
4.2 ฐานข้อมูล *D* 24
4.3 ตาราง C_1 24
4.4 ตาราง L_1 24
4.5 ตาราง C_2 25
4.6 ตาราง C_2 25
4.7 ตาราง L_2 25
4.8 ตาราง $\cup_k L_k$ 25
4.9 ตารางกฎความสัมพันธ์ ที่ได้จากตาราง $\cup k\ Lk$ 26
4.10 ตารางกฎความสัมพันธ์ ที่ได้จากตาราง $\cup k\ Lk$ ผ่านค่ามินิมัมคอนฟิเดนซ์ 28
5.1 โครงสร้างของตารางที่ใช้เก็บรายละเอียดของงาน 31
5.2 โครงสร้างของตารางที่ใช้เก็บรายละเอียดข้อมูลกราฟแบ่งหมวดหมู่ 31
5.3 โครงสร้างของตารางที่ใช้เก็บรายละเอียดโครงสร้างกราฟแบ่งหมวดหมู่ 32
5.4 โครงสร้างของตารางที่ใช้เก็บ C_k ไอเท็มเซต ที่ใช้ขณะประมวลผล 32
5.5 โครงสร้างของตารางที่ใช้เก็บ L_k Itemset ที่ใช้ขณะประมวลผล 33
5.6 โครงสร้างของตารางที่ใช้เก็บรายละเอียดของกฎความสัมพันธ์ 34
6.1 โครงสร้างของตารางที่ใช้เก็บรายละเอียดสินค้า 44
6.2 โครงสร้างของตารางที่ใช้เก็บรายละเอียดข้อมูลการจัดหมวดหมู่สินค้า 45
6.3 โครงสร้างของตารางที่ใช้เก็บรายละเอียดการขายสินค้า 45

สารบัญภาพ

หน้า

ภาพที่

2.1 กระบวนการทำดาต้าไมน์นิ่ง 4
3.1 ภาพผลการขายสินค้าประเภทผ้าอ้อมและเบียร์ 9
3.2 อัลกอริทึมอพริออริ 13
3.3 การหาสมาชิกของ C_k 13
3.4 อัลกอริทึมอพริออริเจน ที่ใช้ในการหาสมาชิก C_k 14
3.5 การสร้าง C_k และ L_k 16
4.1 ฐานข้อมูล *D* 19
4.2 กราฟแบ่งหมวดหมู่ 19
4.3 อัลกอริทึมเบสิค 21
4.4 อัลกอริทึมคิวมูเลท 23
4.5 แสดงการเพิ่มบรรพบุรุษของไอเท็มลงในทรานเซคชั่น 24
5.1 แบบจำลองความสัมพันธ์ของฐานข้อมูล 30
5.2 หน้าจอหลักของระบบ 34
5.3 หน้าจอสำหรับรับข้อมูลงานไมน์นิ่งใหม่ 35
5.4 หน้าจอสำหรับเลือกไฟล์ฐานข้อมูลที่ต้องการทำไมน์นิ่ง 35
5.5 หน้าจอสำหรับตารางและฟิลด์ข้อมูลที่จะนำมาทำการไมน์นิ่ง 36
5.6 ข้อความเตือนในกรณีที่เลือกฟิลด์ข้อมูลไม่ตรงกับฟิลด์ทรานเซคชั่น 36
5.7 หน้าจอสำหรับตารางและฟิลด์ข้อมูลที่เลือกนำมาทำการไมน์นิ่ง 37
5.8 แสดงข้อมูลฟิลด์ทรานเซคชั่นไอดีและไอเท็มไอดี 37
5.9 แสดงข้อมูลตารางและฟิลด์ที่สัมพันธ์กับข้อมูล ไอเท็มไอดีที่เลือกไว้ 38
5.10 แสดงข้อมูลฟิลด์ ไอเท็มไอดีและข้อมูลลำดับชั้นทั้งหมด 38
5.11 แสดงข้อมูลงานไมน์นิ่งที่สร้างรวมถึงลำดับชั้นและข้อมูลไอเท็มทั้งหมดที่เลือกไว้ 39
5.12 แสดงการเลือกฟิลด์ที่ไม่ต้องการที่จะแสดงผล 39
5.13 ข้อมูลกราฟ “Data Hierarchy Detail” ที่แสดงหลังจากที่ลบบางฟิลด์ออกแล้ว 40
5.14 หน้าจอรับข้อมูลไอเท็มใหม่ที่ผู้ใช้ต้องการสร้างเพิ่ม 40

สารบัญภาพ (ต่อ)

หน้า

ภาพที่

5.15 แสดงข้อมูลไอเท็มที่ผู้ใช้สร้าง 41
5.16 แสดงข้อมูลไอเท็ม ที่ผู้ใช้ทำการจัดกลุ่มใหม่ตามที่ต้องการ 41
5.17 แสดงผลกฎความสัมพันธ์ที่ได้จากการทำดาต้าไมน์นิ่ง 42
6.1 แบบจำลองความสัมพันธ์ของฐานข้อมูลที่จะนำมาทำการไมน์นิ่ง 43
6.2 แสดงรายละเอียดลำดับชั้นของหมวดหมู่สินค้าและชื่อสินค้า 46
6.3 แสดงผลกฎความสัมพันธ์ที่ได้จากการทำดาต้าไมน์นิ่ง 47

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ปัจจุบันนี้การดำเนินงานแทบจะทุกด้านมีการเก็บข้อมูลในรูปแบบดิจิตอลกันมากมาย ซึ่งมีการนำข้อมูลที่เก็บไว้เหล่านั้นมาวิเคราะห์ได้มากมายหลายด้าน เพื่อให้ได้ข้อเท็จจริงเกี่ยวกับข้อมูลและสาระสำคัญ(Information) ของข้อมูลนั้นๆออกมา การคิดค้นและพัฒนาเครื่องมือเพื่อช่วยในการวิเคราะห์ข้อมูลมีมาอย่างแพร่หลาย

ดาต้าไมน์นิ่งก็เป็นวิธีการหนึ่งที่นิยมใช้ในการนำใช้ในการวิเคราะห์ข้อมูลต่างๆ ให้ได้ข้อมูลที่เป็นประโยชน์กับธุรกิจได้ เช่นถ้าต้องการทราบแนวโน้มการบริโภคของลูกค้านั้นก็อาจจะนำดาต้าไมน์นิ่งมาวิเคราะห์กับข้อมูลงานขายหรือให้บริการต่างๆได้ ซึ่งการวิเคราะห์หาความสัมพันธ์ของข้อมูลก้อเป็นวิธีการหนึ่งที่สามารถทราบว่าข้อมูลรายการใดมีความสัมพันธ์กับรายการใด ทำให้ธุรกิจต่างๆสามารถนำสาระสำคัญตรงนี้มาใช้ประกอบการพิจารณาจัดโปรโมชั่นให้ตรงตามความต้องการของผู้บริโภคได้

ในยุคแรกการทำดาต้าไมน์นิ่งที่นิยมใช้เพื่อค้นหาความสัมพันธ์ของข้อมูลนั้นนิยมพิจารณาจากรายการสินค้าเป็นหลัก ซึ่งประเภท ชนิด และยี่ห้อของสินค้าแต่ละรายการนั้นมีรายละเอียดแตกต่างกันออกมามากมาย ซึ่งทำให้เมื่อวิเคราะห์ข้อมูลแค่ลำดับเดียว (Single Level) แล้วจะได้ความสัมพันธ์ของข้อมูลที่กระจัดกระจาย ซึ่งอาจจะตกหล่นความสัมพันธ์บางประเด็นที่สำคัญไป ดังนั้นจึงมีการพัฒนาต่อมาเป็น กฎความสัมพันธ์จากข้อมูลแบบหลายลำดับชั้น (multi-level association) ซึ่งวิเคราะห์ข้อมูลร่วมกับประเภท ชนิดของสินค้าทำให้ได้ข้อเท็จจริงหลักๆออกมาได้ละเอียดยิ่งขึ้น

1.2 วัตถุประสงค์

1. เพื่อศึกษาวิธีการและขั้นตอนการทำงานของดาต้าไมน์นิ่ง
2. เพื่อให้เข้าใจถึงวิธีการวิเคราะห์ข้อมูลเพื่อหาความสัมพันธ์ของข้อมูลแบบทรานเซคชั่น
3. เพื่อให้เข้าใจถึงการทำงานของอัลกอริทึมคิวมูเลท (Cumulate algorithm) ซึ่งเป็นอัลกอริทึมที่ใช้หาความสัมพันธ์ของข้อมูลในแบบหลายลำดับชั้น (multi-level association) ชนิดหนึ่ง

4. เพื่อพิจารณาหาแนวทางในการพัฒนาโปรแกรม การค้นหากฎความสัมพันธ์หลายลำดับชั้นที่เหมาะสมต่อไป

1.3 ขอบเขตของโครงการ

โครงการนี้เป็นการศึกษาและพัฒนาระบบที่ใช้ในการวิเคราะห์ความสัมพันธ์ของข้อมูลโดยใช้อัลกอริทึมดิวมูเลท ซึ่งเป็นการทำดาต้าไมน์นิ่งโมเดลประเภท วิเคราะห์ความสัมพันธ์ของข้อมูล (Link Analysis) ที่ทำงานแบบค้นหากฎความสัมพันธ์ของข้อมูลแบบหลายลำดับชั้น เพื่อใช้ในการหาความสัมพันธ์ของข้อมูลในหลายลำดับชั้น มีขอบเขตของโครงการดังนี้

1. สามารถพัฒนาโปรแกรมตามทฤษฎี และอัลกอริทึมได้ถูกต้อง
2. สามารถนำเสนอผลลัพธ์ให้กับผู้ใช้งานนำไปใช้ประโยชน์ได้ถูกต้อง

1.4 ขั้นตอนการดำเนินงาน

การจะทำโครงการให้สำเร็จลุล่วงตามวัตถุประสงค์ได้ ต้องมีการกำหนดการดำเนินงานเพื่อให้การทำงานเป็นไปตามลำดับ เป็นขั้นตอน ดังนี้

1. กำหนดหัวข้อ เป้าหมาย และวัตถุประสงค์ ตลอดจนขอบเขตของโครงการ โดยระบบงานนี้ได้กำหนดเป้าหมาย เพื่อทำการสร้างกฎความสัมพันธ์ของข้อมูลแบบหลายลำดับชั้นนั่นเอง
2. ศึกษาทฤษฎี และงานวิจัยที่เกี่ยวข้องกับหลักการของกฎความสัมพันธ์ (Association Rules), กระบวนการทางด้านดาต้าไมน์นิ่งและอัลกอริทึมที่เกี่ยวข้อง เช่น อัลกอริทึมอพริออริ, อัลกอริทึมเบสิค และอัลกอริทึมดิวมูเลท
3. ทำการ ออกแบบและพัฒนาโปรแกรมให้เป็นไปตามอัลกอริทึมดิวมูเลทได้อย่างถูกต้อง
4. ทดสอบโปรแกรมเพื่อหาข้อบกพร่องของโปรแกรม, ปรับปรุงและแก้ไขหากมีข้อผิดพลาดเกิดขึ้น
5. จัดทำเอกสารประกอบ และสรุปผลการดำเนินงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

การพัฒนาระบบดาต้าไมน์นิ่งเพื่อค้นหาความสัมพันธ์แบบหลายลำดับชั้นนี้ คาดว่าจะได้รับประโยชน์ต่างๆ ดังนี้

1. ทำให้เข้าใจถึงหลักการ, วิธีการ รวมไปถึงขั้นตอนการทำงานของดาต้าไมน์นิ่ง
2. สามารถนำระบบที่พัฒนาขึ้นไปใช้ในการวิเคราะห์หาความสัมพันธ์ของข้อมูลไปประยุกต์ใช้ให้ก่อประโยชน์ในด้านธุรกิจต่างๆได้

บทที่ 2

ดาต้าไมน์นิ่ง

การสืบค้นหาความรู้ที่เป็นประโยชน์ และน่าสนใจบนฐานข้อมูลขนาดใหญ่หรือที่เรียกกันว่าดาต้าไมน์นิ่งถือเป็นวิทยาศาสตร์คอมพิวเตอร์ (Computer Science) สาขาหนึ่ง ที่ได้รับความสนใจอย่างมาก เนื่องจากมีการนำเอาเทคนิคของดาต้าไมน์นิ่งมาใช้วิเคราะห์สืบค้นหาข้อเท็จจริงที่ซ่อนอยู่ในฐานข้อมูลออกมาประยุกต์ใช้ให้เป็นประโยชน์แก่ธุรกิจซึ่งในปัจจุบันมีการแข่งขันกันสูงได้ เช่น การวิเคราะห์หาความต้องการของลูกค้า หรือการทำนายเหตุการณ์ที่กำลังจะเกิดขึ้น เป็นต้น

2.1. ดาต้าไมน์นิ่งคืออะไร

ดาต้าไมน์นิ่งคือกระบวนการในการค้นหาความรู้ที่น่าสนใจ เช่น รูปแบบ, ความสัมพันธ์, การเปลี่ยนแปลง, ความผิดปกติ และลักษณะโครงสร้างที่สำคัญจากข้อมูลจำนวนมากที่เก็บในฐานข้อมูล, คลังข้อมูล หรือแหล่งที่เก็บข้อมูลอื่นๆ เพื่อให้ได้สารสนเทศที่ยังไม่เคยทราบออกมา ทำให้สามารถนำไปประยุกต์ใช้ในธุรกิจต่างๆ ได้ง่าย

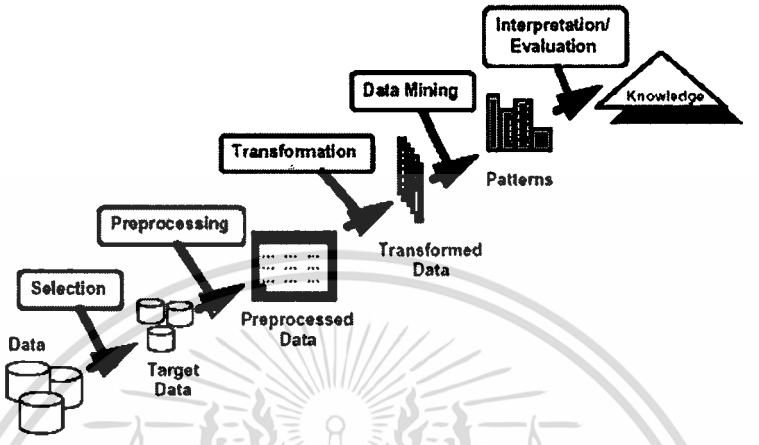
ปัจจัยที่ทำให้ดาต้าไมน์นิ่งได้รับความสนใจมีดังต่อไปนี้

- ในปัจจุบันธุรกิจต่างๆให้ความสำคัญกับการเก็บข้อมูลของลูกค้ามากมาย จึงทำให้ข้อมูลที่เก็บมีการขยายตัวอย่างรวดเร็ว ถ้าปล่อยไว้เฉยๆ จะทำให้เปล่าประโยชน์ จึงมีการคิดหากระบวนการในการวิเคราะห์ข้อมูลที่เก็บไว้เพื่อให้ได้ข้อเท็จจริงออกมา
- กระบวนการทำดาต้าไมน์นิ่งจะให้ข้อเท็จจริงที่อ่านแล้วตีความง่าย ซึ่งสามารถนำไปประยุกต์ใช้ในระบบสนับสนุนการตัดสินใจได้ทันที เพื่อความสะดวกและรวดเร็วในการแข่งขันทางธุรกิจ

2.2. กระบวนการทำงานของดาต้าไมน์นิ่ง (Han and Kamber. 2000)

เนื่องจากดาต้าไมน์นิ่งเป็นกระบวนการที่ทำงานกับฐานข้อมูลต่างๆ จึงเป็นที่รู้จักกันอีกชื่อหนึ่งว่า การค้นหาความรู้จากฐานข้อมูล (Knowledge Discovery in Database) ซึ่งฐานข้อมูลแต่ละประเภทมีลักษณะแตกต่างกันออกไป ดังนั้นก่อนจะทำดาต้าไมน์นิ่งได้นั้นต้องมีวิธีการเตรียมข้อมูลและมีกระบวนการในการเตรียมข้อมูลต่างๆให้มีความพร้อมในการที่จะทำดาต้าไมน์นิ่ง เพื่อให้ได้ข้อมูลที่เหมาะสมออกมาได้ก่อน ขั้นตอนต่างๆ ให้พร้อมจึงจะทำดาต้าไมน์นิ่งและวิเคราะห์ผลลัพธ์ที่ได้เป็นลำดับสุดท้าย

กระบวนการที่กล่าวมานี้ ประกอบด้วย 5 ขั้นตอนดังแสดงในภาพที่ 2.1



ภาพที่ 2.1 กระบวนการทำดาต้าไมน์นิ่ง

2.2.1 การเลือกข้อมูล (Data selection)

การเลือกข้อมูลเฉพาะที่ต้องการเพื่อที่จะนำมาวิเคราะห์ ให้ตรงกับจุดประสงค์ในการทำดาต้าไมน์นิ่งทำการแยกข้อมูลที่ไม่ต้องการออกไป ซึ่งจะเป็นการเริ่มต้นของการเตรียมการไมน์นิ่งการเลือกข้อมูลนั้นแตกต่างไปตามวัตถุประสงค์ของแต่ละธุรกิจที่ได้กำหนดไว้ตอนต้น การเลือกข้อมูลจำเป็นที่จะต้องมีความเข้าใจกับชนิดของข้อมูลและประเภทของข้อมูลที่จะต้องนำมาใช้ด้วย เช่นการทำดาต้าไมน์นิ่งประเภทวิเคราะห์ความสัมพันธ์ของข้อมูล (link analysis) นั้นต้องใช้ข้อมูลประเภททรานเซคชั่นเป็นต้น

2.2.2 การเตรียมข้อมูล (Data Preprocessing)

ตรวจสอบข้อมูล และแก้ไขเพื่อทำให้ได้ข้อมูลที่มีคุณภาพดี และทำให้ข้อมูลถูกเลือกนั้นถูกต้อง ครบถ้วนตามที่จะต้องใช้ในการทำดาต้าไมน์นิ่งเนื่องจากข้อมูลที่ถูกเลือกมาจากกระบวนการเลือกข้อมูล ซึ่งอาจมีบางข้อมูลที่มีจุดบกพร่อง, ขาดหายไป หรือข้อมูลที่เก็บไว้ล้าสมัย ดังนั้นในขั้นตอนนี้จะต้องพิจารณาเพิ่มเติมหลักๆ อยู่ 3 ประเด็น ได้แก่

- **การกำจัดค่าของข้อมูลที่ผิดพลาด (Noisy Data)**

ข้อมูลที่มีลักษณะแตกต่างจากข้อมูลที่คาดการณ์เอาไว้ หรือ ค่าของข้อมูลอาจจะผิดไปจากที่ควรจะเป็น ซึ่งอาจจะเกิดจากการป้อนข้อมูลผิด เช่น ข้อมูลอายุเป็น 650 ปี หรือป้อนรายได้เป็นข้อมูลติดลบ เป็นต้น ซึ่งข้อมูลที่ผิดนี้ อาจเป็นเหตุให้การวิเคราะห์ผิดพลาดได้ จึงต้องทำการกำจัดข้อมูลที่ผิดนี้ออกไป

- **การจัดการกับค่าของข้อมูลที่สูญหาย (Missing Value)**

ค่าของข้อมูลที่ไม่ได้ถูกเลือกมาจากขั้นตอนเลือกข้อมูลหรืออาจจะเป็นค่าที่ไม่สมบูรณ์ ที่เราทำการลบออกไประหว่างการกำจัดค่าของข้อมูลที่ผิดพลาด ทำให้บางค่าอาจสูญหายไปจากความผิดพลาดในการเก็บข้อมูล หรือเกิดจากความผิดพลาดในการบันทึกข้อมูล ซึ่งสามารถแก้ไขได้โดยทำการลบข้อมูลนั้นทิ้งทั้งรายการ หรือทำการบันทึกแทนที่ในค่าที่ขาดหายไปด้วยค่าเฉลี่ย, ค่าที่ปรากฏบ่อย หรืออาจบันทึกเป็น "Unknown" เป็นต้น

- **การจัดการข้อมูลที่ไม่ถูกต้อง (Inconsistence Data)**

เนื่องจากข้อมูลที่เลือกมาอาจจะนำมาจากหลายๆแหล่งมารวมกัน จึงอาจมีข้อมูลที่ไม่ถูกต้องตรงกัน เช่น ชื่อของลูกค้าในไอดี (ID) เดียวกันอาจจะไม่ตรงกัน สามารถแก้ไขได้โดยพิจารณาว่าเป็นลูกค้าคนเดียวกันหรือไม่ ถ้าใช่ก็ทำการนำข้อมูลที่ได้มีการบันทึกล่าสุดมาแทนค่าในข้อมูลที่เก่ากว่า หรือถ้าไม่ใช่ลูกค้าคนเดียวกันก็ต้องทำการเพิ่มไอดีให้เป็นลูกค้าอีกคนหนึ่ง

2.2.3 การแปลงข้อมูล (Data Transformation)

การแปลงข้อมูลที่มี นำมาเปลี่ยนแปลงทำให้อยู่ในรูปแบบของข้อมูลที่พร้อมจะนำไปวิเคราะห์กับ อัลกอริทึมที่ใช้กับต่างๆของดาต้าไมน์นิ่งเช่นการแปลงตัวเลขให้เป็นช่วงๆ เพื่อใช้กับอัลกอริทึมของดิซิชั่นทรี (Decision Tree) หรือการปรับอัตราส่วนตัวเลขให้อยู่ในช่วง 0-1 เพื่อใช้กับนิวรอลเนทเวิร์ค (Neural Network) เป็นต้น

2.2.4 การทำดาต้าไมน์นิ่ง (Data Mining)

คือการประมวลผลข้อมูลตามอัลกอริทึมที่ได้กำหนดเอาไว้ ซึ่งในขั้นตอนนี้จะมีความสัมพันธ์กับการวิเคราะห์ข้อมูลและขั้นตอนการแปลงข้อมูลที่ผ่านมา ในการพัฒนาในส่วนของดาต้าไมน์นิ่งจะเกี่ยวข้องกับการใช้ อัลกอริทึมหลายๆแบบ

กระบวนการในการวิเคราะห์ข้อมูลของดาต้าไมน์นิ่ง แบ่งออกเป็น 4 ประเภทหลักๆ ได้แก่

- **การทำดาต้าไมน์นิ่งเพื่อการพยากรณ์ (Predictive Modeling)** คือการทำนายแนวทางของข้อมูลที่ควรจะเกิดขึ้นต่อไปในอนาคต เช่น ทำนายภูมิอากาศ
- **การแบ่งกลุ่มข้อมูล (Data Segmentation)** เป็นวิธีในการจัดกลุ่มข้อมูล ใช้ในการแบ่งประเภทสินค้า หรือลูกค้าเพื่อการทำประวัติลูกค้า (Customer profile) เพื่อแบ่งกลุ่มลูกค้าชั้นดี หรือหากลุ่มลูกค้าเป้าหมายให้ตรงกับความต้องการในการทำโปรโมชั่นได้
- **การวิเคราะห์ความสัมพันธ์ของข้อมูล (Link Analysis)** เป็นที่นิยมใช้ในการทำการตลาด เช่น อาจจะนำไปใช้ในการทำครอสมาร์เก็ต (cross market)
- **วิเคราะห์หาสิ่งที่ผิดปกติ (Deviation Detection)** ใช้วิเคราะห์หาพฤติกรรมผิดปกติที่เกิดขึ้นในข้อมูล ได้แก่การตรวจสอบการฟอกเงิน หรือตรวจจับการโกง เช่น บริษัทประกันใช้ในการตรวจสอบอาการของโรคที่คนไข้เบิกรักษากับปริมาณและประเภทของยาที่หมอจ่ายให้

การนำดาต้าไมน์นิ่งไปใช้จริงในบางโอกาสอาจต้องประยุกต์ใช้หลายประเภทด้วยกัน เช่น อาจจะจัดกลุ่มข้อมูลประเภทสินค้าก่อน จึงจะนำไปวิเคราะห์ความสัมพันธ์ต่อไป เนื่องจากถ้าใช้ข้อมูลสินค้าทั้งหมดในการวิเคราะห์ความสัมพันธ์นั้น ข้อเท็จจริงที่ได้ออกมาอาจจะมีหลากหลายมากจนทำให้ไม่ได้ข้อเท็จจริงที่เป็นประโยชน์มากๆออกมา เช่น การวิเคราะห์ความสัมพันธ์ในการซื้อสินค้าของลูกค้า ทางบริษัทร้านค้าอาจเก็บข้อมูลแยกสินค้าประเภทนม ออกเป็นหลายยี่ห้อและหลายประเภท ถ้าวิเคราะห์จัดกลุ่มสินค้าประเภทนมให้เป็นกลุ่มเดียวกัน เมื่อผ่านกระบวนการดาต้าไมน์นิ่งแล้วจะทำให้ได้ข้อเท็จจริงที่ชัดเจนยิ่งขึ้น เป็นต้น

2.2.5 การวิเคราะห์ผลลัพธ์ (Interpretation / Evaluation)

หลังจากกระบวนการทำไมน์นิ่งแล้ว ต้องมีการวิเคราะห์ผลลัพธ์ที่ได้ของการประมวลผล ซึ่งจะทำการตีความและประเมินผลลัพธ์ที่ได้จากขั้นตอนการทำดาต้าไมน์นิ่ง ว่าสามารถนำไปใช้ได้ตามวัตถุประสงค์ที่ต้องการได้หรือไม่ รวมทั้งเป็นการวิเคราะห์ถึงความถูกต้องของผลที่ได้จากการทำ พิจารณาความเป็นไปได้ของการเกิดผลลัพธ์ดังกล่าว เพราะบางครั้งผลที่ได้จากการทำไมน์นิ่ง อาจจะนำไปใช้ประโยชน์ไม่ได้ เพราะฉะนั้นการทำงานในส่วนนี้จึงจำเป็นที่จะต้องใช้ทักษะและประสบการณ์ในการวิเคราะห์ผลที่ได้ก่อนนำไปใช้งาน

บทที่ 3

การค้นหากฎความสัมพันธ์จากข้อมูล

การค้นหาข้อมูลจากข้อมูลแบบหลายลำดับชั้นเป็นดาต้าไมน์นิ่งประเภทลิงค์อนาไลซิส (link analysis) คือใช้วิเคราะห์หาความสัมพันธ์ ซึ่งนิยมใช้ในการหาความสัมพันธ์ของสินค้า (data ไอเท็ม) ที่เกิดขึ้นในรายการเดียวกัน ที่มีแนวโน้มว่าจะเกิดขึ้นพร้อมๆกัน เช่น พิจารณาสินค้าที่มักจะถูกซื้อควบคู่กันไปในคราวเดียวกัน

การวิเคราะห์ในลักษณะนี้เรียกว่า "Market Basket Analysis" ซึ่งจะนำไปใช้วิเคราะห์การซื้อของจากลูกค้าว่ารายการสินค้าที่มักจะถูกซื้อควบคู่กัน (item set) การพิจารณางานหรือเหตุการณ์ที่เกิดขึ้นในคราวเดียวกัน เรียกว่าทรานเซคชั่น (transaction) ทำให้ผู้ประกอบธุรกิจสามารถนำไปช่วย ในการวางแผนทางการตลาด หรือกำหนดกลยุทธ์ทางด้านการจำหน่ายสินค้าและบริการได้ เช่น

- **การจัดโปรโมชั่น** ถ้าลูกค้ามีแนวโน้มที่จะซื้อสินค้าชนิดหนึ่งคู่กับสินค้าอีกชนิดพร้อมกัน ควรที่จะมีการจัดโปรโมชั่นให้กับสินค้าชนิดใดชนิดหนึ่ง เพื่อดึงยอดขายของสินค้าอีกชนิดหนึ่ง
- **การวางตำแหน่งของสินค้า** การจัดวางสินค้าที่มีความสัมพันธ์กันเอาไว้ใกล้กัน ย่อมทำให้ลูกค้าหยิบสินค้าได้สะดวกขึ้น

3.1. กฎความสัมพันธ์ (Han and Kamber. 2000)

ผลลัพธ์ที่ได้จากการทำดาต้าไมน์นิ่งประเภทนี้จะได้ออกมาเป็นการหากฎความสัมพันธ์ (association rules) ซึ่งรูปแบบของกฎความสัมพันธ์ จะถูกเขียนแทนด้วยสัญลักษณ์ ต่างๆได้ดังนี้

- If X Then Y
- When Condition1 then Condition2
- X => Y

เรียก X หรือ Condition1 ว่า ตัวกฎ (Rule Body) หรือ เหตุการณ์ที่เกิดขึ้นก่อน (Antecedent)
เรียก Y หรือ Condition2 ว่า หัวของกฎ (Rule Head) หรือ ผลที่ตามมา (Consequent)

ซึ่งสัญลักษณ์ของกฎที่แสดงออกมานั้นขึ้นอยู่กับผู้พัฒนาระบบว่าจะให้แสดงออกมาในรูปแบบใด แต่ทั้งนี้ทั้งนั้น X และ Y คือข้อมูลที่เกิดขึ้นพร้อมกันในทรานเซคชั่นเดียวกัน ซึ่งการเขียนกฎนั้นฝั่ง X และ Y อาจเป็นสินค้าชนิดเดียวหรือหลายชนิดก็ได้

ตัวอย่างเช่น "If A, B then C, D"

ปัจจัยที่เกี่ยวข้องในการสร้างกฎนี้ ได้แก่

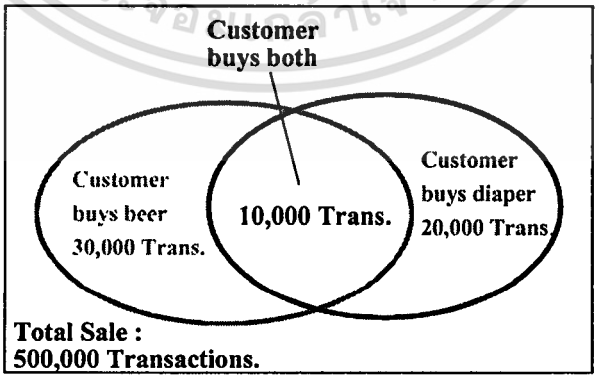
- ค่าความถี่ของเหตุการณ์ X และ Y ขึ้นจากจำนวนเหตุการณ์ทั้งหมด (Support Factor)
- ค่าความน่าเชื่อถือของเหตุการณ์ Y ที่น่าจะเกิดขึ้น เทียบจากเหตุการณ์ X และ Y พร้อมกัน (Confidence Factor)
- ค่าที่บอกได้ว่ากฎนี้มีประโยชน์เพียงพอหรือไม่ (Lift Factor)

ค่าซัพพอร์ท และคอนฟิเดนซ์ นั้นนิยมเขียนในวงเล็บไว้ข้างหลังกฎ ดังเช่นตัวอย่างต่อไปนี้

buys (x, "diapers") => buys (x, "beers") [2%, 50%]

แปลความหมายได้ว่าเหตุการณ์ที่ผ่านมา มีลูกค้าที่นิยมซื้อผ้าอ้อมและเบียร์ไปพร้อมกัน (support) เป็นจำนวน 2% ของรายการซื้อสินค้าทั้งหมดและมีรายการที่ซื้อผ้าอ้อมแล้วซื้อเบียร์ไปด้วย (confidence) เป็นจำนวนถึง 50% ด้วยกัน

การขายสินค้า ประเภทผ้าอ้อมและเบียร์ไปพร้อมๆกัน แสดงได้ดังภาพที่ 3.1



ภาพที่ 3.1 ภาพผลการขายสินค้าประเภทผ้าอ้อมและเบียร์

การคำนวณค่า ซัพพอร์ท, คอนฟิเดนซ์ และค่าลิฟต์สามารถอธิบายได้จากตัวอย่างการซื้อสินค้าของลูกค้าข้างต้น

สมมติ ตัวอย่างรายการซื้อสินค้าของลูกค้าที่ผ่านมาทั้งหมด 500,000 รายการ

- รายการซื้อผ้าอ้อม 20,000 รายการ (4%ของรายการทั้งหมด)
- รายการซื้อเบียร์ 30,000 รายการ (6%ของรายการทั้งหมด)
- รายการซื้อทั้งผ้าอ้อมและเบียร์ 10,000 รายการ (2%ของรายการทั้งหมด)

- ซัพพอร์ท (Prevalence)

เป็นค่าของสัดส่วนของจำนวนเหตุการณ์ซื้อผ้าอ้อมกับเบียร์คู่กัน เทียบกับจำนวนเหตุการณ์ขายสินค้าทั้งหมด

$$\text{ซัพพอร์ท} = \frac{\text{จำนวนชุดข้อมูลที่มีรายการผ้าอ้อมและเบียร์คู่กัน}}{\text{จำนวนชุดข้อมูลทั้งหมด}} = \frac{10,000}{500,000} = 2\%$$

- คอนฟิเดนซ์ (Predictability)

เป็นค่าสัดส่วนของจำนวนเหตุการณ์ซื้อผ้าอ้อมคู่กับเบียร์คู่กัน เทียบกับจำนวนของเหตุการณ์ซื้อผ้าอ้อมเพียงอย่างเดียว

$$\text{คอนฟิเดนซ์} = \frac{\text{จำนวนชุดข้อมูลที่มีรายการผ้าอ้อมและเบียร์คู่กัน}}{\text{จำนวนชุดข้อมูลที่มีรายการซื้อผ้าอ้อม}} = \frac{10,000}{20,000} = 50\%$$

ทั้งหมดสามารถนำมาแปลงเป็นกฎได้ว่า “ลูกค้า 50% ที่ซื้อผ้าอ้อมมักจะซื้อเบียร์ด้วย” และในทางกลับกัน (Reverse Rule) จะได้กฎอีกกฎคือ “ลูกค้าที่ซื้อเบียร์ จะมีโอกาสที่จะซื้อผ้าอ้อมด้วย 33.33%” แต่กฎทั้งสองข้อนั้น มีค่าซัพพอร์ทที่เท่ากันคือ 2 %

โดยปกติ กฎที่น่าสนใจ คือกฎที่มีค่าคอนฟิเดนซ์ ที่สูง เนื่องจากมีโอกาสที่จะเกิดขึ้นสูงตามด้วย และนอกจากตัววัดคือซัพพอร์ทและ คอนฟิเดนซ์ สองตัวนี้แล้ว ยังมีตัววัดค่าความน่าเชื่อถือของกฎที่สร้างขึ้น เรียกว่าลิฟต์

- ลิฟต์

เป็นค่าที่แสดงความน่าเชื่อถือของความสัมพันธ์ระหว่างเหตุการณ์ ยิ่งค่าลิฟต์มีค่าที่สูงเท่าไร ก็จะน่าเชื่อถือมากขึ้นเท่านั้น โดยหาได้จากสัดส่วนระหว่างค่าคอนฟิเดนซ์ ของกฎกับค่า เหตุการณ์ที่เกิด rule head ขึ้นจากเหตุการณ์ทั้งหมด

จากตัวอย่างกฎข้างต้น คอนฟิเดนซ์ มีค่า 50% ส่วนจำนวนชุดของการซื้อเบียร์ คือ 6 % จะคำนวณค่า ลิฟต์ ได้ดังนี้

$$\text{ลิฟต์} = \frac{\text{คอนฟิเดนซ์}}{\text{\% ของจำนวนชุดข้อมูลที่มีรายการซื้อเบียร์}} = \frac{50\%}{6\%} = 8.33$$

ค่า ลิฟต์ ที่ได้ จะแสดงถึงความสำคัญของความสัมพันธ์ หรือ เหตุการณ์ที่ว่ามีมากน้อยแค่ไหน จากข้อมูลที่ได้ ทำให้ได้กฎ "คนซื้อผ้าอ้อมนั้นมักจะซื้อเบียร์ด้วย" มีความน่าเชื่อถือมาก ซึ่งเมื่อพิจารณาร่วมกับค่า คอนฟิเดนซ์ ที่มีค่ามากถึง 50% ความสัมพันธ์ระหว่างเหตุการณ์มีความน่าเชื่อถือมากยิ่งขึ้น

ในกรณีที่ ค่า ลิฟต์ มีค่าน้อยกว่า 1 หมายถึงเหตุการณ์เหล่านั้นน่าสนใจ และในกรณีที่ค่าลิฟต์ มากหรือน้อยเกินไป อาจพิจารณาได้ว่า กฎนั้นไม่เป็นความจริง ซึ่งปกตินักวิเคราะห์ข้อมูลมักจะสนใจกฎที่มีค่า ลิฟต์ ในระดับที่สูง เนื่องจากทำให้หาความสัมพันธ์ได้ง่าย เช่นในกรณีข้างต้น จากกฎ ถ้ามีผู้ที่ซื้อผ้าอ้อมแล้วมักจะซื้อเบียร์ไปด้วย 50% แต่ถ้าสมมติเปลี่ยนจำนวนผู้ที่ซื้อเบียร์อย่างเดียวให้มีปริมาณ 60% ก็จะตีความหมายได้อีกแบบหนึ่งคือ ผู้ที่ซื้อเบียร์อย่างเดียวมีอยู่มากกว่าเพราะลูกค้าที่ซื้อเบียร์นั้นไม่จำเป็นต้องซื้อผ้าอ้อมไปด้วยก็ได้

3.2. การคัดเลือกกฎ

ผลที่ได้จากการทำงานของกฎความสัมพันธ์จะทำให้กฎที่เกิดจำนวนมากมาย หลากหลาย จึงต้องมีการกำจัดหรือตัดกฎที่ไม่น่าสนใจ หรือมีความน่าสนใจน้อยๆออกเพื่อเป็นการลดจำนวนกฎที่เกิดจากเหตุการณ์ที่มีโอกาสที่จะเกิดขึ้นน้อยออกไป ซึ่งสามารถทำได้โดยกำหนดค่า 2 ค่าคือ ค่า มินิมัมซัพพอร์ท (minimum support) และ ค่า มินิมัมคอนฟิเดนซ์ (minimum confidence) ซึ่งถ้าหากว่าค่าซัพพอร์ท หรือค่าคอนฟิเดนซ์ของกฎที่ได้นั้นต่ำกว่าค่าที่กำหนดไว้ ให้กำจัดออกได้ โดยไม่ต้องนำมาพิจารณา ทำให้เวลาที่ใช้ในการคำนวณหากฎความสัมพันธ์นั้นสั้นลง

ตัวอย่างเช่น การพิจารณากฎที่มีความสัมพันธ์จากรายการที่เกิดขึ้นทั้งหมด 1,000,000 รายการ ในการหากฎความสัมพันธ์นั้น ถ้ามีการกำหนดค่า มินิมัมซัพพอร์ท = 1% หมายความว่า กฎที่

จะนำมาพิจารณานั้น ทุกๆไอเท็มในไอเท็มเซต ต้องมีการเกิดของเหตุการณ์ อย่างน้อยที่สุด 10,000 ครั้ง ของจำนวนรายการของเหตุการณ์ที่เกิดขึ้นทั้งหมด และถ้าหากว่าไอเท็มใดมีการเกิดน้อยกว่า 10,000 ครั้ง ก็จะถูกตัดทิ้งไม่นำมาพิจารณา เพราะถือว่ามีความสำคัญน้อย

ในการทำงานจริงนั้น การกำหนดค่ามินิมัมซัพพอร์ท ขึ้นอยู่กับข้อมูลและสถานการณ์ ซึ่งสามารถปรับเปลี่ยนได้ในแต่ละระดับของการทำงานดังที่กล่าวไปตอนต้น เช่นถ้าหากระบุค่ามินิมัมซัพพอร์ท ให้มีค่าต่ำ จะทำให้รายการหรือเหตุการณ์ที่ไม่เกิดขึ้นบ่อยครั้งปรากฏออกมา หรือให้ทางกลับกัน ถ้าหากเรากำหนดค่ามินิมัมซัพพอร์ท ให้มีค่ามากขึ้น ก็จะปรากฏเหตุการณ์ที่เกิดขึ้นเป็นประจำๆเท่านั้น

3.3. ข้อดีและข้อเสียของ Association Rules

ข้อดี

- ผู้ใช้สามารถควบคุมจำนวนผลลัพธ์ได้ โดยระบุค่ามินิมัมซัพพอร์ท และมินิมัมคอนฟิเดนซ์
- สามารถทำงานได้ดี กับข้อมูลขนาด
- ในกรณีที่มีข้อมูลไม่สมบูรณ์ ก็สามารถทำการไมน์นิ่งกับข้อมูลบางส่วนได้
- ไม่จำเป็นต้องระบุขอบเขตของกลุ่มข้อมูล
- สามารถจัดเก็บข้อมูลที่อยู่ในรูปแบบที่ต่างกันได้
- มีการแสดงผลด้วยสัญลักษณ์ ทำให้ง่ายต่อการทำความเข้าใจกว่าผลลัพธ์ที่ได้จากเทคนิคอื่นๆ

ข้อเสีย

- ในการกำหนดค่ามินิมัมซัพพอร์ท และมินิมัมคอนฟิเดนซ์ เพื่อจำกัดจำนวนของกฎที่ถูกสร้างขึ้นจำนวนมากนั้น อาจทำให้กฎที่ได้เกิดความผิดเพี้ยนจากความเป็นจริง เนื่องจากผู้ใช้กำหนดค่าสูงหรือต่ำจนเกินไป
- กฎที่ได้มานั้น อาจเป็นกฎที่เกิดขึ้นบ่อยๆ และกฎที่เกิดขึ้นจากความบังเอิญ ทำให้มีความยากในการบอกความแตกต่างของกฎที่ได้มาค่อนข้างยาก
- กฎที่ได้มาสามารถบอกได้เพียงแค่แนวโน้มที่จะเกิดขึ้นด้วยกัน ไม่ได้บอกเรื่องของความเป็นเหตุเป็นผลของกฎซึ่งต้องอาศัยประสบการณ์ในการหาเหตุผลเอาเอง

3.4. อัลกอริทึมอพริออริ (Han and Fu. 1999)

ตัวอย่างการค้นหากฎความสัมพันธ์ที่เป็นที่นิยมนำมาเป็นต้นแบบในการวิเคราะห์คือการทำงานของอัลกอริทึมอพริออริ(Algorithm Apriori) ซึ่งเป็นอัลกอริทึมพื้นฐานในยุคเริ่มต้นของการทำการพัฒนาทางด้านดาต้าไมน์นิ่ง

มีหลักการทำงานอยู่ 2 ขั้นตอน คือ

- การหาชุดของข้อมูล หรือ ไอเท็มเซต (item set) ที่เกิดขึ้นในรายการซื้อ ว่ามีค่าของความถี่ที่เกิดขึ้นจากการขายทั้งหมด ที่ไม่น้อยกว่ามินิมัมซัพพอร์ทที่กำหนดเอาไว้
- การสร้างกฎความสัมพันธ์โดยนำ L_k ไอเท็มเซต (Frequent itemset) ที่ได้มาสร้างบนพื้นฐานที่ว่า กฎที่ได้นั้นจะต้องมีค่าคอนฟิเดนซ์ ไม่น้อยกว่าค่ามินิมัมคอนฟิเดนซ์

$L_1 = \{frequent\ 1\text{-}itemsets\}$
For $(k = 2; L_{k-1} \neq \phi\ ; k++)$ do begin
  C_k = apriori- gen $(L_{k-1}, \text{min\_sup})$;
  forall transaction $t \in D$ do begin
    C_t = subset (C_k, t)
    forall candidates $c \in C_t$ do
      $c.count++$;
    end
  end
  $L_k = \{c \in C_k | c.count \geq minsup\}$
end
$Answer = \bigcup_k L_k$;

ภาพที่ 3.2 อัลกอริทึมอพริออริ

insert into C_k
select $p.item_1, p.item_2, p.item_{k-1}, q.item_{k-1}$
from $L_{k-1}p, L_{k-1}q$
where $p.item_1 = q.item_1, \ldots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$

ภาพที่ 3.3 การหาสมาชิกของ C_k

```
forall itemsets c ∈ C_k do
    forall (k-1)-subsets s of c do
        if (s ∉ L_{k-1}) then
            delete c from C_k
```

ภาพที่ 3.4 อัลกอริทึมอพริออริเจนที่ใช้หาสมาชิก Ck (Prune Step)

3.4.1 สัญลักษณ์ที่ใช้ในอัลกอริทึมอพริออริ

ตัวแปรที่จะต้องพิจารณา คือ

D	คือฐานข้อมูลที่เก็บรายการของทรานเซคชั่นทั้งหมด ที่เก็บ <TID,items>
TID	คือ ตัวเลขที่ใช้ระบุแต่ละรายการของทรานเซคชั่น
K-itemset	คือ เซตของข้อมูลที่มีจำนวนสมาชิกจำนวน k ตัว
Lk	คือ เซตของ K-ไอเท็มเซต ที่เกิดขึ้นบ่อย (Frequent K-itemset) คือ แต่ละเซตจะมีไอเท็ม เป็นสมาชิกทั้งหมด k ตัวและทุกเซตจะต้องมีความถี่ในการเกิด มากกว่าหรือเท่ากับค่ามินิมัมซัพพอร์ท และมินิมัมคอนฟิเดนซ์ ซึ่งสมาชิกในเซตจะประกอบด้วย 2 fields คือ ไอเท็มเซต และ ค่าซัพพอร์ท
Ck	คือ เซตของ K-ไอเท็มเซต ที่คัดไว้ (Candidate K-itemset) ซึ่งเป็นเซตที่ถูกเลือกมาจาก Lk

3.4.2 การทำงานของอัลกอริทึมอพริออริ

การทำงานของอัลกอริทึม เริ่มจากการหา Lk ไอเท็มเซต หรือ L1 โดยการนับค่าความถี่ของเหตุการณ์หรือไมเท็มแต่ละตัวที่เกิดขึ้นในฐานข้อมูลจากแต่ละทรานเซคชั่น แล้วทำการเลือกเฉพาะ ไอเท็ม ที่มีค่าซัพพอร์ทมากกว่าค่ามินิมัม ซัพพอร์ทที่กำหนดไว้จากนั้นจึงเข้าสู่กระบวนการทำงานเป็นรอบการทำงานขั้นตอนนี้ของอัลกอริทึมจะทำการสร้าง Ck ขึ้น ดังภาพที่ 3.3 ตามและมีการคัดตัว Ck บางตัวออกตามอัลกอริทึมอพริออริเจน (algorithm apriori-gen) ดังภาพที่ 3.4 ซึ่งมีขั้นตอนการทำงานดังนี้

ขั้นที่ 1 สร้าง C_k (Join Step)

L_{k-1} จะนำไปใช้สร้าง C_k ไอเท็มเซต โดยการใช้อัลกอริทึมอพริออริเจนโดยการนำแต่ละ ไอเท็มเซต ของ L_{k-1} มาทำการ Join กัน

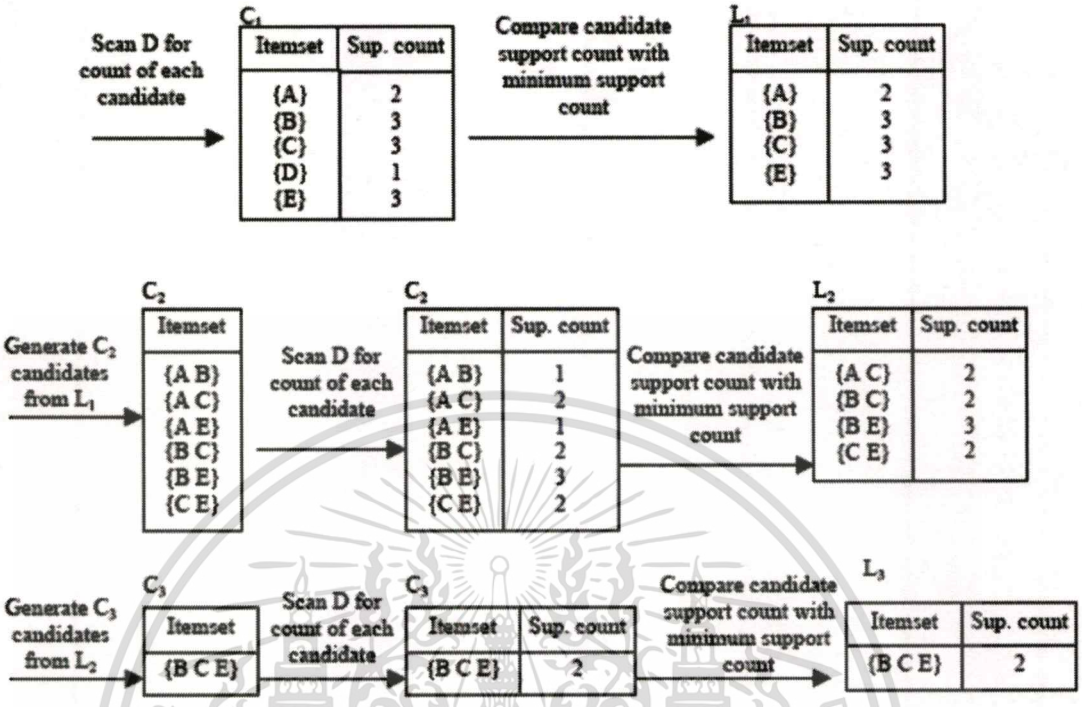
ขั้นที่ 2 คัดเลือก C_k (Prune Step)

นำ C_k ที่สร้างมาทำการกำจัดออกบางส่วนจากการคัดตัวไอเท็มเซต ของ k-1 ที่ไม่ได้อยู่ใน L_{k-1}

หลังจากอัลกอริทึมนับค่าซัพพอร์ท ที่เกิดขึ้นในฐานข้อมูลแต่ละ k-1 ไอเท็มเซต ใน C_k โดยเลือกมาเฉพาะ k-1 ไอเท็มเซต ที่มีค่ามากกว่าค่า มินิมัม ซัพพอร์ท มาสร้างเป็น L_k เรียบร้อยแล้ว ก็จะวนรอบโดยทำการเพิ่มค่า k ขึ้น 1 ค่า และทำตามขั้นตอนเดิมใน k-Loop ต่อไปจนไม่สามารถหาไอเท็มเซต L_k ได้ต่ออีกแล้ว จึงสิ้นสุดการทำไมน์นิ่ง อธิบายได้จากตัวอย่าง ต่อไปนี้

ตารางที่ 3.1 ตัวอย่างฐานข้อมูล (D) ที่ใช้ในการทำอัลกอริทึมอพริออริ

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



ภาพที่ 3.5 การสร้าง C_k และ L_k

จากภาพ เป็นข้อมูลของทรานเซคชั่นของการซื้อสินค้าที่เกิดขึ้นในฐานข้อมูล ซึ่งสามารถใช้อัลกอริทึมอพริออริหา L_k ไอเท็มเซต ใน D ได้ตามขั้นตอนดังต่อไปนี้

1.ขั้นแรกเพียงแค่ นำข้อมูลจากฐานข้อมูลมานับจำนวนของไอเท็ม แต่ละรายการจาก ทรานเซคชั่นทั้งหมด จะได้ C_1 ออกมา

2.ให้พิจารณาค่าความถี่ หรือค่าซัพพอร์ท ที่เกิดขึ้นของแต่ละไอเท็มเซต ซึ่งถ้ามีค่าซัพพอร์ทต่ำกว่าค่ามินิมัมซัพพอร์ท ซึ่งสมมติให้มีการกำหนดไว้เท่ากับ 50% นั่นก็คือ ถ้านับได้น้อยกว่า 2 ทรานเซคชั่นก็จะถูกกำจัดออก และนำค่าที่มากกว่า มาสร้างเป็น L_1

3. ค้นหาไอเท็มเซต ของ L_2 โดยขั้นตอนการหา C_k ของอัลกอริทึมอพริออริเจน โดยทำการนำ L_1 กับ L_1 มารวมกันและผ่านขั้นตอนคัดเลือก C_k ซึ่งจะได้ C_2 ออกมา

4.จากนั้นกำจัดบางค่าใน C_2 ที่มีค่า ซัพพอร์ท น้อยกว่าค่า มินิมัม ซัพพอร์ท ออก ซึ่งจะได้ L_2 ออกมา

5. สร้าง L_3 ขึ้นมา โดยนำ L_2 มารวมเข้าด้วยกัน โดยทำการพิจารณาจากคุณสมบัติของอัลกอริทึมอพริออริที่ว่าซับเซตทั้งหมดของแต่ละไอเท็มเซต ของ C_k นั้นจะต้องปรากฏอยู่ใน L_2 ทั้งหมด เช่นสับเซตของ C_3 ทั้งหมดที่ได้มาจากการ Join ต้องปรากฏอยู่ใน ไอเท็มเซตในตาราง L_2 ด้วย แต่ถ้าไม่ปรากฏให้ทำการกำจัด ไอเท็มเซต นั้นออก

6. นับค่าซัพพอร์ทของแต่ละไอเท็มเซต ใน C_3 จาก *D*

7. L_3 จะได้จากไอเท็มเซตทั้งหมดของ C_3 ที่มีค่าซัพพอร์ทมากกว่าค่ามินิมัมซัพพอร์ทที่กำหนดไว้

8. สุดท้ายแล้วเราได้ L_3 เป็นไอเท็มเซตเพียงเซตเดียว ไม่สามารถทำการคำนวณหา C_4 ได้ จึงถือว่าสิ้นสุดกระบวนการไมน์นิ่งที่ L_3 นี้

3.4.3 การสร้างกฎความสัมพันธ์

การสร้างกฎความสัมพันธ์ สามารถทำได้โดยนำ L_k ไอเท็มเซต ที่ได้จากอัลกอริทึมอพริออริในหัวข้อ 3.4.2 โดยนำค่า L_k ไอเท็มเซต ตั้งแต่ L_2 มาคำนวณหา สับเซตโดยนำสับเซตเหล่านั้นมาสร้างเป็นกฎความสัมพันธ์ ได้ดังนี้

ตารางที่ 3.2 ผลลัพธ์ของ L_k ไอเท็มเซต

L_k	Support
A C	50%
B C	50%
B E	75%
C E	50%
B C E	50%

จากผลที่ได้สามารถนำมาสร้างเป็นกฎได้ดังนี้

L2-Itemsets : 4 Frequents

A C : total 2 transactions (50%)

Rule A => C [50, 100] : lift factor = 1.33

Rule C => A [50, 67] : lift factor = 1.33

B C : total 2 transactions (50%)

Rule B => C [50, 67] : lift factor = 0.89

Rule C => B [50, 67] : lift factor = 0.89

B E : total 3 transactions (75%)

Rule B => E [75, 100] : lift factor = 1.33

Rule E => B [75, 100] : lift factor = 1.33

C E : total 2 transactions (50%)

Rule C => E [50, 67] : lift factor = 0.89

Rule E => C [50, 67] : lift factor = 0.89

B C E : total 2 transactions (50%)

Rule B => C, E [50, 67] : lift factor = 1.33

Rule C => B, E [50, 67] : lift factor = 0.89

Rule E => B, C [50, 67] : lift factor = 1.33

Rule B, C => E [50, 100] : lift factor = 1.33

Rule B, E => C [50, 67] : lift factor = 0.89

Rule C, E => B [50, 100] : lift factor = 1.33

ซึ่งถ้าพิจารณาตามความน่าเชื่อถือของกฎแล้ว ควรตัดกฎความสัมพันธ์ที่มีค่า ลิฟต์ factor น้อยกว่า 1 ออกไปเพราะถือว่าเป็นกฎความสัมพันธ์ที่ไม่น่าสนใจ ดังที่ได้อธิบายมาแล้วข้างต้น

บทที่ 4

การค้นหากฎความสัมพันธ์จากข้อมูลแบบหลายลำดับชั้น

ในอัลกอริทึมอพริออรินั้นพิจารณาจากไอเท็มเพียงอย่างเดียว ทำให้ได้กฎความสัมพันธ์ ที่มีความกระจัดกระจายสูง และอาจหลุดข้อเท็จจริงที่สำคัญๆไปได้ ซึ่งถ้าพิจารณาต่อไปถึงกลุ่มของข้อมูลก็จะทำให้ได้ข้อเท็จจริงที่ละเอียดและตรงประเด็นมากขึ้น เช่น ถ้ารายการสินค้าประเภทนมแบ่งเป็น นมรสจืด, รสหวานหรือแม้กระทั่งนมพร่องมันเนย ก็จัดเป็นสินค้าประเภทนมเหมือนกัน ถ้านำมาพิจารณาร่วมในการคำนวณด้วย จะได้กฎความสัมพันธ์ที่มีความน่าสนใจมากขึ้น

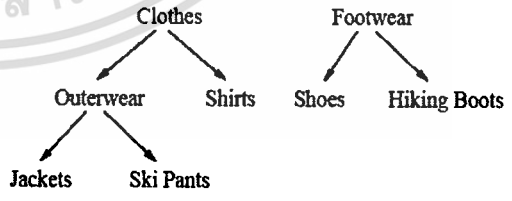
4.1. การค้นหากฎความสัมพันธ์จากข้อมูลแบบหลายลำดับชั้น (Srikant and Agrawal. 1995)

ในการค้นหากฎความสัมพันธ์แบบหลายลำดับชั้นนั้น ค่าซัพพอร์ท และคอนฟิเดนซ์ จำเป็นจะต้องมีค่าที่มากกว่าหรือเท่ากับค่า มินิมัมซัพพอร์ท และ มินิมัมคอนฟิเดนซ์ ที่ผู้ใช้ได้กำหนดไว้ ซึ่งการทำงานของกฎความสัมพันธ์ของข้อมูลแบบหลายลำดับชั้นนี้อยู่บนพื้นฐานของอัลกอริทึมอพริออริ

โดยปกติข้อมูลแต่ละทรานเซคชั่นที่เก็บไว้ในฐานข้อมูล (ภาพที่ 4.1) จะเกิดขึ้นจากเซตของไอเท็ม ซึ่งมีการแบ่งหมวดหมู่ (Taxonomy) เอาไว้เป็นลำดับชั้นชั้น (Hierarchy) ดังแสดงในภาพที่ 4.2

Transaction	Items Bought
100	Shirt
200	Jacket, Hiking Boots
300	Ski Pants, Hiking Boots
400	Shoes
500	Shoes
600	Jacket

ภาพที่ 4.1 ฐานข้อมูล D



ภาพที่ 4.2 กราฟแบ่งหมวดหมู่

จากฐานข้อมูล *D* เมื่อนำไปหาความสัมพันธ์ โดยอัลกอริทึมอพริออริซึ่งมีค่ามินิมัมซัพพอร์ท = 2 จะเห็นได้ว่าไม่สามารถสามารถหาความสัมพันธ์ของไอเท็มได้ เนื่องจากทรานเซคชั่นในฐานข้อมูล *D* นั้น มีเพียงทรานเซคชั่นที่ 100 และ 200 ที่เกิดความสัมพันธ์ระหว่าง 2 ไอเท็ม ขึ้นคือ {Jacket, Hiking Boots} และ {Ski Pants, Hiking Boots} ตามลำดับ ซึ่งแต่ละไอเท็มเซต จะมีค่า ซัพพอร์ท เพียงแค่ 1 เท่านั้น จึงไม่ถึงค่า มินิมัม ซัพพอร์ท ที่กำหนดเอาไว้ ทำให้กฎเหล่านี้ถูกกำจัดออกไป

แต่เมื่อพิจารณาจาก กราฟแบ่งหมวดหมู่ จะเห็นได้ว่าทั้ง Jackets และ Ski Pants ต่างก็เป็นไอเท็ม ชนิดเดียวกันคือ Outerwear กล่าวคือ ทั้ง Jackets และ Ski Pants ต่างก็เป็นลูกของ Outerwear และ Outerwear ก็เป็นบรรพบุรุษ (ancestor) ของทั้ง Jackets และ Ski Pants และกล่าวต่อไปได้อีกว่าในเมื่อ Outerwear มี Clothes เป็น Parent เพราะฉะนั้นทั้ง Jackets และ Ski Pants ก็จะต้องมี Clothes เป็นบรรพบุรุษของ ไอเท็ม ทั้ง 2 ด้วย เป็นการสืบทอด

จึงทำให้กฎที่บอกว่า

"เมื่อลูกค้าซื้อ Jacket แล้วจะซื้อ Hiking Boot ด้วย" (Jacket → Hiking Boots)

และ กฎที่ว่า

"เมื่อลูกค้าซื้อ Outerwear แล้วจะซื้อ Hiking Boots ด้วย" (Outerwear → Hiking Boots)

มีความหมายเดียวกัน จึงสามารถสรุปได้ว่าทรานเซคชั่นที่ 100 และ 200 นั้น เกิดความสัมพันธ์ (Outerwear → Hiking Boots) ในลำดับชั้นบนที่เหมือนกัน เมื่อพิจารณาในลำดับชั้นบนจะเห็นได้ว่าค่า ซัพพอร์ท ในฐานข้อมูล *D* มีค่าเท่ากับ 2 ซึ่งเท่ากับค่ามินิมัมซัพพอร์ท จึงทำให้สามารถหาความสัมพันธ์ในฐานข้อมูล *D* นี้ได้

4.2 อัลกอริทึมเบสิค (Algorithms Basic)

ในการค้นหากฎความสัมพันธ์แบบหลายลำดับชั้นนั้นค่าซัพพอร์ท และคอนฟิเดนซ์ จำเป็นจะต้องมีเหมือนเดิมเช่นที่มีในอพริออริอัลกอริทึม คือต้องมีค่าที่มากกว่าหรือเท่ากับค่ามินิมัมซัพพอร์ทและมินิมัมคอนฟิเดนซ์ที่ผู้ใช้งานได้กำหนดไว้ ซึ่งการทำงานอัลกอริทึมจะมีการทำงานอยู่บนพื้นฐานของอพริออริอัลกอริทึม และได้มีการเพิ่มบรรพบุรุษ (ancestor) ของ ไอเท็ม จากกราฟแบ่งหมวดหมู่ที่เตรียมไว้ของแต่ละไอเท็ม เข้าไปในทุกทรานเซคชั่นของฐานข้อมูล

การทำงานในรอบแรก (1st-Pass) นั้นเพียงแค่หา L_1 ไอเท็มเซต โดยทำการนับความถี่ (Support) ที่เกิดขึ้นของแต่ละไอเท็ม และนับรวมไปถึงบรรพบุรุษของแต่ละไอเท็มในแต่ละทรานเซคชั่นซึ่งทุกไอเท็มในไอเท็มเซต นั้นสามารถที่จะนำมาจากทุกลำดับชั้น (level) ของกราฟแบ่งหมวดหมู่ได้ทั้งหมด ไม่จำเป็นที่จะต้องนำมาจากลำดับล่าง (Low level) เพียงระดับเดียว

ในส่วนถัดมา จะเป็นการทำงานของ K รอบซึ่งประกอบด้วย 2 ขั้นตอนการทำงานคือ

1. ใช้อัลกอริทึมอพริออริเจน (Apriori Candidate Generations) สร้าง L_{k-1} ไอเท็มเซตในรอบการทำงาน (k-1)th ซึ่งได้มาจาก C_k ไอเท็มเซตโดยที่แต่ละไอเท็มใน C_k นั้นสามารถที่จะนำมาได้จากทุกลำดับชั้นใน กราฟจัดหมวดหมู่

3. นับค่าความถี่ของแต่ละไอเท็มเซต C_k โดยนับจากแต่ละทรานเซคชั่นของฐานข้อมูลที่เพิ่มตัวบรรพบุรุษเข้าไปด้วย จากนั้นให้ทำการกำจัด C_k บางไอเท็มเซตออก จึงได้ L_k ออกมาเช่นเดียวกันกับอัลกอริทึมอพริออริ

```
L1 = {frequent 1-itemsets}
For ( k = 2; Lk-1 ≠ φ ; k++) do begin
    Ck = apriori-gen (Lk-1, min_sup );
    forall transaction t ∈ D do begin
        t = add - ancestor(t, T)
        Ct = subset (Ck, t)
        forall candidates c ∈ Ct do
            c.count++;
        end
    end
    Lk = { c ∈ Ck | c.count ≥ minsup}
end
Answer = ⋃k Lk;
```

ภาพที่ 4.3 อัลกอริทึมเบสิค

4.3 อัลกอริทึมคิวมูเลท (Algorithm Cumulate)

จากอัลกอริทึมเบสิคนั้นสามารถพัฒนา โดยการเพิ่มประสิทธิภาพการคำนวณเข้าไปในขั้นตอนการทำงานของอัลกอริทึมทำให้ใช้เวลาในการทำงานสั้นลง โดยการเพิ่มประสิทธิภาพที่ว่ามีอยู่ 3 ขั้นตอนดังนี้

4.3.1 การหาบรรพบุรุษของข้อมูลเพื่อเพิ่มในทรานเซคชั่น (Filtering the ancestor added to transaction)

ในการทำงานของคิวมูเลทอัลกอริทึมเพียงแค่ทำการเพิ่มบรรพบุรุษของ ไอเท็ม ที่ปรากฏอยู่ในไอเท็มเซต ของ C_k ที่กำลังนับอยู่ในรอบการทำงานปัจจุบัน แทนการเพิ่มทุกบรรพบุรุษ ของแต่ละไอเท็ม ในแต่ละทรานเซคชั่นและถ้าไอเท็ม ไม่ได้อยู่ใน ไอเท็มเซตของ C_k ก็ให้กำจัดออกจากทรานเซคชั่น

4.3.2 การเตรียมบรรพบุรุษของข้อมูลก่อนการประมวลผล (Pre-computing ancestors)

ทำกระบวนการการเตรียมการประมวลผลโดยการหาบรรพบุรุษของแต่ละไอเท็ม สร้างเซตของบรรพบุรุษจากการสำรวจ กราฟแบ่งหมวดหมู่หลังจากนั้นให้ทำการกำจัด บรรพบุรุษที่ไม่ได้อยู่ใน C_k ไอเท็มเซต ออกไปก่อนที่จะตรวจสอบฐานข้อมูล และเพิ่มข้อมูลในแต่ละทรานเซคชั่นทำให้ลดขั้นตอนลงได้บางส่วน

4.3.3 ลบไอเท็มเซตที่มีตัวไอเท็มและบรรพบุรุษอยู่ด้วยกัน (Pruning Itemsets containing an item and it ancestor)

เมื่อพิจารณาค่าซัพพอร์ทของ ไอเท็มเซต X ซึ่งประกอบด้วย ไอเท็ม x และ บรรพบุรุษของ ไอเท็ม x($\hat{x}$) จะมีค่า ซัพพอร์ท เท่ากับ ไอเท็มเซต ที่ประกอบด้วย ไอเท็มเซต $X - \hat{x}$ เนื่องจาก $X - \hat{x} \subset X$

ดังนั้นในการค้นหา C_k จะไม่ทำการนับค่า ไอเท็มเซต ที่มี ไอเท็ม x และบรรพบุรุษของ x เข้าไปด้วย ($\hat{x}$) ทำให้ลดขั้นตอนการประมวลผลได้น้อยลง

Compute T^*, the set of ancestors of each item, *from* T // Optimiz 2
L_1 = *{frequent 1-itemsets}*
For *(* $k = 2$; $L_{k-1} \neq \phi$; $k++$*)* do begin
    C_k = apriori- gen *(*L_{k-1}, min_sup *)*;
    if (k = 2) then prune(C_2) // Optimiz 3
    T^* = remove - unnecessary(T^*, C_k) // Optimiz 1
    forall transaction $t \in D$ do begin
        t = add - ancestor(t, T^*)
        C_t = subset *(*C_k, t*)*
        forall candidates $c \in C_t$ do
            $c.count++$;
        end
    end
    $L_k = \{ c \in C_k | c.count \geq minsup \}$
end
$Answer = \bigcup_k L_k$;

ภาพที่ 4.4 อัลกอริทึมคิวมูเลท

4.4 การทำงานของอัลกอริทึมคิวมูเลท

จากภาพที่ 4.2 กราฟแบ่งหมวดหมู่สามารถสร้างตาราง T* ได้ดังนี้

ตารางที่ 4.1 ตาราง T* ที่สร้างมาจากกราฟบางหมวดหมู่

Item	Ancestors Set
Shirt	{Clothes}
Jacket	{Outerwear, Clothes}
Hiking Boots	{Footwear}
Ski Pants	{Outerwear, Clothes}
Shoes	{Footwear}

เมื่อมีการเริ่มประมวลผลอ่านฐานข้อมูลจะมีการเพิ่ม บรรพบุรุษ ของแต่ละไอเท็มในทรานเซคชั่นเข้าในการคำนวณด้วย ดังภาพ

ตารางที่ 4.2 ฐานข้อมูล *D*

Transaction	Items Bought
100	Shirt
200	Jacket, Hiking Boots
300	Ski Pants, Hiking Boots
400	Shoes
500	Shoes
600	Jacket

↓

T100	Shirt, ***Clothes***
T200	Jacket, Hiking Boots, ***Outwear, Clothes, Footwear***
T300	Ski Pants, Hiking Boots, ***Outerwear, Clothes, Footwear***
T400	Shoes, ***Footwear***
T500	Shoes, ***Footwear***
T600	Jacket, ***Outwear, Clothes***

ภาพที่ 4.5 แสดงการเพิ่มบรรพบุรุษของไอเท็ม ลงในแต่ละทรานเซคชั่น

ตารางที่ 4.3 ตาราง C_1

Itemset	Sup.count
{Shirt}	1
{Jacket}	2
{Hiking Boots}	2
{Ski Pants}	1
{Shoes}	2
{Clothes}	4
{Outerwear}	3
{Footwear}	4

→

ตารางที่ 4.4 ตาราง L_1

Itemset	Sup.count
{Jacket}	2
{Hiking Boots}	2
{Shoes}	2
{Clothes}	4
{Outerwear}	3
{Footwear}	4

ตารางที่ 4.5 ตาราง C_2

Itemset
{Jacket, Hiking Boots}
{Jacket, Shoes}
{Jacket, Footwear}
{Hiking Boots, Clothes}
{Hiking Boots, Outerwear}
{Shoes, Clothes}
{Shoes, Outerwear}
{Clothes, Footwear}
{Outerwear, Footwear}

ตารางที่ 4.6 ตาราง C_2

Itemset	Support
{Jacket, Hiking Boots}	1
{Jacket, Shoes}	0
{Jacket, Footwear}	1
{Hiking Boots, Clothes}	2
{Hiking Boots, Outerwear}	2
{Shoes, Clothes}	0
{Shoes, Outerwear}	0
{Clothes, Footwear}	2
{Outerwear, Footwear}	2

ตารางที่ 4.7 ตาราง L_2

Itemset	Support
{Hiking Boots, Clothes}	2
{Hiking Boots, Outerwear}	2
{Clothes, Footwear}	2
{Outerwear, Footwear}	2

ตารางที่ 4.8 ตาราง $\cup_k L_k$

Itemset	Sup.count
{Jacket}	2
{Hiking Boots}	2
{Shoes}	2
{Clothes}	4
{Outerwear}	3
{Footwear}	4
{Hiking Boots, Clothes}	2
{Hiking Boots, Outerwear}	2
{Clothes, Footwear}	2
{Outerwear, Footwear}	2

ตารางที่ 4.9 ตารางกฎความสัมพันธ์ที่ได้จากตาราง $\cup_k L_k$

Rule	Support	Conf.
Hiking Boots --> Clothes	33%	100%
Hiking Boots --> Outerwear	33%	100%
Clothes --> Footwear	33%	50%
Outerwear --> Footwear	33%	66%
Clothes --> Hiking Boots	33%	50%
Outerwear --> Hiking Boots	33%	66%
Footwear -->Clothes	33%	50%
Footwear -->Outerwear	33%	50%

ตารางที่ 4.3 ถึงตารางที่ 4.8 เป็นการทำงานของอัลกอริทึมคิวมูเลทในการหาค้นหากฎความสัมพันธ์ของสินค้าจากทรานเซคชั่นของการซื้อสินค้าที่จัดเก็บไว้ในฐานข้อมูล ซึ่งมีการทำงานอยู่บนพื้นฐานของอัลกอริทึมอพริออริ สามารถที่จะหา L_k ไอเท็มเซต และความสัมพันธ์ของสินค้าใน D ได้ตามขั้นตอนดังต่อไปนี้

1. ขั้นแรกสร้างตารางของไอเท็มและบรรพบุรุษ จาก กราฟแบ่งหมวดหมู่

2. ทำการอ่านข้อมูลจากฐานข้อมูล D มาแต่ละทรานเซคชั่นและพิจารณาว่า แต่ละไอเท็มมีบรรพบุรุษหรือว่าอยู่ในหมวดหมู่อะไรบ้าง จากตาราง T* และทำการเพิ่มบรรพบุรุษของแต่ละ ไอเท็ม เข้าไปในทรานเซคชั่นโดยมีเงื่อนไขที่ว่า แต่ละทรานเซคชั่นนั้นสามารถที่จะมี บรรพบุรุษ ชนิดเดียวกันได้เพียงจำนวนหนึ่งตัวเท่านั้น ถ้ามี บรรพบุรุษ เกิดขึ้นซ้ำกันจำนวนหลายตัว ให้ทำการกำจัด บรรพบุรุษ ที่เหลือออก

3. นับจำนวนความถี่ของไอเท็มที่เกิดขึ้น จากแต่ละทรานเซคชั่น ให้พิจารณาค่าความถี่ หรือ ค่าซัพพอร์ท ที่เกิดขึ้นของแต่ละไอเท็ม ซึ่งขั้นตอนนี้จะเหมือนอัลกอริทึมอพริออริ แตกต่างกันเพียงที่อัลกอริทึมคิวมูเลทนั้นมีการให้บรรพบุรุษ ของแต่ละ ไอเท็ม มารวมอยู่ในแต่ละ ไอเท็มเซต ในตาราง C_1 ได้ ถ้า ไอเท็มเซต ใด ในตาราง C_1 นั้นมีค่า ซัพพอร์ท ที่ต่ำกว่าค่า มินิมัม ซัพพอร์ท ซึ่งกำหนดไว้เท่ากับ 2 ไอเท็มเซต นั้นจะถูกกำจัดออกไป โดยจะนำเฉพาะไอเท็มเซต ที่มีค่า ซัพพอร์ทมากกว่าค่า มินิมัม ซัพพอร์ท มาสร้างเป็นตาราง L_1

4. เข้าสู่การประมวลผลตั้งแต่การสร้างตาราง L_k ซึ่งมีการกำหนดค่าเริ่มต้นคือ k=2 โดยมีเงื่อนไขกำหนดไว้ว่า ถ้าหากว่า $L_K = L_2$ จะต้องเข้าสู่ขั้นตอนการสร้าง C_k ซึ่งมีการหลัก

การทำงานอยู่บนอัลกอริทึมอพริออริเจน โดยจะทำการรวมระหว่าง L_k กับ L_k ก็คือ L_1 กับ L_1 และผ่านขั้นตอนการคัดตัว C_k ออกซึ่งจะได้ C_2 ออกมาและเนื่องจากค่าเริ่มต้นคือ k=2 จึงเข้าขั้นตอนการทำงานของอัลกอริทึมคิวมูเลทโดยจะทำการลบ C_2 บางตัวที่ในไอเท็มเซตของ C_k นั้นประกอบด้วยไอเท็มและบรรพบุรุษของตัวไอเท็ม เองออก จาก C_2 โดยถ้าหาก $k > 2$ ก็ไม่จำเป็นที่จะต้องทำซ้ำในขั้นตอนนี้อีก เนื่องจากหลังจากที่ทำขั้นตอนที่ k=2 แล้วจะไม่มี ไอเท็มเซต และ บรรพบุรุษ ของตัว ไอเท็มเซต เอง ปรากฏอยู่ใน C_k ที่ $k > 2$ อีกต่อไป หลังจากนั้นให้ทำการลบบาง บรรพบุรุษ ใน T* ที่ไม่ได้อยู่ใน ไอเท็มเซต ของ C_2 ออก

5. ทำการนับค่าซัพพอร์ทของแต่ละไอเท็มเซต ของ C_2 โดยมีวิธีการทำงานเหมือนกับอัลกอริทึมอพริออริโดยค่าซัพพอร์ทที่ได้นั้น เกิดจากจำนวนไอเท็มเซตและบรรพบุรุษ ที่ปรากฏอยู่ในตาราง T* ในแต่ละทรานเซคชั่นจากนั้นให้กำจัด C_2 บางตัวที่มีค่า ซัพพอร์ท น้อยกว่าค่า มินิมัม ซัพพอร์ท ออก ซึ่งจะได้ L_2 ออกมา

6. ทำการวนรอบไปที่การทำงาน K รอบการทำงานของ C_k ในข้อ 4 ใหม่อีกครั้ง ซึ่งจะเพิ่มค่า k ขึ้นหนึ่งค่า ซึ่งในตอนนี้เท่ากับ 3

และทำตามขั้นตอนเดิม และทำการสร้างตาราง L_3 ขึ้นมาโดยนำแต่ละ ไอเท็มเซตของตาราง L_2 มา Join เข้าด้วยกัน หลังจากนั้นให้พิจารณาจากพื้นฐานคุณสมบัติของอพริออริที่ว่า สับเซต ทั้งหมดของแต่ละไอเท็มเซต ของ C_k นั้นจะต้องปรากฏอยู่ใน L_2 ทั้งหมด เช่น สับเซต ของ C_3 ทั้งหมดที่ได้มาจากการ Join ต้องปรากฏอยู่ในไอเท็มเซต ในตาราง L_2 ทั้งหมดด้วย ซึ่งถ้าหากไม่ปรากฏอยู่ใน L_2 ทั้งหมดแล้วละก็ให้ทำการกำจัดไอเท็มเซตนั้นออกไป หลังจากได้ C_3 แล้ว ก็ให้ทำการวนรอบทำงานต่อไปจนกว่า $C_k = 0$ ซึ่งจะทำให้ $L_k = 0$ ด้วย ซึ่งจะตรงกับเงื่อนไขที่จะต้องออกจาก k-Loop ของอัลกอริทึม และให้ไปทำเงื่อนไขสุดท้าย คือให้แสดงผลลัพธ์ของทุก L_k ไอเท็มเซตนั้นเอง

4.4.1 การสร้างกฎความสัมพันธ์

การสร้างกฎความสัมพันธ์ สามารถทำได้โดยนำทุกๆ L_k ไอเท็มเซตที่ได้จากการทำงานของอัลกอริทึมคิวมูเลทในหัวข้อ 4.3 โดยนำทุกๆ L_k ไอเท็มเซต ตั้งแต่ตาราง L_2 มาคำนวณหาค่า ซัพพอร์ท และค่าคอนฟิเดนซ์และนำมาแสดงผลเป็นกฎความสัมพันธ์ ในตารางที่ 4.9

สังเกตุได้ว่าผลลัพธ์ที่ได้ ไม่มีกฎความสัมพันธ์ ที่เกิดจากข้อมูลระดับล่างเพียงระดับเดียวแต่อย่างใดซึ่งถ้าทำการพิจารณาเดียงแค่ความสัมพันธ์ระดับเดียว ก็จะไม่ได้ผลลัพธ์ที่น่าสนใจจากการทำไมน์นิ่งครั้งนี้

ถ้ามีการตั้งค่า มินิมัมคอนฟิเดนซ์ ไว้ที่ 60% ต้องลบไอเท็มเซตที่มีค่าคอนฟิเดนซ์น้อยกว่า 60% ออกดังจะเห็นได้ในตารางที่ 4.10 ดังต่อไปนี้

ตารางที่ 4.10 ตารางกฎความสัมพันธ์ที่ได้จาก $\cup_k L_k$ ที่ผ่านค่า มินิมัมคอนฟิเดนซ์

Rule	Support	Conf.
Hiking Boots --> Clothes	33%	100%
Hiking Boots --> Outerwear	33%	100%
Outerwear --> Footwear	33%	66%
Outerwear --> Hiking Boots	33%	66%

เห็นได้ว่าผลที่ได้ออกมานั้นมีการแสดงความสัมพันธ์ของแต่ละไอเท็ม ในลักษณะความสัมพันธ์แบบหลายลำดับชั้นมากขึ้น และกฎความสัมพันธ์บางกฎที่ไม่สามารถปรากฏในการค้นหาความสัมพันธ์แบบลำดับชั้นเดียวได้นั้น จะสามารถปรากฏออกมาให้ทำการวิเคราะห์หาความสัมพันธ์ได้

บทที่ 5

การวิเคราะห์และออกแบบระบบ

โครงงานนี้ได้นำเอาอัลกอริทึมคิวมูเลทมาใช้ เพื่อค้นหากฎความสัมพันธ์จากข้อมูลแบบหลายลำดับชั้นมาเป็นต้นแบบในการพัฒนาระบบ ซึ่งการทำงานของระบบงานนี้ มีการทำงานของระบบหลักๆอยู่ 2 ขั้นตอน ดังนี้

1. การนำเข้าข้อมูลจากระบบอื่นๆ ได้ โดยทำการเลือกมาจากฐานข้อมูลต่างๆ
2. การวิเคราะห์หากฎความสัมพันธ์จากข้อมูลทั้งแบบลำดับชั้นเดียวและหลายลำดับชั้น

5.1 สภาพแวดล้อมการพัฒนาระบบ

สภาพแวดล้อมของระบบที่ต้องใช้ประกอบไปด้วย 2 ส่วนหลัก คือ ทางด้านฮาร์ดแวร์ และทางด้านซอฟต์แวร์

5.1.1 ด้านฮาร์ดแวร์ ประกอบด้วย

- เครื่องคอมพิวเตอร์แพนเทียม (Pantium) 4.2.0 MHz
- ฮาร์ดดิสก์ที่มีหน่วยความจำไม่น้อยกว่า 512 Mb
- เม้าส์และคีย์บอร์ด
- ซีดีรอม

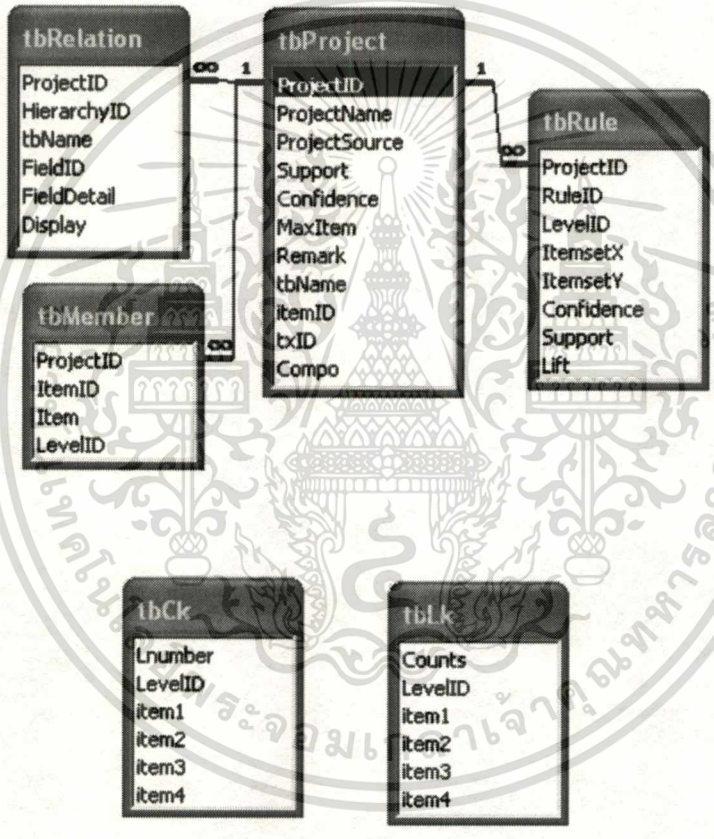
5.1.2 ด้านซอฟต์แวร์ ประกอบด้วย

- ระบบปฏิบัติการวินโดว์ (Window XP)
- วิชวลเบสิค เวอร์ชั่น 6 (Visual Basic6.0)
- คริสตัลรีพอร์ท (Crystal Report)
- ไมโครซอฟต์แอคเซส (Microsoft Access)

5.2 โครงสร้างตารางที่ใช้ในระบบ

ในระบบได้มีการใช้งานฐานข้อมูลภายในระบบเอง เพื่อเก็บข้อมูลในการประมวลผลต่างๆ รวมถึงเก็บข้อมูลที่ต้องการใช้ เช่น สร้างงานไมน์นิ่งเก็บไว้เพื่อนำมาวิเคราะห์ภายหลังได้

ฐานข้อมูลนี้ใช้ ไมโครซอฟต์แอคเซสซึ่งหาได้ง่าย เหมาะสำหรับการใช้งานทั่วๆไปได้ ชื่อของฐานข้อมูลที่ใช้ในระบบงานนี้คือ “MLMiningData.mdb” ซึ่งมีโครงสร้างตารางดังนี้



ภาพที่ 5.1 แบบจำลองความสัมพันธ์ของฐานข้อมูล

ตารางที่ 5.1 โครงสร้างของตารางที่ใช้เก็บรายละเอียดของงาน

ชื่อตาราง : tbProject

ใช้สำหรับ : เก็บข้อมูลงานที่ทำการวิเคราะห์ รวมถึงรายละเอียดของงานที่ทำการวิเคราะห์

ชื่อฟิลด์	รายละเอียด	ประเภท	ชนิดของคีย์	ตารางที่อ้างถึง
ProjectID	รหัสของงาน	Integer	PK	
ProjectName	ชื่อของงานที่ทำการไมน์นิ่ง	Varchar(255)		
ProjectSource	แหล่งที่อยู่ของฐานข้อมูลที่ทำการไมน์นิ่ง	Varchar(255)		
Support	ค่า มินิมัมซัพพอร์ตของงานไมน์นิ่ง	Integer		
Confidence	ค่า มินิมัมคอนฟิเดนซ์ ของงานไมน์นิ่ง	Integer		
MaxItem	จำนวนของไอเท็มที่สามารถมีได้มากที่สุดในกฎความสัมพันธ์ ของงานไมน์นิ่ง	Integer		
Remark	หมายเหตุ	Varchar(255)		
TbName	ชื่อตารางที่เก็บทรานเซคชั่นและไอเท็ม	Varchar(255)		
ItemID	ชื่อฟิลด์ที่เก็บข้อมูลไอเท็ม	Varchar(255)		
TxID	ชื่อฟิลด์ที่เก็บข้อมูลทรานเซคชั่น	Varchar(255)		

ตารางที่ 5.2 โครงสร้างของตารางที่ใช้เก็บรายละเอียดข้อมูลกราฟแบ่งหมวดหมู่

ชื่อตาราง : tbMember

ใช้สำหรับ : เก็บข้อมูลโครงสร้างของ กราฟแบ่งหมวดหมู่

ชื่อฟิลด์	รายละเอียด	ประเภท	ชนิดของคีย์	ตารางที่อ้างถึง
ProjectID	รหัสของงาน	Integer	FK	tbProject
ItemID	ข้อมูลไอเท็มไอดีที่เก็บอยู่ในกราฟแบ่งหมวดหมู่	Varchar(255)		
Item	ข้อมูลไอเท็ม ที่เก็บอยู่ในกราฟแบ่งหมวดหมู่	Varchar(255)		
LevelID	ลำดับของข้อมูลที่อยู่ใน กราฟแบ่งหมวดหมู่	Integer		

ตารางที่ 5.3 โครงสร้างของตารางที่ใช้เก็บรายละเอียดโครงสร้างกราฟแบ่งหมวดหมู่

ชื่อตาราง : tbRelation

ใช้สำหรับ : เก็บข้อมูลโครงสร้างของ กราฟแบ่งหมวดหมู่

ชื่อฟิลด์	รายละเอียด	ประเภท	ชนิดของคีย์	ตารางที่อ้างถึง
ProjectID	รหัสของงาน	Integer	FK	tbProject
HierarchyID	ลำดับของโครงสร้าง กราฟแบ่งหมวดหมู่	Integer		
TbName	ชื่อตารางที่เก็บข้อมูลของ กราฟแบ่งหมวดหมู่	Varchar(255)		
FieldID	ชื่อฟิลด์ที่เก็บข้อมูล ID ที่เป็นโครงสร้าง กราฟแบ่งหมวดหมู่	Integer		
FieldDetail	ชื่อฟิลด์ที่เก็บรายละเอียดของข้อมูลที่เป็นโครงสร้าง กราฟแบ่งหมวดหมู่	Varchar(255)		
Display	เป็นตัวบอกว่าฟิลด์นี้จะแสดงใน กราฟแบ่งหมวดหมู่ หรือไม่ (ใช้ในกรณีที่ข้อมูลมีโครงสร้างมากเกินความต้องการ)	Boolean		

ตารางที่ 5.4 โครงสร้างของตารางที่ใช้เก็บ Ck ไอเท็มเซต ที่ใช้ขณะประมวลผล

ชื่อตาราง : tbCk

ใช้สำหรับ : เก็บ Ck ไอเท็มเซต ขณะที่ประมวลผล (ใช้ชั่วคราวเท่านั้น)

ชื่อฟิลด์	รายละเอียด	ประเภท	ชนิดของคีย์	ตารางที่อ้างถึง
Lnumber	รหัส Candidate	Integer		
K	จำนวนของ ไอเท็ม ใน Candidate ไอเท็มเซต นี้	Integer		
item1	ข้อมูล ไอเท็ม ที่ 1	Varchar(255)		
item2	ข้อมูล ไอเท็ม ที่ 2	Varchar(255)		
...	...	...		
item (k)	ข้อมูล ไอเท็ม ที่ k	Varchar(255)		

ตารางที่ 5.5 โครงสร้างของตารางที่ใช้เก็บ L_k ไอเท็มเซต ที่ใช้ขณะประมวลผล

ชื่อตาราง : tbLk **ใช้สำหรับ :** เก็บ L_k ไอเท็มเซต ขณะที่ประมวลผล (ใช้ชั่วคราวเท่านั้น)				
ชื่อฟิลด์	**รายละเอียด**	**ประเภท**	**ชนิดของคีย์**	**ตารางที่อ้างถึง**
Count	จำนวนความถี่ของ L_k ไอเท็มเซต ที่ได้จากฐานข้อมูล	Long Integer		
K	จำนวนของ ไอเท็ม ใน L_k ไอเท็มเซต นี้	Integer		
item1	ข้อมูลไอเท็ม ที่ 1	Varchar(255)		
item2	ข้อมูลไอเท็ม ที่ 2	Varchar(255)		
...	...	...		
item (k)	ข้อมูลไอเท็ม ที่ k	Varchar(255)		

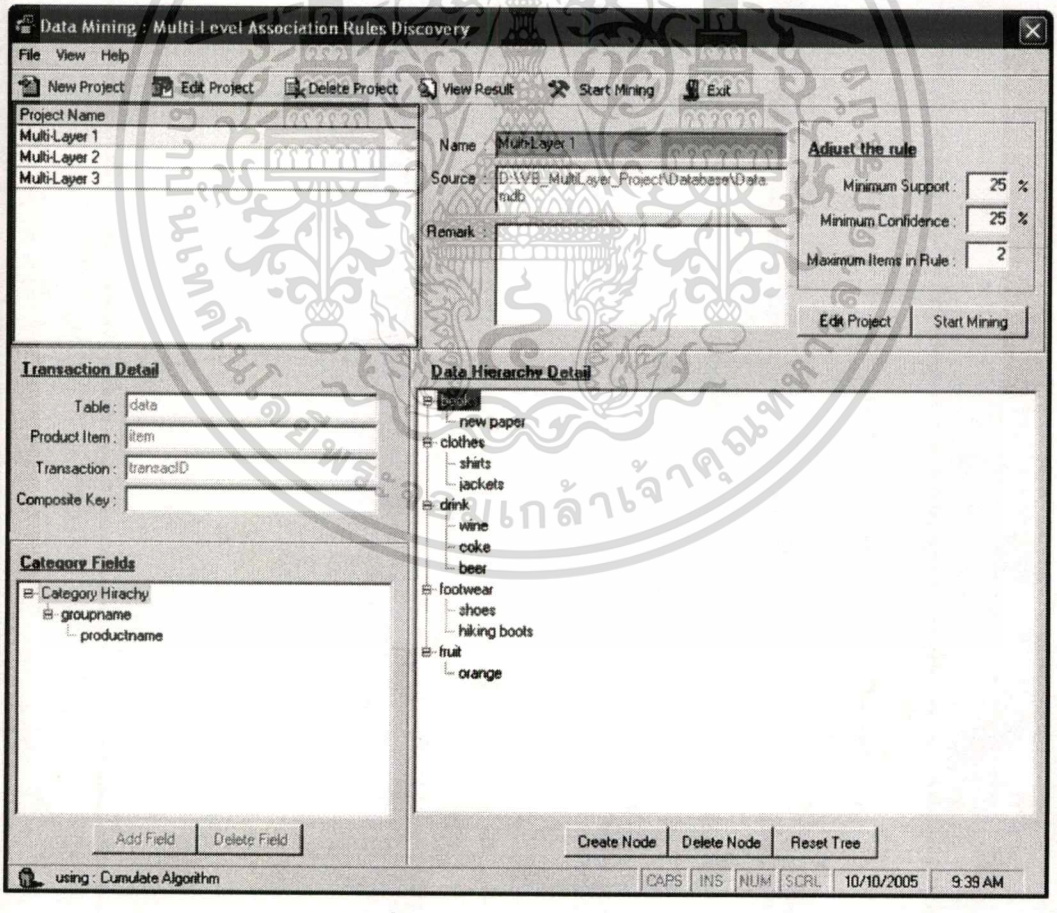
ตารางที่ 5.6 โครงสร้างของตารางที่ใช้เก็บรายละเอียดของกฎความสัมพันธ์

ชื่อตาราง : tbRule **ใช้สำหรับ :** เก็บรายละเอียดของกฎความสัมพันธ์ที่สร้างได้				
ชื่อฟิลด์	**รายละเอียด**	**ประเภท**	**ชนิดของคีย์**	**ตารางที่อ้างถึง**
ProjectID	รหัสของงาน	Integer	FK	tbProject
RuleID	รหัสของกฎความสัมพันธ์	Integer		
K	จำนวนของ ไอเท็ม ทั้งหมดที่มีในกฎความสัมพันธ์	Integer		
ItemsetX	แสดงข้อมูล ไอเท็มเซต Y	Varchar(255)		
ItemsetY	แสดงข้อมูล ไอเท็มเซต Y	Varchar(255)		
Support	ค่าซัพพอร์ท ของกฎความสัมพันธ์	Integer		
Confidence	ค่าคอนฟิเดนซ์ ของกฎความสัมพันธ์	Integer		
Lift	ค่าลิฟต์ ของกฎความสัมพันธ์	Integer		

5.3 การพัฒนาระบบและหน้าจอการทำงาน

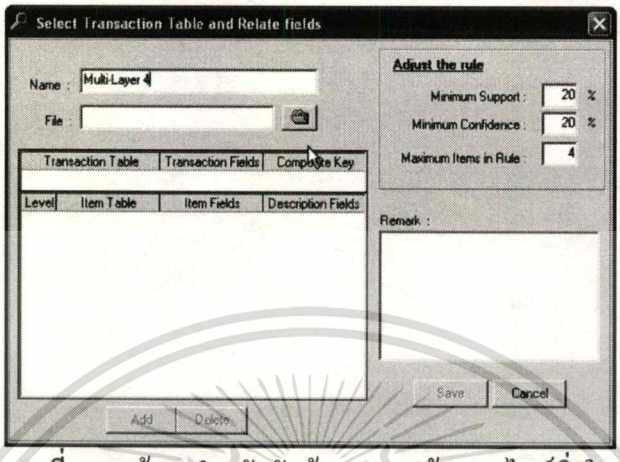
เมื่อเข้าสู่ระบบดาต้าไมน์นิ่งเพื่อค้นหาความสัมพันธ์แบบหลายลำดับชั้นนั้น ผู้ใช้จะเห็นหน้าจอหลักของโปรแกรม(ภาพที่ 5.2) ซึ่งแสดงข้อมูลของงานดาต้าไมน์นิ่งที่เคยสร้างไว้แล้ว (Data Mining Project) หรือที่เคยมีการทำไมน์นิ่งมาก่อนหน้านี้ ซึ่งระบบจะแสดงให้เห็นถึงข้อมูล ไอเท็ม และข้อมูลของ บรรพบุรุษ ที่กำหนดไว้ แหล่งที่เก็บฐานข้อมูล รวมทั้งรายละเอียดของความสัมพันธ์ของ ไอเท็ม

ถ้าต้องการทำการไมน์นิ่งใหม่ ก็สามารถทำได้เพียงเปลี่ยนค่า มินิมัมซัพพอร์ท, มินิมัม คอนฟิเดนซ์ และค่า จำนวนไอเท็มที่ต้องการให้มีได้มากที่สุดในกฎความสัมพันธ์ได้ตามต้องการและกดปุ่ม “Start Mining” ได้เลย หรือถ้าต้องการดูผลลัพธ์การไมน์นิ่งครั้งล่าสุดของงานไมน์นิ่งนั้นๆ ก็สามารถทำได้โดยการกดปุ่ม “View Result” ที่แถบเครื่องมือของหน้าจอหลักก็ได้



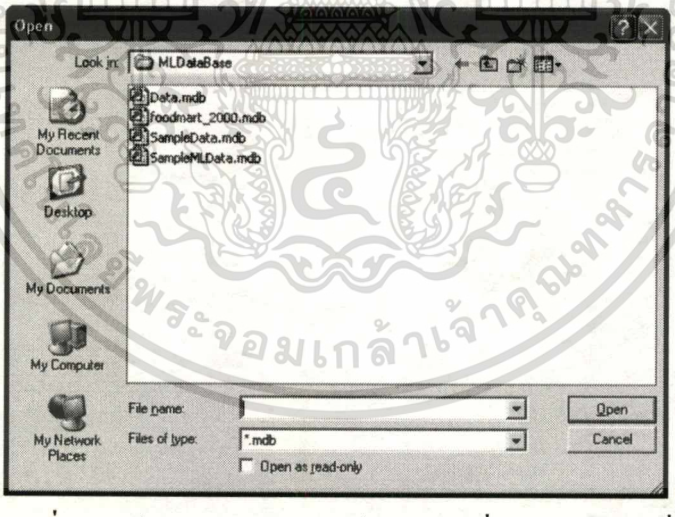
ภาพที่ 5.2 หน้าจอหลักของโปรแกรม

ถ้าต้องการสร้างงานไมน์นิ่งใหม่ สามารถทำได้โดยกดปุ่ม “New Project” ระบบจะแสดงหน้าจอให้ป้อนรายละเอียดเกี่ยวกับข้อมูลที่จะใช้ในการทำไมน์นิ่ง ดังนี้



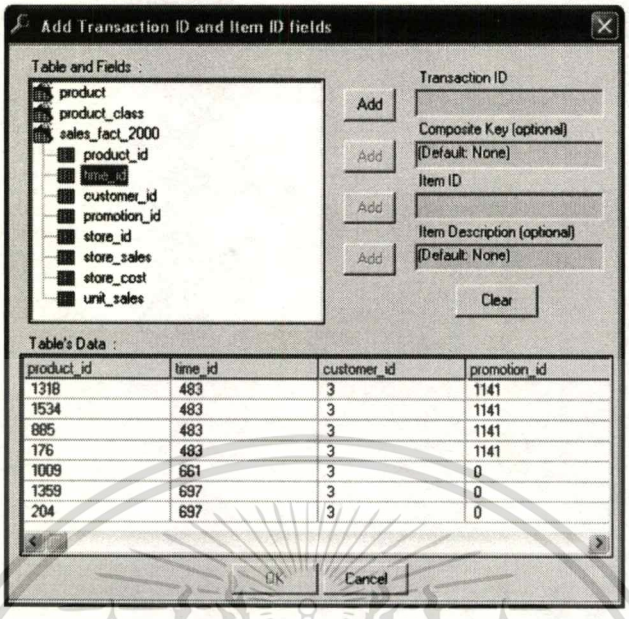
ภาพที่ 5.3 หน้าจอสำหรับรับข้อมูลการสร้างงานไมน์นิ่งใหม่

ผู้ใช้ต้องเป็นผู้เลือกฐานข้อมูลที่จะนำมาทำการไมน์นิ่งเอง โดยการกดที่ปุ่ม  เพื่อที่จะเปิดเลือกไฟล์ข้อมูลได้ ซึ่งจะแสดงไฟล์ข้อมูลของฐานข้อมูล ไมโครซอฟต์แอคเซสที่อยู่ในแฟ้มข้อมูลออกมา



ภาพที่ 5.4 หน้าจอสำหรับเลือกไฟล์ฐานข้อมูลที่ต้องการทำไมน์นิ่ง

เมื่อเลือกไฟล์ฐานข้อมูลได้แล้วระบบจะแสดงข้อมูลตารางและฟิลด์ข้อมูลของตารางทั้งหมดของฐานข้อมูลนั้นมาแสดง รวมไปถึงตัวอย่างของข้อมูลในตารางถูกเลือกขณะนั้นไว้ด้วย

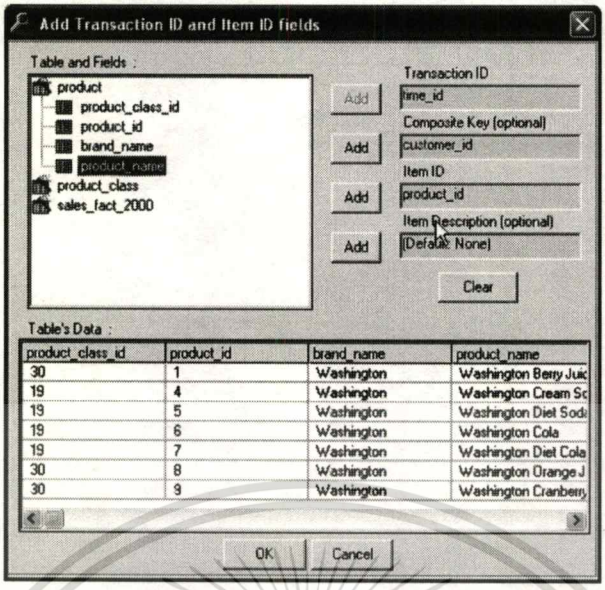


ภาพที่ 5.5 หน้าจอสำหรับเลือกตารางและฟิลด์ข้อมูลที่จะนำมาทำการไมน์นิ่ง

ในเบื้องต้น ระบบจะบังคับให้เลือกฟิลด์ที่เป็นทรานเซคชั่นไอดี (TransactionID) ก่อนจึงจะทำการเลือกฟิลด์ข้อมูลอื่นต่อไปได้

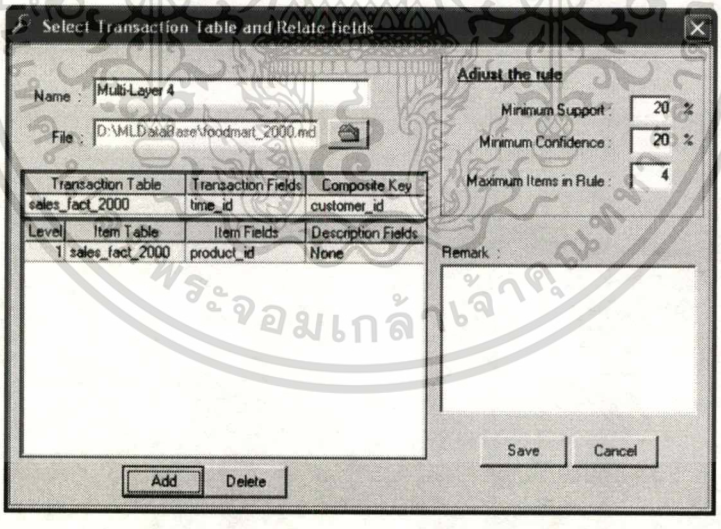
ซึ่งฟิลด์ไอเท็มไอดีที่จะเลือกต่อไปต้องเป็นฟิลด์ในตารางเดียวกับฟิลด์ของทรานเซคชั่นไอดี ถ้าเลือกผิด ระบบจะมีข้อความเตือนดังนี้

ภาพที่ 5.6 ข้อความเตือน ในกรณีที่เลือกฟิลด์ข้อมูลไม่ตรงกับฟิลด์ทรานเซคชั่น



ภาพที่ 5.7 หน้าจอสำหรับตารางและฟิลด์ข้อมูลที่เลือกนำมาทำการไมน์นิ่ง

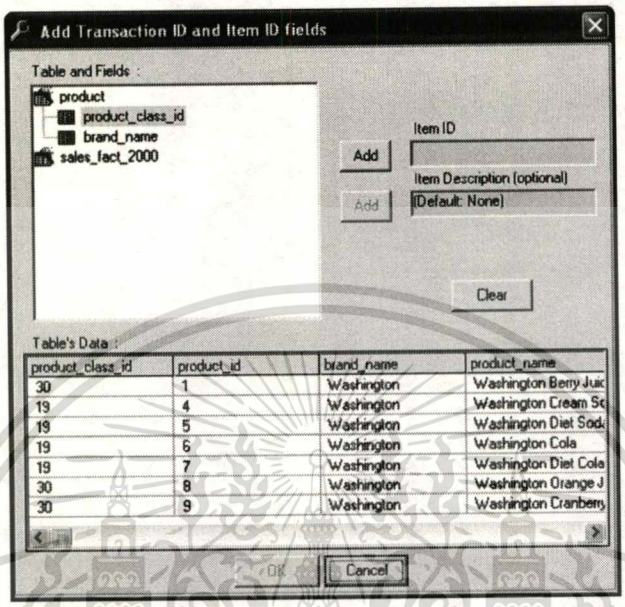
เมื่อเลือกข้อมูลได้แล้วดังภาพที่ 5.7 แล้วทำการกดปุ่ม "OK" เรียบร้อยแล้ว หน้าจอสำหรับรับข้อมูลฟิลด์ก็จะหายไป ข้อมูลทั้งหมดจะถูกจัดแสดงในหน้าจอของการรับรายละเอียดงานไมน์นิ่งดังแสดงในภาพที่ 5.8



ภาพที่ 5.8 แสดงข้อมูลฟิลด์ทรานเซคชั่นไอดีและไอเท็มไอดี

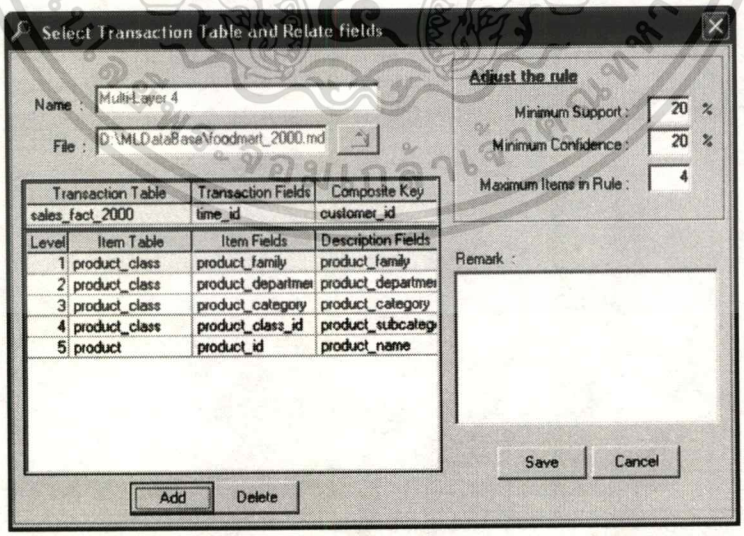
ผู้ใช้สามารถทำการจัดลำดับชั้นของข้อมูลโดยการกดปุ่ม "Add" ระบบจะทำการเลือกตารางที่เกี่ยวข้องกับ ฟิลด์ของไอเท็มที่เลือกไปก่อนหน้านี้แล้วขึ้นมา ตัวอย่างจากภาพที่ 5.8 ได้มีการเลือก product_id เป็น ไอเท็มไอดีผู้ใช้ต้องทำการเลือกฟิลด์ที่จะมาเป็นบรรพบุรุษของ product_id

เป็นลำดับต่อไป ระบบจะเลือกตารางที่มีฟิลด์ product_id อยู่ในตารางนั้นขึ้นมาแสดง ดังภาพที่ 5.9 นั่นก็คือตาราง product ซึ่งสามารถเลือกได้ว่าจะนำฟิลด์ product_class_id หรือฟิลด์ brand_name มาเป็นบรรพบุรุษได้

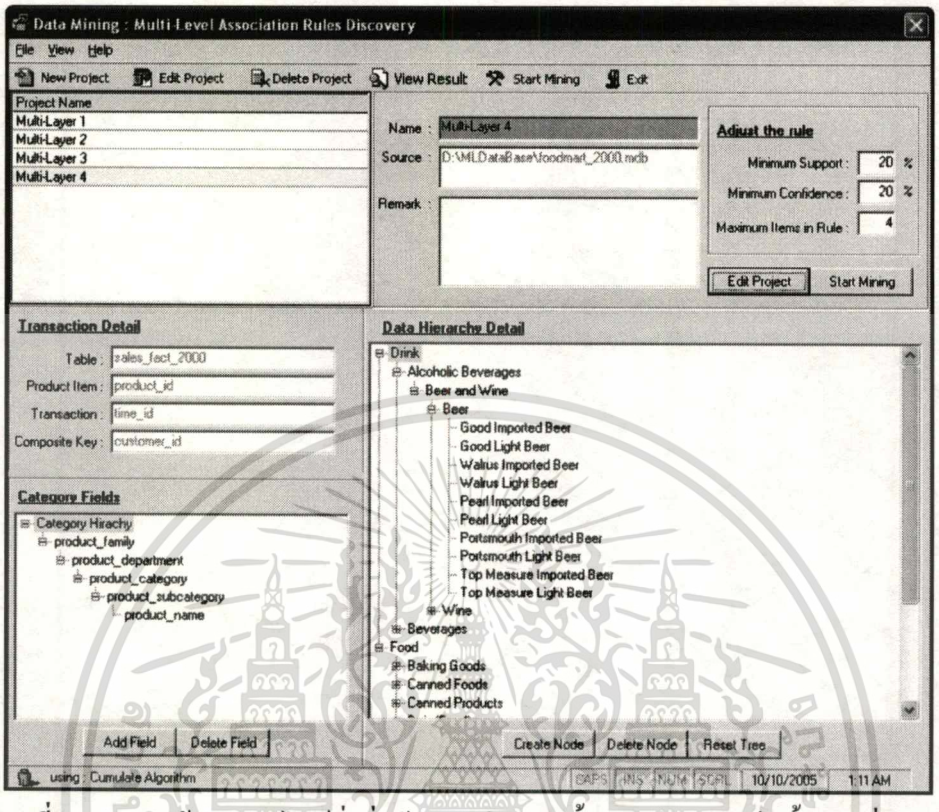


ภาพที่ 5.9 แสดงข้อมูลตารางและฟิลด์ที่สัมพันธ์กับข้อมูล ไอเท็มไอดีที่เลือกไว้

การสร้างลำดับชั้นของข้อมูลนี้สามารถทำได้เรื่อยๆจนกระทั่งไม่มีฟิลด์ที่สัมพันธ์กับข้อมูล ไอเท็มไอดี ที่เลือกไว้ ดังภาพที่ 5.10

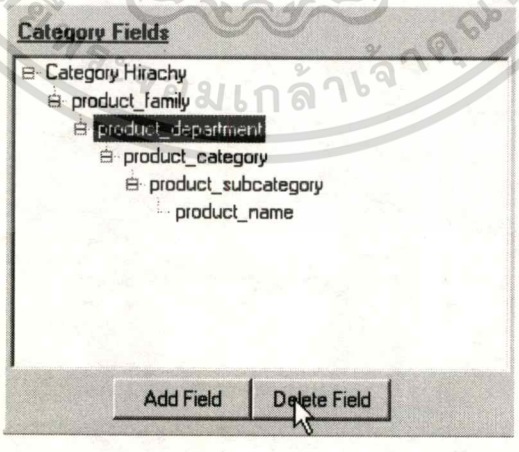


ภาพที่ 5.10 แสดงข้อมูลฟิลด์ ไอเท็มไอดีและข้อมูลลำดับชั้นทั้งหมด



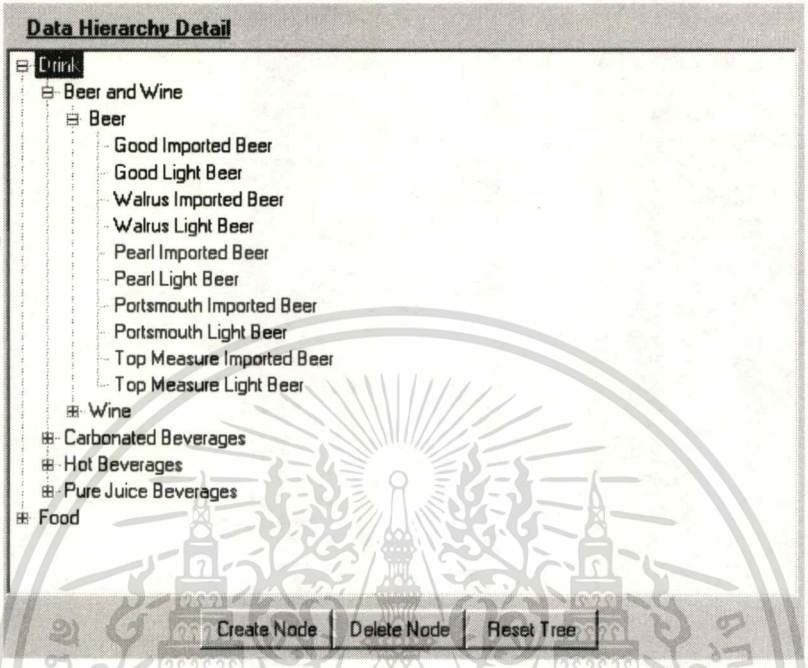
ภาพที่ 5.11 แสดงข้อมูลงานไมน์นิ่งที่สร้างรวมถึงลำดับชั้นและข้อมูลไอเท็มทั้งหมดที่เลือกไว้

ซึ่งถ้าผู้ใช้เห็นว่าลำดับชั้นของข้อมูลมีมากไป สามารถเลือกที่จะไม่แสดงได้โดยการกดปุ่ม "Delete Field"



ภาพที่ 5.12 แสดงการเลือกฟิลด์ที่ไม่ต้องการที่จะแสดงผล

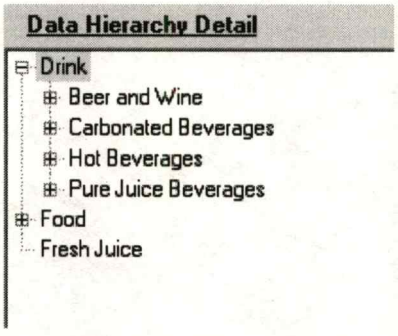
ข้อมูลของ ไอเท็ม ในส่วนที่เป็น product_department ก็จะถูกลบออกไปจากกราฟ “Data Hierarchy Detail” ดังภาพที่ 5.13



ภาพที่ 5.13 ข้อมูลกราฟ “Data Hierarchy Detail” ที่แสดงหลังจากที่ลบบางฟิลด์ออกแล้ว

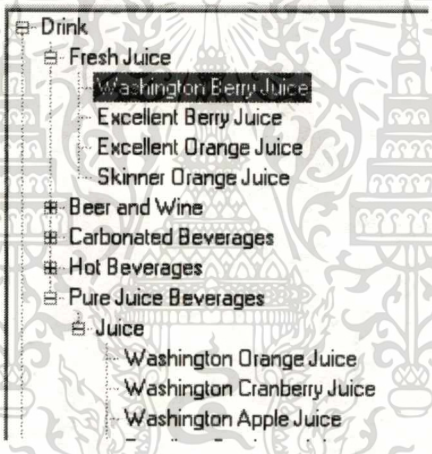
หากผู้ใช้มีความประสงค์จะเพิ่มข้อมูลไอเท็ม สามารถทำได้โดยกดที่ปุ่ม “Create Node” ระบบจะปรากฏหน้าจอให้กรอกข้อมูล

ภาพที่ 5.14 หน้าจอรับข้อมูลไอเท็มใหม่ที่ผู้ใช้ต้องการสร้างเพิ่ม



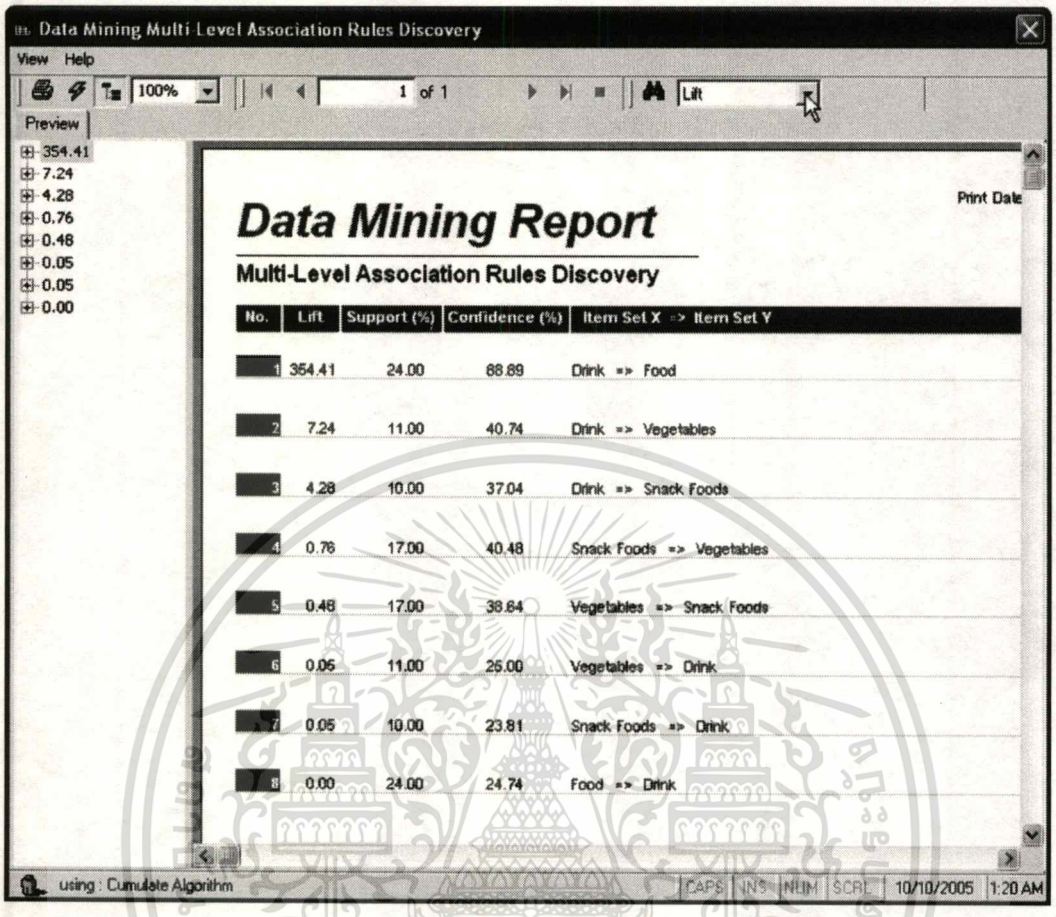
ภาพที่ 5.15 แสดงข้อมูลไอเท็มที่ผู้ใช้สร้าง

ซึ่งผู้ใช้สามารถที่จะจัดกลุ่มใหม่ได้ โดยการเลือกที่ไอเท็มที่ต้องการแล้วลากเม้าส์ไปวางยังที่ต้องการให้เป็นบรรพบุรุษได้



ภาพที่ 5.16 แสดงข้อมูลไอเท็ม ที่ผู้ใช้ทำการจัดกลุ่มใหม่ตามที่ต้องการ

เมื่อผู้ใช้ปรับแต่งข้อมูลตามที่ต้องการแล้วกดปุ่ม "Start Mining" ที่หน้าจอหลักเพื่อทำการประมวลผลไมน์นิ่งข้อมูลแบบหลายลำดับชั้น จะได้ผลลัพธ์เป็นกฎความสัมพันธ์ออกมา ซึ่งผู้ใช้สามารถเลือกได้ว่าต้องการที่จะให้เรียงลำดับจากค่า ลิฟต์, ซัพพอร์ท หรือคอนฟิเดนซ์ ของกฎได้



ภาพที่ 5.17 แสดงผลกฎความสัมพันธ์ที่ได้จากการทำดาต้าไมน์นิ่ง

บทที่ 6

การประยุกต์ใช้งาน

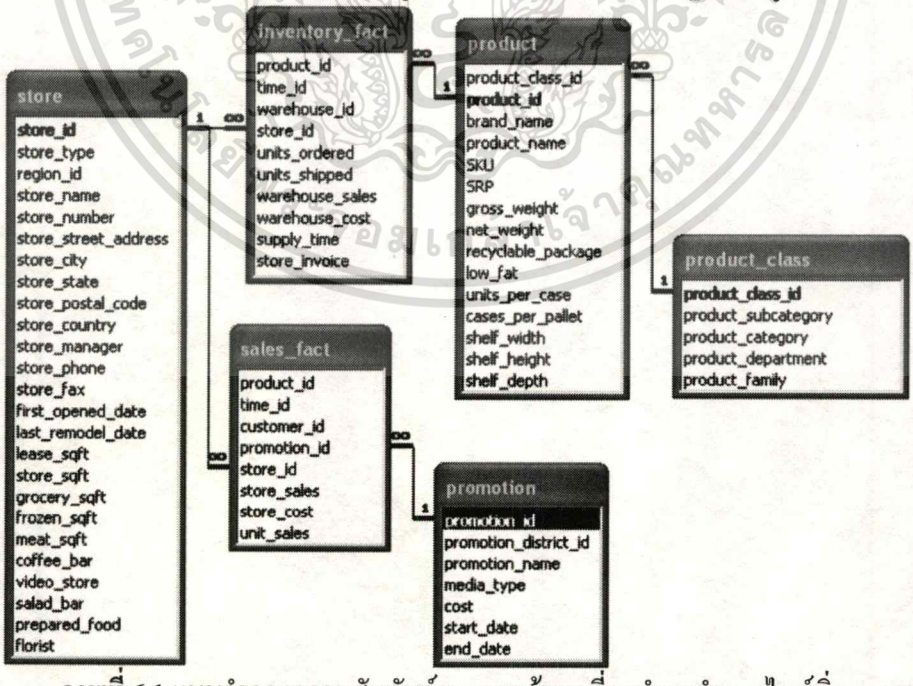
ห้างสรรพสินค้าแห่งหนึ่ง ต้องการทำยอดขายสินค้าอาหารประเภทต่างๆ ในซุปเปอร์มาร์เก็ตของทางห้าง ซึ่งต้องการทราบความนิยมในการจับจ่ายซื้อของของลูกค้า เพื่อนำมาจัดโปรโมชั่นเพื่อดึงยอดขายให้สูงขึ้น จึงได้จัดให้มีการนำระบบดาต้าไมน์นิ่งเพื่อค้นหาความสัมพันธ์แบบหลายระดับนี้มาใช้ เพื่อให้ได้ข้อเท็จจริงที่มีประโยชน์มาประยุกต์ใช้ร่วมในการวิเคราะห์ด้วย ดังนี้

6.1 วัตถุประสงค์ในธุรกิจ

- ต้องการทราบถึงความนิยมในการบริโภคของลูกค้าอย่างแท้จริง
- เพื่อคัดสรรสิ่งที่ดีที่สุดมาทำการจัดโปรโมชั่น

6.2 การเตรียมข้อมูล

เมื่อศึกษาข้อมูลจากฐานข้อมูลซึ่งเกี่ยวข้องกับสินค้าและการขายสินค้าของซุปเปอร์มาร์เก็ตของห้างสรรพสินค้ามีโครงสร้างของฐานข้อมูลและรายละเอียดตารางในฐานข้อมูล ดังนี้



ภาพที่ 6.1 แบบจำลองความสัมพันธ์ของฐานข้อมูลที่จะนำมาทำการไมน์นิ่ง

จากภาพที่ 6.1 นั้น ตารางที่เกี่ยวข้องกับข้อมูลของรายการขายสินค้าและข้อมูลสินค้าที่น่าสนใจมี 3 ตาราง คือ ตาราง "sales_fact" ซึ่งใช้เก็บข้อมูลขายสินค้า, ตาราง "product" ซึ่งใช้เก็บข้อมูลสินค้าและตาราง "product_class" ซึ่งใช้เก็บข้อมูลการจัดหมวดหมู่สินค้า ดังนี้

ตารางที่ 6.1 โครงสร้างของตารางที่ใช้เก็บรายละเอียดสินค้า

ชื่อตาราง : product

ใช้สำหรับ : รายละเอียดสินค้า

ชื่อฟิลด์	รายละเอียด	ประเภท	ชนิดของคีย์	ตารางที่อ้างถึง
product_id	รหัสสินค้า	Integer	PK	
product_class_id	รหัสหมวดหมู่สินค้า	Integer	FK	product_class
brand_name	ยี่ห้อสินค้า	Varchar(100)		
product_name	ชื่อสินค้า	Varchar(100)		
SKU	รหัสที่เก็บในคลังสินค้า (Stock keeping number)	Integer		
SRP	ราคาเสนอขาย (Suggest retail price)	Integer		
gross_weight	น้ำหนักทั้งหมด	Integer		
net_weight	น้ำหนักสุทธิ	Integer		
recyclatable_package	สามารถทำห่อบรรจุกลับมาใช้ได้อีก (หรือไม่)	Boolean		
low_fat	เป็นสินค้าไขมันต่ำ (หรือไม่)	Boolean		
unit_per_case	ขนาดบรรจุต่อชิ้น	Integer		
cases_per_pallet	จำนวนชิ้นที่บรรจุในหีบห่อ	Integer		
shelf_width	ความกว้างของขนาดที่วางขายในชั้นขายสินค้า	Integer		
shelf_height	ความสูงของขนาดที่วางขายในชั้นขายสินค้า	Integer		
shelf_dept	ความลึกของขนาดที่วางขายในชั้นขายสินค้า	Integer		

ตารางที่ 6.2 โครงสร้างของตารางที่ใช้เก็บข้อมูลการจัดหมวดหมู่สินค้า

ชื่อตาราง : product_class

ใช้สำหรับ : รายละเอียดการจัดหมวดหมู่สินค้า

ชื่อฟิลด์	รายละเอียด	ประเภท	ชนิดของคีย์	ตารางที่อ้างถึง
product_class_id	รหัสหมวดหมู่สินค้าย่อย	Integer	PK	
product_subcategory	หมวดหมู่สินค้าย่อย	Varchar(100)		
product_category	หมวดหมู่สินค้า	Varchar(100)		
product_department	ชนิดของสินค้า	Varchar(100)		
product_family	ประเภทของสินค้า	Varchar(100)		

ตารางที่ 6.3 โครงสร้างของตารางที่ใช้เก็บรายละเอียดการขายสินค้า

ชื่อตาราง : sales_fact

ใช้สำหรับ : รายละเอียดการขายสินค้า

ชื่อฟิลด์	รายละเอียด	ประเภท	ชนิดของคีย์	ตารางที่อ้างถึง
product_id	รหัสสินค้า	Varchar(50)	FK	product
time_id	วันและเวลาที่ขาย	Integer		
customer_id	ลำดับลูกค้า	Integer		
promotion_id	รหัสโปรโมชั่น	Integer		
store_id	รหัสคลังสินค้า	Integer		
store_sales	สาขาที่ขายสินค้า	Integer		
store_cost	ราคาขาย	Integer		
unit_sales	จำนวนที่ขาย	Integer		

6.3 การไมน์นิ่งข้อมูล

เมื่อทำการพิจารณารายละเอียดของข้อมูลการจัดหมวดหมู่สินค้าและข้อมูลการขาย และทำการเลือกฟิลด์ข้อมูลที่จะใช้ในการทำไมน์นิ่งครั้งนี้คือ

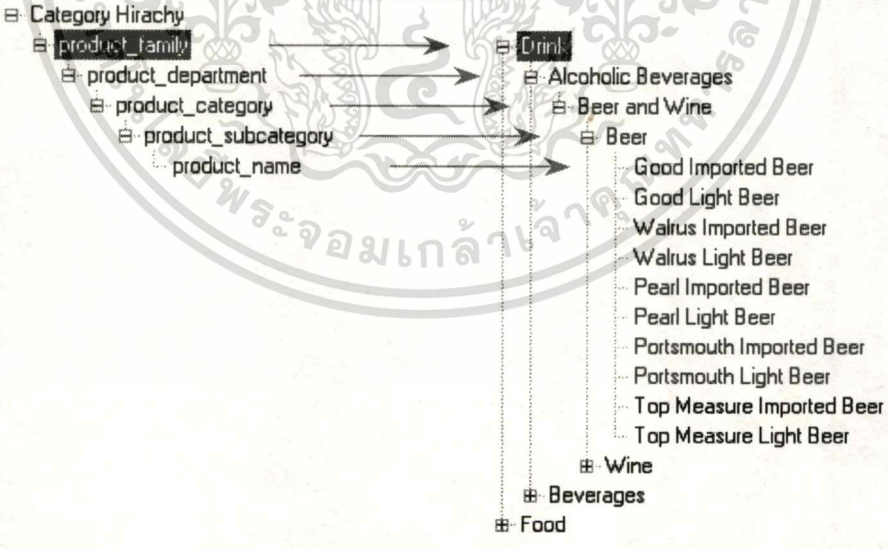
6.3.1 ข้อมูลทรานเซกชั่น

- customer_id
- time_id

6.3.2 ข้อมูลไอเท็ม

- product_name (product_id)
- product_subcategory (product_class_id)
- product_category
- product_department
- product_family

ลำดับต่อไปคือการจัดกลุ่มข้อมูลตามที่เลือกมาโดยใช้ระบบงานที่พัฒนาขึ้นมาทำการจัดกลุ่ม ซึ่งได้จัดให้มีลำดับของหมวดหมู่สินค้า ดังภาพที่ 6.2



ภาพที่ 6.2 แสดงรายละเอียดลำดับชั้นของหมวดหมู่สินค้าและชื่อสินค้า

6.4 วิเคราะห์ผลลัพธ์ของการทำไมน์นิ่ง

ผลลัพธ์ของการไมน์นิ่งครั้งนี้ ได้ผลลัพธ์ออกมาเป็นกฎความสัมพันธ์ทั้งหมด 17 กฎ และในจำนวนนี้มีกฎความสัมพันธ์ที่มีค่าลิฟต์น้อยกว่า 1 ซึ่งถือเป็นกฎที่ไม่มีประโยชน์อยู่เพียง 1 กฎเท่านั้น ซึ่งหมายความว่ามีกฎความสัมพันธ์ที่น่าสนใจทั้งหมดเป็นจำนวน 16 กฎด้วยกัน ดังภาพ

No.	Lift	Support (%)	Confidence (%)	Item Set X => Item Set Y
1	51064.81	2.70	100.00	Tell Tale Red Pepper => Walrus Chardonnay
2	51064.81	2.70	100.00	Walrus Chardonnay => Tell Tale Red Pepper
3	17021.60	2.70	100.00	Tell Tale Red Pepper => Wine
4	16472.52	2.70	100.00	Tell Tale Red Pepper => Beer and Wine
5	8371.28	2.70	100.00	Walrus Chardonnay => Fresh Vegetables
6	5937.77	2.70	100.00	Walrus Chardonnay => Vegetables
7	210.14	2.70	33.33	Wine => Tell Tale Red Pepper
8	178.37	2.70	32.26	Beer and Wine => Tell Tale Red Pepper
9	34.46	2.70	33.33	Wine => Fresh Vegetables
10	29.24	2.70	32.26	Beer and Wine => Fresh Vegetables
11	24.44	2.70	33.33	Wine => Vegetables
12	20.74	2.70	32.26	Beer and Wine => Vegetables
13	6.05	2.70	16.39	Fresh Vegetables => Walrus Chardonnay
14	2.02	2.70	16.39	Fresh Vegetables => Wine
15	1.95	2.70	16.39	Fresh Vegetables => Beer and Wine
16	1.09	2.70	11.63	Vegetables => Walrus Chardonnay
17	0.36	2.70	11.63	Vegetables => Wine

ภาพที่ 6.3 แสดงผลกฎความสัมพันธ์ที่ได้จากการทำดาต้าไมน์นิ่ง

สำหรับกฎความสัมพันธ์ลำดับที่ 17 นั้นไม่น่าสนใจเพราะมีค่าลิฟต์น้อย ถึงแม้ว่าจะมีค่าซัพพอร์ทหรือความถี่ที่ลูกค้าซื้อสินค้า Vegetables และ Wine พร้อมกันเป็นจำนวน 2.7% เท่ากับกฎตัวอื่นๆก็ตาม นั่นวิเคราะห์ได้อีกอย่างหนึ่งว่า มีลูกค้าที่ซื้อไวน์เยอะมากอยู่แล้ว โอกาสที่ลูกค้าจะซื้อไวน์ร่วมกับสินค้าชนิดอื่นๆจึงมีเยอะตามไปด้วย แต่ไม่จำเป็นจะต้องซื้อไวน์คู่กับผักแต่อย่างใด ดังนั้นจึงถือว่ากฎนี้ไม่น่าให้ความสำคัญ

สำหรับกฎมีค่าคอนฟิเดนซ์สูงๆนั้น ย่อมมีความน่าสนใจสูงกว่ากฎอื่น จึงจะพิจารณาจากกฎที่มีค่าลิฟต์ที่สูงที่สุดและค่าคอนฟิเดนซ์สูงที่สุด(100%) ก่อน ซึ่งได้แก่

Tell Tale Red Pepper => Walrus Chardonnay [2.7%, 100%]

Walrus Chardonnay => Tell Tale Red Pepper [2.7%, 100%]

ซึ่งหมายความว่า คนที่ซื้อสินค้าพริกไทยยี่ห้อ Tell Tale Red Pepper นั้นจะซื้อไวน์ยี่ห้อ Walrus Chardonnay ทุกครั้ง และคนที่ซื้อไวน์ Walrus Chardonnay จะซื้อ Tell Tale Red Pepper ทุกครั้งเช่นเดียวกัน เพราะว่ามีค่าคอนฟิเดนซ์ 100% เท่ากันทั้งสองกฎ สำหรับสินค้าประเภท Walrus Chardonnay เป็นสินค้าใน ประเภท Wine ซึ่งอยู่ในหมวดหมู่ Beer and Wine และสินค้า Tell Tale Red Pepper เป็นสินค้าประเภท Fresh Vegetable ในหมวดหมู่ Vegetable
ดังนั้น ทั้งสองกฎข้างต้น จึงเป็นประโยชน์มากกว่ากฎความสัมพันธ์ที่ได้ ต่อไปนี้

Tell Tale Red Pepper => Wine [2.7%, 100%]

Tell Tale Red Pepper => Beer and Wine [2.7%, 100%]

Walrus Chardonnay => Fresh Vegetables [2.7%, 100%]

Walrus Chardonnay => Vegetables [2.7%, 100%]

6.5 สรุปผลการทำไมน์นิ่ง และข้อเสนอแนะในการนำไปใช้

การทำไมน์นิ่งครั้งนี้สรุปผลได้ว่า ผู้บริโภคนิยมซื้อสินค้า Tell Tale Red Pepper และสินค้า Walrus Chardonnay ไปด้วยกัน และจากกฎความสัมพันธ์ข้อสุดท้ายที่ที่ได้กล่าวในข้างต้นว่าเป็นกฎที่ไม่มีประโยชน์แต่ทำให้เราทราบได้ว่ามีผู้ที่นิยมบริโภคไวน์เป็นจำนวนมาก ดังนั้น จึงไม่แนะนำให้จัดโปรโมชั่นเพื่อลดราคาไวน์ เพราะถึงแม้ว่าจะไม่ลดราคา ก็มีผู้บริโภคนิยมซื้อสินค้าชนิดนี้อยู่แล้ว แต่จะแนะนำให้ลดราคาหรือ Tell Tale Red Pepper แทน เพราะว่ามีโอกาสเป็นไปได้สูงมากที่ผู้บริโภคที่ซื้อ Tell Tale Red Pepper นั้นจะซื้อไวน์ไปด้วย จะทำให้ยอดจำหน่ายไวน์ของทางซุปเปอร์มาเก็ตสูงขึ้นโดยที่ไม่ต้องลดราคาไวน์แต่อย่างใด

บทที่ 7

สรุปผลการดำเนินงาน

โครงการนี้จัดทำขึ้น เพื่อเสนอให้เห็นถึงการนำทฤษฎีดาต้าไมน์นิ่ง มาประยุกต์ใช้ในธุรกิจได้จริง ซึ่งถ้านำไปช่วยในการวิเคราะห์ข้อมูลเพื่อประกอบในการวางแผนกลยุทธ์ทางการตลาดของบริษัท จะทำให้ผู้ประกอบการเข้าถึงความต้องการของลูกค้ามากยิ่งขึ้น

7.1 ผลการวิเคราะห์และออกแบบระบบ

การพัฒนาโปรแกรมเพื่อค้นหากฎความสัมพันธ์จากข้อมูลแบบหลายลำดับชั้นนี้ เป็นการใช้อัลกอริทึมคิวมูเลทในการค้นหากฎความสัมพันธ์ ซึ่งโปรแกรมที่พัฒนานี้สามารถติดต่อกับฐานข้อมูลที่เป็นฐานข้อมูลเชิงสัมพันธ์ (relational databse) โดยผู้ใช้งานเป็นผู้กำหนดว่าจะวิเคราะห์ฐานข้อมูลใดฟิลด์ไหนได้ตามต้องการ ซึ่งง่ายต่อการใช้งานเพราะให้ผู้ใช้สามารถกำหนดเองได้ จึงสามารถนำไปใช้งานได้กับฐานข้อมูลที่มีความสัมพันธ์กันหลายๆ รูปแบบ ตามความเหมาะสม

ผลลัพธ์ที่ได้จะแสดงออกมาในรูปแบบของกฎความสัมพันธ์แบบหลายลำดับชั้น แสดงผลออกมาในรูปแบบของสัญลักษณ์ซึ่งเข้าใจง่าย

7.2 ข้อเสนอแนะ

ระบบงานนี้สามารถทำงานได้อย่างมีประสิทธิภาพก็ต่อเมื่อทำการติดต่อกับฐานข้อมูลแบบไมโครซอฟต์แอคเซสเท่านั้น แต่ผู้ใช้สมารถกำหนดได้ว่าต้องการวิเคราะห์ข้อมูลจากฐานข้อมูลใดจากตารางและฟิลด์ใดก็ได้ ซึ่งหากต้องการนำไปใช้จริงควรนำไปประยุกต์ให้มีการติดต่อกับฐานข้อมูลแบบอื่นได้ด้วย และถ้าใช้งานกับฐานข้อมูลที่มีการประมวลผลข้อมูลได้เร็วจะทำให้ระบบทำงานได้ดีกว่าเดิม

7.3 ข้อจำกัด

อย่างไรก็ตามการทำงานของระบบงานนี้มีข้อจำกัดคือ เมื่อประมวลผลกับฐานข้อมูลขนาดใหญ่มาก หรือต้องการให้วิเคราะห์ความสัมพันธ์ให้ขนาดของไอเท็มเซตมีจำนวนหลายไอเท็ม นั้นจะใช้เวลาประมวลผลค่อนข้างนาน เนื่องจากการประมวลผลของไมโครซอฟต์แอคเซสในการอ่าน

ฐานข้อมูลขนาดใหญ่นั้นใช้เวลาค่อนข้างนาน และหากใช้งานให้การประมวลผลมีการจัดกลุ่ม (group by) โดยให้เงื่อนไขไปมากๆ นั้นจะยิ่งทำให้การประมวลผลช้าขึ้นมาก

บรรณานุกรม

Agrawal, Rakesh and Ramakrishnan, Srikant. 1994. **Fast algorithms for mining association rules** In Proc. of the VLDB Conference, Santiago, Chile. [Online]. Available: http://www.almaden.ibm.com/software/quest/Publications/papers/vldb94.pdf.

Agrawal, Rakesh. and Ramakrishnan, Srikant. 1995. **Mining Generalized Association Rules.** IBM Almaden Research Center. [Online]. Available: http://www.almaden.ibm.com/software/quest/Publications/papers/vldb95_tax.pdf.

Han, Jiawei and Kamber, Micheline. 2000. **Data Mining: Concept and Techniqies**. San Diego: Morgan Kaufmann Publishers.

Han, Jiawei and Fu, Yongjian. 1999. **Mining Multiple-Level Association Rules in Large Databases**. Washington, DC: IEEE Computer Society.

ประวัติผู้เขียน

ชื่อ-นามสกุล	นางสาวนภาพร รุ่งแสงเพชร
สถานที่เกิด	จังหวัดจันทบุรี
ประวัติการศึกษา	
ระดับประถมศึกษา	โรงเรียนสตรีมารดาพิทักษ์ จังหวัดจันทบุรี
ระดับมัธยมศึกษา	โรงเรียนเบญจมราชูทิศ จังหวัดจันทบุรี
ระดับอุดมศึกษา	คณะวิทยาศาสตร์ ภาควิชาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยบูรพา
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตรบัณฑิต (คอมพิวเตอร์)